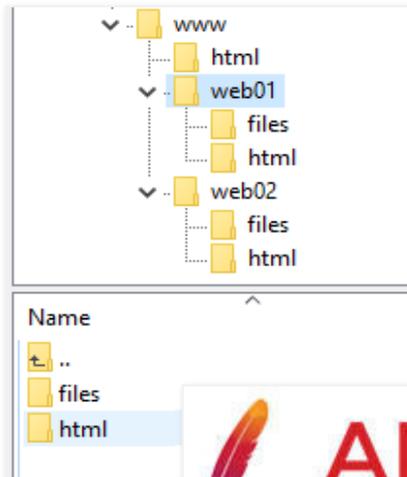
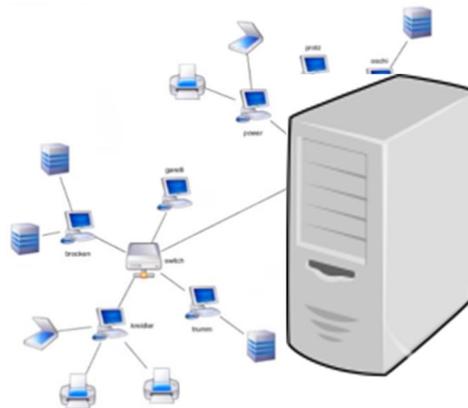
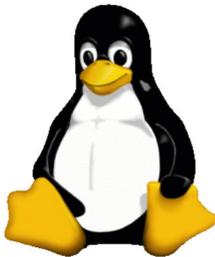


# Linux - Serverdienste

SSH-, Web-, DB und Fileserver

Netzwerk- und Serverinstallation zur Demonstration  
gängiger Serverdienste mit Windows- und Linux-Betriebssystemen



## Anmerkungen zur Dokumentation

In der Dokumentation werden die Begriffe Unix und Linux für das Betriebssystem verwendet. Die meisten Erläuterungen gelten sowohl für Unix als auch für Linux gleichermaßen.

### *Übungsaufgaben*

Die beschriebenen Kommandos und Installationsschritte wurden mit virtuellen Maschinen (VM's) in einem separaten LAN-Segment getestet. Hierbei wurde eine Netzwerkbrücke eingerichtet, die alle VM's miteinander verbindet. Der Internetzugang wurde über einen separaten Router gewährleistet.

### *Dokumenteninfos*

Dateiname: Linux-Ubuntu-Server-Projekt\_V1906.docx

Erstellt von H. Koch: 07/2018, überarbeitet 05/2019

aktueller Stand: 20.07.2019

Druckdatum: 14.12.2019

## Copyrighthinweise

Das Script ist überprüft und getestet. Es kann aber weder eine Garantie für Fehlerfreiheit, noch für inhaltliche Richtigkeit gegeben werden. Für technische Fehler, fehlerhafte Angaben sowie deren Folgen wird keine Haftung übernommen. Etwaige Ersatzansprüche, gleich welcher Art, sind damit ausgeschlossen.

## Literaturverzeichnis

- [1] L. Ewing, S. Budig, A. Gerwinsky und S. Hagel, „Linux Kernel Struktur,“ 31 12 2008. [Online]. Available: [http://de.wikipedia.org/w/index.php?title=Datei:Linux\\_Kernel\\_Stuktur.svg&filetimestamp=20091107100810](http://de.wikipedia.org/w/index.php?title=Datei:Linux_Kernel_Stuktur.svg&filetimestamp=20091107100810). [Zugriff am 12 04 2012].
- [2] Creative Commons PoBox 1866, Mountain View, CA 94042, „CC creative commons,“ [Online]. Available: <https://creativecommons.org/licenses/by/2.5/>. [Zugriff am 01 Mai 2019].
- [3] GNU\_Free\_Documentation\_License, „WIKIMEDIA Commons,“ [Online]. Available: [https://en.wikipedia.org/wiki/en:GNU\\_Free\\_Documentation\\_License](https://en.wikipedia.org/wiki/en:GNU_Free_Documentation_License).

