Zürcher Hochschule
für Angewandte Wissenschaften

School of
Management and Law

AACSB ACCREDITED
Seit 2015

DevOps
Container Virtualisierung

Building Competence. Crossing Borders.

Adrian Moser
mosa@zhaw.ch, FS2024

Docker und Docker Compose

Wir verlassen das Thema «Dev» und bewegen und Richtung «Ops».

Docker

Ziel

Docker ist eine Open-Source Software zur Isolierung von Anwendungen mit Containervirtualisierung.

Idee

Docker vereinfacht die Bereitstellung von Anwendungen, weil sich Container, die alle nötigen Pakete enthalten, leicht als Dateien transportieren und installieren lassen.

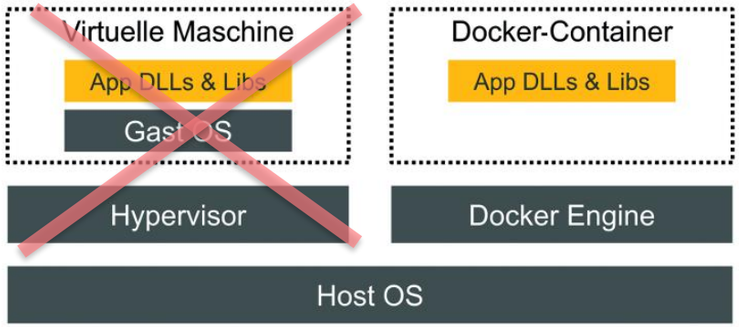
Docker Konzepte

Docker Konzepte

- Umgebung um Prozesse zu starten
- Saubere Trennung von Umgebungen
- Sandbox mit Zugriff auf definierte Ressourcen
- Einfaches Interface um Applikationen zu starten

Was Docker NICHT ist

- (Fast) keine Virtualisierung
  (von Prozessoren, z.B. x86, ARM)
- Keine getrennte Kernel
- Kein Hypervisor (Virtual-Machine-Monitor,
  Gast-OS auf fremdem OS)



Kubernetes und Docker Swarm

Kubernetes, oder k8s (k - 8 Buchstaben – s, alles klar?), oder “Kube” als Abkürzung, ist eine Open-Source Plattform die Linux Container Operationen automatisieren kann.

Anders gesagt, man kann Gruppen von Linux Containern zusammenfassen und **Kubernetes** hilft diese zu verwalten.

Docker Swarm ist eine Alternative zu Kubernetes. Im Unterricht brauchen wir nur Docker und weder k8s noch Swarm.

Begriffe: Image und Container

Docker Image

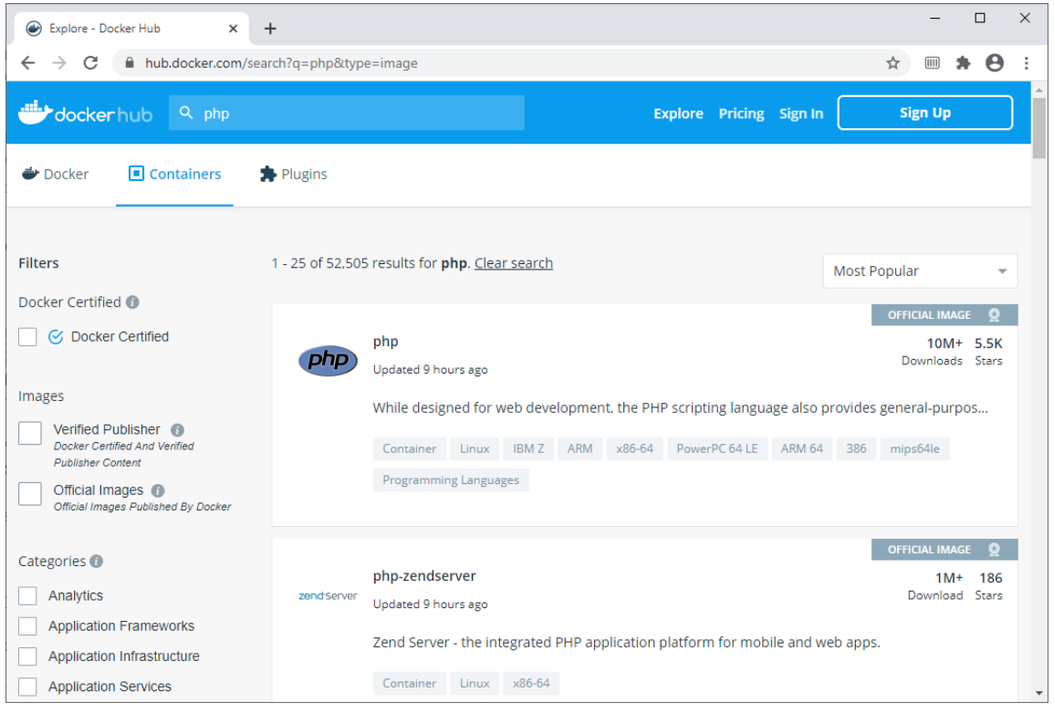
- Ein unveränderbares Template für Container
- Kann von einer Registry geladen bzw. dort gespeichert werden (z.B. Docker Hub)
- Image Namen haben die Form [registry/][user/]name[:tag]
- Der Standard-Tag ist latest
- Architektur-Support beachten: amd64 (PC, Intel Mac), arm64 (M1 Mac)

Docker Container

- Eine Instanz eines Image
- Lässt sich starten, stoppen, neustarten, …
- Verwaltet Änderungen auf dem File System
- Neues Image könnte vom aktuellen Container-Stand gemacht werden (nicht empfohlen, stattdessen Dockerfile verwenden)

Docker Hub

Docker Hub ist ein Verzeichnis von Docker Images. Mittlerweile gibt es für praktisch jede Software ein Docker Image.



https://hub.docker.com/

Ausgewählte Docker Befehle

Docker

docker version	Docker Version
docker ps	Zeigt alle laufenden Container
docker ps –a	Zeigt alle Container

