

# **Git / Docker Images**

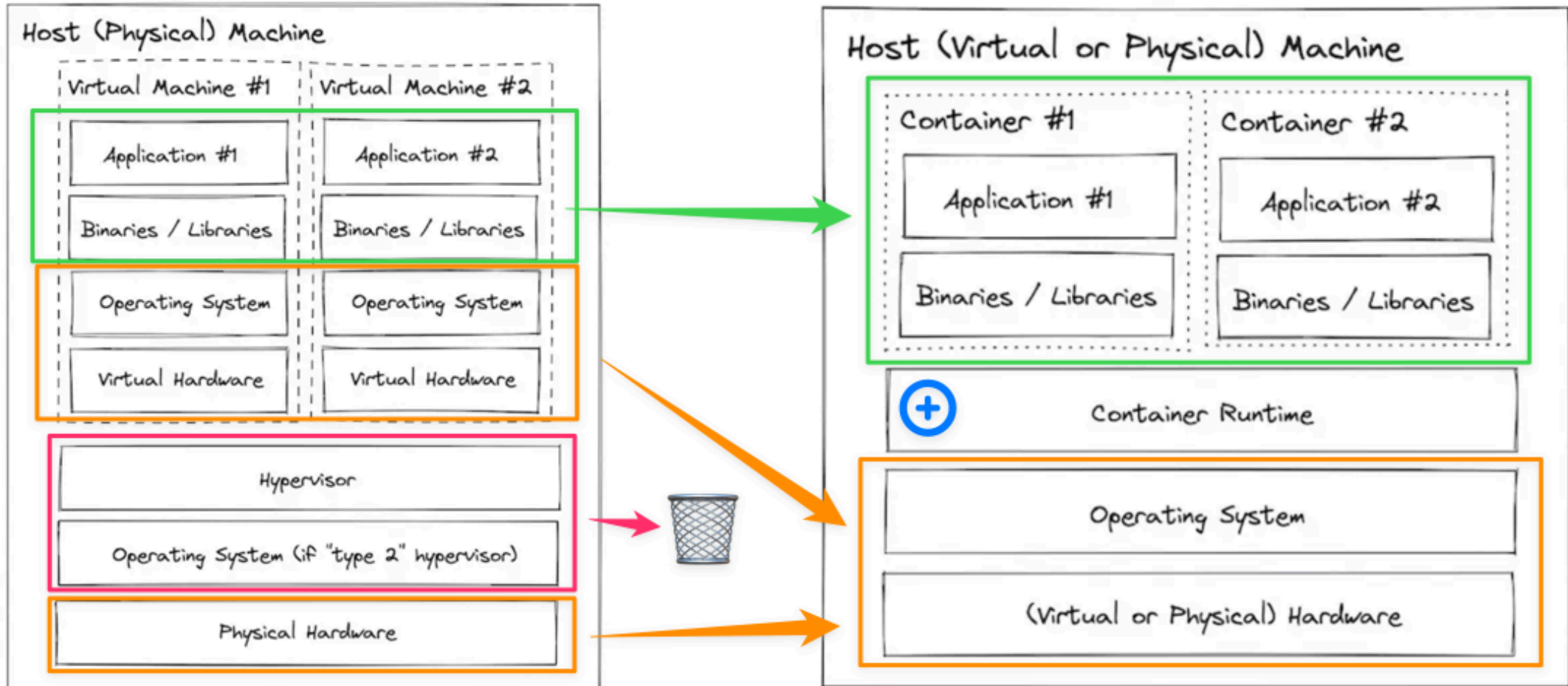
## **Modul 169**

# Inhalt

- **Repetition**
- **Modul 169 VM**
- **Git**
  - Grundlagen
  - Installieren
  - Konfigurieren
  - SSH-Key
- **Docker Image**  
*Schichtenarchitektur / Eigenschaften / Bilden eines Image*
- **Dockerfile**  
*Befehle / Caching / Multistage Build*
- **Übungen**  
*zu Docker Image und Dockerfile*

# Was versteht Ihr unter einem Container?

# Virtuelle Maschine / Container





# Container teilen

*\*physikalische* **Ressourcen!**

\* unter Linux 🧐



# **Modul 169 VM**

## **Live Demo der Umgebung durch Lehrperson**

## **Auftrag**

Lesen Sie das Kapitel "Git"

- **Git Start**

-  Einzelarbeit
-  10 Min



# GIT Demo

- GitHub Projekte zeigen
- Commits
- Änderungen sehen
- PR vs Mergen

## **Auftrag**

Machen Sie die Übung "Git: Start".