## Quellangaben der im Skript verwendeten Grafiken

	<p>Das Logo des Linux-Tux wurde freigegeben von Larry Ewing unter folgenden Bedingungen: „Es ist erlaubt, diese Grafik zu verwenden und/oder zu verändern. Bedingung ist jedoch: Falls jemand fragt, muss man mich – lewing@isc.tamu.edu – als Urheber nennen und auf GIMP hinweisen.“ [1]</p>
<p>Linux-UBUNTU</p> 	<p>Grafik gemeinfrei Datei:  <a href="https://de.wikipedia.org/wiki/Ubuntu#/media/File:Ubuntu_logo.svg">https://de.wikipedia.org/wiki/Ubuntu#/media/File:Ubuntu_logo.svg</a></p>
<p>Webserver Apache</p> 	<p>Datei: <a href="https://de.wikipedia.org/wiki/Apache_Software_Foundation#/media/File:Apache_Software_Foundation_Logo_(2016).svg">https://de.wikipedia.org/wiki/Apache_Software_Foundation#/media/File:Apache_Software_Foundation_Logo_(2016).svg</a></p> <p>Licensed under the Apache License, Version 2.0 (the "License");you may not use this file except in compliance with the License.You may obtain a copy of the License at <a href="http://www.apache.org/licenses/LICENSE-2.0">http://www.apache.org/licenses/LICENSE-2.0</a> Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.</p>
<p>PHP Skriptsprache</p>  <p>Datenbankserver</p>  <p>DB-Admin</p> 	<p>Datei php: <a href="https://de.wikipedia.org/wiki/PHP#/media/File:PHP-logo.svg">https://de.wikipedia.org/wiki/PHP#/media/File:PHP-logo.svg</a>  Datei DB: <a href="https://de.wikipedia.org/wiki/MariaDB#/media/File:MariaDB_Logo.png">https://de.wikipedia.org/wiki/MariaDB#/media/File:MariaDB_Logo.png</a>  Datei : <a href="https://de.wikipedia.org/wiki/PhpMyAdmin#/media/File:PhpMyAdmin-Logo.svg">https://de.wikipedia.org/wiki/PhpMyAdmin#/media/File:PhpMyAdmin-Logo.svg</a></p> <p>Attribution-ShareAlike 4.0 International (CC BY-SA 4.0) [2]</p> <p>You are free to: Share — copy and redistribute the material in any medium or format  Adapt — remix, transform, and build upon the material  for any purpose, even commercially. This license is acceptable for Free Cultural Works. The licensor cannot revoke these freedoms as long as you follow the license terms. Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.  ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.  No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.  <a href="https://creativecommons.org/licenses/by-sa/4.0/legalcode">https://creativecommons.org/licenses/by-sa/4.0/legalcode</a></p>
<p>WinSCP</p> 	<p>Datei: <a href="https://de.wikipedia.org/wiki/WinSCP#/media/File:WinSCP_Logo.png">https://de.wikipedia.org/wiki/WinSCP#/media/File:WinSCP_Logo.png</a>  Lizenz: <a href="https://www.gnu.org/licenses/gpl-3.0.html">https://www.gnu.org/licenses/gpl-3.0.html</a></p>
<p>PuTTY</p> 	<p>Datei: <a href="https://de.wikipedia.org/wiki/PuTTY#/media/File:PuTTY_icon_128px.png">https://de.wikipedia.org/wiki/PuTTY#/media/File:PuTTY_icon_128px.png</a>  Lizenz: <a href="https://opensource.org/licenses/mit-license.php">https://opensource.org/licenses/mit-license.php</a></p>
<p>RaspBerryPi</p> 	<p>Lizenzen: <a href="https://www.raspberrypi.org/trademark-rules/">https://www.raspberrypi.org/trademark-rules/</a>  <a href="https://www.raspberrypi.org/">https://www.raspberrypi.org/</a></p>
<p>WebAdmin</p> 	<p>Datei: <a href="http://www.webmin.com/graphics/webmin-full.jpg">http://www.webmin.com/graphics/webmin-full.jpg</a>  Lizenz: <a href="http://www.webmin.com/graphics.html">http://www.webmin.com/graphics.html</a></p>