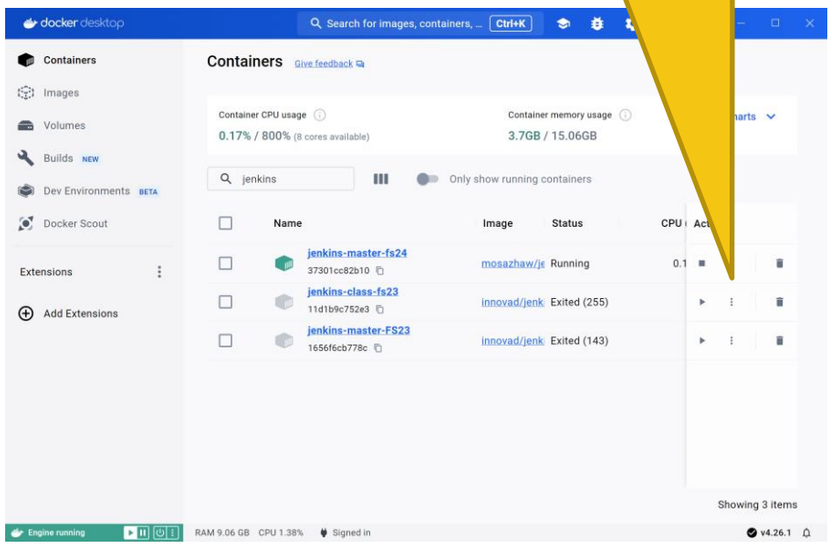
Docker Hub / Images

docker pull XYZ	Ein Image von DockerHub laden
docker image ls	Alle geladenen Images auflisten
docker image rm ID	Image löschen
docker image prune -a	alle Images aufräumen

Docker Container

docker container ls

Auch über das Docker Dashboard lassen sich immer mehr Befehle ausführen



Docker Tutorial

Mediawiki installieren (Wikipedia-Software)

Docker Bedienung

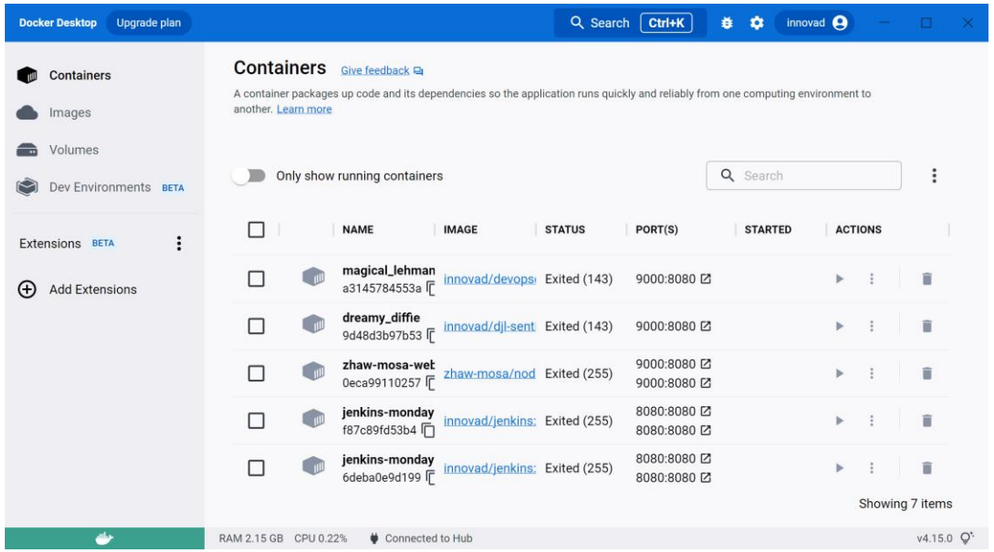
Tutorial

Docker

Die Bedienung erfolgt grundsätzlich über die Konsole (Windows cmd) oder Terminal (macOS).

Dashboard

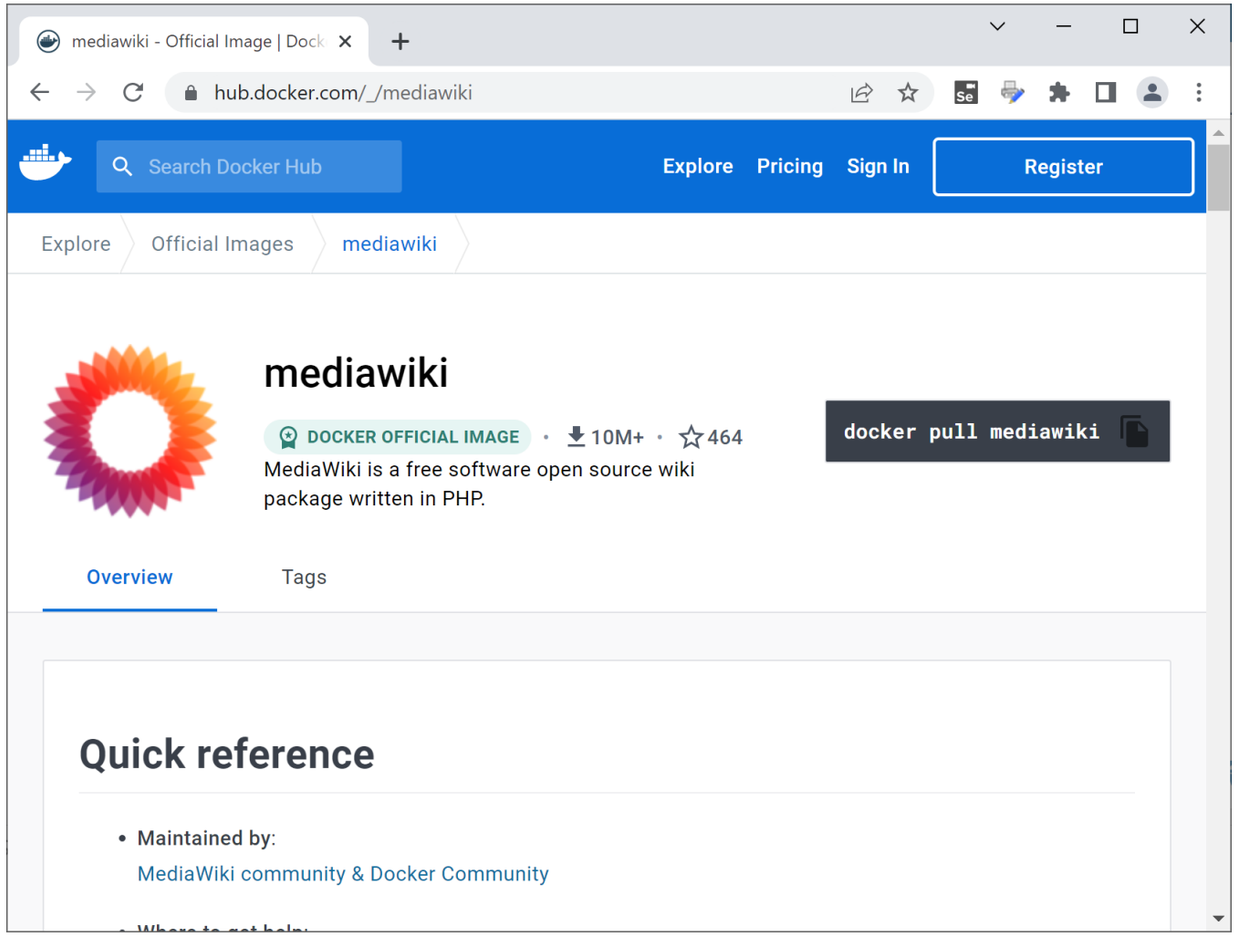
Das Dashboard bietet eine Übersicht sowie die Ausführung wichtiger Befehle.



