



# Workshop Docker – Teil 1

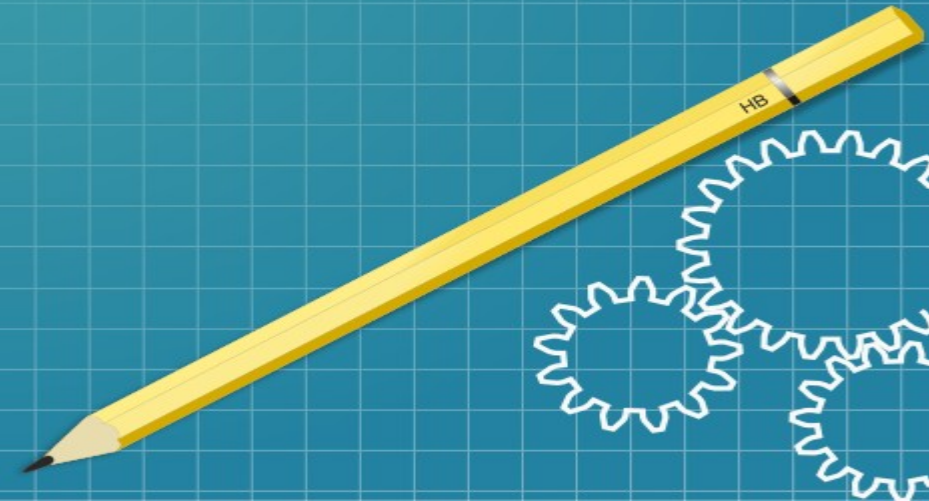
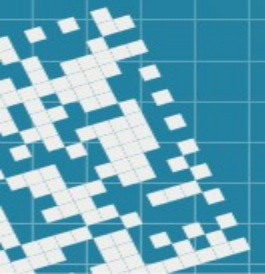
UN-Hack-Bar e.V.

# Ablauf



- Theorie und Wissenswertes
  - Was sind Container
  - Was braucht man für einen Dienst
  - Docker Umgebungsvariablen
  - Docker Volumes
  - Docker Netzwerke
  - Docker Compose
  - Reverse Proxy
- Handarbeit!
  - VM updaten
  - Docker installieren
  - Reverse Proxy installieren
  - Let's encrypt installieren
  - Ggfs. Watchtower installieren

# Wissenswertes



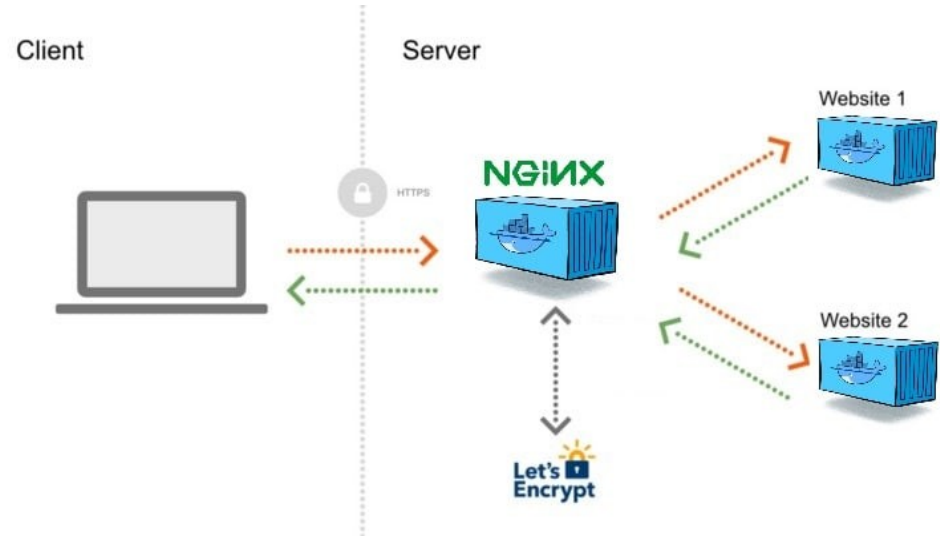
# Docker, Container, was?