## Inhalt

1	Übersicht .....	6
1.1	Beispielprojekt .....	6
1.2	Installationsübersicht .....	7
2	Installation der VM's .....	8
2.1	VM neu erstellen .....	8
2.2	Anpassung einer VM .....	10
2.3	Server-Grundinstallation .....	11
3	Netzwerk konfigurieren .....	12
3.1	Konfiguration anzeigen .....	12
3.2	Netzwerkmanager .....	12
3.3	Remote-Netzwerkzugriff ermöglichen .....	15
3.4	Root-Zugriff auf Ubuntu .....	19
4	Webserver .....	20
4.1	Webserverfunktionen .....	20
4.2	Übertragungsmethoden .....	21
4.3	Installation und Funktionstest .....	21
4.4	Apache starten/stoppen .....	22
4.5	Apache für VHOSTS vorbereiten .....	22
4.6	Planung der Web-User (Website-Kunden) .....	23
4.7	Web-User erstellen .....	23
4.8	VHOSTS für den Apache-Server konfigurieren .....	25
4.9	Websites testen .....	26
4.10	Funktionsweise der Apache-Konfiguration .....	27
5	Scriptsprache PHP .....	29
5.1	PHP Versionsgeschichte .....	29
5.2	PHP7 auf dem Server installieren .....	29
5.3	PHP-Erweiterungen installieren .....	30
5.4	PHP-Funktionstest .....	31
6	Datenbankserver installieren .....	32
6.1	Vergleich MySQL vs. MariaDB .....	32
6.2	MariaDB-Installation .....	33
6.3	MariaDB-Funktionstest .....	34
6.4	Module nachinstallieren .....	35
7	PHP MyAdmin installieren .....	36
7.1	Installation auf dem Ubuntu-Server .....	36
7.2	Administrator für den Remotezugriff erstellen .....	37
7.3	Funktionsüberblick phpMyAdmin .....	38

## Abbildungsverzeichnis

Abbildung 1: Netzwerkplanung .....	6
Abbildung 2: Planung und Dokumentation eines Hosts .....	6
Abbildung 3: VirtualBox Manager .....	8
Abbildung 4: Name der VM .....	8
Abbildung 5: Arbeitsspeicher RAM .....	8
Abbildung 6: Festplatte erzeugen .....	9
Abbildung 7: Festplatte Speicherplatz .....	9
Abbildung 8: Startmedium "einlegen" .....	10
Abbildung 9: Weitere Festplatte erzeugen .....	10
Abbildung 10: Anzeige der Festplatten .....	10
Abbildung 11: Netzwerkeinstellungen .....	10
Abbildung 12: Anmeldung an der Konsole .....	11
Abbildung 13: Netzwerkeinstellungen anzeigen .....	12
Abbildung 14: yaml-Datei zur Netzwerkeinstellung .....	13
Abbildung 15: Beispiel mit statischen IP-Adressen .....	14
Abbildung 16: Verbindungstests .....	14
Abbildung 17: traceroute unter Ubuntu .....	15
Abbildung 18: Servicestatus ssh .....	16
Abbildung 19: ssh-Verbindung mit puTTY .....	17
Abbildung 20: PuTTY mit Ubuntu-Desktop .....	17
Abbildung 21: sftp-Dateiaustausch .....	18
Abbildung 22: WinSCP-Datenaustausch .....	18
Abbildung 23: Anmeldung mit dem Systemverwalter root .....	19
Abbildung 24: root-Fernzugriff mit Windows-PuTTY .....	19
Abbildung 25: Ordnerstruktur für Webserver unter XAMPP und T-Online .....	20

Abbildung 26: Apache-Funktionstest .....	21
Abbildung 27: zusätzliche IP-Adressen planen .....	22
Abbildung 28: ping-Test vom Windows Client aus.....	23
Abbildung 29: VHOSTs konfigurieren.....	25
Abbildung 30: Startseite .....	26
Abbildung 31: Test der Webseite .....	26
Abbildung 32: php-Versionstest .....	30
Abbildung 33: php-Erweiterungen .....	30
Abbildung 34: Abruf einer PHP-Datei .....	31
Abbildung 35: PHP-Quellcode .....	31
Abbildung 36: Ausgabe der php-Seite.....	31
Abbildung 37: MariaDB mit mySQL auf der Testseite.....	35
Abbildung 38: root Zugriff gesperrt .....	37
Abbildung 39: Zugriff möglich.....	37
Abbildung 40: mySQL-Grundeinstellung.....	37

# 1 Übersicht

Das vorliegende Skript möchte in einfachen Schritten die Installation verschiedener Serverdienste auf Linux-Betriebssystemen und den Zugriff darauf von verschiedenen Clients beispielhaft darstellen. Es erhebt keinen Anspruch auf Vollständigkeit und zeigt einige wenige Möglichkeiten der Einarbeitung in die Thematik anhand eines Beispielprojekts auf. Ein kleines lokales Netzwerk soll virtuell entstehen. Hierzu werden virtuelle Client- und Server-Rechner mit unterschiedlichen Betriebssystemen ausgestattet, um möglichst viele Möglichkeiten der Konfiguration und des Zugriffs darzustellen.

