



DOCKER.

Eine Präsentation von Marcel Langoth

INHALT

- ▶ Wie funktioniert Docker?
- ▶ Was ist Kubernetes & Swarm?
- ▶ Wie baue ich Docker in meine Projekt ein?

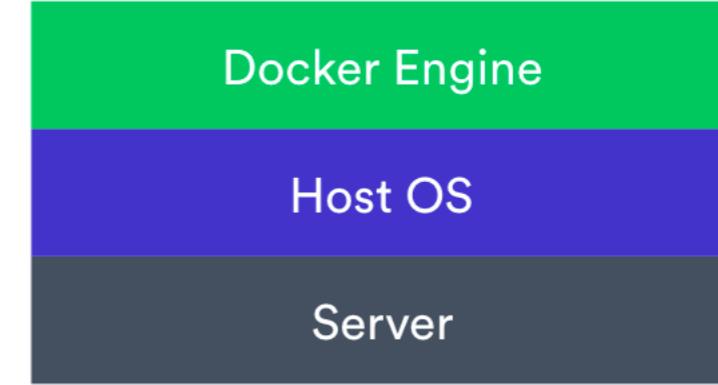
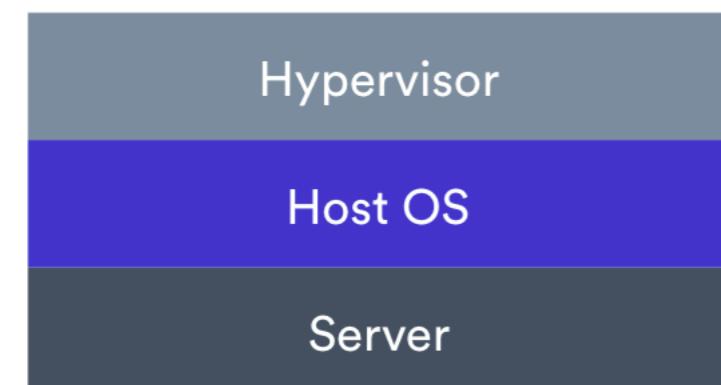
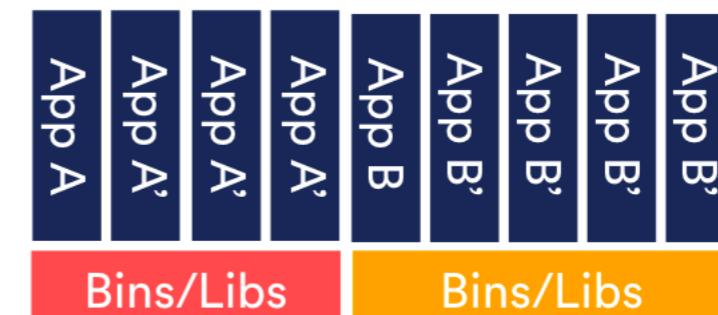
WIE FUNKTIONIERT DOCKER?

Virtual Machine



VS.

Docker



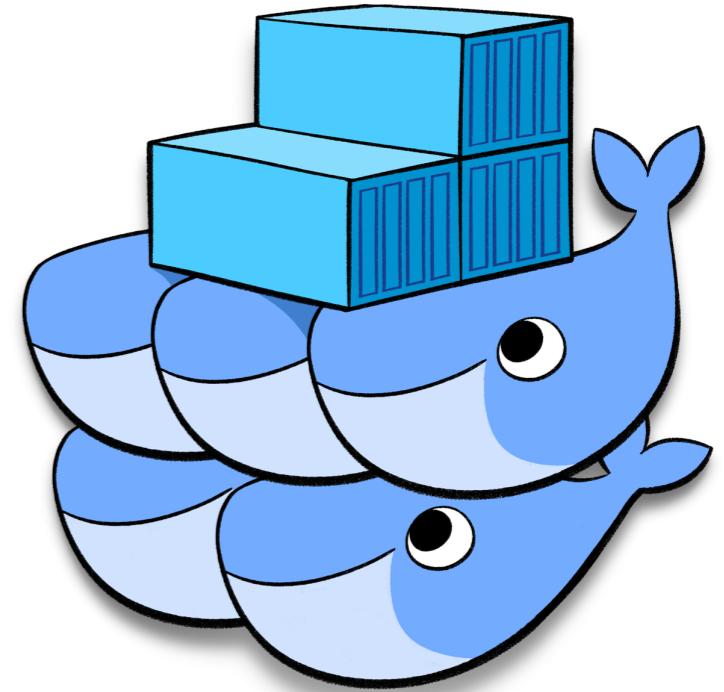
DIE VERSPRECHEN VON DOCKER



KUBERNETES VS. SWARM



vs.



