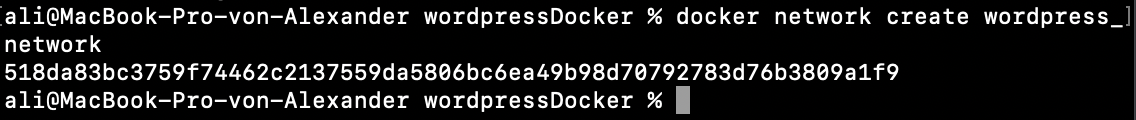
1) Im Terminal: Netzwerk vorbereiten (zum Verbinden der Container)  
docker network create wordpress\_network

1. Netzwerk vorbereiten
   1. Führe obiges Kommando aus (Screenshot)



* 1. Rufe Hilfe für „network“ im Terminal aus (Screenshot)

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

* 1. Suche den Befehl für das Auflisten der Netzwerke (Screenshot)

Ein Bild, das Text, Screenshot, Schrift, Schwarz enthält.

Automatisch generierte Beschreibung

* 1. Was bedeutet obiger Befehl? Recherchiere im Internet oder über die Terminal-Hilfe und beschrifte jeden Teil des Befehls:  
     „docker network“

Erstellt eine Software-Bridge, die es ermöglicht mit anderen verbundenen Containern zu kommunizieren  
„create“

Erstellt einen Container, ohne ihn auszuführen  
„wordpress\_network“

Ein Netzwerk von Webseiten

2) Im Terminal: Datenbank Container laden und starten  
docker run -d --name mysql\_server –e MYSQL\_ROOT\_PASSWORD=my\_secret\_pw -e MYSQL\_DATABASE=db\_wordpress --network **???** mysql:latest

1. Datenbank-Container laden/starten
   1. Welche Text gehört in obige Lücken (**???**)?

Wie der Container heißen soll

* 1. Führe obiges Kommando aus (Screenshot) Ein Bild, das Text, Screenshot, Schrift enthält.

     Automatisch generierte Beschreibung
  2. Was bedeutet obiger Befehl? Recherchiere im Internet, über die Terminal-Hilfe oder über <https://hub.docker.com/_/mysql> und beschrifte jeden Teil des Befehls:

„docker run“: Der Befehl zum Starten eines neuen Docker-Containers.

-d: Die Option, die den Container im Hintergrund (detach mode) ausführt, ohne die Ausgabe auf dem Bildschirm anzuzeigen.

--name mysql\_server: Der Name, der dem Container zugewiesen wird. In diesem Fall "mysql\_server".

-e: Die Option zum Setzen von Umgebungsvariablen innerhalb des Containers.

-e MYSQL\_ROOT\_PASSWORD=my\_secret\_pw: Setzt die Umgebungsvariable "MYSQL\_ROOT\_PASSWORD" auf den Wert "my\_secret\_pw". Dies ist das Passwort für den MySQL-Root-Benutzer.

-e MYSQL\_DATABASE=db\_wordpress: Setzt die Umgebungsvariable "MYSQL\_DATABASE" auf den Wert "db\_wordpress". Dies erstellt eine neue Datenbank mit dem Namen "db\_wordpress".

--network ???: an der Stelle der ??? muss der Name angegeben werden

mysql:latest: Der Name des Docker-Images, das verwendet werden soll, um den Container zu erstellen. "latest" ist hier eine Versionsnummer, die angibt, dass die neueste Version des MySQL-Images verwendet werden soll.

3) Im Terminal: Wordpress Container laden und starten  
docker run -d --name wordpress -p 8080:80 -e WORDPRESS\_DB\_HOST=mysql\_server:3306 -e WORDPRESS\_DB\_USER=root -e WORDPRESS\_DB\_PASSWORD=**???** –e WORDPRESS\_DB\_NAME=**???**--network wordpress\_network wordpress:latest

1. Wordpress-Container laden/starten
   1. Welche Text gehört in obige Lücken (**???**)?

Das Passwort für die Datenbank

* 1. Führe obiges Kommando aus (Screenshot)Ein Bild, das Text, Screenshot, Schwarzweiß enthält.

