

Die Web-Server Survival Checklist

1.) Was man braucht

- **Einen Webserver zur Miete.**
Für einfache Einzelseiten reichen Pauschalangebote mit Unterstützung für PHP und MySQL.
Wer flexibel sein möchte, mietet einen Rootserver. Virtuelle Rootserver bieten die gleiche Funktionalität wie physische Angebote.
- **Apache 2, PHP5 und MySQL5.**
Weitere Werkzeuge wie SSH und Perl sind auf Linux-Webservern Standardmäßig mitinstalliert. Andere Serversoftware wie alternative Datenbanken (PostgreSQL, SQLite etc.), Umgebungen für diverse Programmiersprachen wie C++ oder Java und tausende weitere Tools lassen sich leicht nachinstallieren. Alle relevanten Softwarepakete sind unter freien Lizenzen verfügbar. Kein relevantes Paket muss aus den Quellcodes installiert werden.

2.) System Setup und Bedienung

- Apache und MySQL sind die eigentlichen Server(Dienste), beide müssen ständig laufen.
Unter Debian und UBUNTU starten die Dienste schon bei der Installation. Per Hand lassen sich alle Dienste unter UNIX mit Skriptprogrammen unter /etc/init.d bedienen. Der Befehl:

```
:~$ /etc/init.d/mysql start
```

startet MySQL. Alle Skripte kennen die Optionen „start“ und „stop“, die meisten kennen auch „restart“. Für Apache ist außerdem das Steuerwerkzeug apache2ctl verfügbar.

```
:~$ apache2ctl graceful
```

startet Apache neu *ohne* laufende Nutzersessions zu beenden.

Immer, wenn Sie etwas an der Konfiguration von Apache oder PHP geändert haben, muss dieses Kommando folgen, damit das System Ihre Änderungen übernimmt. Ein Rechner-Neustart ist **nicht** erforderlich.

- Unter Linux/UNIX arbeiten Sie mit den Dateien für Ihre Webseite *immer* als einfacher Nutzer und bei der Systemkonfiguration *immer* als Administrator root. Sobald Sie einen Rootserver gemietet haben, bekommen Sie von Ihrem Provider Zugangsdaten zugeschickt, in denen steht, wie Sie Ihr System erreichen können. Sie sollten erst ein neues Passwort für root vergeben und dann einen neuen einfachen Nutzer anlegen. Das geht oft über ein Browserinterface und in einem Terminal wie Putty oder dem Mac-Terminal via ssh:

```
:~ ssh root@rechnername.de
Password
Anmeldung erfolgt
# passwd
erst das alte, dann ein neues Passwort eingeben
# useradd -D normalnutzer
# passwd normalnutzer
Passwort eingeben
```

Damit haben Sie sich via ssh als root angemeldet, ein neues Passwort für root vergeben und einen neuen Normalnutzer namens „normalnutzer“ angelegt, der schließlich ein eigenes Passwort bekommen hat. Dieser Nutzer hat auch ein eigenes Verzeichnis /home/normalnutzer bekommen. Das ist ein guter Ort für die Document Root Ihrer Webseite.

- Document Root ist das Verzeichnis, aus dem Apache die Dateien holt, wenn ein Besucher Ihren Webserver mit http://rechnername.de aufruft. Es gibt Voreinstellungen für Document Root aber Sie sollten das Verzeichnis aktiv selbst kontrollieren. Das tun Sie in so genannten vhost-Dateien in einem Unterverzeichnis von /etc/apache2. Das Unterverzeichnis für die vhost-Dateien kann verschiedene Namen haben, das gilt auch für die Vorlagen, die die Distribution Ihres Apache mitliefert. Auch der Inhalt der Dateien kann etwas unterschiedlich sein, in den wichtigsten Zeilen ist er aber immer gleich:

```
<VirtualHost meineseite.de:80>
    ServerAdmin webmaster@meineseite.de
    ServerName meineseite.de
    # DocumentRoot: The directory out of which you will serve your
    # documents. By default, all requests are taken from this directory.
    DocumentRoot /home/normalnutzer/public_html/
    # Diverse weitere Anweisungen(Direktiven) für diesen virtuellen Host.
</VirtualHost>
```

Diese Datei sorgt dafür, dass die index.php-Datei aus /home/normalnutzer/public_html von Apache geliefert wird, wenn jemand den Rechner unter dem Namen meineseite.de aufruft. Für „www.meineseite.de“ müssen Sie eine weitere Datei anlegen. Haben Sie zusätzliche Domains registriert, die auf diesen Rechner zeigen, legen Sie für diese ebenfalls solche Dateien an, wobei Sie meineseite.de durch den jeweiligen Domainnamen

ersetzen und ein anderes Verzeichnis für Document Root eintragen, falls die anderen Domains andere Webseiten anzeigen sollen. So lassen sich auf einem Root-Server dutzende verschiedene Webseiten betreiben.

