

Docker-Compose LAMP

Von Fabrizio



[Datum]

[Firmenname]

[Firmenadresse]

Inhalt

[2. Einleitung 2](#_Toc58242865)

[3. Vorbereitung 2](#_Toc58242866)

[Vorbedingung : 2](#_Toc58242867)

[Benutzer mit root Rechten erstellen 2](#_Toc58242868)

[Installation der Tools 3](#_Toc58242869)

[Aktualisieren 3](#_Toc58242870)

[Installation Docker-Compose 3](#_Toc58242871)

[4. Docker 3](#_Toc58242872)

[YML File erstellen 3](#_Toc58242873)

[5. GitHub 4](#_Toc58242874)

[Repository anlegen 4](#_Toc58242875)

[6. Ausführung 5](#_Toc58242876)

[Git Clone 5](#_Toc58242877)

[Container Starten 5](#_Toc58242878)

[My SQL Datenbank befüllen 5](#_Toc58242879)

[PHPMyAdmin 5](#_Toc58242880)

[Shell SSH 6](#_Toc58242881)

[Kontrolle der Einträge 7](#_Toc58242882)

[7. Fazit 7](#_Toc58242883)

DOCKERCOMPOSE LAMP von Milchtank

# Einleitung

In dieser Anleitung wird beschrieben, wie man mit Docker-Compose einen LAMP Server einrichten kann.

Es wird erst ein YML File erstellt, dass nachher dann in einem Github Repository abgelegt ist.

Eine Demonstration der Installation und ein paar Beispiele per Mysql.

# Vorbereitung

Zur vorbereitung wird ein blanker Server erwartet. In diesem Fall ein ubuntu Server.

## Vorbedingung :

In diesem Beispiel steht ein ubuntu 20.04.1 lts (focal fossa) Server blank zur Verfügung.

Es wird ein User benötigt der root Rechte hat.

### Benutzer mit root Rechten erstellen

Login mit dem User root

#### User erstellen:

Eingabe in Konsole:

Adduser milchtank // Dann muss noch das Passwort 2x eingegeben werden und noch diverse Infos zum Benutzer

Adding user `milchtank' ...

Adding new group `milchtank' (1000) ...

Adding new user `milchtank' (1000) with group `milchtank' ...

Creating home directory `/home/milchtank' ...

Copying files from `/etc/skel' ...

Enter new UNIX password:

Retype new UNIX password:

passwd: password updated successfully

Changing the user information for milchtank

Enter the new value, or press ENTER for the default

Full Name []: milchtank

Room Number []:

Work Phone []:

Home Phone []:

Other []:

Is the information correct? [Y/n] y

#### User zu Gruppe root zufügen:

Damit der User auch root Rechte hat, packen wir diesen mit folgendem Befehl in die Benutzergruppe «sudo» Wenn dann Rechte benötigt werden um etwas auszuführen, kann vor den Befehl «sudo» geschrieben werden. Dann muss man sich nochmals mit seinem Passwort verifizieren, kann nachher aber den Befehl als Administrator(root) ausführen.

usermod -aG sudo milchtank

## Installation der Tools

Auf dem blanken Server müssen vorab zwingend Tools installiert werden, damit wir unser Milchtank LAMP auch umsetzen können.

### Aktualisieren

Aktualisieren des Servers, damit dieser auf dem neusten Stand ist:

Mit dem Befehl:

sudo apt-get update && apt-get upgrade

Werden automatisch alle relevanten Updates für das System gesucht und auch installiert.

### Installation Docker-Compose

Mit dem Befehl:

sudo apt install docker docker-compose

wird die Docker-Compose installiert

Weitere Informationen zur Installation [bietet die Dokumentation](https://docs.docker.com/installation).

# Docker-Compose

Wir werden unseren LAMP Server anhand eines docker-compose.yml File erstellen und laufen lassen. Anschliessend kommt das YML File in ein Github Repository damit dies auch einfach verteilt und installiert werden kann.

Um Docker zu installieren folgender Code eingeben:

sudo apt-get docker-compose

Dannach wird Docker-Compose installiert und wir sind Startbereit.

## YML File erstellen

Nun werden wir unseren LAMP anhand einer .yml Datei erstellen. Der Vorteil darin ist, das nichts vorinstalliert werden muss. Sondern nur die .yml Datei korrekt geschrieben werden muss.