Dockerfiles nicht kompatibel
Schnell

Mehr Funktionen als Docker-Swarm
etwas langsamer bei Deployments
entwickelt von Google

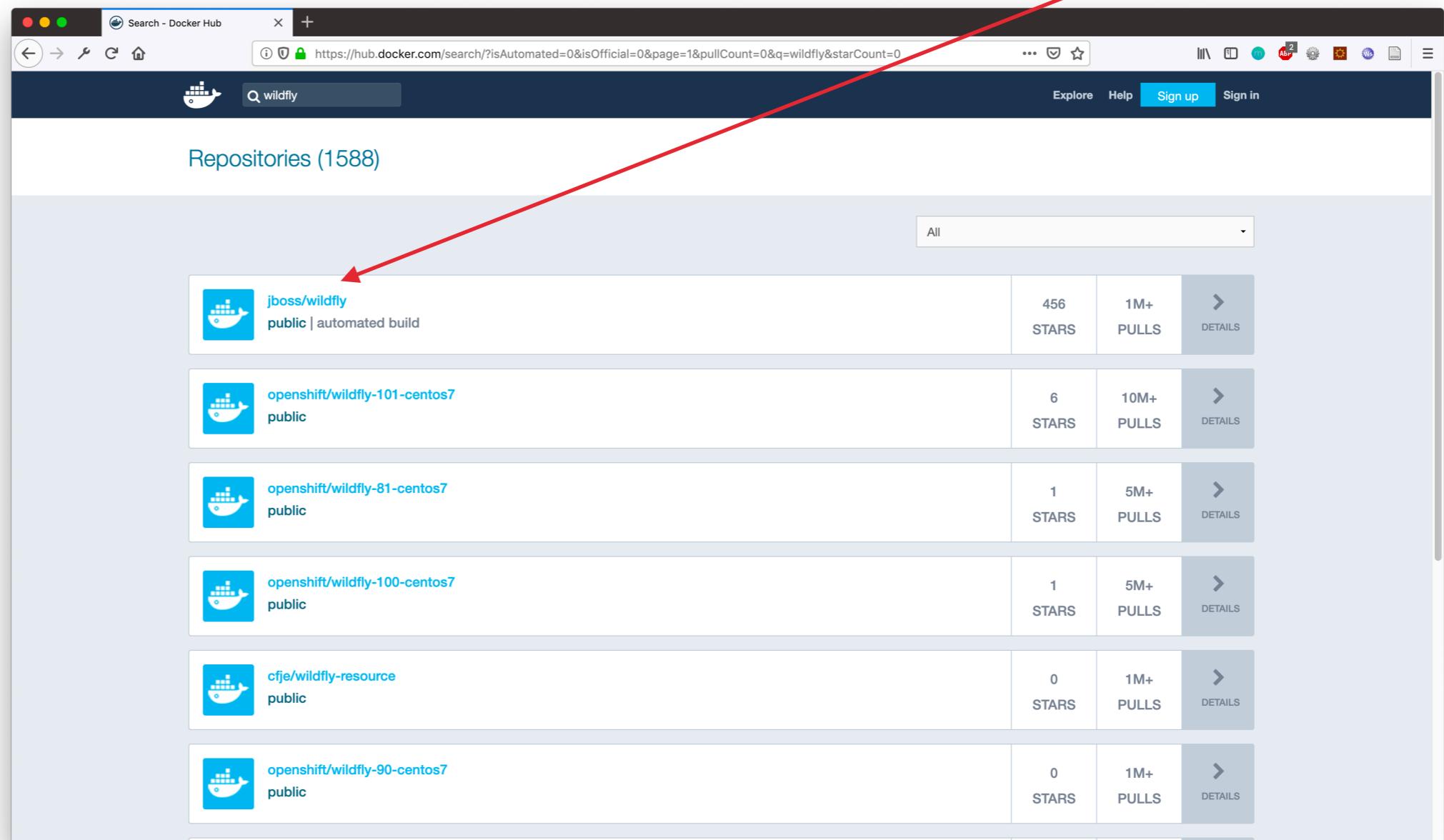
Einfach
Schnell

Funktioniert gleich wie Docker
entwickelt von Docker

WIE BAUE ICH DOCKER IN MEIN PROJEKT EIN?