- Die Einrichtung von MySQL wird von MySQL selbst in einer speziellen Tabelle gespeichert. Dort sind vor allem MySQL-Nutzerkonten gespeichert, die nichts mit den UNIX-Nutzerkonten von Linux zu tun haben. Zur Bedienung von MySQL empfiehlt sich für Einsteiger PHPmyAdmin, das sich als eine Art Webseite in einem Verzeichnis unter DocumentRoot ablegen lässt. Unter Debian und UBUNTU lässt sich PHPmyAdmin auch mit dem Paketmanager installieren:

```
# apt-get install phpmyadmin
```

Unter Suse und Fedora sind ähnlich simple Installationsprogramme verfügbar.

- Das Paketmanagement installiert unter Linux automatisch Programme aus geprüften und gewarteten Paketsammlungen im Internet. Diese so genannten „Repositories“ enthalten alles, was Sie benötigen. Der Download von Software von Webseiten ist unter Linux nicht erforderlich und auf Webservern ist es auch nicht sinnvoll, Programme aus den Quellcodes selbst zu bauen. Über das Paketmanagement halten Sie Ihren Server auch aktuell. Solche Updates sind absolut lebensnotwendig um die Sicherheit des Servers zu garantieren. Unter Open Suse geschieht das so:

```
# yast2 -online-update
```

Das Programm öffnet eine grafische Anwendung im Terminal, in der Sie sich mit der Tabulatortaste und den Pfeiltasten bewegen und mit Leertaste und Enter Optionen einschalten und Befehle auslösen.

Unter Debian und UBUNTU halten Sie Ihr System mit apt-get auf dem Stand der Dinge:

```
#apt-get update  
Holt Infos über verfügbare Updates.  
#apt-get upgrade  
Installiert alle verfügbaren Updates
```

- PHP richten Sie mit der Datei /etc/php5/apache/php.ini ein. Die Datei bearbeiten Sie am besten mit dem UNIX-Standard-Editor vi
- Vi ist auf Webservern meist in der Variante vim installiert. Vim starten Sie am besten mit der Datei, die Sie bearbeiten möchten:

```
# vim /etc/php5/apache/php.ini
```

Vim kennt verschiedene Arbeitsmodi und sein Standardmodus ist erst einmal ein Befehlsmodus, in dem Sie Zeichen zwar löschen, kopieren und aus dem Zwischenspeicher einfügen aber nicht schreiben können. Sie sollten daher zuerst durch Druck auf die Taste „I“ (für insert) in seinen Eingabemodus wechseln. Unten links erscheint dann das Wort „Einfügen“ oder „Insert“. In diesem Modus lässt sich vim wie ein simpler Texteditor bedienen, was fürs erste schon für die meisten Konfigurationsaufgaben genügen sollte. Zum Speichern Ihrer Änderungen kehren Sie in den Befehlsmodus zurück mit Druck auf ESC (links oben auf der Tastatur). Geben Sie dann einen Doppelpunkt (:) ein. Links unten im Terminal sehen Sie jetzt den Doppelpunkt, hinter dem sich Befehle für vim eingeben lassen. Hier die wichtigsten:

```
:q!  
Beenden ohne zu Speichern  
:w  
Datei speichern  
:w dateiname  
Datei unter dem neuen Dateinamen „dateiname“ speichern  
:wq  
Speichern und Beenden.
```

Vorsicht!! Vim sieht simpel aus, hat aber eine unbekannte Masse von Funktionen unter der Haube, die er meist ohne Nachfrage ausführt. Ein paar wütende Tastaturanschläge können eine wichtige Konfigurationsdatei komplett zerstören. Zum Beispiel jede zweite Zeile löschen oder jedes Vorkommen von „x“ durch „u“ ersetzen. Haben Sie doch einmal eine ungewollte Aktion verursacht, aktivieren Sie mit ESC den Kommandomodus und drücken Sie „U“ (für „undo“), bis alles wieder OK ist.

