# Webserver

## 1. Allgemeines

Diese Anleitung wurde nach den Grundlagen-Anleitungen (ehemals Streamingclient) geschrieben und enthält nur die zusätzlichen Infos. Auf die bereits bekannten Dinge werde ich verweisen.

**Bitte vorher die Grundlagen ansehen!**

**Hier die Themen, die in dieser Anleitung vorkommen werden:**

1. Allgemeines
2. Apache 2 als Webserver
3. PHP5 wegen Nutzung von Content-Management-Software
4. MySQL wegen Nutzung von Content-Management-Software
5. phpMyAdmin zur Verwaltung von MySQL
6. FTP-Server zum Upload
7. Joomla 3.15 als Content-Management-Software
8. Nutzung von Libre-Office
9. Kompletter Webserver mit Lighttpd, PHP, SQLite, WordPress und SearchWP

Noch einmal etwas zu meinen Kenntnissen: Ich habe von Linux fast keine und von Python gar keine Ahnung! Das hier verbreitete „Wissen" stammt fast komplett aus irgendwelchen Beiträgen im Internet und ist dementsprechend durchwachsen! 🙂

Diese Anleitung dient hauptsächlich der Nachbausicherheit bei gleicher Hard- und Software. Wer Fehler bekommt, die vermutlich nicht auf Fehler in meinen Beschreibungen zurückzuführen sind, sollte lieber gleich nach Beiträgen im Internet suchen!

In meiner weiteren Anleitung müssen immer wieder Linux-Befehle eingegeben werden. Groß-/Kleinschreibung sind bei Linux genauestens einzuhalten! Jedem Befehl stelle ich ein **$** vorweg, welches **nicht** mit eingegeben werden darf!

**Für alle meine Einstellungen und meine Vorgehensweisen gilt: Viele Wege führen nach Rom!** 🙂

**Los geht's!**

## 2. Installation der Software Apache 2

Es gibt diverse Möglichkeiten für eine derartige Software, aber ich habe mich für den Standard „Apache 2" entschieden. Sollte ich später Performanceprobleme beim Raspberry feststellen, muss ich wechseln.

Ich gehe davon aus, dass die Installation des Wheezy-Raspbian und des RDP-Servers bereits nach meinen Anleitungen auf der ersten Seite erfolgt ist und man die ersten Erfahrungen beim Editor „nano" gemacht hat.

Um die Installation zu starten, gibt man einfach folgende Befehle beim Raspberry ein:

```
$sudo bash
$groupadd webusergroup
$usermod -a -G webusergroup www-data
$apt-get install apache2
$nano /etc/apache2/apache.conf
```

In dieser Datei muss ganz am Ende folgende Zeile eingetragen werden:

ServerName localhost

Es folgt ein Neustart:

$reboot

Anschließend gibt man im Browser seiner Wahl die IP-Adresse des Raspberry ein und wenn alles richtig gemacht wurde, bekommt man die Standardseite mit

**„It works!"**

**Weiter mit PHP 5 —>**

### 3. Installation von PHP5

Für ein aktuelles Content-Management-System und auch für jede andere moderne Internetseite ist PHP angesagt!

Bitte beim Raspberry eingeben:

```
$sudo bash
$apt-get install php5
```

Das ist schon alles! Die Installation wird wie folgt geprüft:

```
$cd /var/www
$nano phpinfo.php
```

```
<?php
phpinfo ();
?>
```

```
$reboot
```

Nach dem Neustart wird als Test die IP-Adresse des Raspberry + „/phpinfo.php" in einem Browser eingegeben (z. B. „http://192.168.1.123/phpinfo.php") und man bekommt eine lange Auflistung von Daten, wenn alles richtig funktioniert!

**Weiter mit MySQL —>**

## 4. Installation von MySQL

Im privaten Bereich geht kaum etwas an MySQL vorbei, wenn es sich um Datenbanken dreht. Aber auch Firmen nutzen es sehr häufig! Für Internetnutzung sehe ich es als Standard bei den Datenbanken.

Zur Installation gibt man beim Raspberry folgendes ein:

```
$sudo bash
$apt-get install mysql-server mysql-client php5-mysql
```

Generell sollte man immer Passworte an geeigneter Stelle deponieren! Das hier verlangte neue SQL-Admin-Passwort ist extrem wichtig! Aber genauso wichtig ist es, den zugehörigen Usernamen mit der gleichen Aufmerksamkeit zu behandeln!

```
$reboot
```

Das war’s hier!