Was möchte ich in Docker laufen lassen? z.B. Application Server, Datenbanken, ...

Im Dockerhub sind bereits vorgefertigte Images für solche Anwendungen, z.B. für **Wildfly**



DOCKER.

► Dockerfiles

Dockerfile — docker_example

```
FROM nginx:stable
COPY default.conf /etc/nginx/conf.d/default.conf
VOLUME /usr/share/nginx/html
```

Ln 4, Col 29 Spaces: 4 UTF-8 LF Dockerfile

0 0 0 1

Dockerfile — docker_example

```
FROM mysql:5
COPY 1-createdb.sql 2-data.sql /docker-entrypoint-initdb.d/
```

Ln 3, Col 60 Spaces: 4 UTF-8 LF Dockerfile

0 0 0 1

DOCKER.

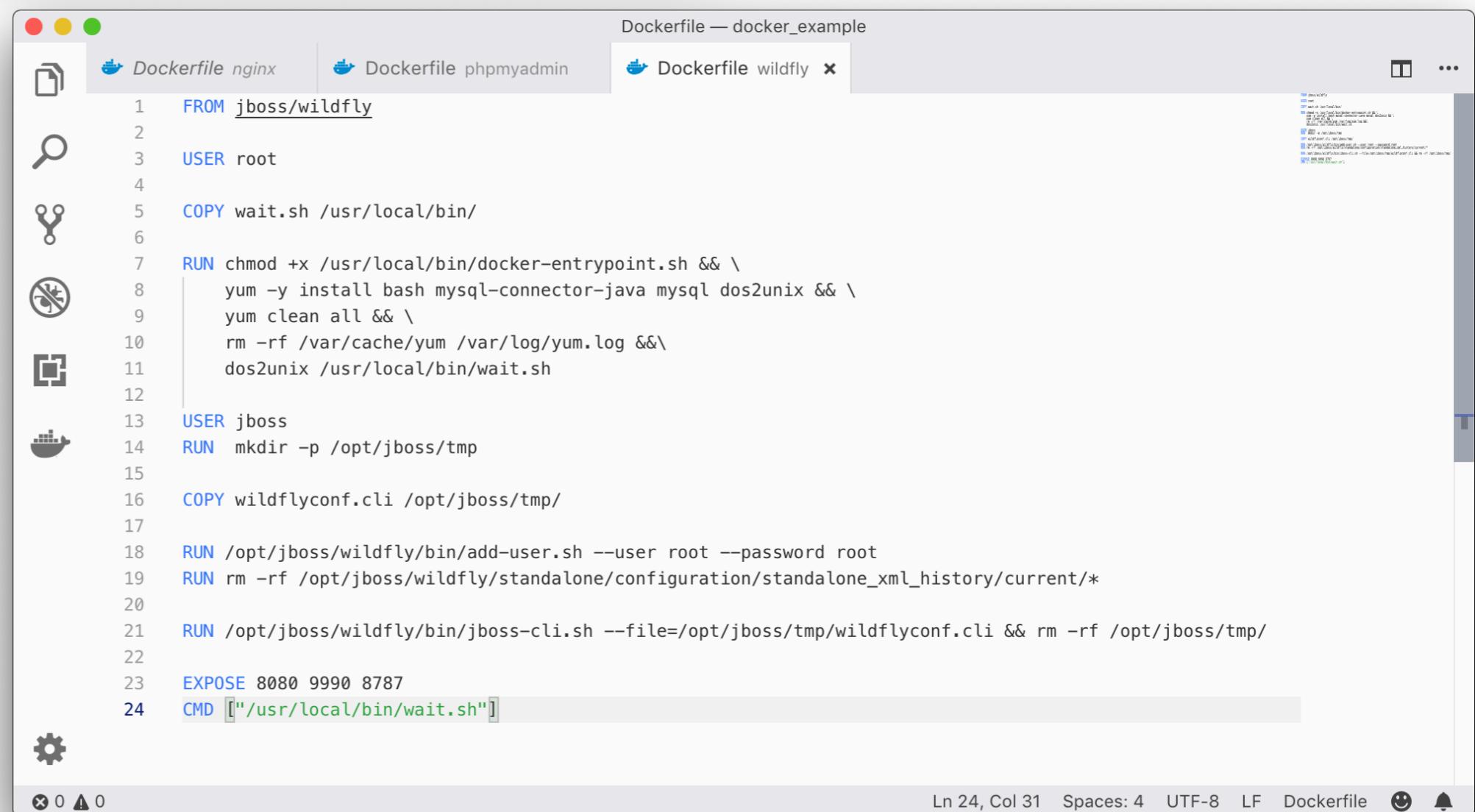
Dockerfiles



A screenshot of a code editor window titled "Dockerfile — docker_example". The editor shows a single tab with the Dockerfile for "phpmyadmin". The code is as follows:

```
FROM phpmyadmin/phpmyadmin
```

The editor interface includes a toolbar with icons for file operations, a sidebar with a gear icon (containing the number 1), and status indicators at the bottom.



A screenshot of a code editor window titled "Dockerfile — docker_example". The editor shows three tabs: "Dockerfile nginx", "Dockerfile phpmyadmin", and "Dockerfile wildfly". The "Dockerfile wildfly" tab is active and displays the following Dockerfile content:

```
FROM jboss/wildfly
USER root
COPY wait.sh /usr/local/bin/
RUN chmod +x /usr/local/bin/docker-entrypoint.sh && \
    yum -y install bash mysql-connector-java mysql dos2unix && \
    yum clean all && \
    rm -rf /var/cache/yum /var/log/yum.log && \
    dos2unix /usr/local/bin/wait.sh
USER jboss
RUN mkdir -p /opt/jboss/tmp
COPY wildflyconf.cli /opt/jboss/tmp/