- Die wichtigsten Verzeichnisse unter UNIX sind:
 - /etc - enthält alle Konfigurationsdateien Ihres Systems. Die Dateien sind alle in menschenlesbarem Textformat.
 - /home - in den Unterverzeichnissen mit den Nutzernamen liegen alle Arbeitsdateien auf dem Rechner. Webseiten, Bilder, Texte etc etc etc. In Dateien und Verzeichnissen, deren Name mit einem Punkt beginnt, sind hier Einstellungen gespeichert, die sich auf den Nutzer beziehen(persönliche Optionen).
 - /usr - Programme , Bibliotheken und Handbücher. Fast alle Anwendungen finden Sie unter usr/bin
 - /var/log - Systemlogbücher, das wichtigste heißt messages. In diesen Dateien protokolliert Linux, was es tut und was schief geht. Hier finden Sie zum Beispiel auch fehlgeschlagene Anmeldeversuche von möglichen Angreifern.

Die 8 wichtigsten Kommandos unter Linux/UNIX

Hinter jedem Kommando muss ein Leerzeichen stehen. Optionen stehen direkt nach dem Leerzeichen. Sie werden mit einem doppelten Minus-Strich eingeleitet, nach dem kein Leerzeichen kommt. Für die meisten Optionen sind auch Kürzel aus einem Buchstaben verfügbar, vor dem dann ein einfacher Strich steht. Diese Kürzel lassen sich meist nach dem Strich aneinanderhängen. So übergibt „-vR“ die beiden Optionen „verbose“ und „recursive“ gleichzeitig. Nach den Optionen lässt sich dort, wo das sinnvoll ist, eine Datei angeben, auf die das Kommando wirken soll. Zwei Beispiele:

```
:~$ ls -l --human-readable /home/semi/public_html
```

Zeigt den Inhalt des Verzeichnis public_html im home-Verzeichnis von semi ausführlich und mit menschenlesbar aufbereiteten Größenangaben (193M statt 202276151Byte).

Hat man die ersten Zeichen eines Kommandos oder eines Namens eines Verzeichnis/Datei eingetippt, holt ein Druck auf die Tabulatortaste automatisch den Rest des Namens. Die letzten 1000 eingegebenen Kommandozeilen lassen sich durch Druck auf die Pfeiltaste nach oben zurückholen. Ein Druck auf STRG+C beendet festgefahrene Kommandos.

1.) sich im Dateisystem bewegen mit cd

```
:~$ cd /etc/apache2
```

Bringt Sie in das Verzeichnis, in dem die Konfigurationsdateien für Apache liegen.

2.) Dateien und Verzeichnisse anzeigen mit ls

```
:~$ ls -lh /home/semi/public_html
```

Identisch mit dem oben gezeigten Kommando aber mit Kurzform von human-readable.

3.) Dateien und Verzeichnisse kopieren mit cp

```
:~$ cp -vR markitup/ web/
```

Kopiert das Verzeichnis „markitup“ ins Verzeichnis „web“, ist dabei geschwätzig (verbose) und geht das ganze markitup-Verzeichnis durch (R steht für rekursiv).

4.) Dateien und Verzeichnisse verschieben mit mv

```
:~$ mv -vv quicktags* web/
```

Schiebt alle Dateien im aktuellen Verzeichnis, die mit „quicktags“ beginnen, in das Verzeichnis web und löscht sie dann im aktuellen Verzeichnis.

5.) Verzeichnisse anlegen mit mkdir

```
:~$ mkdir -vp web/unterweb/webber
```

Legt im Verzeichnis web die Unterverzeichnisse unterweb und webber an. Das geht nur Dank der Option p - „parents“.

6.) Dateirechte anpassen mit chmod

```
:~$ chmod -v uga+w web/unterweb/*
```

Gibt dem Nutzer(u), der Gruppe(g) und allen Anderen(a) das Schreibrecht w für alle Dateien und Verzeichnisse in unterweb, nicht aber an unterweb selbst. Chmod arbeitet mit so genannten Modi, die ihm mitteilen, wie genau die Rechte nach der Aktion aussehen sollen. Diese Modi stehen hinter den Optionen mit Leerzeichen aber ohne Strich davor, danach kommt ein Leerzeichen und die Datei/das Verzeichnis, auf die chmod wirken soll.

7.) Informationen über Kommandos ausgeben lassen mit man

```
:~$ man ls
```

Gibt das erstaunlich lange Handbuch (manual) für ls aus.

8.) Archive einpacken/auspacken mit tar.

```
:~$ tar -xzf archiv.tar.gz
```

Packt das Archiv archiv.tar.gz im aktuellen Ordner aus. Dabei entsteht unter Linux fast immer ein Unterverzeichnis „archiv“. Die Reihenfolge der Optionen: x - eXtract(auspacken) z (gZip-Kompression auflösen) f (mit dem direkt folgenden File arbeiten) muss genau eingehalten werden.