Docker Einführung

Docker installieren:

```
curl -fsSL https://get.docker.com -o get-docker.sh
sudo sh ./get-docker.sh
```

Falls kein curl installiert: `apt install curl`

Wenn man nicht mit dem Nutzer Root arbeitet, sollte man den aktuellen Benutzer berechtigen:

```
sudo usermod -aG docker $USER
```

Mit Docker arbeiten

- Laufende Container auflisten: `docker ps`
- Alle Container auflisten (auch gestoppte): `docker ps -a`
- Einen Container anhalten: `docker stop <Containername>` (den Namen findet man mit `docker ps heraus`)
- Einen gestoppten Container endgültig löschen: `docker rm <Containername>`

Einen simplen Webserver starten

Der Container aus dem Image nginx fährt mit folgendem Befehl hoch:

```
docker run -p 80:80 nginx
```

Die eigene IP-Adresse erhält man mit `ip a`

Arbeiten mit Docker-Compose

Legt euch am besten einen eigenen Ordner für das Docker-Projekt an, um Ordnung zu halten. Die Datei docker-compose.yml enthält die Definition der Container.

Bearbeitet wird die Datei mit:

```
nano docker-compose.yml
```

Die Inhalte findet ihr unten in diesem GitHub-Gist. Den Texteditor Nano beendet man mit: `Strg+X`, dann `Y`

Die Compose-Zusammenstellung hochfahren:

```
docker-compose up -d
```

Will man die Container updaten, lädt man die neuen Images mit

```
docker-compose pull
```

Eine oder mehrere Docker-Compose-Dateien?

Das ist definitiv Geschmachssache und hängt von der Umgebung ab. Wenn man mehr als ein Projekt (zum Beispiel einen Blog und ein Pihole) auf einem Server betreibt, sollte man für jedes einen Ordner anlegen und darin eine Docker-Compose-Datei ablegen. Die nützlichen Helfer wie Portainer und Watchtower kommen zusammen in eine weitere Datei. Dann kann man mit `docker compose down` gezielt Teile der Umgebung herunterfahren.

```
# Die Docker-Compose-Zusammenstellung für Pihole (https://hub.docker.com/r/pihole/pihole)
services:
  pihole:
    container_name: pihole
    image: pihole/pihole:latest
    ports:
      - "53:53/tcp"
      - "53:53/udp"
      - "67:67/udp"
      - "80:80/tcp"
    environment:
      TZ: 'Europe/Berlin'
```

```
    # WEBPASSWORD: 'set a secure password here or it will be random'
  volumes:
    - './etc-pihole:/etc/pihole'
    - './etc-dnsmasq.d:/etc/dnsmasq.d'
  cap_add:
    - NET_ADMIN
  restart: unless-stopped
```

```
# Docker-Compose-Datei für Portainer (https://hub.docker.com/r/portainer/portainer-ce). Antwortet auf Port 9000 des Servers
services:
  portainer:
    image: portainer/portainer-ce
    ports:
     - 9000:9000
    volumes:
     - /var/run/docker.sock:/var/run/docker.sock
     - ./portainer_data:/data
    restart: always
```

```
# Docker-Compose-Datei für Watchtower (https://hub.docker.com/r/containrrr/watchtower)
services:
  watchtower:
    image: containrrr/watchtower
    volumes:
     - /var/run/docker.sock:/var/run/docker.sock
```

Revision #1
Created 17 October 2024 12:30:19 by Hermann
Updated 17 October 2024 12:32:37 by Hermann