

Tutorial: Raspberry Pi - Webserver Apache | PHP | MariaDB | PHP MyAdmin

Tutorial: Raspberry Pi als Webserver mit Apache, PHP, MariaDB und phpMyAdmin installieren und einrichten.

Webserver - Installation und Grundkonfiguration von Apache2, PHP7, Datenbankserver MariaDB und phpMyAdmin auf dem Raspberry Pi.

Da der Raspberry Pi ja praktisch wie ein vollwertiger PC arbeitet, drängt sich eine Nutzung als kleiner Webserver mit PHP und Datenbankserveranbindung geradezu auf. Bis Raspbian Jessie wurde noch die klassische Konfiguration mit PHP5 und MySQL als Datenbankserver unterstützt.

Seit Raspbian Stretch werden MariaDB als Datenbank und PHP7 als Scriptinterpreter verwendet.

In diesem Tutorial möchte ich Euch die Einrichtung von Apache, PHP7, MariaDB und PHP MyAdmin als Webserver auf dem Raspberry Pi beschreiben. Am Ende dieser Anleitung haben wir auf unserem Raspberry Pi einen voll funktionsfähigen Webserver zu laufen.

## Voraussetzungen

Um unseren Raspberry Pi zu einem vollwertigen Webserver zu machen, benötigen wir zunächst einige Voraussetzungen.

Der Raspberry Pi sollte über folgende Elemente verfügen:

1. Grundlagen-Tutorial: Tutorial - Raspberry Pi Betriebssystem einrichten
2. Grundlagen-Tutorial: Tutorial: onboard Netzwerkkarten (LAN, WLAN) einrichten
3. Grundlagen-Tutorial: Tutorial: Fernsteuerung mit SSH und XRDP
4. Aktualisierung (Update und Upgrade) sollte durchgeführt sein

Unser Raspberry Pi sollte jetzt also über ein fertig eingerichtetes Betriebssystem, eine funktionierende Internetverbindung und die Möglichkeit zur Fernsteuerung über SSH und/oder RDP verfügen. Wenn Sie den Pi direkt mit Maus/Tastatur und Bildschirm betreiben, können Sie den Schritt Fernsteuerung mit SSH und XRDP überspringen.

## Webserver Apache 2 installieren

Als Webserver bietet sich der klassische und bewährte Apache an. Dieser Webserver lässt sich ganz einfach per Kommandozeile installieren.

Um die Kommandozeile einzugeben stellen wir entweder eine SSH-Verbindung zu unserem Pi her oder öffnen im Desktop (direkt am Bildschirm oder über RDP) ein Terminalfenster.

Das Kommando für die Installation des Apache lautet wie folgt:

```
pi@rechnername: ~ $sudo apt-get install apache2 -y
```

Die Installation geht relativ schnell vonstatten.

Um die Funktion des Webserver zu testen, rufen Sie einfach auf dem Desktop des Pi den Webbrowser auf und schreiben dort in die Adressleiste das Wort: localhost (enter). Jetzt sollte sich eine Startseite des Apache Webservers öffnen.

Alternativ dazu können Sie von einem PC im Netzwerk aus auch einen Webbrowser öffnen und dort in die Adressleiste die Zeile http://ip-adresse-des-raspberry-pi (enter) eingeben. Wenn sich jetzt die Startseite des Apache im Webbrowser zeigt, ist der Apache fertig installiert.

Anschliessend wechseln wir gleich zum nächsten Schritt, der Installation der Datenbank MariaDB.

### Datenbank MariaDB Server und Client installieren und einrichten

Der nächste Schritt auf dem Weg zu unserem Webserver ist jetzt die Installation des Datenbank Servers und Clients. Als Software kommt hier MariaDB zum Einsatz.