Passendes Image suchen

Tutorial

https://hub.docker.com/_/mediawiki: Docker Pull Command für das gewünschte Image kopieren



docker pull und run

Tutorial

pull

docker pull mediawiki

1. Pull ausführen (mit Tag)

Mit **tag** kann gewünschte Version geladen werden

z.B. docker pull mediawiki:1.41

run

docker run --name some-mediawiki -p 8080:80 -d mediawiki:1.41

-d detached (nicht im Vordergrund)

image

Port des Docker Host

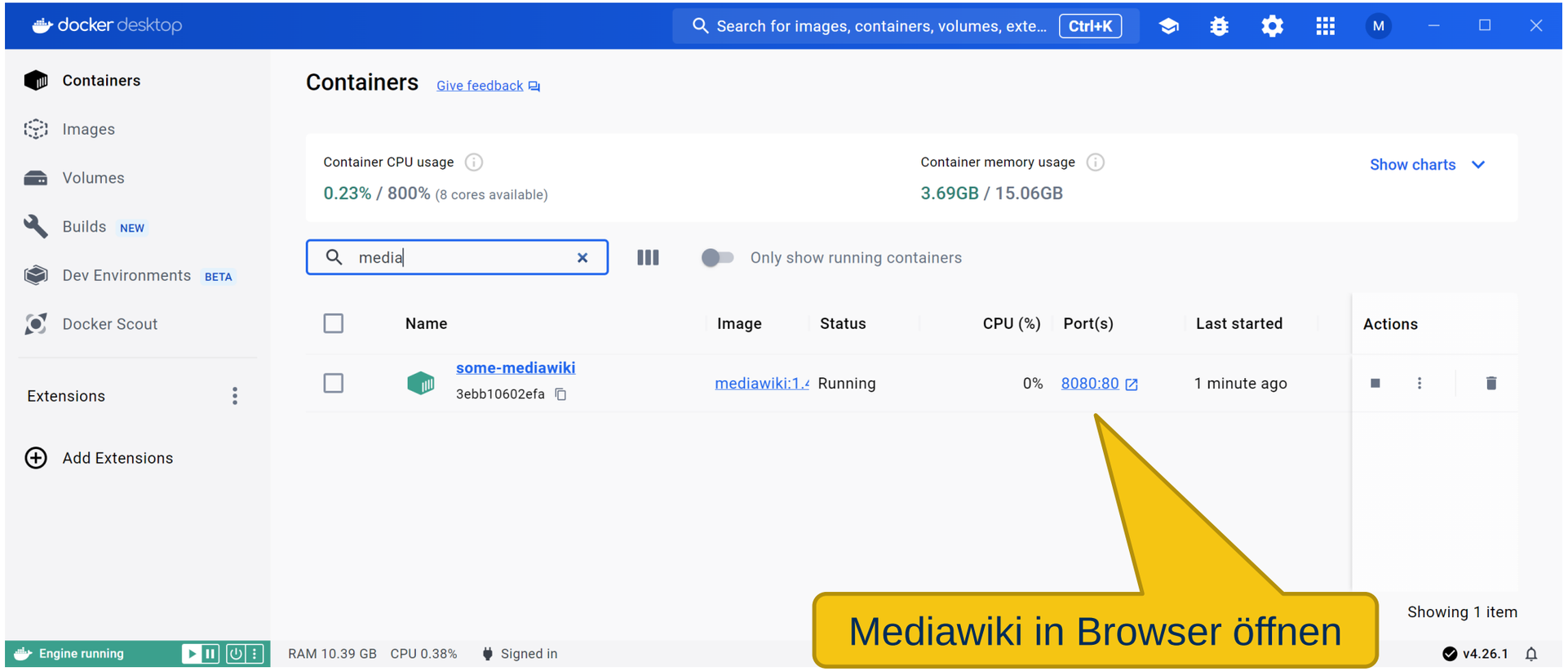
Interner Port innerhalb Container

2. Starten

Dashboard

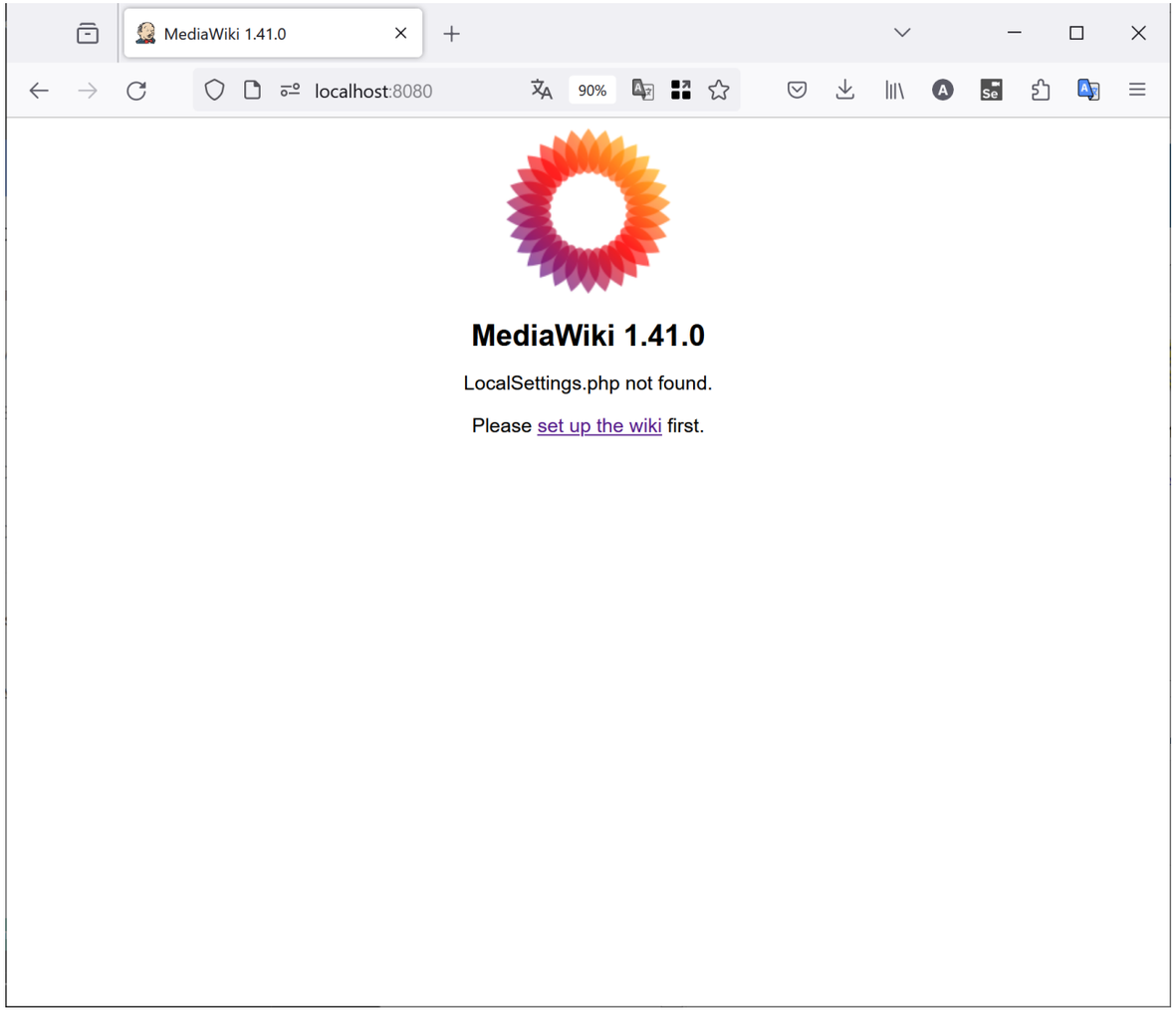
Tutorial

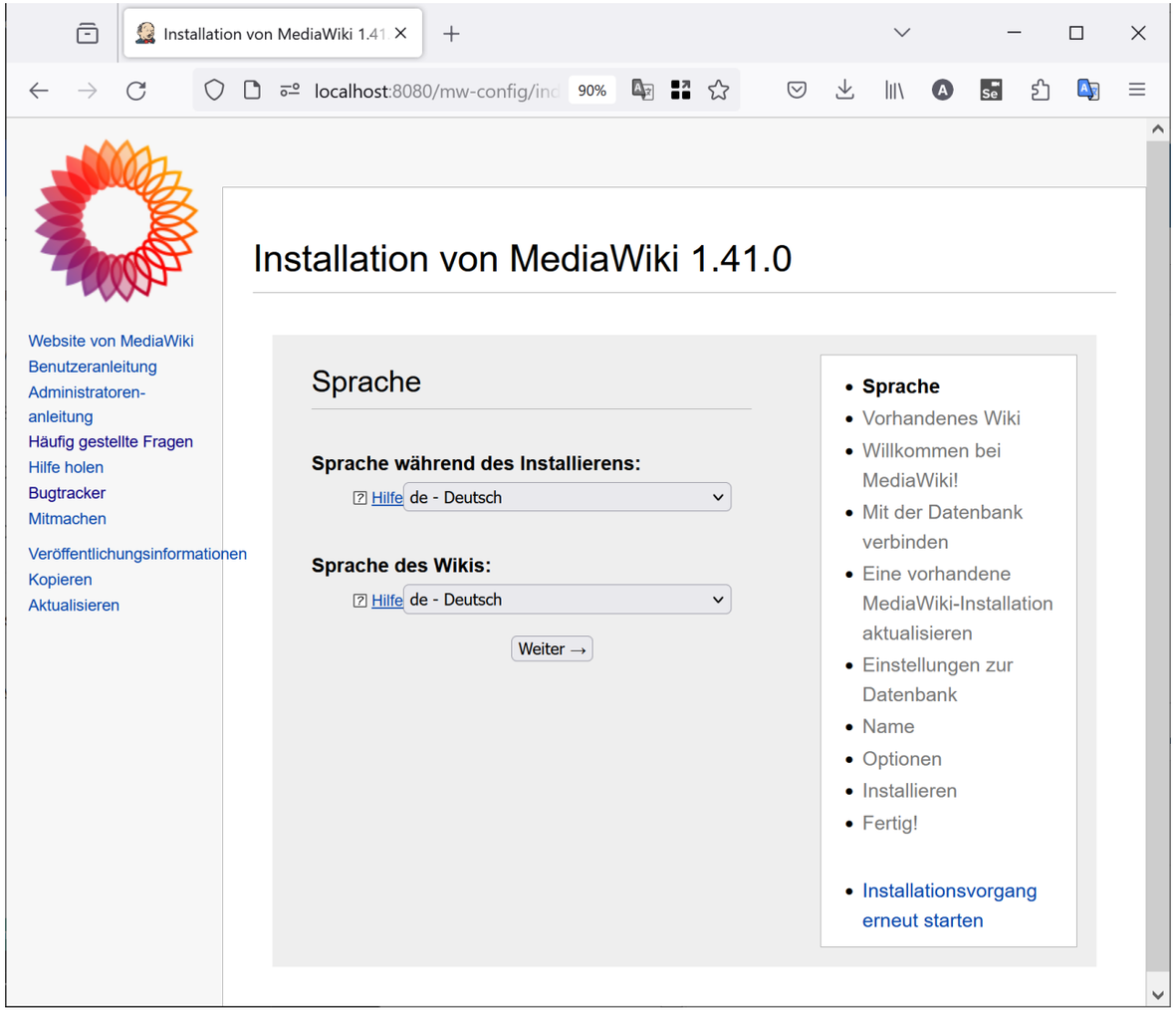
Nun sollte der Container im Dashboard erscheinen



