



---

# DOCKER-COMPOSE LAMP

---

Von Fabrizio



21. DEZEMBER 2020

MY-SOLUTION  
Fabrizio Piacente

## 1. Inhalt

2.	Einleitung.....	2
3.	Vorbereitung .....	2
	Vorbedingung : .....	2
	Benutzer mit root Rechten erstellen.....	2
	Installation der Tools.....	3
	Aktualisieren.....	3
	Installation Docker-Compose .....	4
4.	Docker-Compose .....	5
	YML File erstellen .....	5
5.	GitHub .....	7
	Repository anlegen.....	7
6.	Ausführung .....	8
	Git Clone .....	8
	Container Starten .....	8
	My SQL Datenbank befüllen.....	8
	PHPMyAdmin.....	8
	Shell SSH .....	9
	Kontrolle der Einträge .....	10
7.	Fazit .....	10

# DOCKERCOMPOSE LAMP von Fabrizio Paicente

## 2. Einleitung

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

L = Linux (ubuntu Betriebssystem)

A= Apache2 (Webserver)

M= MYSQL (MariaDB)

P= PHP (PHP Version 7.3)

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.

## 3. Vorbereitung

Zur Vorbereitung wird ein blanker Server erwartet. In diesem Fall ein ubuntu 20.04.1 lts (focal fossa) 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 sudo 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.

## 4. Docker-Compose

Damit die Funktionen einfach verteilt und auch genutzt werden können, werden anhand Docker-Compose ein YML File erstellen.

Das File übernimmt anschliessend ein Grossteil der Arbeit. Daher ist es wichtig die YML Datei korrekt zu erstellen.

### YML File erstellen

Nun werden wir unseren LAMP anhand einer .yml Datei erstellen. Der Vorteil darin ist, dass 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"         << Pfad des Webservers

  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:

## 5. GitHub

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

### Repository anlegen und initial Push

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:

```
milchtank@v22017124433158432:~$ dir
LAMP  README.md
milchtank@v22017124433158432:~$
```

- 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 <a href="https://github.com/Milchtank/LAMPCA.git">https://github.com/Milchtank/LAMPCA.git</a>	<< Github Repository linken
git push -u origin main	<< Push auf github.com

Anschliessend den Ordner LAMP per «git add LAMP» hinzufügen.

(Wichtig: hierzu im richtigen Ordner sein)

Folgende Abfolge muss eingehalten werden:

git add LAMP	<< README.md ins Repository aufnehmen
git commit -m "Add LAMP Folder"	<< Commit erstellen -m « Nachricht»
git push -u origin main	<< Push auf github.com



## 6. 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
```

Mittels des Befehles den Container starten:

```
Docker-compose up -d
```

### My SQL-Datenbank befüllen

Anschliessend eine kurze Anleitung wie die Mysql Datenbank befüllt werden kann.

### PHPMyAdmin

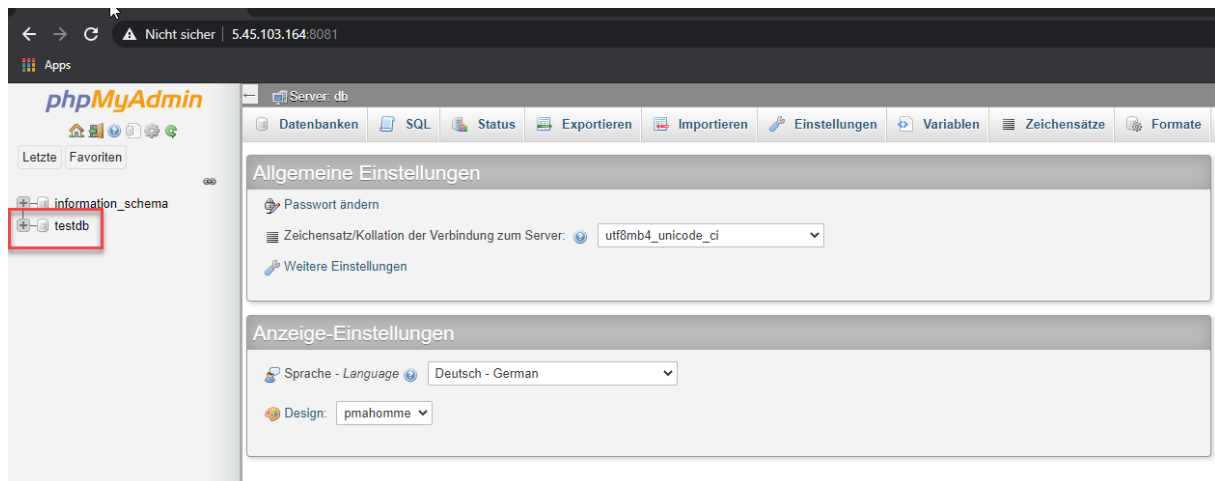
Login:

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

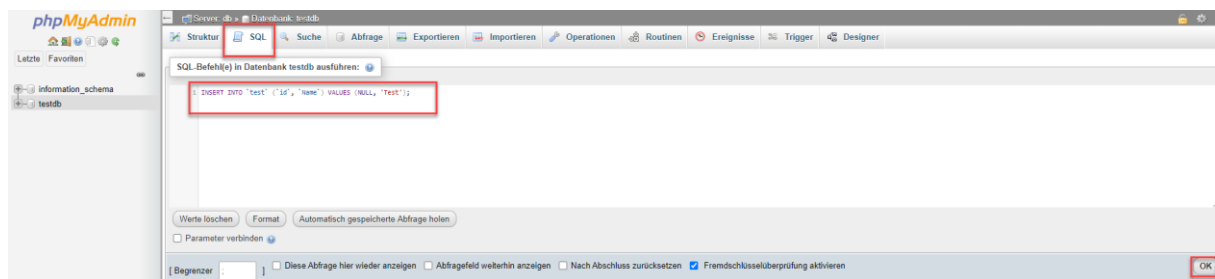
User: testuser

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;

© Fabrizio Piacente

```
INSERT INTO `test` (`id`, `Name`) VALUES (NULL, 'Hier_dein_Wert');
```

## Kontrolle der Einträge

In beiden Tools kann mittels folgenden Befehles der Inhalt abgefragt werden:

```
SELECT * FROM `test`;
```

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

## 7. Fazit

Es ist sehr interessant mit Docker-Compose Files zu arbeiten. Es erspart enormen Aufwand zum Aufsetzen von Systemen. Mit nur einer Datei kann schon sehr viel gemacht werden.