

Compose – Übungsaufgaben Lösungen

In diesem Assessment wirst du das bekannte Wordpress mit einer MySQL Datenbank installieren, um Blogs und Web Seiten zu erstellen. Du wirst sehen, wie einfach das mit Docker Compose geht. 😊

1. Erstelle ein neues Verzeichnis `wordpress` und wechsele in dieses neue Verzeichnis. Erstelle darin eine leere `docker-compose.yml` Datei.

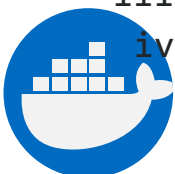
```
mkdir wordpress  
cd wordpress  
touch docker-compose.yml
```

2. Erstelle jetzt schrittweise die `docker-compose.yml` Datei. Beginne damit, die Version und die Services zu initialisieren.

```
version: '3'  
services:
```

Definiere danach den ersten Docker Container `db` für die MySQL Datenbank. Dieser Container soll die folgenden Eigenschaften haben:

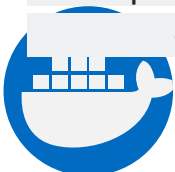
- Name des Service “`db`”
- Verwende das Image `mysql:5.7`
- Binde das folgende Laufwerk ein `db_data:/var/lib/mysql`
- Setze den `restart` auf `always`
- Setze die folgenden Umgebungsvariablen:
 - i. `MYSQL_ROOT_PASSWORD: einPasswort`
 - ii. `MYSQL_DATABASE: wordpress`
 - iii. `MYSQL_USER: wordpress`
 - iv. `MYSQL_PASSWORD: wordpress`



```
db:
  image: mysql:5.7
  volumes:
    - db_data:/var/lib/mysql
  restart: always
  environment:
    MYSQL_ROOT_PASSWORD: somewordpress
    MYSQL_DATABASE: wordpress
    MYSQL_USER: wordpress
    MYSQL_PASSWORD: wordpress
```

3. Definiere danach den zweiten Container `wordpress` und setze die folgenden Parameter:
- Name des Service "`wordpress`"
 - Dieser Container hängt vom MySQL container `db` ab
 - Verwende das Image `wordpress:latest`
 - Leite den internen Wordpress Port 80 auf deinen Host Port 8000 weiter
 - Setze den `restart` auf `always`
 - Setze die folgenden Umgebungsvariablen:
 - i. `WORDPRESS_DB_HOST: db:3306`
 - ii. `WORDPRESS_DB_USER: wordpress`
 - iii. `WORDPRESS_DB_PASSWORD: wordpress`
 - iv. `WORDPRESS_DB_NAME: wordpress`

```
wordpress:
  depends on:
    - db
  image: wordpress:latest
  ports:
    - "8000:80"
```



```
restart: always
environment:
  WORDPRESS_DB_HOST: db:3306
  WORDPRESS_DB_USER: wordpress
  WORDPRESS_DB_PASSWORD: wordpress
  WORDPRESS_DB_NAME: wordpress
```

4. Binde das folgende Volume unter der Service Ebene ein `db_data: {}`

Hinweis: Die `{}` bedeuteten nur, dass dem benannten Volume keine zusätzlichen Optionen gegeben wurden. Dies unterscheidet sich nicht davon, es leer zu lassen.

```
volume:
  db_data: {}
```

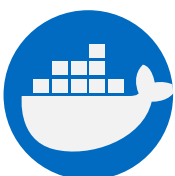
5. Starte jetzt die erstellte Compose Datei im Hintergrund Modus.