Mediawiki: Up and running

Tutorial





Docker Tutorial «mehrere Container»

Mediawiki nutzt integrierte Datenbank

Typischerweise aber mehrere «Server», z.B. Applikation und Server

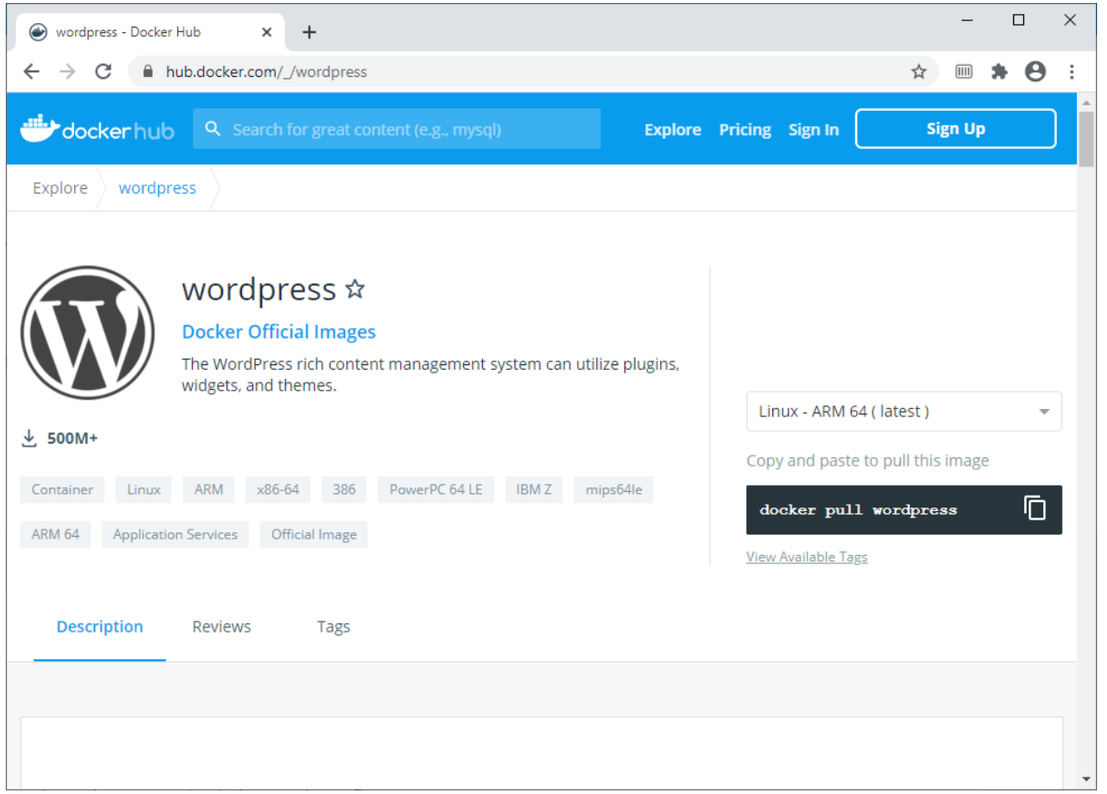
Mehrere Docker Instances

Netzwerke

Wordpress Image

Tutorial

Wir möchten Wordpress (**wordpress:6.4.2-php8.3**) installieren und brauchen dazu eine Datenbank (**mysql:8.2.0**)



MySQL Datenbank installieren & starten

Tutorial

MySQL

docker run --name local-mysql -e MYSQL_ROOT_PASSWORD=12345 -e MYSQL_DATABASE=wordpress -e MYSQL_USER=wordpress -e MYSQL_PASSWORD=wordpress -d mysql:8.2.0

Pull wird automatisch gemacht

Der Container legt beim Starten automatisch eine Datenbank mit dem Namen wordpress und dem Benutzer/Passwort wordpress/wordpress an. Diese Werte können konfiguriert werden.

→ wenn image nicht verfügbar ist, dann erfolgt **pull** automatisch

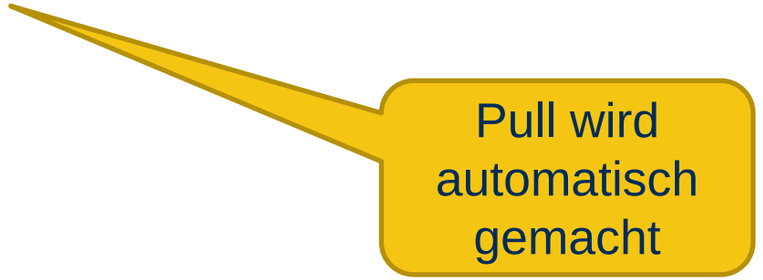
→ Image verlangt das Setzen diverser Parameter