## 1.1 Beispielprojekt

Das Netzwerk besteht im ersten Ausbau aus einem Server-PC (virtuell) und zwei Client-PC's, die für jeweils zwei Benutzer eingerichtet werden. Die Verbindung der (virtuellen) Rechner erfolgt über einen (virtuellen) Switch, eine Anbindung ans Internet wird über eine DSL-Routerverbindung realisiert. Der Netzwerkplan zeigt den Grundaufbau.

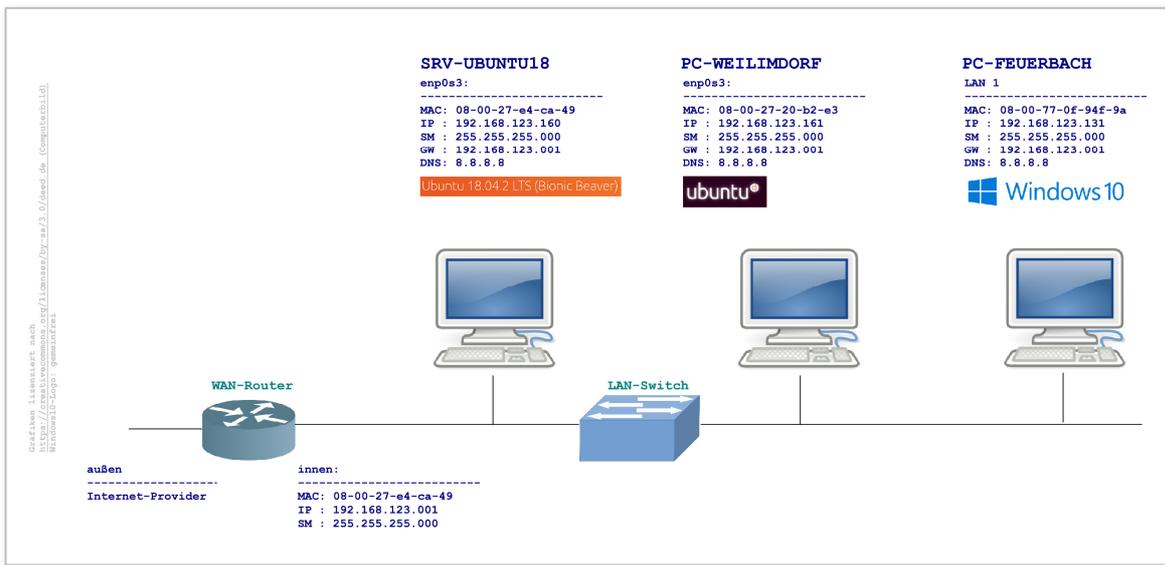


Abbildung 1: Netzwerkplanung

Zu jedem Host erfolgt eine genaue Dokumentation der zu verwendenden Rechnernamen, Netzwerkadressen und der installierten Software mit Benutzernamen und Passwörtern...

Die Hardwareadressen können aus Lieferpapieren oder im virtualisierten Fall aus den Einstellungen des VBox-Managers entnommen werden.

```

SRV-UBUNTU18
enp0s3:
-----
MAC: 08-00-27-e4-ca-49
IP : 192.168.123.160
SM : 255.255.255.000
GW : 192.168.123.001
DNS: 8.8.8.8
Ubuntu 18.04.2 LTS (Bionic Beaver)
Betriebssystem: Ub.18.04.2 LTS
SSH-Server
Benutzer: pinguin [P@ssW0rt]
    
```

Abbildung 2: Planung und Dokumentation eines Hosts

## 1.2 Installationsübersicht

---

Die folgende Auflistung zeigt die voraussichtlich notwendigen Arbeitsschritte, um das Beispielprojekt zu realisieren.