**Weiter mit phpMyAdmin —>**

## 5. Installation von phpMyAdmin

Dieses Produkt ist zur Administration von MySQL gedacht. Auch hier ist es wieder ein Standardprodukt.

Wie immer geht es wieder um Eingaben beim Raspberry:

$sudo bash
$apt-get install libapache2-mod-auth-mysql phpmyadmin

„Apache 2" auswählen.
Die Datenbanken automatisch erstellen lassen.
Das erste Passwort ist das bei der MySQL-Installation vergebene Passwort. Danach denkt man sich ein Passwort zur Nutzung von PHPmyAdmin aus. (Dieser User hat allerdings nicht das Recht neue Datenbanken anzulegen. Das geht nur mit „root".)
Dann muss noch phpMyAdmin mit Apache zusammenarbeiten, indem man eine Datei editiert:

$nano /etc/php5/apache2/php.ini

Im Editor wird diese Zeile an das Dateiende angehangen:

extension=mysql.so

Für einen Test ruft man den Raspberry im Browser auf. Hier wird als Beispiel die IP-Adresse „192.168.1.10" genutzt:

http://192.168.1.10/phpmyadmin

Die Anmeldung geschieht entweder mit „root" oder „phpmyadmin" mit dem passenden Passwort. Für die nächste Installation braucht aber keine Datenbank vorbereitet zu werden.

**Weiter mit FTP —>**

## 6. Installation des FTP-Servers

Einen FTP-Server braucht man z. B. dann, wenn man Dateien zum Webserver hochladen will. Ich nutze auf der Clientseite (ein Windows-PC) Filezilla.

Wieder beginnt die Installation mit dem Befehl „apt-get install", den man beim Raspberry eingibt:

```
$sudo bash
$apt-get install proftpd
$nano /etc/proftpd/proftpd.conf
```

Es muss wieder einmal Text am Ende einer Datei eingegeben werden:

```
DefaultRoot ~
AuthOrder mod_auth_file.c mod_auth_unix.c
AuthUserFile /etc/proftpd/ftpd.passwd
AuthPAM off
RequireValidShell off
```

Nach dem Abspeichern geht es weiter:

```
$cd /etc/proftpd
$ftpasswd –passwd –name piforum –uid 33 –gid 33 –home /var/www/ –shell /bin/false
$chmod g+s /var/www
$chmod 775 /var/www
$chown -R www-data:webusergroup /var/www
$reboot
```

Das wäre auch erledigt!

**Weiter mit Joomla —>**

## 7. Installation von Joomla

Es ging um eine Entscheidung zwischen Joomla und WordPress. Deshalb hatte ich Joomla auf dem Raspberry installieren wollen. Aber gleich nach der Installation ging schon das Installieren von Templates nicht. Wer also Fehler bei der Installation findet, bitte melden! (Auf einem Server meines Providers hat es funktioniert.) Vielleicht liegt der Fehler schon in den vorangegangenen Installationen.

In diesem Fall gibt es kein Installations-Paket für den Raspberry. Ich habe zuerst auf einem Windows-PC Joomla heruntergeladen und entpackt. Dann habe ich mich mittels Filezilla mit dem Raspberry verbunden. Da ich das mit dem User „Pi" mache, habe ich nicht genug Rechte, um alles in das endgültige Verzeichnis zu kopieren. (Wie erwähnt habe ich nur minimale Kenntnisse bei diesen Themen.) Insofern habe ich das entpackte Joomla nach /pi/home/joomla315 kopiert. Anschließend habe ich mich mittels RDP am Raspberry angemeldet und alles nach /var/www kopiert:

```
$sudo bash
$cd joomla315
$cp -r . /var/www
$cd ..
$rm -r joomla315
$cd /etc/php5/apache2
$sudo nano php.ini
```

Mittels <Strg>+W sucht man nach „output_buffering". Der Abschnitt wird wie folgt geändert:

output buffering
Default Vaule: off
Development Value: 0
Production Value: 0

Nach dem Abspeichern wird der Raspberry neu gestartet.

```
$sudo reboot
```

Nach dem Reboot:

```
$sudo bash
$cd /var/www
$touch configuration.php
$chmod 777 configuration.php
$rm index.html
```

In einen Webbrowser gibt man die IP-Adresse des Raspberry ein und bekommt die erste Joomla-Seite zur Konfiguration. Diese und die folgenden Seiten sind nach Joomla-Anleitung (oder intuitiv) auszufüllen. Sie enden hiermit:

Ich musste beim Raspberry das Verzeichnis manuell löschen:

$rm -r installation

Wie schon anfangs erwähnt, geht bei mir die Template-Installation und auch die Mail nicht. Bei der Mail muss man mal mit den verschiedenen Mailing-Sorten spielen. Beim Template habe ich derzeit nichts gefunden. Da die Sache eilt, habe ich auf den Raspberry verzichtet und nutzt meine Möglichkeiten über meinen Hoster/Provider. Aber wer Vorschläge hat, wie ich den Raspberry doch noch mit Joomla 3.x flott bekomme, bitte mailen!