In unserem Fall werden wir erst die Ordnerstruktur erstellen.

Im Homedir können wir den ordner LAMP erstellen:

mkdir LAMP

Anschliessend in den Ordner wechseln:

cd LAMP

Dann das docker-compose.yml File erstellen:

Touch docker-compose.yml

Nano docker-compose.yml

Fertiges YML File für unser Vorhaben:

version: '3.7' << Version des YML Files

services: << In dieser Rubrik werden die Services aufgelistet die als Package laufen so

php-httpd: << Name des Containers

image: php:7.3-apache << Name des Images mit : kann die version gewählt werden (-apache)

ports: << Werden die Ports definiert Extern:Intern

- 80:80

volumes: << Wird die Ablage/Speicherort definioert

- "./DocumentRoot:/var/www/html"

mariadb:

image: mariadb:10.5.2

volumes:

- mariadb-volume:/var/lib/mysql

environment: << Unter enviroment können Einstellungen vorkonfiguriert werden.

TZ: "Europe/Rome"

MYSQL\_ALLOW\_EMPTY\_PASSWORD: "no"

MYSQL\_ROOT\_PASSWORD: "rootpwd"

MYSQL\_USER: 'testuser'

MYSQL\_PASSWORD: 'testpassword'

MYSQL\_DATABASE: 'testdb'

ports:

- 3306:3306

phpmyadmin:

image: phpmyadmin/phpmyadmin

links:

- 'mariadb:db'

ports:

- 8081:80

volumes:

mariadb-volume:

# GitHub

Wir werden ein Githubt Repository erstellen, damit wir unseren LAMP einfach vervielfachen können.

## Repository anlegen

Erstmal auf Github.com ein neues Repository anlegen.

Sobald dies angelegt wurde, auf dem Server zum Speicherort navigieren. (Ein Ordner bevor der Ordner LAMP kommt)

Mit einer dir Ausgabe sollte folgendes kommen:



* README.md wird noch nicht da sein.

In diesem Ordner folgende Befehle ausführen

echo "# LAMPCA" >> README.md << README.md Datei erstellen

git init << Ordnerstruktur auf Git umstellen

git add README.md << README.md ins Repository aufnehmen

git commit -m "first commit" << Commit erstellen -m « Nachricht»

git branch -M main << Branch wählen/erstellen

git remote add origin <https://github.com/Milchtank/LAMPCA.git> << Github Repository linken

git push -u origin main << Push auf github.com

# Ausführung

Nun zur Installation und Vervielfältigung des LAMP Servers.

Dazu brauchen wir einen blanken Server.

## Git Clone

In den Ordner wechseln, in dem der Container laufen soll. Anschliessend folgenden Befehl eingeben:

git clone <https://github.com/Milchtank/LAMPCA.git>

Nun sind wir startbereit!

## Container Starten

In das Verzeichnis wechseln wo die YML Datei liegt:

cd LAMPCA/LAMP

Container starten:

Docker-compose up -d

## My SQL Datenbank befüllen

### PHPMyAdmin

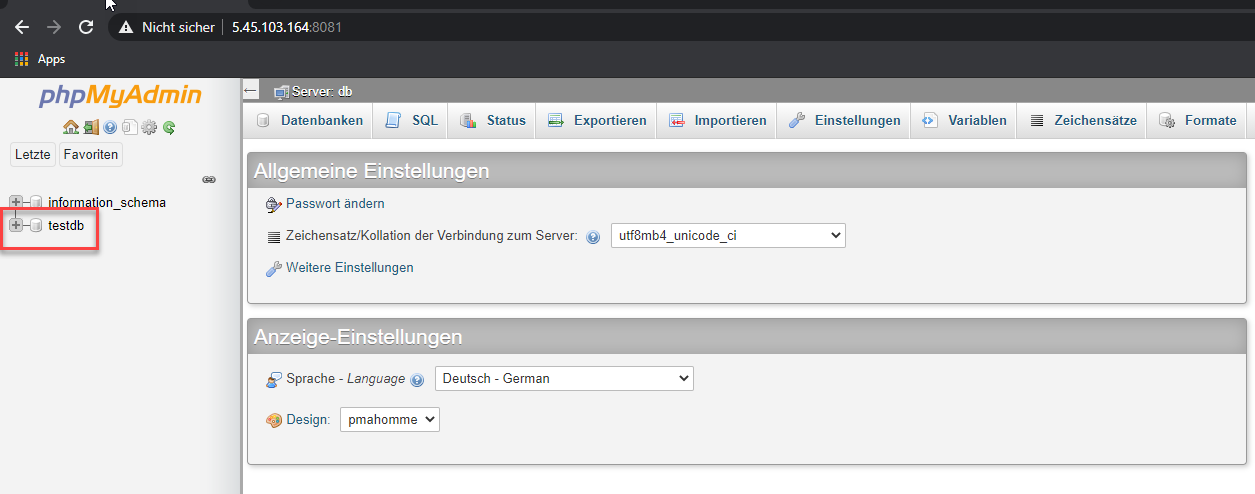
Login:

URL: <http://5.45.103.164:8081/>

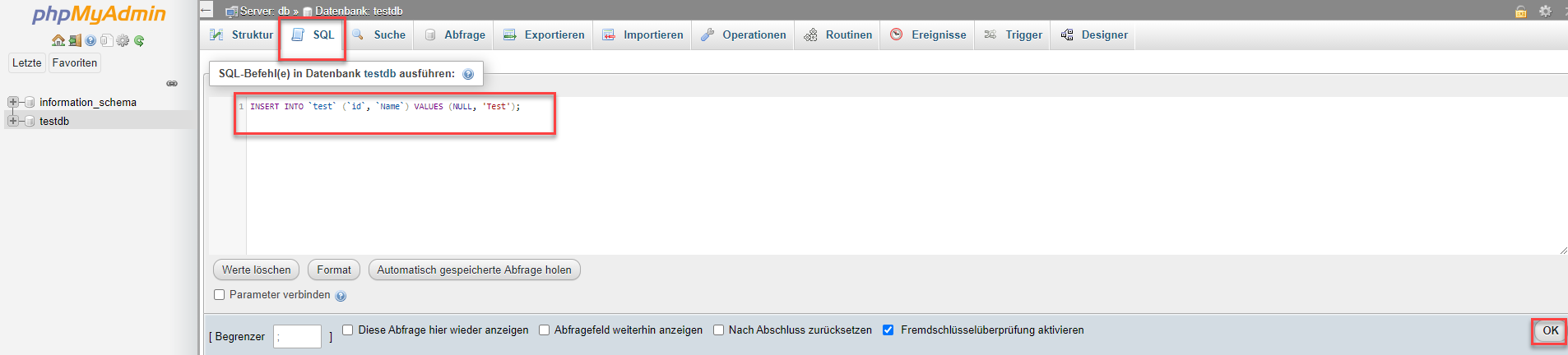
User: testuser

Pasword: testpassword

Auswahl testdb:



Dann SQL auswählen um die SQL Befehle einzugeben:



Befehl zum Eintragen:

INSERT INTO `test` (`id`, `Name`) VALUES (NULL, 'Hier\_dein\_Wert');

Falls direkt aus SQL geklickt wird braucht es folgende eingabe:

Use testdb;

INSERT INTO `test` (`id`, `Name`) VALUES (NULL, 'Hier\_dein\_Wert');

### Shell SSH

Per Shell auf 5.45.103.164 verbinden(Zb: Putty):

Benutzer: schule

Passwort: schule

Anschliessend in der Konsole folgenden Befehl eingeben:

Mysql -h 5.45.103.164 -P 3306 -u testuser -p

Anschliessend seit ihr eingeloggt in Mysql per Shell.

Befehl zum Eintragen: (Hier muss immer erst die Datenbank angegeben werden)

use testdb;

INSERT INTO `test` (`id`, `Name`) VALUES (NULL, 'Hier\_dein\_Wert');

## Kontrolle der Einträge

In beiden Tools kann mittels folgendem Befehl der Inhalt abgefragt werden:

SELECT \* FROM `test` ;

Allenfalls braucht es wieder den Link zur Datenbank mittels use testdb;

# Fazit

Es ist sehr intéressant mit Docker-Compose Files zu arbeiten. Es erspart enormen aufwand zum Ausfsetzen von Systemen. Da mit fast nur einer Datei schon sehr viel gemacht werden kann.