- Server – Grundinstallation
- Benutzer festlegen
- Netzwerk konfigurieren, Testverbindungen
- Server - Zusatzpakete zur Vorbereitung der weiteren Dienste
  - Perl
  - Fernverwaltungstools einrichten
    - SSH für den Fernzugriff mit den Tools Putty und SCP
    - zusätzlich Webmin
- Server – Webdienste
  - Apache Webserver Grundinstallation
  - Apache VHOSTS zum IP-basierten Webhosting
- PHP als Skriptsprache
  - Installation, Funktionstest
- Datenbankserver
  - Installation, Funktionstest
- ftp-Server ProFtpD
  - Installation, Funktionstest

Nach Installation dieser Serverdienste ist es möglich, weitere webfähige Geräte in das Netzwerk zu integrieren.

## 2 Installation der VM's

Die Installation erfolgt im dargestellten Beispiel in virtuellen Maschinen (VM's). Die Einrichtung auf „Hardware-Maschinen“ verläuft in ähnlicher Weise.

### 2.1 VM neu erstellen

Im Oracle VM VirtualBox Manager werden eventuell schon vorhandene VMs angezeigt. Über die Werkzeuge können neben Änderung, Import und Export auch neue VMs erstellt werden.

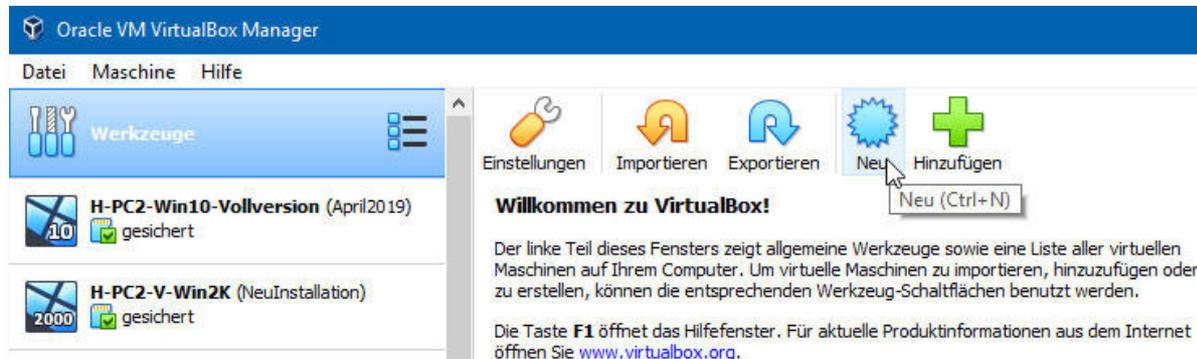


Abbildung 3: VirtualBox Manager

Der Name der VM ist für die Zuordnung im VBox Manager wichtig.

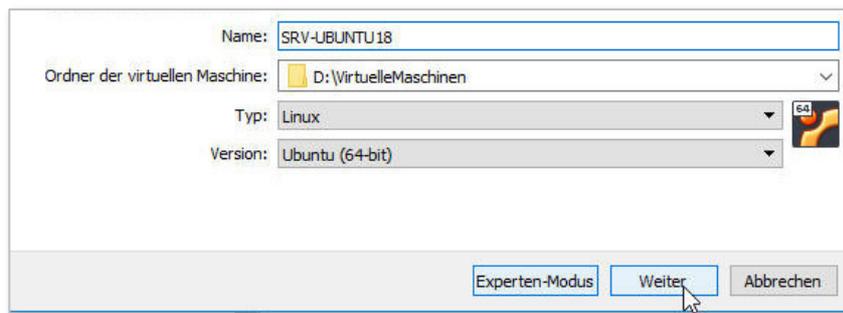


Abbildung 4: Name der VM

Der Arbeitsspeicher darf nicht zu groß gewählt werden, damit der Haupt-PC noch genügend Ressourcen behält. Wird der Speicher im grünen Bereich gehalten, ist er auch für den Wirts-PC noch ausreichend. ACHTUNG: Sollen mehrere VMs gleichzeitig gestartet werden, muss dies bei der Wahl des RAM für die VM beachtet werden!