- Begriffe: Image und Container:
  - Image: Unveränderbarer „Snapshot“ eines Dateisystem
  - Container: Dateisystem das auf einem Image basiert aber änderbar ist (Stichwort: „Overlay filesystem“)
  - Aktuell vor allem OCI format („Open Container Initiative“)
- docker vs. kubernetes vs. podman
  - Alle drei nutzen sind ein CRI („Container Runtime Interface“)
  - Wir nutzen Heute docker
  - Spannende alternative: podman
  - Kubernetes vor allem in großen Umgebungen (→ Cluster)

# Übliche Architektur

- Proxy-Container bindet auf Port 80 und 443
- Weiterer Container kümmert sich um Let's encrypt
- Alle andere Docker-Container sind nur im Docker Netz verfügbar (und werden von nginx angesprochen)



# Reverse Proxy



- Viele Dienste wollen Port 80/HTTP bzw. 443/HTTPS nutzen
- Problem: Nur ein Prozess darf die Ports nutzen
- Lösung: HTTP/HTTPS Reverse Proxy nimmt entgegen und leitet Anfragen anhand der Domain an die Container weiter (wird dann auch „Edge Router“ genannt)
- Verbreitete Container dafür:
  - [nginxproxy/nginx-proxy](#) + [nginxproxy/acme-companion](#)
  - [traefik](#)

# Docker Umgebungsvariablen



- Wir wissen: Images sind nicht veränderbar
- Problem: Wie kann man sie dann sinnvoll konfigurieren?
- Lösung: Umgebungsvariablen

HOSTNAME: `www.un-hack-bar`, `www.unhb.de`

# Docker Volumes



- Wir wissen: Images sind nicht veränderbar
- Problem: Wie können wir Daten dauerhaft speichern?

- Lösung: Volumes

```
volume: /srv/docker-volumes/db:/opt/db
```

(Auf dem Docker-Host)

(im Container)

- Mehrere Dateisystem „übereinander“
  - Image → Container → Volume



# Docker Netzwerke



- Wir wissen: Container müssen miteinander sprechen
- Problem: Wie legen wir fest wer mit wem sprechen darf?
- Lösung: Netzwerke
- networks :  
    wordpress\_net :
- Ein Container kann in einem (default) oder mehreren unterschiedlichen Netzwerken sein

# Docker Compose

- Wir wissen: Es muss viel pro Container eingestellt werden
- Vorige Daten aber auch zB. Datenbank-Zugangsdaten
- Problem: Wie kann man das übersichtlich erledigen?
- Lösung: docker compose

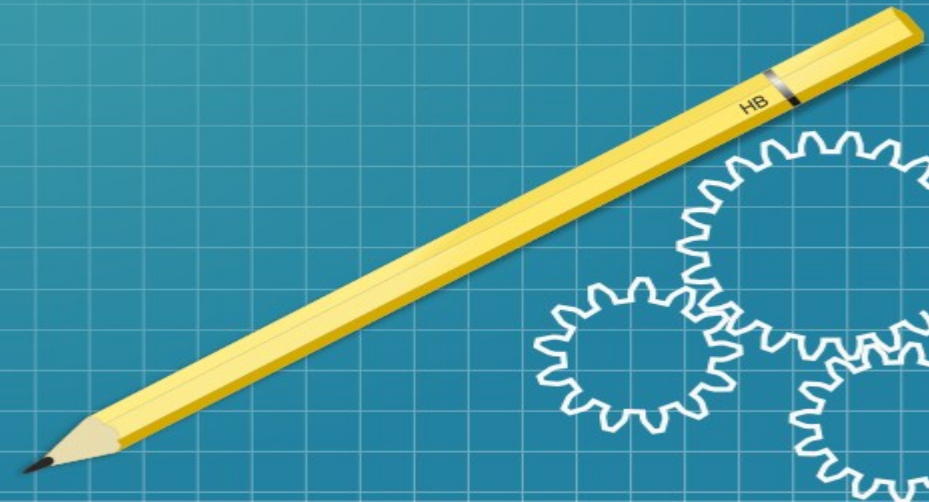
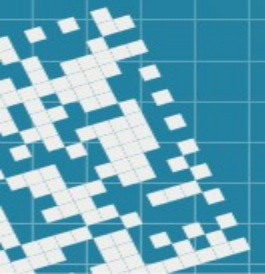
```
version: '3.5'
services:
  wordpress:
    image: wordpress:6.1.1-php8.1-apache
    container_name: wordpress.server4
    restart: always
    networks:
      wordpress_net:
      nginx_net:
    environment:
      WORDPRESS_DB_HOST: wordpress_sql.server4
      WORDPRESS_DB_USER: user_unhb_blog
      WORDPRESS_DB_PASSWORD: **geheim**
      WORDPRESS_DB_NAME: db_unhb_blog
      VIRTUAL_PORT: 80
      VIRTUAL_HOST: un-hack-bar.de,www.un-hack-bar.de,unhb.de,www.unhb.de
      LETSENCRYPT_HOST: un-hack-bar.de,www.un-hack-bar.de,unhb.de,www.unhb.de
      LETSENCRYPT_EMAIL: letsencrypt@un-hack-bar.de
    volumes:
      - /srv/docker-volumes/wordpress/html:/var/www/html

wordpress_db: [...]

networks:
  wordpress_net:
    external: false
    name: wordpress_net
  nginx_net:
    external: true
    name: nginx_net
```