**Weiter mit Libre Office —>**

## 8. Installation von LibreOffice

Zu Testzwecken für meine Kräuterdatenbank brauchten wir eine gemeinsame Grundlage um Daten zu editieren. Uns beiden war Excel am Geläufigsten und so wurde nach einer preiswerten Alternative gesucht, da nur ein Excel vorhanden war. Das war wieder einmal die Stunde des Raspberry!

apt-get install LibreOffice

und alle zusätzlichen Pakete bestätigen. Ich habe das per RDP gemacht und der Raspberry tat mir echt leid, als ich die Auslastung im Taskmanager gesehen hatte! Ich habe dann nach einigen Minuten die Software gestartet. Allerdings war alles in Englisch, was ich gerne auf Deutsch geändert hätte. Ein Blick ins Internet ergab dieses:

apt-get install libreoffice-l10n-de

Nun musste man zwar noch über Tools und Language alles manuell auf Deutsch umstellen und einmal gabs auch Mecker, aber danach hatte es funktioniert. Und das Calc macht erstmal einen guten Eindruck! Die Test im Echtbetrieb dauern noch an…

**Weiter mit Neuinstallation des Webservers mit anderen Tools —>**

## 9. Kompletter Webserver mit Lighttpd, PHP, SQLite, WordPress und SearchWP

Für meine Kräuterdatenbank und deren Suchfunktion habe ich den Raspi zu einem Webserver gemacht, mit dem ich auch PDF-Dateien durchsuchen kann. Auch wenn die Performance nicht so toll ist, reicht es mir vorerst.

Da ich die Installation von Webserverkomponenten schon in den vorherigen Seiten beschrieben habe, gibt es dazu nur eine kurze Doku.

Bis auf SearchWP ist die Software kostenlos. Aber ich hatte keine Software gefunden, die PDFs so durchsucht, dass ich zu brauchbaren Ergebnissen komme.

Zuerst wird das aktuelle Raspbian heruntergeladen und per WIN32DiskImager auf die SD-Karte gebracht. Aus meinen bisherigen Erfahrungen heraus und aus Performancegründen nutze ich eine SanDisk-8GB-Extreme-Karte. Solange die Karte noch am PC steckt, ändert man mittels Editor die **\config.txt**, indem man dort mittels Notepad++ einen Eintrag hinzufügt, was aber auch später mittels Nano funktionieren würde:
gpu_mem=16

Nach dem Booten von dieser Karte macht man die gängigen Internationalisierungs-Einstellungen, wie ich sie hier für Punkt 4 beschrieben habe. Auch das Ändern des Passworts und das Erweitern des Filesystems sind dort beschrieben. Nach einem Reboot kann ein Update und Upgrade nicht schaden.

Jetzt wird die nötige Software installiert und konfiguriert. Ich nutze ein lokal angeschlossenes Keyboard, man kann aber auch über Putty arbeiten.
$sudo bash
$apt-get install lighttpd php5-cgi php-apc php5
$lighty-enable-mod fastcgi
$lighty-enable-mod fastcgi-php
$service lighttpd force-reload
$apt-get install sqlite3 php5-sqlite

Anschließend wird der Seitencache APC durch das Editieren der Datei **/etc/php5/conf.d/20-apc.ini** mittels nano konfiguriert:
extensions=apc.so
apc.enabled=1
apc.file_update_protection=2
apc.optimization=0
apc.shm_size=32M
apc.include_once_override=0
apc.shm_segments=1
apc.gc_ttl=7200
apc.num_files_hint=1024
apc.enable_cli=0

Ein Neustart von lighttpd aktiviert diese Einstellungen und dann folgen weitere Befehle zur Installation und Konfiguration von WordPress:
$service lighttpd restart

```
$cd /var/www
$wget http://wordpress.org/latest.zip
$unzip latest.zip
$chown -R www-data:www-data wordpress
$cd wordpress
$cp wp-config-sample.php wp-config-php
$nano wp-config.php
```