- **Git Start**



-  Einzelarbeit
-  Bis zur grossen Pause

 **20 Min Pause**

## **Auftrag**

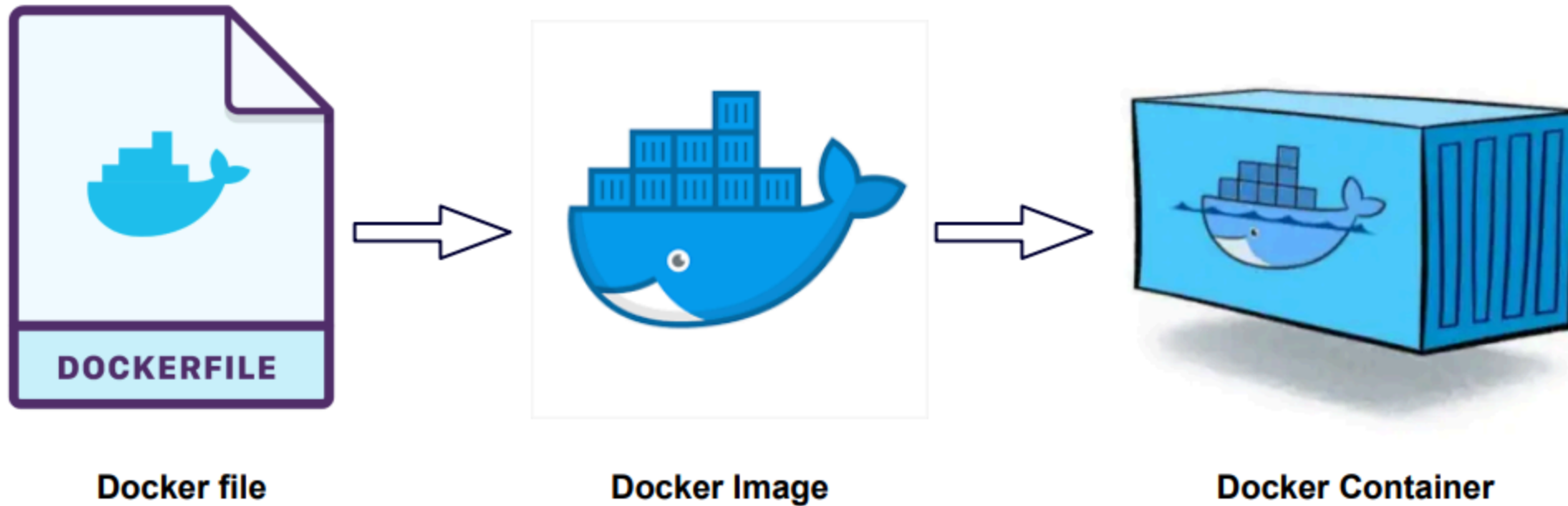
Lesen und Machen Sie die Übung "Git: Übungs-Repository".

- **Git: Übungs-Repository**

-  Einzelarbeit
-  bis zur Grossen Pause

# Wie werden Docker Images erstellt?

# Docker: vom File zum Container



# Docker: Hauptmerkmale

## Dockerfile

```
FROM ubuntu
COPY . /app
RUN pip install -r requirements.txt
CMD ["python", "app.py"]
```

## Vom File zum Image

```
docker build . -t webapp:1.12.5
```



## Image

- **Schichtenarchitektur:**
  - *FROM, RUN, COPY, ADD, WORKDIR*  
→ **verändern Dateisystem**
- **Portabilität:**
  - on premise / cloud (kubernetes)
- **Versionierung:**
  - *tag: z.B. webapp:1.12.5*
- **Verteilung:** *Container Registry*

## **Auftrag**

Lesen Sie auf der Modulwebseite Woche 2

- **Docker Images**
  - Bis und Mit "**Dockerfile**"

-  Einzelarbeit
-  10 Min

# Dockerfile Demo

## **Auftrag**

Machen Sie die **Übung 1** unter Einfaches Dockerfile.

- Öffnen Sie den Ordner der Übung01
- Kopieren Sie die Lösung
- Passen Sie den Author an
- Bilden Sie das Docker Image
- Listen Sie das Image
- Erstellen Sie einen Container

## **Git Bash Befehle**

```
cd ../02-dockerimage/uebung01
cp ../solution/Dockerfile Dockerfile
code Dockerfile # öffnet VS-Code
docker buildx build -t uebung01 .
docker images
docker run uebung01
docker ps
```



# Warum jetzt Git für Docker?

- Git eignet sich für die Versionierung von Textfiles
- Das Dockerfile ist ein Textfile.
- Docker Images sind auch getagget 🤝.
- Es ist klar, welchen Code welches Image gebaut hat.
- Automatisierung braucht immer beschreibenden Code.

## **Auftrag**

Lesen Sie auf der Modulwebseite Woche 2



- Image builden
- Caching beim Erstellen von Docker-Images
- Multistage Builds

-  Einzelarbeit
-  10 Min

## **Auftrag**

Lösen Sie die weiteren Aufgaben von "Einfaches Dockerfile" und **mindestens eine** von "Multistage Dockerfile".

- Einfaches Dockerfile
  - Übungen 1 - 10
- Multistage Dockerfile
  - Übungen 11 - 12

-  Einzelarbeit
-  60 Min

# Hausaufgaben

- Üben, üben, üben.
- Es gibt genug Übungen.
- Sie können sie auch wiederholen! Das schadet nicht ;)