RUN /opt/jboss/wildfly/bin/add-user.sh --user root --password root
RUN rm -rf /opt/jboss/wildfly/standalone/configuration/standalone_xml_history/current/*
RUN /opt/jboss/wildfly/bin/jboss-cli.sh --file=/opt/jboss/tmp/wildflyconf.cli && rm -rf /opt/jboss/tmp/
EXPOSE 8080 9990 8787
CMD ["/usr/local/bin/wait.sh"]
```

The editor interface includes a sidebar with various icons, a status bar at the bottom, and a small preview window on the right side.

► Docker-compose

The screenshot shows a code editor window titled "docker-compose.yml — docker_example". The file contains a YAML configuration for a multi-service Docker application. The services defined are wildfly, nginx, mysql, and phpmyadmin. The wildfly service uses the docker/wildfly image and exposes ports 8080, 9990, and 8787. It depends on the mysql service and has an environment variable MySQL_ROOT_PASSWORD set to root. The nginx service uses the docker/nginx image and exposes port 80. It depends on the wildfly service and mounts the www directory from the host to /usr/share/nginx/html. The mysql service uses the docker/mysql image and exposes port 3306. It has a volume mysql_data mounted at /var/lib/mysql and an environment variable MySQL_ROOT_PASSWORD set to root. The phpmyadmin service uses the docker/phpmyadmin image and exposes port 5050. It depends on the mysql service and has environment variables PMA_HOST set to mysql and MySQL_ROOT_PASSWORD set to root. A volume mysql_data is also defined for this service. The bottom of the editor shows status icons and text indicating 0 errors, 0 warnings, and 40 lines of code.