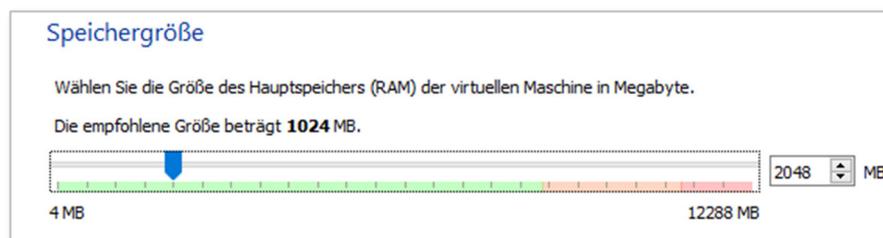


Abbildung 5: Arbeitsspeicher RAM

Die virtuelle Festplatte wird als einzelne Datei auf der Festplatte des Wirts-PC abgespeichert. Der Standard-Speicherpfad kann im VBox-Manager unter dem Hauptmenü Datei... angepasst werden.

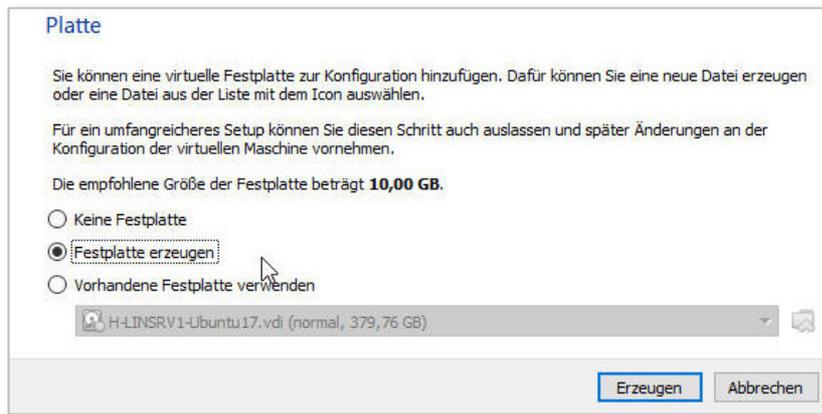


Abbildung 6: Festplatte erzeugen

Als Dateityp können neben VDI auch Dateitypen von Fremdsystemen (VHD, VMDK) verwendet werden. VDI ist jedoch die beste Wahl. Die Größe der Festplatte ist stark vom späteren Anwendungsfall abhängig.

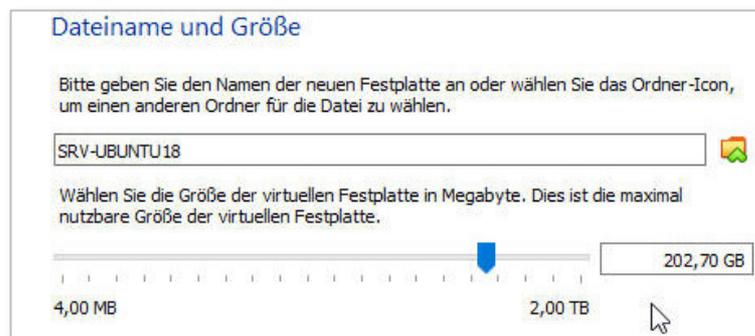
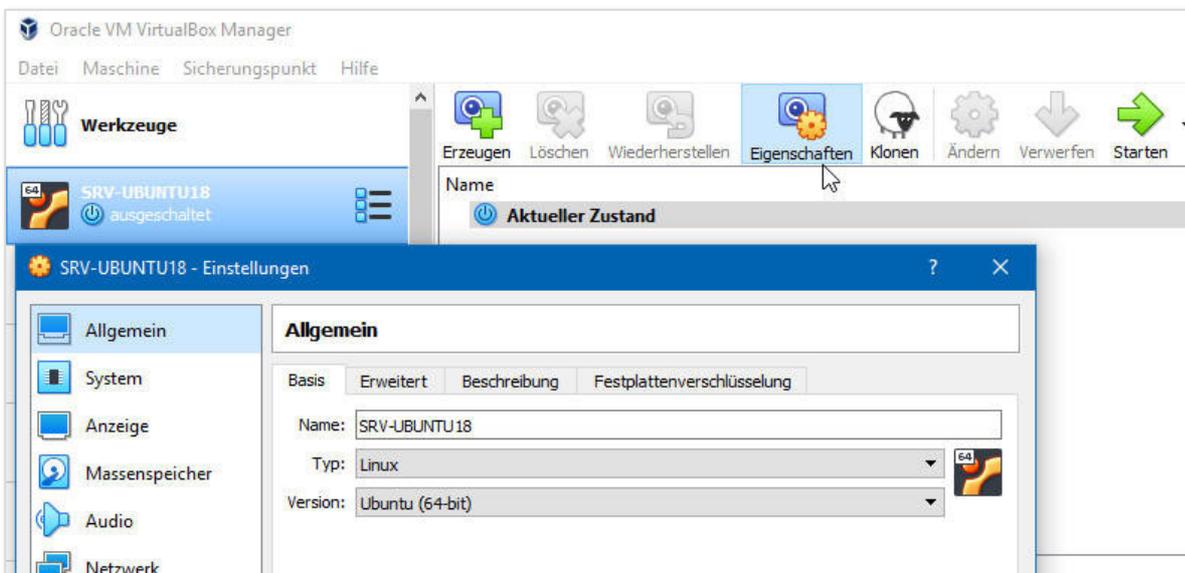