Wordpress installieren & starten

Tutorial

Wordpress

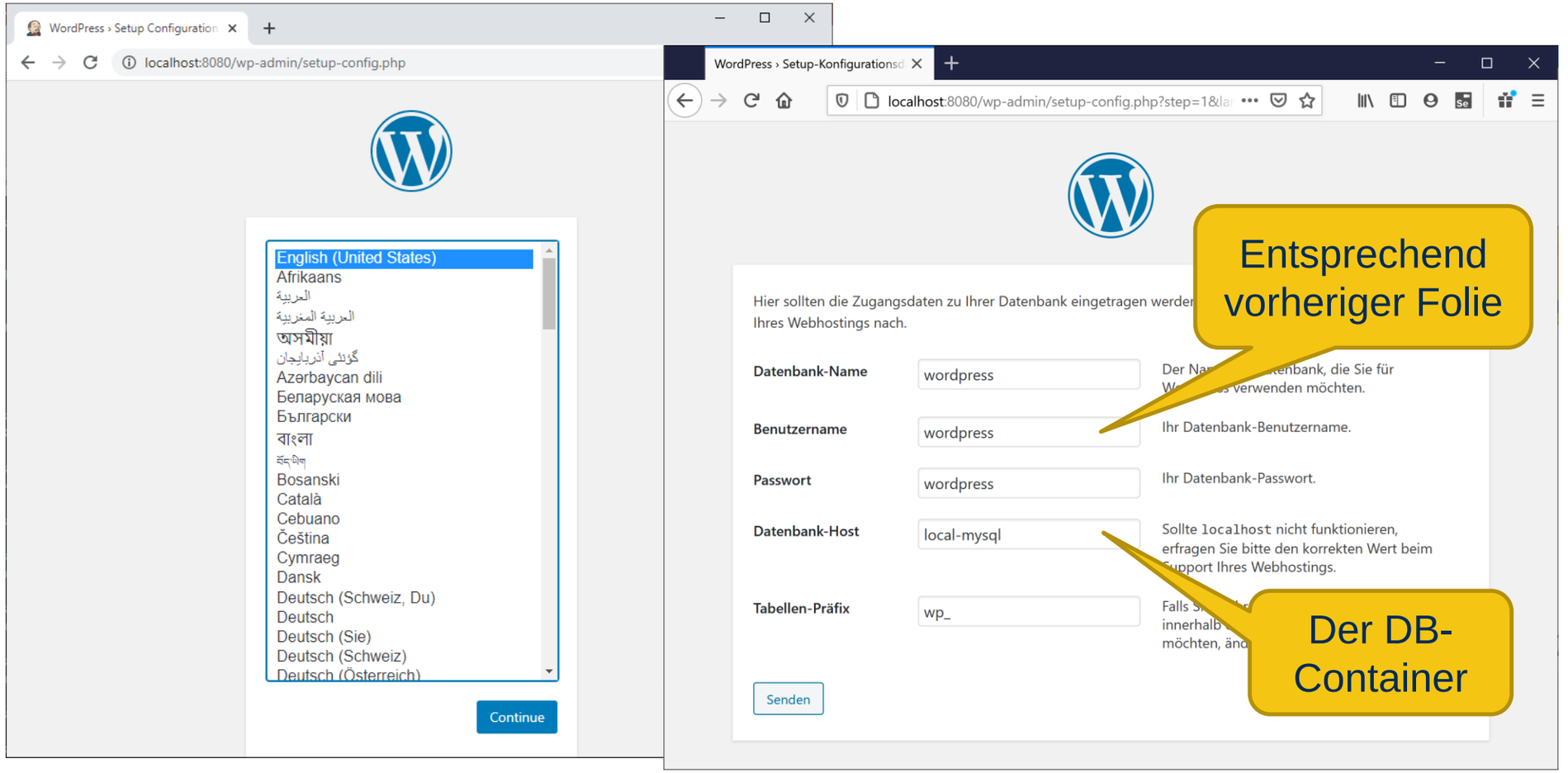
docker run --name local-wordpress -p 8080:80 -d wordpress:6.4.2-php8.3



Wordpress im Browser öffnen

Tutorial

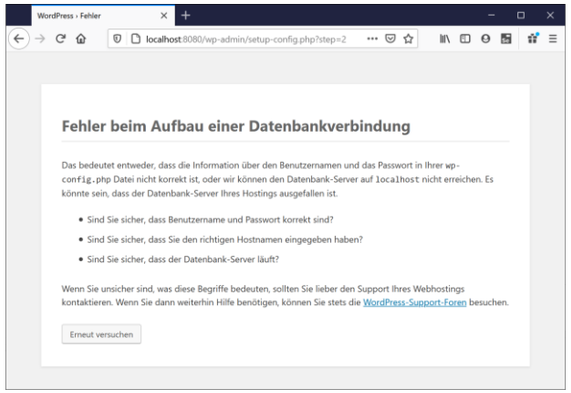
http://localhost:8080 öffnen und die Datenbank konfigurieren, aber …



Docker Networks

Tutorial

Die beiden Container können nicht miteinander kommunizieren.