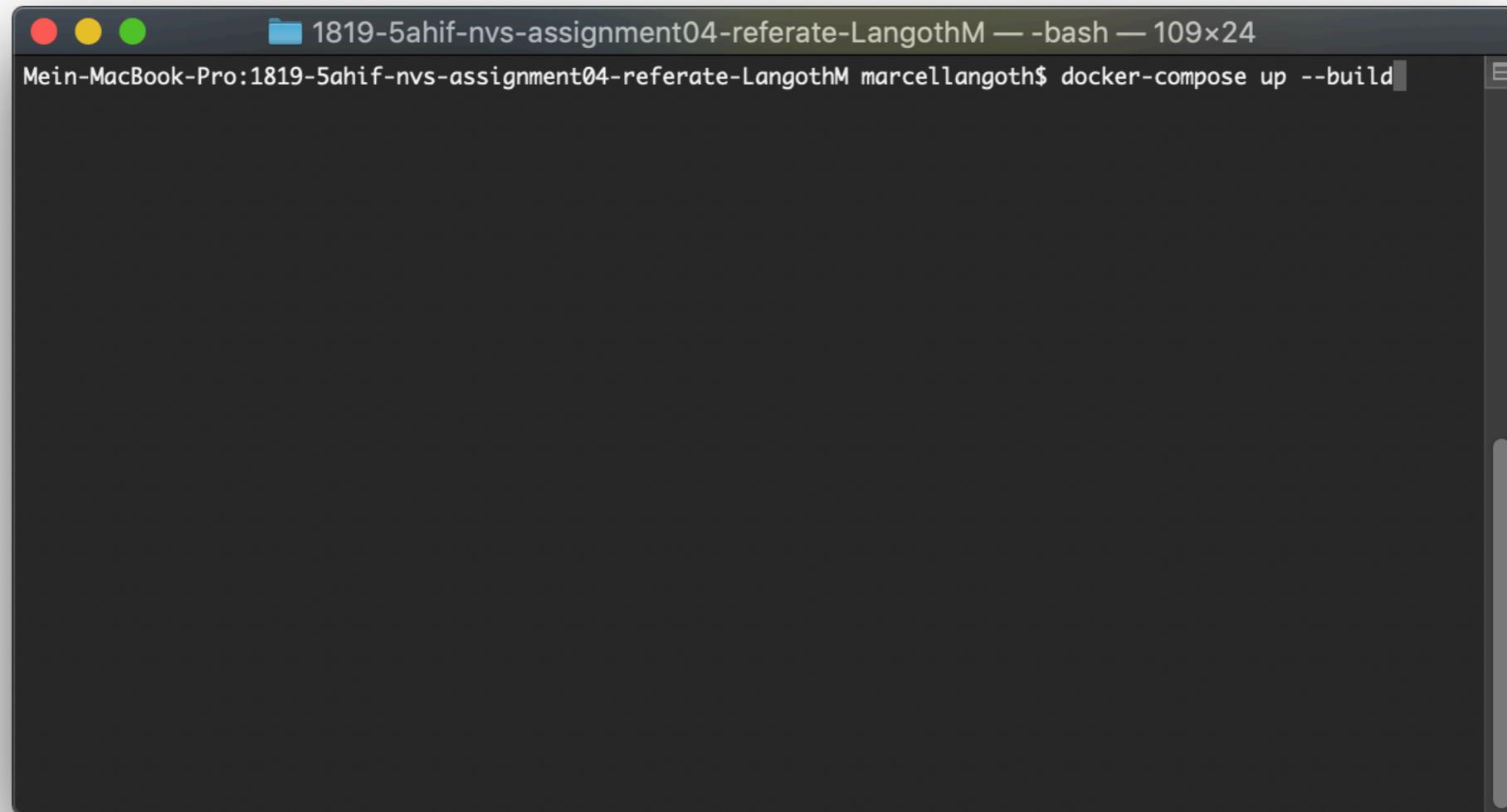
```
version: '2.0'

services:
  wildfly:
    build: docker/wildfly
    ports:
      - 8080:8080
      - 9990:9990
      - 8787:8787
    depends_on:
      - mysql
    environment:
      - MYSQL_ROOT_PASSWORD=root
  nginx:
    build: docker/nginx
    ports:
      - 80:80
    volumes:
      - ./www:/usr/share/nginx/html
    depends_on:
      - wildfly
  mysql:
    build: docker/mysql
    ports:
      - 3306:3306
    volumes:
      - mysql_data:/var/lib/mysql
    environment:
      - MYSQL_ROOT_PASSWORD=root
  phpmyadmin:
    build: docker/phpmyadmin
    ports:
      - 5050:5050
    depends_on:
      - mysql
    environment:
      - PMA_HOST=mysql
      - MYSQL_ROOT_PASSWORD=root
  volumes:
    mysql_data:
```

Ln 40, Col 14 Spaces: 2 UTF-8 LF YAML ⚙️ 🔔

DOCKER.

Im Terminal starten & fertig!



```
1819-5ahif-nvs-assignment04-referate-LangothM — -bash — 109x24
Mein-MacBook-Pro:1819-5ahif-nvs-assignment04-referate-LangothM marcellangoth$ docker-compose up --build
```