

LAMP Installation (MySQL, Apache, PHP)

Einleitung

In dieser Anleitung beschreibe ich kurz wie wir einen **LAMP Webserver** installieren.

Ein **LAMP Webserver** ist eine Installationskombination von **Apache**, **MySQL / MariaDB** und **PHP**.

Voraussetzungen

Um diese Installation durchführen zu können, sind folgende Voraussetzungen gegeben:

- Zugriff auf die Konsole über SSH / Telnet / Lokal
 - Root Rechte / sudo Rechte
 - Debian 11
 - Internet Anbindung des Servers
-

Apache Webserver Installation

Um den Apache Webserver zu installieren, aktualisierst du zuerst die Paketquellen und installierst ggf. Updates und entfernst nicht mehr benötigte Pakete.

```
sudo apt-get update && apt-get upgrade -y && apt-get autoremove -y
```

Als Nächstes installierst du den Apache2 Webserver.

```
sudo apt install apache2 -y
```

Du kannst jetzt die Installation mit dem Befehl überprüfen.

Wenn du eine UFW Firewall mit installiert und aktiviert hast, musst du die Ports auf der Firewall freigeben!

```
sudo ufw allow 80/tcp && \  
sudo ufw allow 443/tcp && \  
sudo ufw reload
```

MariaDB Datenbank Installation

Um jetzt die MariaDB Datenbank zu installieren, aktualisierst du zuerst wieder die Paketquellen, installierst Updates und entfernst nicht mehr benötigte Pakete.

```
sudo apt-get update && apt-get upgrade -y && apt-get autoremove -y
```

Als Nächstes installierst du jetzt die MariaDB Datenbank.

```
sudo apt install mariadb-server -y
```

Und jetzt führen wir ein Skript aus. Dieses lässt uns Sicherheitseinstellungen für unseren Datenbankserver einstellen. Dieses öffnen wir mit folgendem Befehl

```
mysql_secure_installation
```

Hier werden einige Dinge abgefragt. Diese Einstellungen werden wir jetzt gemeinsam setzen.

- Zuerst Enter drücken (Wir haben bisher kein root Kennwort gesetzt für den Datenbankbenutzer)
- Debian 11: Die Unix Authentifizierung lehnen wir mit **n** ab.
- Jetzt **Y** eingeben. Wir setzen nun ein Kennwort für den **Root** Datenbank Benutzer.
- Als nächstes **Y** eingeben. Wir wollen alle unbekannten Benutzer löschen.
- Dann geben wir wieder **Y** ein. Wir unterbinden damit eine Anmeldung des **Root Benutzers** außerhalb unseres Servers.
- Und wieder geben wir **Y** ein. Damit wird die Testdatenbank und die Rechte dorthin gelöscht.
- Als Letztes geben wir wieder ein **Y** ein. Damit werden die Berechtigungen einmal neu geladen.

Jetzt kannst du dich mit folgendem Befehl auf dem SQL Server einloggen. Du wirst nach der Eingabe nach dem Kennwort des Root Benutzers gefragt. Sobald du dieses eingegeben hast, kannst du SQL Befehle absetzen.

```
mysql -u root -p
```

Wenn kein Passwort für Root angegeben wurde, loggt sich der Benutzer automatisch auf dem Server ein.

Installation von PHP 7.4

Um unsere Installation abzuschließen, installieren wir jetzt PHP7.4

PHP ist eine serverseitige Programmiersprache. Damit können Befehle direkt auf dem Server ausgeführt werden, z.B. werden Datenbankabfragen häufig über PHP durchgeführt.

```
sudo apt install php7.4 php7.4-cli php7.4-common php7.4-curl php7.4-gd php7.4-intl php7.4-json php7.4-mbstring php7.4-mysql php7.4-opcache php7.4-readline php7.4-xml php7.4-xsl php7.4-zip php7.4-bz2 libapache2-mod-php7.4 -y
```

Du hast jetzt erfolgreich PHP 7.4 mit Modulen installiert!

Installation von PHP 8.0

Um unsere Installation abzuschließen, installieren wir jetzt PHP7.4

PHP ist eine serverseitige Programmiersprache. Damit können Befehle direkt auf dem Server ausgeführt werden, z.B. werden Datenbankabfragen häufig über PHP durchgeführt.

```
sudo apt install php8.0 php8.0-cli php8.0-common php8.0-curl php8.0-gd php8.0-intl php8.0-mbstring php8.0-mysql php8.0-opcache php8.0-readline php8.0-xml php8.0-xsl php8.0-zip php8.0-bz2 libapache2-mod-php8.0 -y
```

Du hast jetzt erfolgreich PHP 8.0 mit Modulen installiert!

Optional: Installation von phpMyAdmin

Du kannst optional auch phpMyAdmin installieren, um deine Datenbank über eine Weboberfläche zu verwalten.

Als Erstes wechselst du in das Verzeichnis, in dem du phpMyAdmin ablegen möchtest.

```
cd /var/www
```

Nun lädst du das verpackte Archiv mit den Dateien für phpMyAdmin herunter.

```
wget https://www.phpmyadmin.net/downloads/phpMyAdmin-latest-all-languages.zip -O phpmyadmin.zip
```

Als Nächstes entpackst du das ZIP Archiv und löschst das alte Verzeichnis.

```
sudo unzip phpmyadmin.zip && sudo rm phpmyadmin.zip
```

Jetzt veränderst du den Namen des Ordners in phpMyAdmin.

```
mv phpMyAdmin-*-all-languages phpmyadmin
```

Jetzt veränderst du die Berechtigungen auf das Verzeichnis.

```
sudo chmod -R 0755 phpmyadmin
```

Und damit wir dann über den Webbrowser auf phpMyAdmin zugreifen können, erstellen wir eine Konfigurationsdatei für den Apache2 Webserver.

```
sudo nano /etc/apache2/conf-available/phpmyadmin.conf
```

Dort fügst du folgende Konfiguration ein.

```
Alias /phpmyadmin /var/www/phpmyadmin

<Directory /var/www/phpmyadmin>
    Options SymLinksIfOwnerMatch
    DirectoryIndex index.php
</Directory>

<Directory /var/www/phpmyadmin/templates>
    Require all denied
</Directory>

<Directory /var/www/phpmyadmin/libraries>
    Require all denied
</Directory>

<Directory /var/www/phpmyadmin/setup/lib>
    Require all denied
</Directory>
```

Sobald du mit der Tastenkombination **STRG + X** und danach **Y** die Datei gespeichert hast, aktivieren wir jetzt die Konfigurationsdatei.

```
sudo a2enconf phpmyadmin && sudo systemctl reload apache2
```

Um die Installation abzuschließen, erstellen wir temporäres Verzeichnis im **phpMyAdmin Verzeichnis** und vergeben für dieses die entsprechende Berechtigung.

```
sudo mkdir /var/www/phpmyadmin/tmp/ && sudo chown -R www-data:www-data /var/www/phpmyadmin/tmp/
```

Du kannst dich jetzt mit den entsprechenden Datenbankbenutzern anmelden. Wenn sich **phpMyAdmin** auf demselben Server wie die Datenbank befindet, musst du keinen Server angeben und du kannst dich mit einem Benutzer anmelden, mit dem Host **localhost**. Du kannst dich also auch als **root** anmelden.

Revision #3

Created 16 March 2022 10:14:39 by Phillip U.

Updated 14 May 2024 08:05:17 by Phillip U.