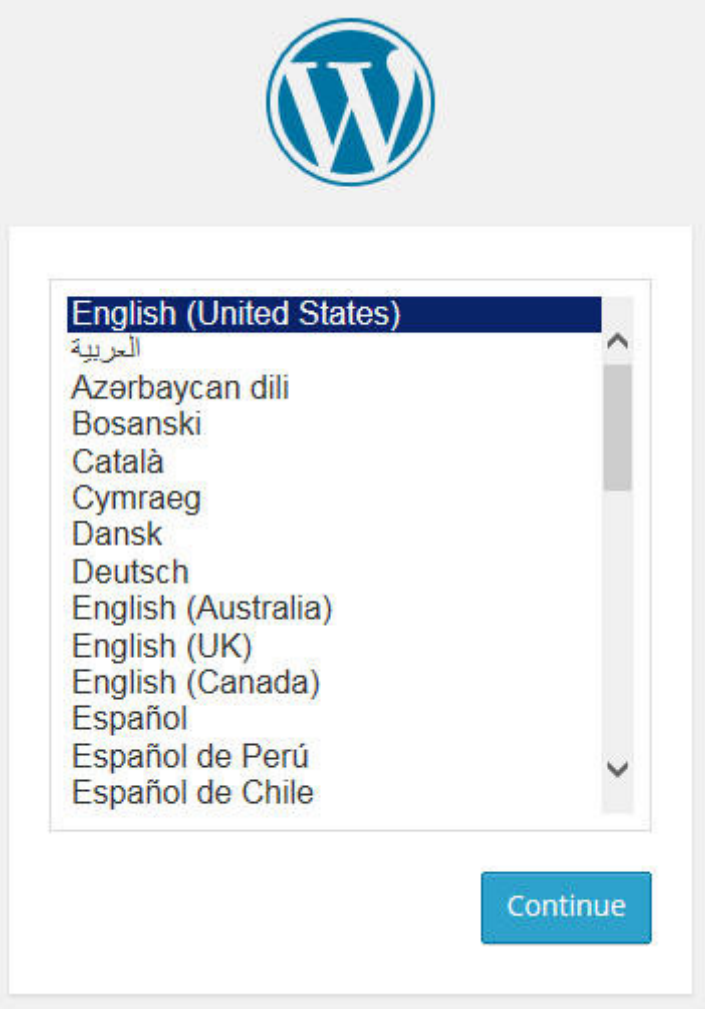
Unter die Zeile mit „define(‚DBCollate…" muss folgende Zeile eingefügt werden:
define(‚DB_TYPE‘, ’sqlite‘);

```
$cd wp-content/plugins
$wget http://downloads.wordpress.org/plugin/sqlite-integration.1.7.zip
$unzip sqlite-integration.1.7.zip
$cd sqlite-integration
$cp db.php ../..
```

**Da ich WordPress bisher noch nicht beschrieben hatte, erfolgt es hier etwas ausführlicher, als der Rest.**

Endlich ist der Raspi bereit, die Installation von WordPress auszuführen. Dazu gibt man diesen Link in einen Browser eines PCs ein, wobei die IP-Adresse die des Raspi sein muss:
http://ip-adresse/wordpress/wp-admin/install.php

Es erscheint das erste Bild der Installation (in der Version 4.0) und man wählt seine Installations-Sprache aus:




Da nichts in Stein gemeißelt und vieles selbsterklärend ist, gehe ich nur auf Besonderheiten ein. Der weitere Ablauf:

## Willkommen

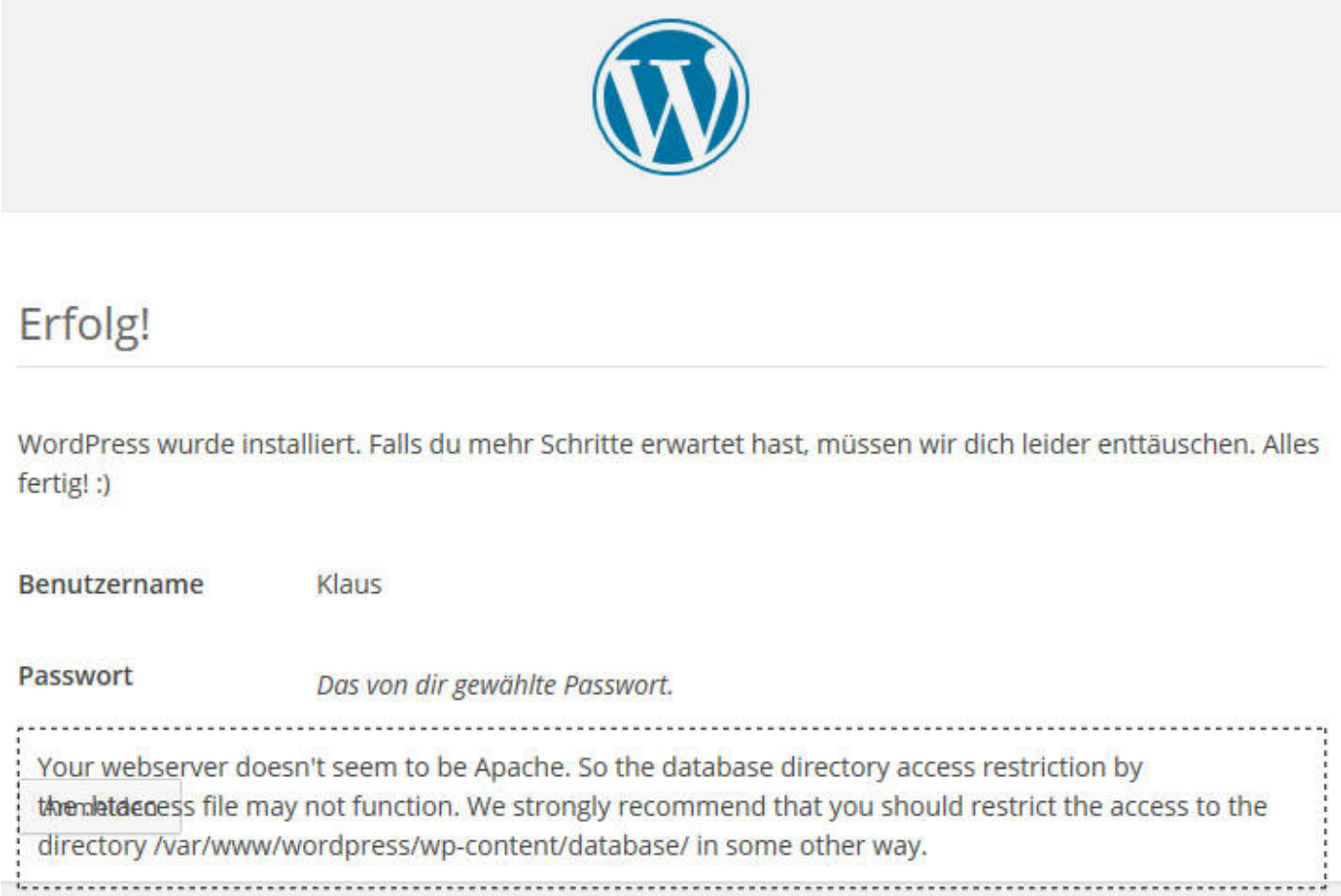
Willkommen zur berühmten 5-Minuten-Installation von WordPress! Gebe unten einfach die benötigten Informationen ein, um danach das mächtigste und flexibelste Weblog-System der Welt benutzen zu können.

## Benötigte Informationen

Bitte trage die folgenden Informationen ein. Keine Sorge, du kannst all diese Einstellungen später auch wieder ändern.

Die Fertigmeldung kommt gut, wenn nur die abschließende Englische Bemerkung nicht gewesen

wäre:

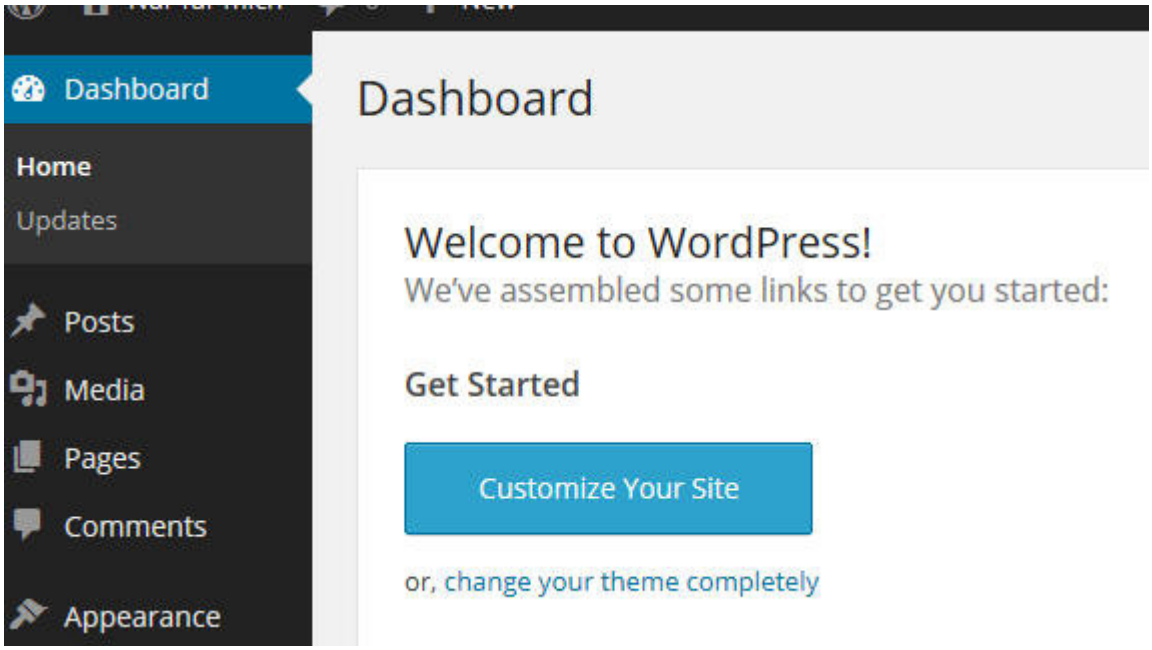



Ich ignoriere das Sicherheitsleck vorerst (und auch später), weil nicht so schnell heraus zu finden war, ob und wie sqlite mit einer htaccess umgehen kann. Angeblich kann man aber Dateien, die mit einem Punkt anfangen, nicht so einfach aufrufen und die Datenbank beginnt mit einem Punkt.

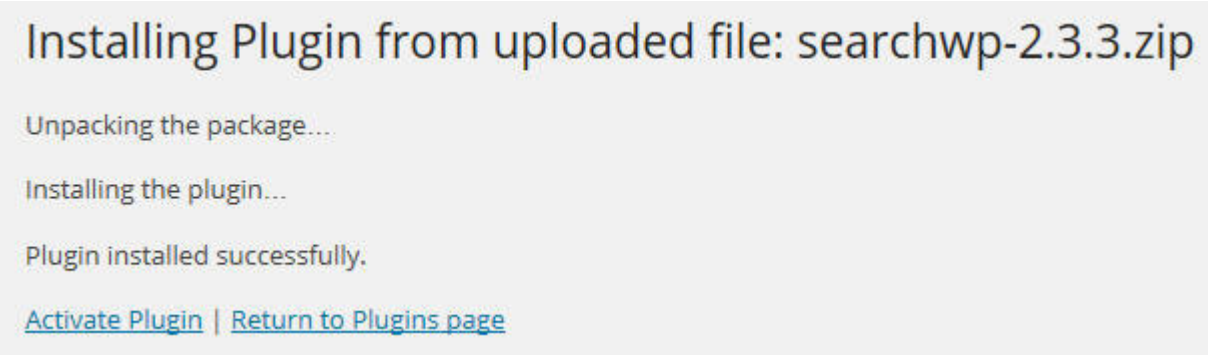
Ich gebe also als Nächstes dieses in meinen Browser ein:
http://ip-adresse/wordpress

Bei der richtig eingegebenen IP-Adresse erscheint das erste Bild von WordPress im Normal-User-Modus. Ich wechsle auf den Admin-Modus:
http://ip-adresse/wordpress/wp-admin

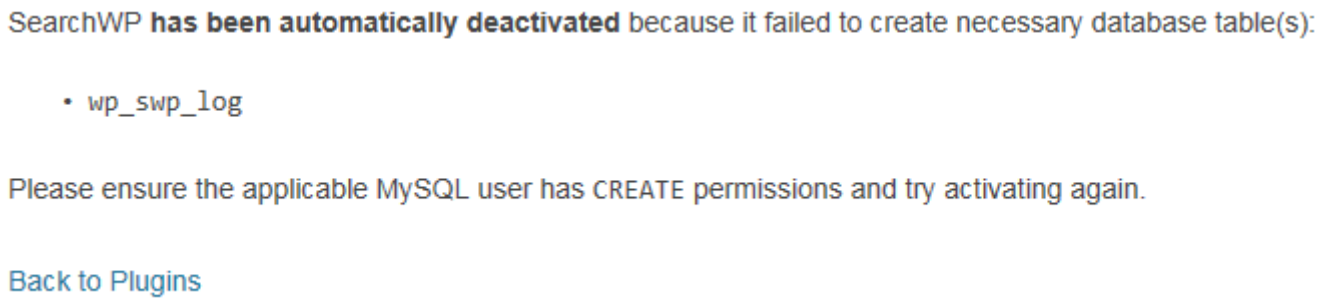
Nachdem ich pfiffig genug war, mir Name und Passwort aufzuschreiben und auch noch richtig einzugeben, erscheint:



