

WordPress mit MariaDB & phpMyAdmin in Docker

Ziel: Web-App mit WordPress + Datenbank in Docker betreiben

- Alle Container sind im **gemeinsamen benutzerdefinierten Netzwerk**
- Keine externen Links, nur interne Kommunikation
- Zugriff auf:
 - phpMyAdmin: <http://localhost:8080>
 - WordPress: <http://localhost:8081>

◇ 1. Netzwerk erstellen

```
docker network create blog-network
```

◇ 2. MariaDB starten

```
docker run -d \  
  --name mariadb \  
  --network blog-network \  
  -e MARIADB_ROOT_PASSWORD=sml12345 \  
  mariadb
```

Check: Ist MariaDB bereit?

```
docker logs mariadb
```

Achte auf die Zeile:

ready for connections

IP-Adresse der Datenbank herausfinden

```
docker inspect -f "{{range.NetworkSettings.Networks}}{{.IPAddress}}{{end}}"  
mariadb
```

Alternativ: In **blog-network** funktioniert auch der Containername **mariadb** als Hostname!

◇ 3. phpMyAdmin starten (extern zugänglich)

```
docker run -d \  
  --name phpmyadmin \  
  -e PMA_HOST=mariadb \  
  -p 8080:80 \  
  phpmyadmin
```

Netzwerk wechseln (von bridge zu blog-network)

```
docker network connect blog-network phpmyadmin  
docker network disconnect bridge phpmyadmin
```

Zugriff über Browser


<http://localhost:8080>

Login:

- Benutzer: **root**
- Passwort: **sml12345**

Neuen Benutzer & DB in phpMyAdmin anlegen

1. Neuer Benutzer: **blog**
2. Passwort generieren (z. B. über Passwortmanager oder **openssl rand**)
3. Gleicher Name für Datenbank: **blog**
4. **Alle Rechte gewähren**

 Passwort notieren, wird gleich gebraucht!

◇ 4. WordPress starten

```
docker run -d \  
  --name wordpress \  
  --network blog-network \  
  -e WORDPRESS_DB_HOST=mariadb \  
  -e WORDPRESS_DB_USER=blog \  
  -e WORDPRESS_DB_PASSWORD="MEIN_SICHERES_PASSWORT" \  
  -e WORDPRESS_DB_NAME=blog \  
  -p 8081:80 \  
  wordpress
```

WordPress einrichten

Gehe zu: <http://localhost:8081>

1. Sprache wählen
 2. Admin-Zugang erstellen (Benutzername, Passwort, E-Mail)
 3. Installation abschliessen
-

☒ Fertig!

Du hast jetzt eine komplett dockerisierte Webanwendung – **modular, vernetzt, einsatzbereit**.