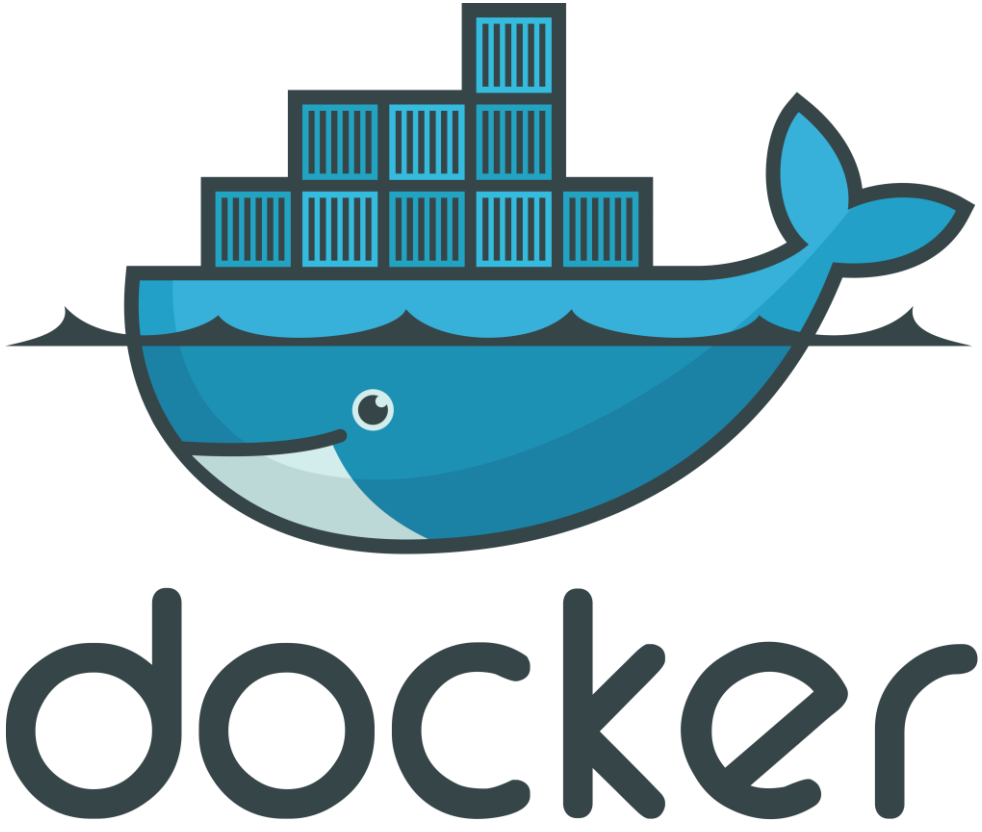
itelio

# Container / Docker

# Container vs Virtuelle Maschinen

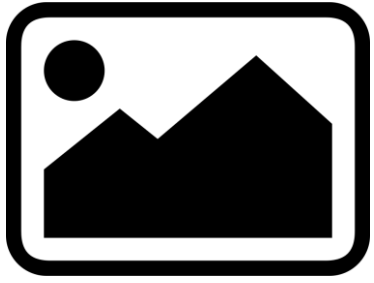
# Vorteile Container

- Wenig Overhead
- Sehr gute Portabilität
- Höhere Effizienz
- Bessere Developer-Experience
- Schnelle Startgeschwindigkeit
- Einfaches Dependencymanagement