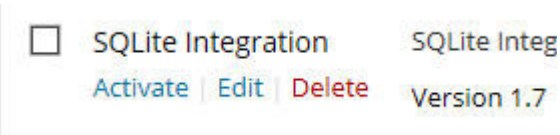

Wie man nun WordPress konfiguriert, sprengt tatsächlich den Rahmen meiner Seiten. Ich baue nur noch das Plugin SearchWP ein, was ich vor einigen Wochen bereits erhalten hatte. Dazu nutzt man links „Plugins", dann „Add New" und „Upload Plugin":




Über „Activate Plugin" wird es gestartet und – UPS – die nächste Fehlermeldung!



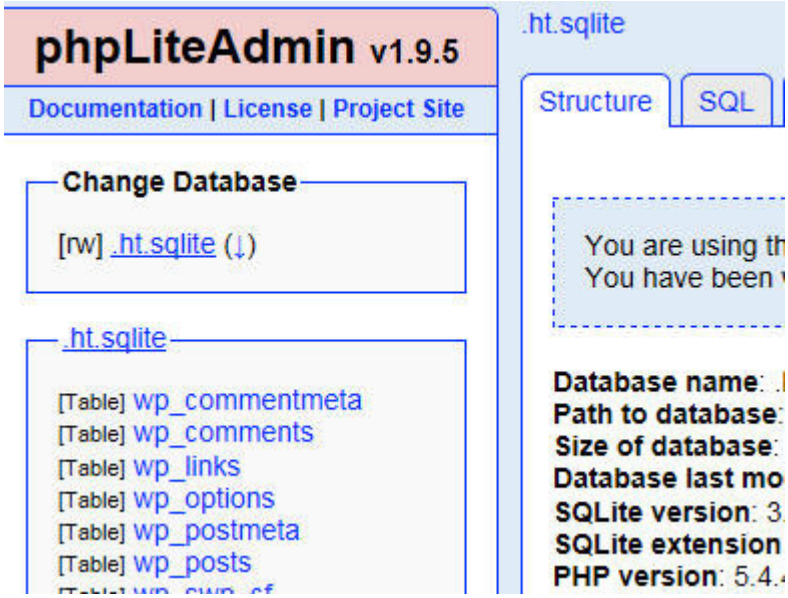

Daraufhin habe ich in den Plugins erst einmal die SQL-Integration aktiviert, aber das alleine hat es nicht gebracht, was auch schon vorher klar war.




Zur Verwaltung von SQLite lade ich mir ein Tool:
cd /var/www/wordpress/wp-content/plugins
wget https://phpliteadmin.googlecode.com/files/phpliteAdmin_v1-9-5.zip
unzip phpliteAdmin_v1-9-5.zip

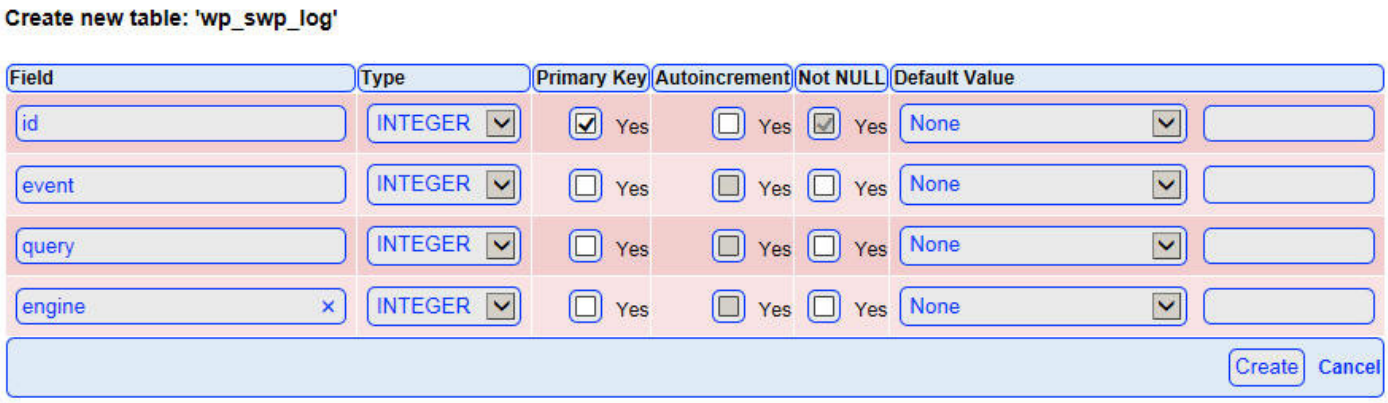
Dort muss man in der Datei phpliteadmin.php mittels nano den Pfad zur Datenbank mitteilen. Der Eintrag beginnt mit „$directory =…“ und muss bei mir so lauten:
$directory = ‚/var/www/wordpress/wp-content/database‘;