Virtuelles Netzwerk erstellen

docker network create --attachable wordpress-network

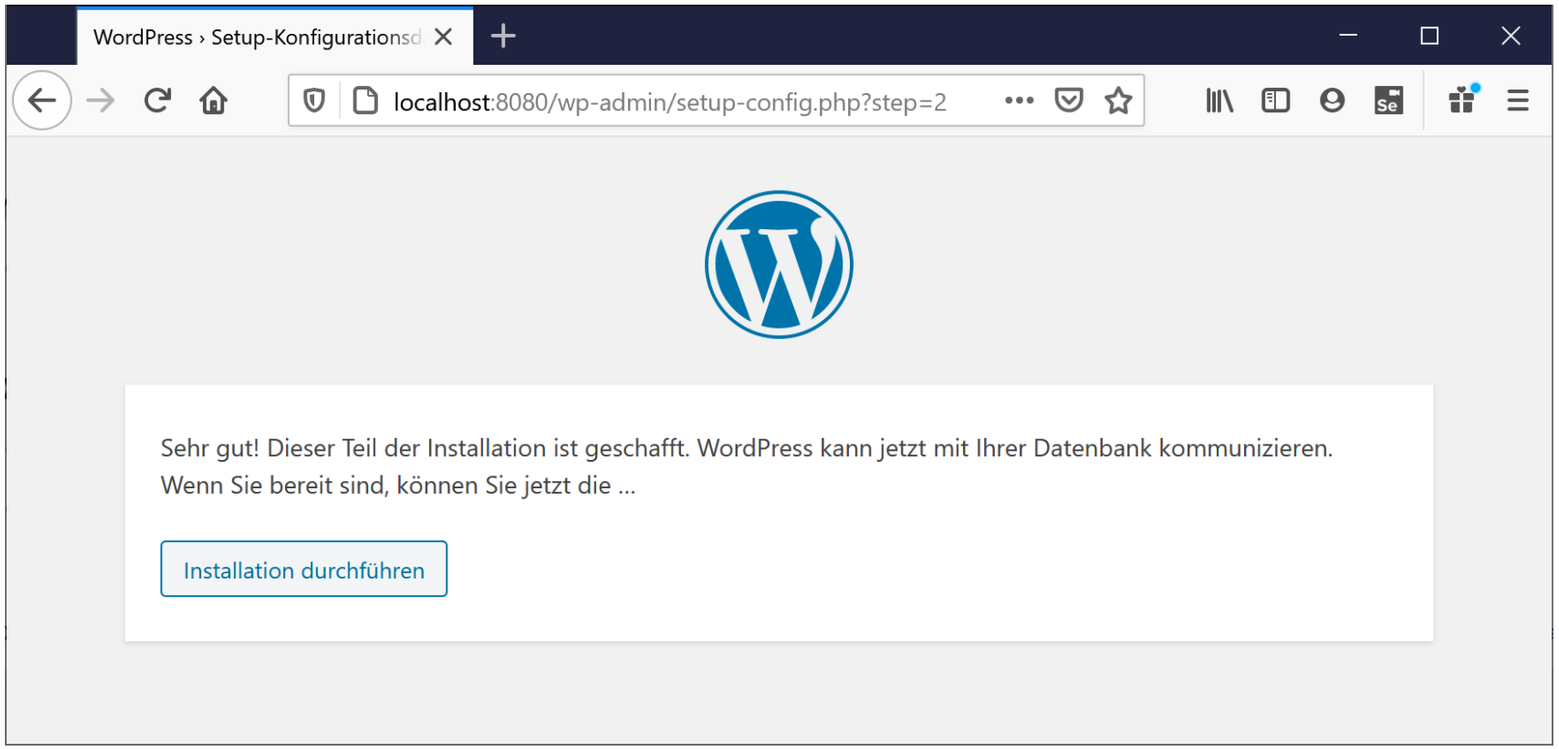
Container zu Netzwerk hinzufügen

docker network connect wordpress-network local-mysql

docker network connect wordpress-network local-wordpress

Tutorial

Erneut Passwort eingeben und der Datenbank-Zugang sollte funktionieren:



Fazit

Mehrere Container

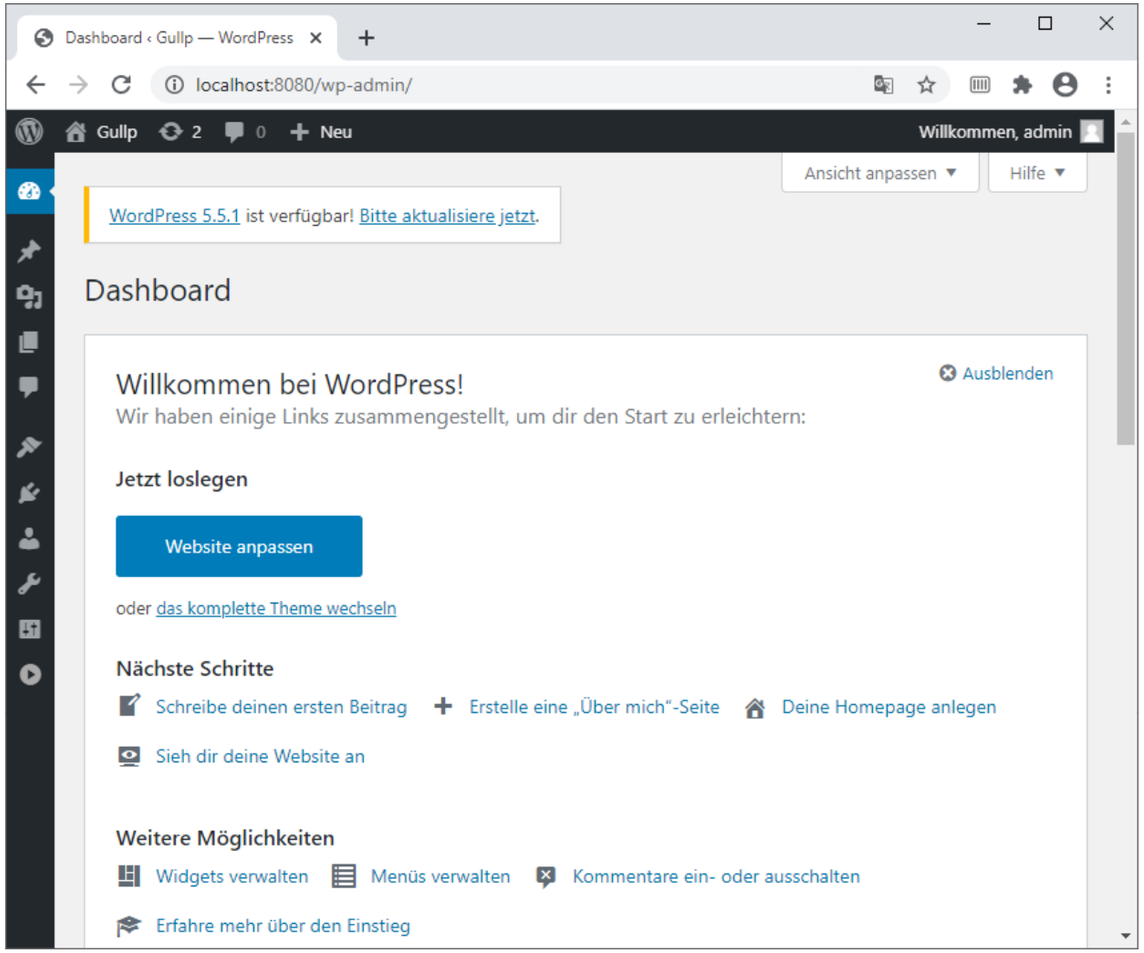
Die meisten Webapps bestehen aus mehreren Containern (z.B. Datenbank und Applikation)

Netzwerke

Docker unterstützt Netzwerke

Konfiguration

Die Konfiguration des Netzwerkes und der beiden Container kann aufwändig und fehleranfällig sein