# Handarbeit



# VM aktualisieren



## 1) Per SSH auf eure VM einloggen

- `sudo -i`
- `apt-get update`
- `apt-get dist-upgrade`
- `apt-get autoremove`
- `shutdown -r now`

# Docker installieren



## 1) Per SSH auf eure VM einloggen

- Sources updaten
- Siehe <https://docs.docker.com/engine/install/debian/>
- `apt-get update`
- `apt-get install docker-ce docker-ce-cli containerd.io  
docker-buildx-plugin docker-compose-plugin`
- `docker run hello-world`
-

# Exkurs: hilfreiche Befehle



## 1) Alle Container anzeigen

➤ `docker ps`

## 2) Shell in einem Container öffnen

➤ `docker exec -it containername /bin/bash`

## 3) Log eines Container anzeigen

➤ `docker logs containername`

# Reverse Proxy installieren (1)



## 1) Docker-Compose Datei anlegen

- `mkdir -p /srv/docker-config/proxy`
- `cd /srv/docker-config/proxy`
- `nano/vi/emacs docker-compose.yaml`
- [Datei auf der nächste Folie]

## 2) Compose nutzen um Container anzulegen

- `docker compose up -d`

# Reverse Proxy installieren (2)



version: '3.5'

<https://pad.unhb.de/p/docker-1>

services:

nginx-proxy:

container\_name: nginx-proxy

image: jwilder/nginx-proxy:alpine

restart: unless-stopped

ports:

- "80:80/tcp"
- "443:443/tcp"

networks:

- nginx\_net

volumes:

- /var/run/docker.sock:/tmp/docker.sock:ro
- /srv/docker-volumes/nginx/conf.d:/etc/nginx/conf.d
- /srv/docker-volumes/nginx/vhost.d:/etc/nginx/vhost.d
- /srv/docker-volumes/nginx/certs:/etc/nginx/certs
- /srv/docker-volumes/nginx/html:/usr/share/nginx/html

environment:

- DEFAULT\_EMAIL=accounts@un-hack-bar.de
- DEFAULT\_HOST=unhb.de

nginx-proxy-letsencrypt:

container\_name: nginx-proxy-le

image: nginxproxy/acme-companion

networks:

- nginx\_net

restart: always

volumes:

- /var/run/docker.sock:/var/run/docker.sock:ro
- /srv/docker-volumes/nginx/vhost.d:/etc/nginx/vhost.d
- /srv/docker-volumes/nginx/certs:/etc/nginx/certs
- /srv/docker-volumes/nginx/html:/usr/share/nginx/html
- /srv/docker-volumes/nginx/acme.sh:/etc/acme.sh

environment:

- DEFAULT\_EMAIL=accounts@un-hack-bar.de
- CERTS\_UPDATE\_INTERVAL=21600
- NGINX\_PROXY\_CONTAINER=nginx-proxy

networks:

nginx\_net:

name: nginx\_net



# Watchtower installieren



## 1) Docker compose Datei anlegen

- `mkdir -p /srv/docker-config/watchtower`
- `cd /srv/docker-config/watchtower`
- `nano/vi/emacs docker-compose.yaml`
- Datei siehe <https://pad.unhb.de/p/docker-2>

## 2) Docker Compose nutzen

- `docker compose up -d`

Watchtower prüft regelmässig ob für die genutzten Container neuere Versionen vorhanden sind und installiert sie dann automatisch. Details dazu nächste mal.



# Fertig für Heute

:-D



This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License. It makes use of the works of Mateus Machado Luna.