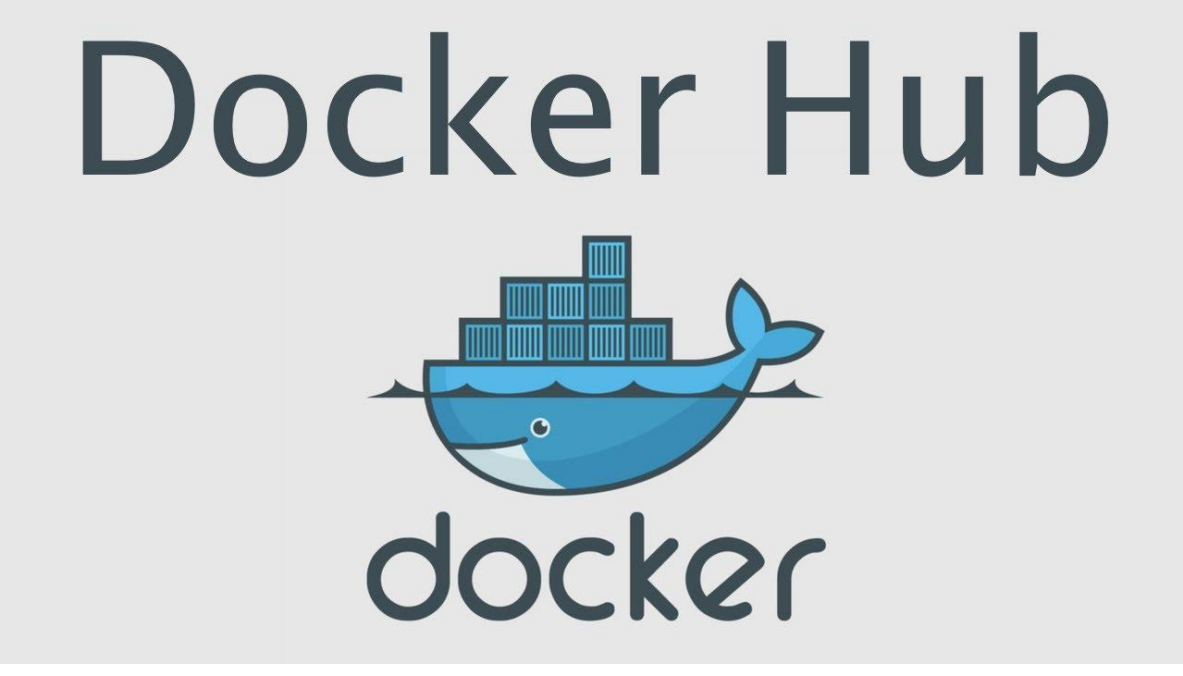

docker

# Begriffe

- Dockerfile
- Image
- Container
- Registry




# Dockerhub



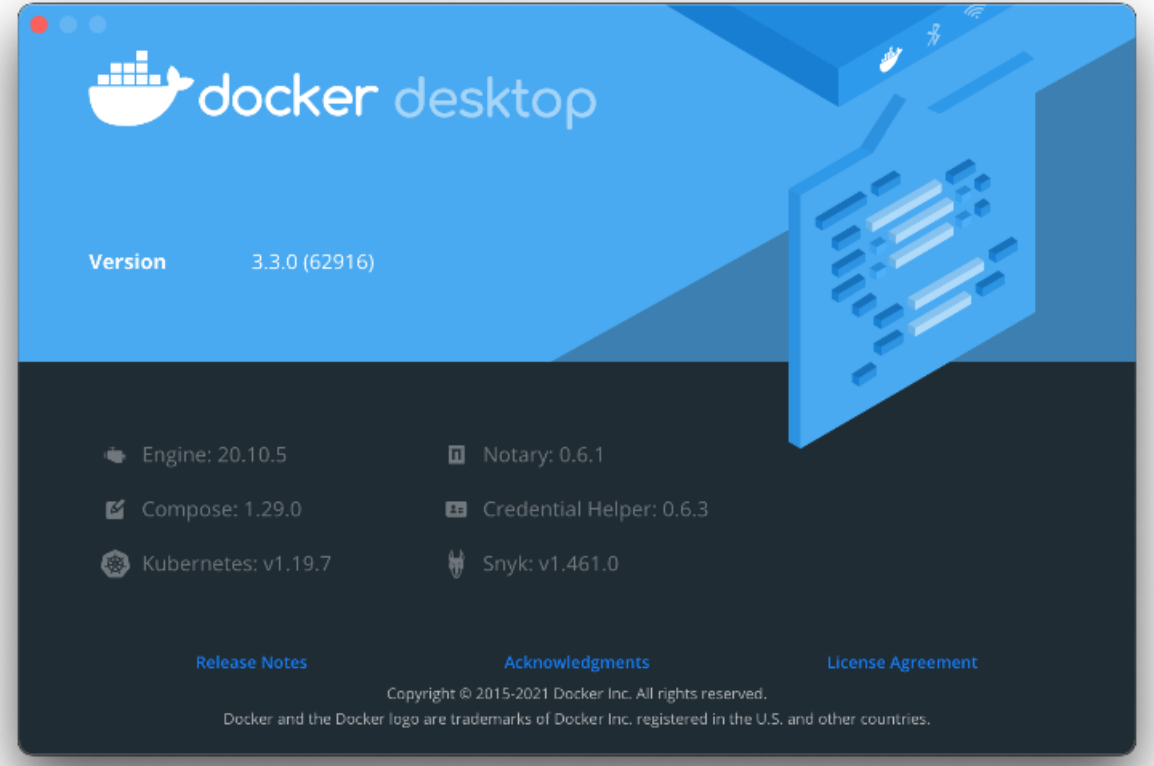
# Docker CLI

- docker ps [OPTIONS]
- docker pull [OPTIONS] NAME[:TAG|@DIGEST]
- docker run [OPTIONS] IMAGE [COMMAND] [ARG...]
- docker images [OPTIONS] [REPOSITORY[:TAG]]

itelio

- Übung 01 - Webserver

# Docker Desktop



# Dockerfile

```
# syntax=docker/dockerfile:1
FROM ubuntu:18.04
COPY . /app
RUN make /app
CMD python /app/app.py
```

- Übung 02 - Dockerfile

itelio

- Übung 03 - Dockerfile

# Storage in Docker

- Volumes
- Bind Mounts
- (Tmpfs Mounts)

# Volumes

- Docker managed
- Mehr Isolation vom Host
- Für Docker gemacht =>
  - Flexibeler
  - Können zwischen Containern geteilt werden