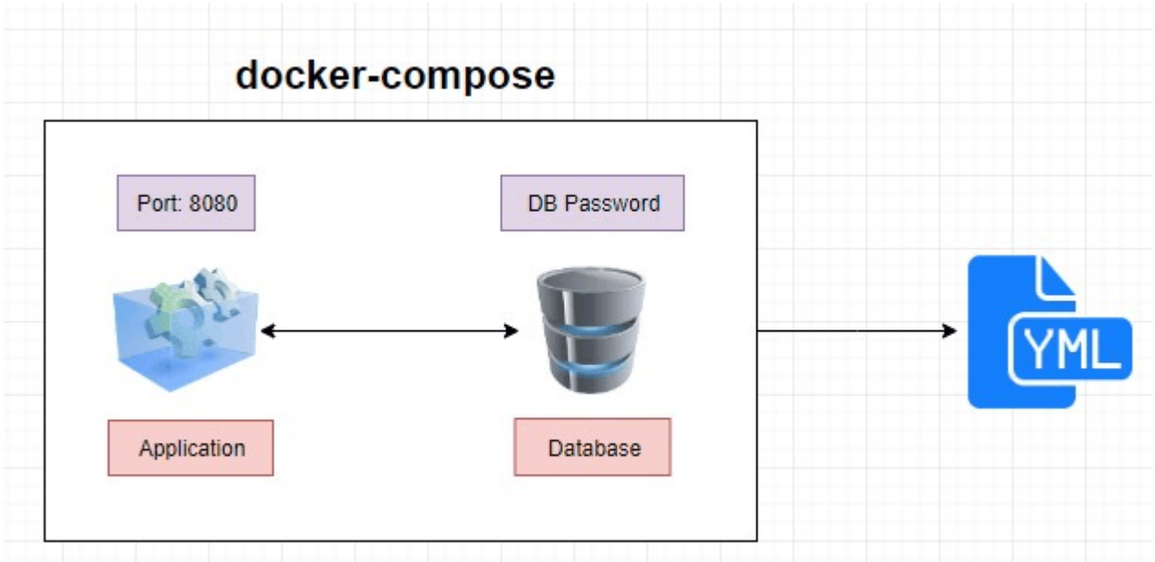


Docker Compose

Für Multi-Container Applications

Idee

Konfiguration mehrerer Container zusammenfassen



Docker Compose File

Tutorial

Docker Compose

Compose ist ein Tool welches Multi-Container Docker Applikationen definieren und ausführen kann.

YAML

Die Services werden in einem YAML-File beschrieben (**docker-compose.yml**).
Mit einem einzigen Befehl können alle Services gestartet werden.

Starten (im Verzeichnis)

docker-compose up -d

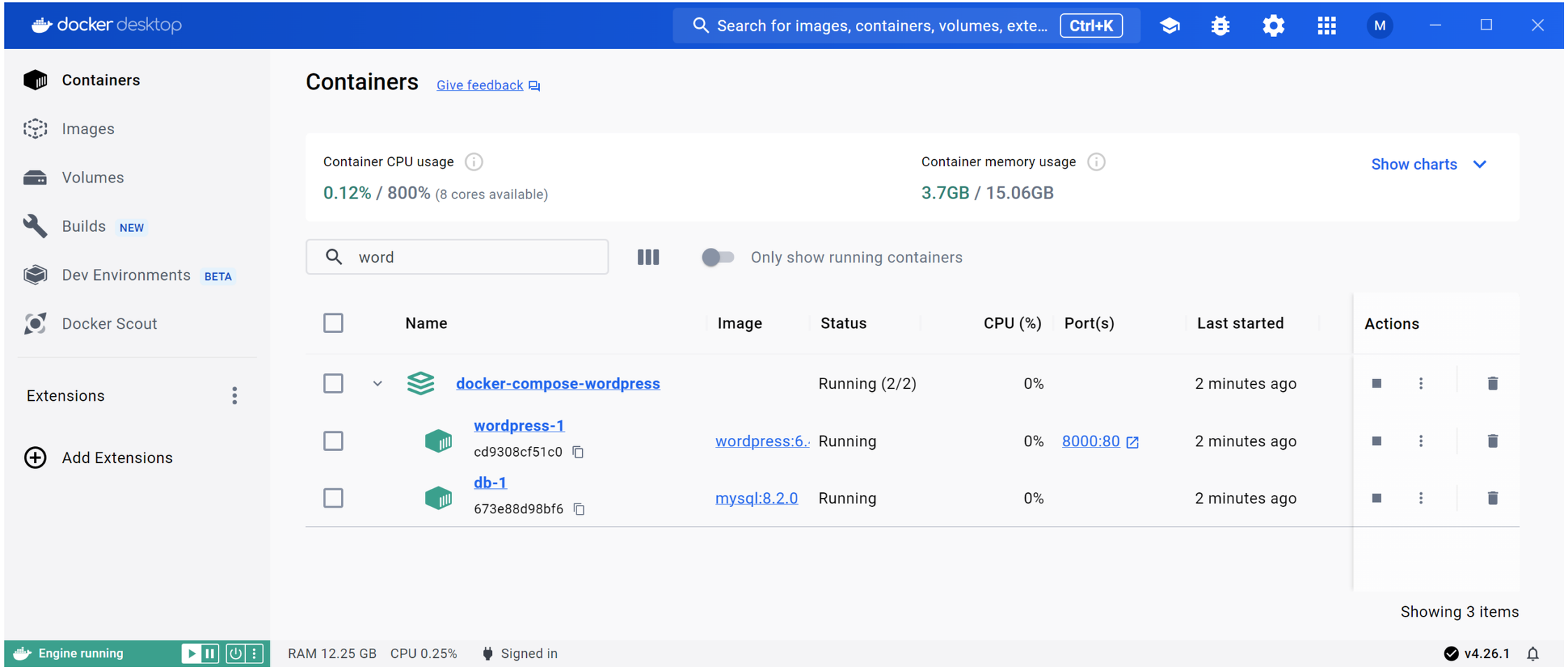
```yaml
version: '3.8'

services:
   db:
     image: mysql:8.2.0
     volumes:
       - db_data:/var/lib/mysql
     restart: always
     environment:
       MYSQL_ROOT_PASSWORD: somewordpress
       MYSQL_DATABASE: wordpress
       MYSQL_USER: wordpress
       MYSQL_PASSWORD: wordpress

   wordpress:
     depends_on:
       - db
     image: wordpress:6.4.2-php8.3
     ports:
       - "8000:80"
     restart: always
     environment:
       WORDPRESS_DB_HOST: db:3306
       WORDPRESS_DB_USER: wordpress
       WORDPRESS_DB_PASSWORD: wordpress
       WORDPRESS_DB_NAME: wordpress
volumes:
    db_data: {}
```

Docker Compose File

Tutorial



Fazit

Docker Compose

Mehrere Container auf einen Docker Host deployen

Docker Swarm (kein Thema dieser Vorlesung)

Container auf mehrere Docker Hosts deployen

Lernjournal

Lernjournal «Docker»

Ziele

- Docker-Konzept verstehen (Images und Container)
- Docker Networks anwenden können
- Docker Compose anwenden können

Checkliste

- ✓ Suche nach Applikation auf Docker Hub (Dokumentation), Prozessor-Architektur beachten
- ✓ Gefundene Webapplikation von Docker Hub installieren
- ✓ Applikation muss mindestens zwei Container umfassen
- ✓ Manuelles Deployment mit Docker testen, nachvollziehbare Dokumentation der lauffähigen Applikation
- ✓ Vorgehen dokumentieren
- ✓ Docker Compose File erstellen, auf GitHub verfügbar machen und dieses testen