Jetzt kann man die Datenbank konfigurieren, in dem man im PC-Browser diesen Link aufruft:
http://ip-adresse/wordpress/wp-content/plugins/phpliteadmin.php

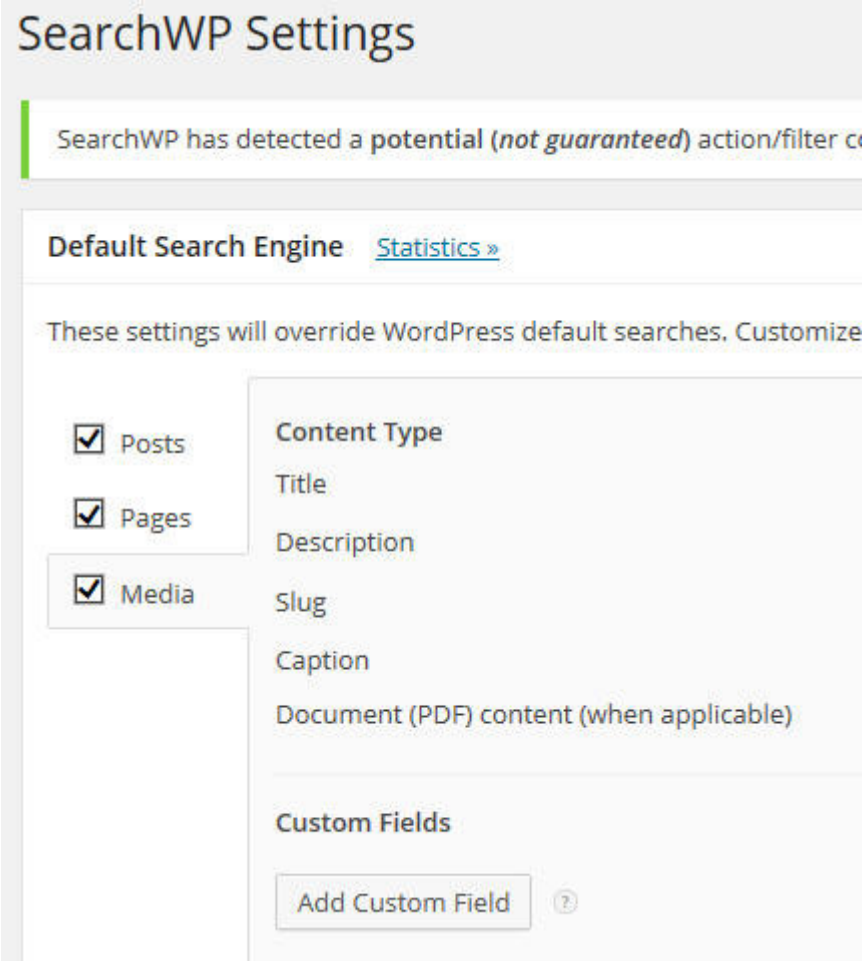



Die Tabelle wp_sw_log gibt es tatsächlich noch nicht. Die Berechtigung des DB-Users kann ich nicht finden – nicht einmal seinen Namen! Für zukünftige Projekte ist es zu überlegen, ob man nicht doch beim nächsten Mal MySQL nutzt!

Ich lege die Tabelle einfach mal nach den Infos an, die ich im Internet finden konnte:




Zumindest ist das darauf folgenden Aktivieren des Plugin durchgelaufen! Die Konfiguration von SearchWP mittels „Settings“ und „SearchWP“:




Der Haken bei „Media" ist der entscheidende für PDF-Dateien!

Ich werde nun mit WordPress und SearchWP experimentieren und schließe diese Seite. Weitere Probleme oder Fehler kann ich derzeit nicht ausschließen!