     Automatisch generierte Beschreibung
  2. Was bedeutet obiger Befehl? Recherchiere im Internet, über die Terminal-Hilfe oder über <https://hub.docker.com/_/wordpress> und beschrifte jeden Teil des Befehls:  
     „--name wordpress“

Name  
„-p 8080:80“

Ip und Port  
Was bedeutet 80 hier (Für welches Protokoll steht der Port?)

HTTP  
Welchen Port muss ich bei lokalem Aufruf nach „localhost“ angeben? Je nachdem welchen man angegeben hat  
„wordpress:latest“. Benützt den letzten Port  
was bedeutet latest hier? Benützt den letzten Port

4) Im Terminal: Container inpizieren

1. Container inspizieren
   1. Suche den Befehl zum inspizieren von Containern (über die docker-Terminal-Hilfe) (Screenshot)Ein Bild, das Text, Screenshot, Schrift enthält.

      Automatisch generierte Beschreibung
   2. Eruiere die ID des Wordpress-Containers

518da83bc3759f74462c2137559da5806bc6ea49b98d70792783d76b3809a1f9

* 1. Führe den Befehl zum Inspizieren des Containers aus (Screenshot)

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

* 1. Überprüfe auf Richtigkeit (alle Angaben des Befehls, auch die Namen der Environment-Variablen) hab ich

5) Im Terminal: Container löschen/Neu erstellen

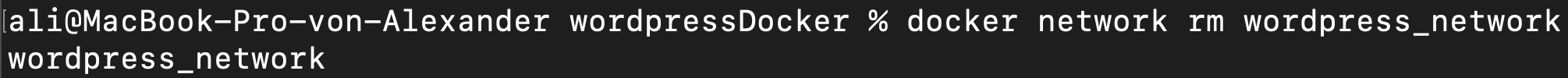
1. Container löschen/Neu erstellen
   1. Eruiere die ID des Wordpress-Containers:

518da83bc3759f74462c2137559da5806bc6ea49b98d70792783d76b3809a1f9

* 1. Wie lautet der Befehl zum löschen eines Containers?

Docker network rm NAME

* 1. Führe den Befehl aus (Screenshot)

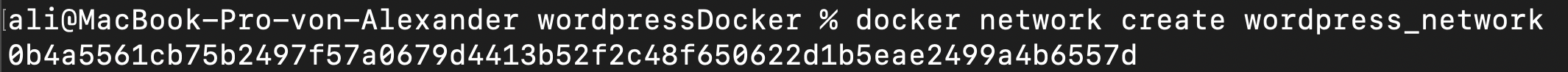


* 1. Liste ALLE Container auf und prüfe, ob das Löschen funktioniert hat (Screenshot)

**Ein Bild, das Text, Screenshot, Schrift, Schwarz enthält.

Automatisch generierte Beschreibung**

* 1. Erstelle den Container neu (Suche mit Arrow-Up-Taste, mglw. Korrektur von Typos) (Screenshot)



1. Im Browser: Konfigurationsseite aufrufen

<http://localhost:8080/>

Im Browser: als Admin im Backend einloggen

<http://localhost:8080/wp-login.php>

Username: <your-user>  
Passwort: <your-password>

Ein Bild, das Text, Screenshot, Schrift, Schwarz enthält.

Automatisch generierte Beschreibung

Ich weiß nicht welchen Befehl oder was auch immer ich eingeben sollte, sodass es funktioniert, weil immer eine Error Message kam.

1. Frontend/Anmeldefenster Backend
   1. Frontend aufrufen
   2. Backend aufrufen und Eingabe User/PW (Screenshot)

7) Im Terminal: Datenbankzugriff (phpMyAdmin)

docker run --name myadmin -d --link mysql\_server:db -p 8081:80 --network  
**???** phpmyadmin

Im Browser: als Admin im phpMyAdmin einloggen

[http://localhost:8081](http://localhost:8081/) (Username: **???,** Passwort: **???)**

1. DB-Zugriff phpMyAdmin
   1. Was bedeutet obiger Befehl?  
      „—name myadmin“:  
      „-d“:  
      „-p 8081:80“:  
      welche ist die richtige Angabe bei„--network“:
   2. Führe obigen Befehl aus (Screenshot)
   3. Finde User/PW heraus – Welche Eingabe ist hier zu machen (Screenshot)