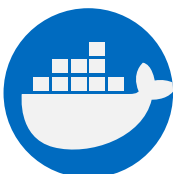
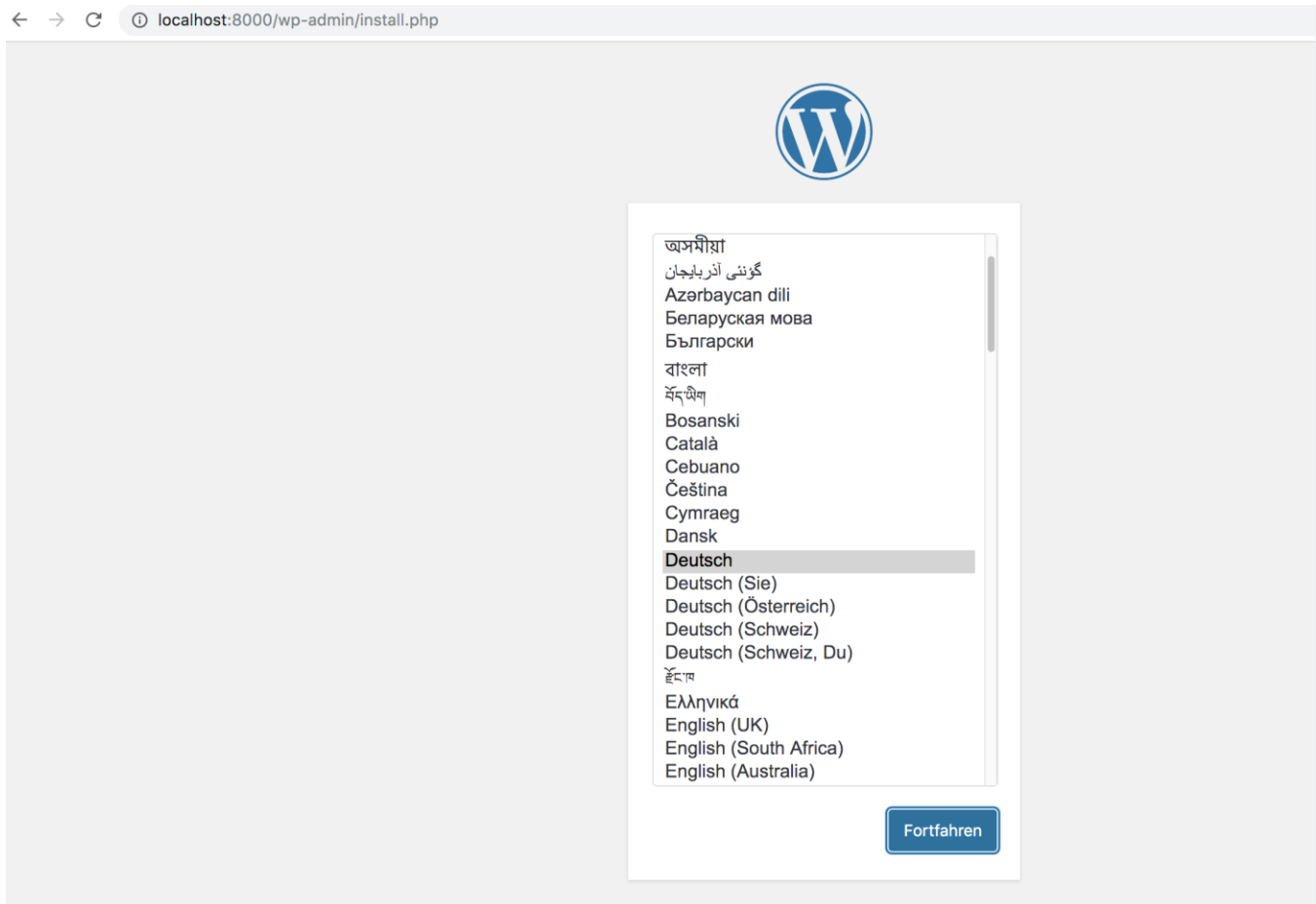
```
docker-compose up -d
```

6. Öffne Wordpress in einem Web Browser.


```
http://localhost:8000
```

7. Jetzt kannst du Wordpress installieren. Das ist aber nicht mehr Teil dieses Kurses... was dich aber nicht daran hindern soll. 😊





← → ↻ ⓘ localhost:8000/wp-admin/install.php?step=1



Willkommen

Willkommen bei der berühmten 5-Minuten-Installation von WordPress! Gib unten einfach die benötigten Informationen ein und schon kannst du starten mit der am besten erweiterbaren und leistungsstarken persönlichen Veröffentlichungsplattform der Welt.

Benötigte Informationen

Bitte trage die folgenden Informationen ein. Keine Sorge, du kannst all diese Einstellungen später auch wieder ändern.

Titel der Website

Benutzername
Benutzernamen dürfen nur alphanumerische Zeichen, Leerzeichen, Unterstriche, Bindestriche, Punkte und das @-Zeichen enthalten.

Passwort [Verbergen](#)
Stark

Wichtig: Du wirst dieses Passwort zum Anmelden brauchen. Bitte bewahre es an einem sicheren Ort auf.

Deine E-Mail-Adresse
Bitte überprüfe nochmal deine E-Mail-Adresse auf Richtigkeit, bevor du weitermachst.

Sichtbarkeit für Suchmaschinen ☐ Suchmaschinen davon abhalten, diese Website zu indexieren.
Es ist Sache der Suchmaschinen, dieser Bitte nachzukommen.

[WordPress installieren](#)

8. Stoppe und räume jetzt wieder die Container auf.

Bonus: Welche Möglichkeiten gibt es und was ist der Unterschied?

- Der Befehl `docker-compose down` entfernt die Container und das Standardnetzwerk, behält jedoch deine WordPress-Datenbank bei.
- Der Befehl `docker-compose down --volumes` entfernt die Container, das Standardnetzwerk und die WordPress-Datenbank.