- Erstellen: docker volume create my_volume
- Benutzen: docker run -d --name my_container -v my_volume:/data my_image

itelio

- Übung 04 - Volumes

# Bind Mounts

- Dateisystemmount
- Kontrolle über den Pfad der Daten
- Weniger Sicherheitsüberprüfungen wie Volumes

- Benutzen: docker run -d --name my_container -v /host/data:/container/data my_image

itelio

- Übung 05 - Mounts

# Docker Compose

- Komplexe Apps
- Mehrere Container
- Einfache (wiederanwendbare) Konfiguration => yaml
- Development oder Singlehost Production Deployments

# Docker Compose

itelio

```yaml
version: '3.8'

services:
  web:
    image: nginx:latest
    ports:
      - "8080:80"
    volumes:
      - ./data:/usr/share/nginx/html
    networks:
      - frontend

  db:
    image: mysql:5.7
    environment:
      MYSQL_ROOT_PASSWORD: example
    volumes:
      - db_data:/var/lib/mysql
    networks:
      - backend

networks:
  frontend:
  backend:

volumes:
  db_data:
```

```yaml
version: '3.8'

services:
  web:
    build:
      context: ./app
    ports:
      - "5000:5000"
    environment:
      - REDIS_HOST=redis
    networks:
      - my_network

  redis:
    image: redis:alpine
    ports:
      - "6379:6379"
    networks:
      - my_network

networks:
  my_network:
```

itelio

- Übung 06 - Compose

# Docker Networks

- Erlaubt Kommunikation
- Verschiedene Driver
  - Bridge (default)
  - Host
  - None
- Commands
  - docker network create --driver <driver_name> <network_name>
  - docker network ls
  - docker network connect <network_name> <container_name>
- Auch in Compose nutzbar

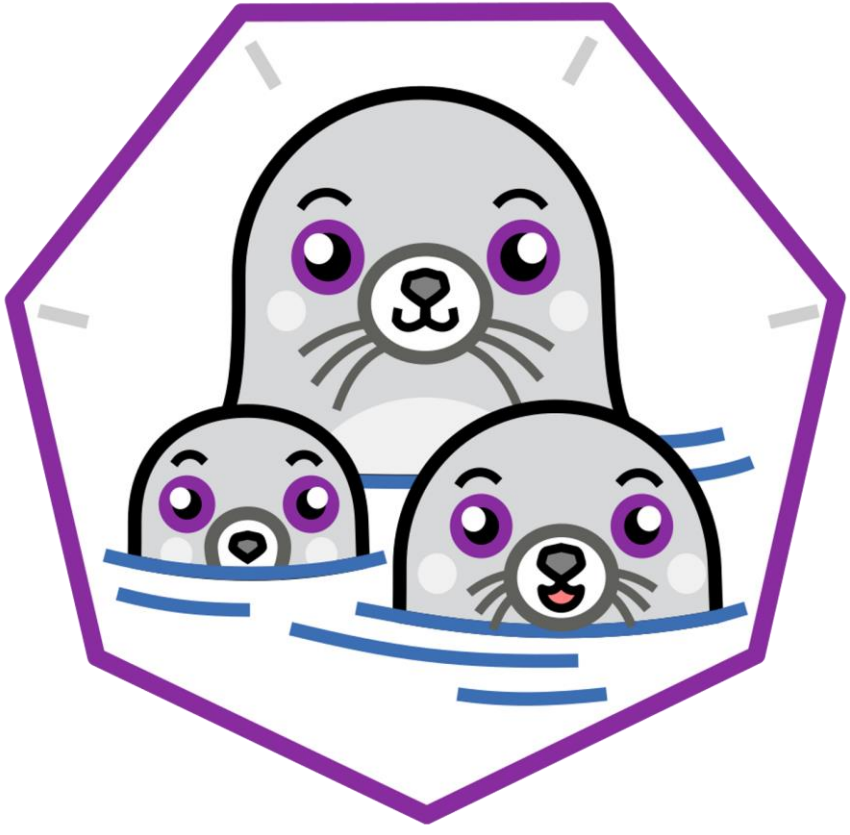
- Übung 07 - Networks

- Übung 08 – Freie Übung

# OCI

- Von Docker ausgehend 2015 ins Leben gerufen
- Ermöglicht Kompatibilität mit anderen Container Runtimes
- Von der Linux Foundation verwaltet
- Gesponsert von: AWS, Docker, Google, RedHat, Microsoft, Oracle…
- Ermöglicht das Ausführen eines Containers mit anderen Container Runtimes
- Nur eine Registry => Verschiedene Container in einer Registry

# Podman: Alternative zu Docker



**Erfahren Sie mehr unter www.itelio.com**



itelio GmbH
Franz-Larcher-Str. 4
D-83088 Kiefersfelden
Fon +49-8033-6978- 0
Fax +49-8033-6978-91
info@itelio.com | www.itelio.com