Die Installation ist auch hier ganz einfach und erfolgt mit der Eingabe der folgenden Kommandozeile:

```
pi@rechnername:~ $sudo apt-get install mariadb-server mariadb-client -y (enter)
```

Nach der Installation startet wir noch das folgende Kommando zur Konfiguration des Datenbankservers:

```
pi@rechnername:~ $sudo mysql_secure_installation (enter)
```

Auf dieses Kommando hin werden uns ein paar Fragen gestellt:

Als erstes werden wir nach dem aktuellen Passwort gefragt. Hier geben wir kein Passwort ein, sondern bestätigen die Frage nur mit der Entertaste.

Die zweite Frage danach, ob wir das Root-Passwort ändern möchten, bestätigen wir mit y (enter) für yes oder ja. Daraufhin werden wir gebeten, ein Root-Passwort festzulegen (das machen wir dann auch). Achtung, während der Eingabe des Passwortes werden aus Sicherheitsgründen die eingegebenen Zeichen nicht angezeigt.

Anschließend werden wir noch gefragt, ob wir den anonymen User löschen möchten. Auch hier bestätigen wir wieder mit y (enter) für: Ja, wir möchten diesen User löschen.

Die folgende Frage, ob wir den externen (von einem anderen Rechner) Root-Logins auf diesen Datenbankserver sperren möchten, beantworten wir auch wieder mit y (enter) und verbieten damit den externen Zugriff mit Root-Rechten.

Auch die letzten zwei Fragen danach, ob wir die Testdatenbank löschen möchten und nach der Aktualisierung der Rechte beantworten wir jeweils mit y (enter).

Damit haben wir unseren MariaDB-Datenbankserver erfolgreich fertig konfiguriert und er ist einsatzbereit. Der nächste Schritt widmet sich der Installation von PHP7.

## PHP 7 installieren

Damit wir auch dynamische Webseiten und Scripte erstellen und verarbeiten können, benutzen wir die Scriptsprache PHP, die wir in diesem Schritt installieren möchten. Die Kommandozeile hierfür lautet wie folgt:

```
pi@rechnername:~ $sudo apt-get install php -y (enter)
```

Nach der PHP-Installation müssen wir jetzt noch den Apache neu starten, damit er seine neuen Möglichkeiten auch kennt und zu nutzen lernt das geht so:

```
pi@rechnername:~ $sudo systemctl restart apache2 (enter)
```

Jetzt sind grundsätzlich Apache Webserver, PHP-Interpreter und MariaDB Datenbanksystem fertig eingerichtet.

Das Web-Basisverzeichnis findet Ihr standardmäßig unter /var/www/html.

Der nächste Schritt führt uns durch die Installation von phpMyAdmin als hilfreiches Werkzeug im Umgang mit dem Datenbankserver.

## phpMyAdmin Installation und Konfiguration

Die Installation von phpMyAdmin erfolgt wieder ganz einfach über die folgende Kommandozeile:

```
pi@rechnername:~ $sudo apt-get install phpmyadmin -y (enter)
```

Während der Installation von phpMyAdmin erscheinen mehrere Auswahlfenster. In der ersten Auswahl werden wir gefragt, für welchen Webserver wir phpMyAdmin einrichten wollen. Hier wählen (Häkchen setzen mit der Leertaste) wir den Eintrag 'Apache2' und drücken dann die Enter-Taste.

Im nächsten Auswahlfenster lassen wir einfach das Ja ausgewählt und bestätigen die Eingabe durch das Drücken der Enter-Taste.

Jetzt sollen wir ein Passwort für den phpMyAdmin-Standardbenutzer festlegen. Nach Eingabe des Passwortes bestätigen wir wieder mit der Enter-Taste.

Hinweis: Wenn wir kein Passwort festlegen, erzeugt die Installationsroutine ein zufälliges Passwort für diesen Benutzer.

Da der Root-Zugriff auf den MariaDB-Server aus Sicherheitsgründen verboten ist, müssen wir im nächsten Schritt noch einen Admin-User für den Datenbankserver erzeugen.

## Erzeugen eines Admin-Users für den MariaDB-Server

Damit wir administrativen Zugriff auf unseren Datenbankserver bekommen (wir melden uns damit zum Beispiel beim phpMyAdmin an), müssen wir noch einen Benutzer anlegen, der administrative Rechte in der Datenbank

bekommt. Dazu loggen wir uns wie folgt in den Datenbankserver ein.

Wir öffnen ein Terminal (SSH oder RDP oder direkt am Desktop des Raspberry Pi) und loggen uns mit der folgenden Kommandozeile in unserem Datenbankserver ein:

```
pi@rechnername:~ $sudo mysql -u root (enter)
```

Danach gebt Ihr bitte die folgenden Befehle an der DB-Konsole ein:

```
CREATE USER ihr_username@localhost IDENTIFIED BY 'ihr_eigenes_passwort'; (enter)
```

Merkt Euch diesen Benutzernamen und das Passwort gut, da Ihr Euch nur damit am phpMyAdmin einloggen könnt, um Datenbanken zu erzeugen und/oder zu bearbeiten.

Als nächstes bekommt unser neuer Benutzer noch Admin-Rechte im Datenbankserver. Das geht dann so:

```
GRANT ALL PRIVILEGES ON *.* TO ihr_username@localhost WITH GRANT OPTION; (enter)
```

Natürlich müsst Ihr für die Zeichenketten 'ihr\_username' und 'ihr\_eigenes\_passwort' Euren gewünschten Benutzernamen und das Passwort für die Datenbank austauschen, die Ihr dann auch benutzen wollt.

Am Ende verlassen wir dann die Datenbank-Konsole durch die Eingabe folgender Zeile:

```
exit (enter)
```

Solltet Ihr eine Fehlermeldung bekommen, die in etwa so beginnt:

```
ERROR 1064 (42000): You have an error in your SQL syntax...
```

dann liegt das wahrscheinlich daran, dass Ihr aus der Ferne (per SSH) auf den Pi zugreift. Da wir externen Root-Zugriff verboten haben, muss also die Eingabe direkt an einer Tastatur oder per Fernsteuerung (RDP oder VNC) direkt über den Desktop des Pi vorgenommen werden.

Nachdem der Benutzer erfolgreich erstellt und mit Rechten versehen wurde, müssen wir nun nur noch den Webserver neu starten. Das geht wieder mit:

```
pi@rechnername:~ $sudo systemctl restart apache2
```

Nachdem Neustart sollte also auch der phpMyAdmin im Browser zu erreichen sein. Zum Testen tippt Ihr einfach in die Adressleiste des Browsers folgende Zeile ein: `http://ip_des_raspberry_pi/pypmyadmin` (enter). Jetzt sollte nach einer kurzen Ladezeit der Login-Bildschirm des phpMyAdmin angezeigt werden, bei dem Ihr Euch dann auch mit Eurem neu erzeugten User und dem zugewiesenen Passwort anmelden könnt.

## Installation des Python Mysql-Connectors

Da Python die bevorzugte Scriptsprache für den Raspberry Pi ist, macht es natürlich auch Sinn, dass Python sich mit der Datenbank verbinden kann, um dort Daten auszulesen oder zu speichern.

Python wird derzeit oft noch mit der (alten) Version 2, immer öfter jedoch mit der aktuellen Version 3 verwendet.

Beide Versionen sind nicht untereinander kompatibel.

Damit beide Versionen funktionieren, müssen wir auch für jede Version einen Mysql-Connector installieren. Zunächst Updaten und Upgraden wir noch unseren Paketkatalog und die installierten Pakete. Das funktioniert wie folgt:

```
pi@rechnername: ~ $ sudo apt-get -y update (enter)
pi@rechnername: ~ $ sudo apt-get -y upgrade (enter)
pi@rechnername: ~ $ sudo apt-get -y install python-mysql.connector (enter)
pi@rechnername: ~ $ sudo apt-get -y install python3-mysql.connector (enter)
pi@rechnername: ~ $ sudo apt-get -y install python-mysqldb (enter)
pi@rechnername: ~ $ sudo apt-get -y install python3-mysqldb (enter)
```