Abbildung 7: Festplatte Speicherplatz

Die fertige VM wird im VBox-Manager angezeigt und kann über Eigenschaften - Ändern angepasst werden.



## 2.2 Anpassung einer VM

Die folgenden Beispiele zeigen einige mögliche Anpassungen der erzeugten VM.

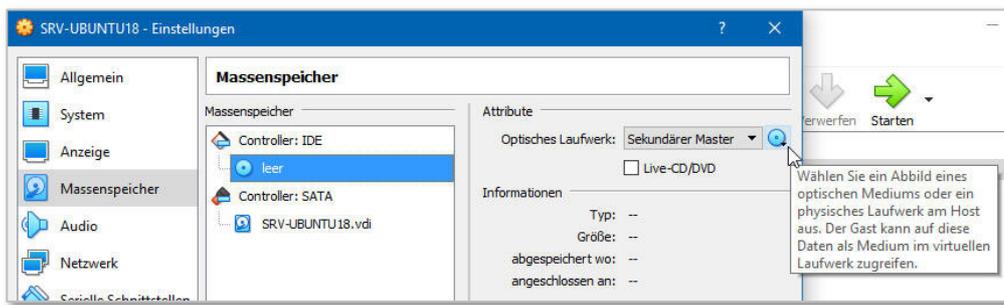


Abbildung 8: Startmedium "einlegen"

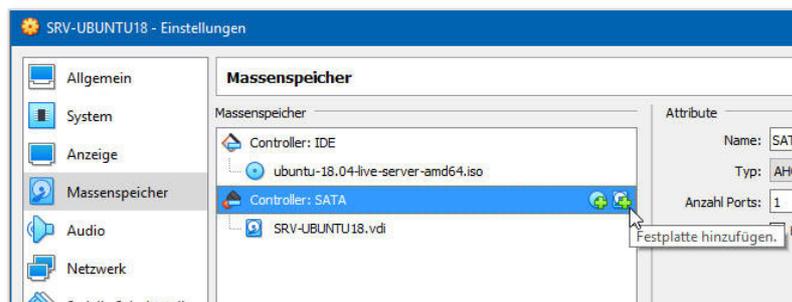


Abbildung 9: Weitere Festplatte erzeugen

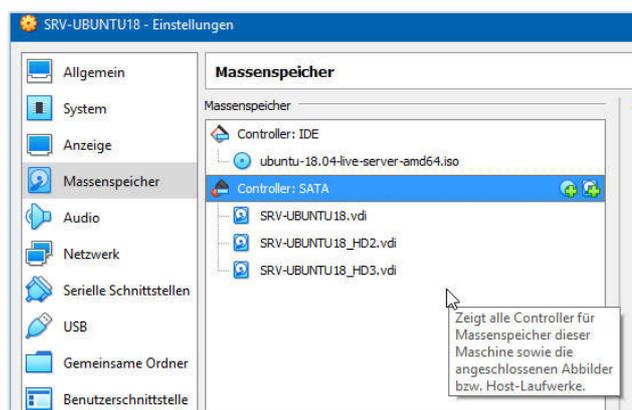


Abbildung 10: Anzeige der Festplatten

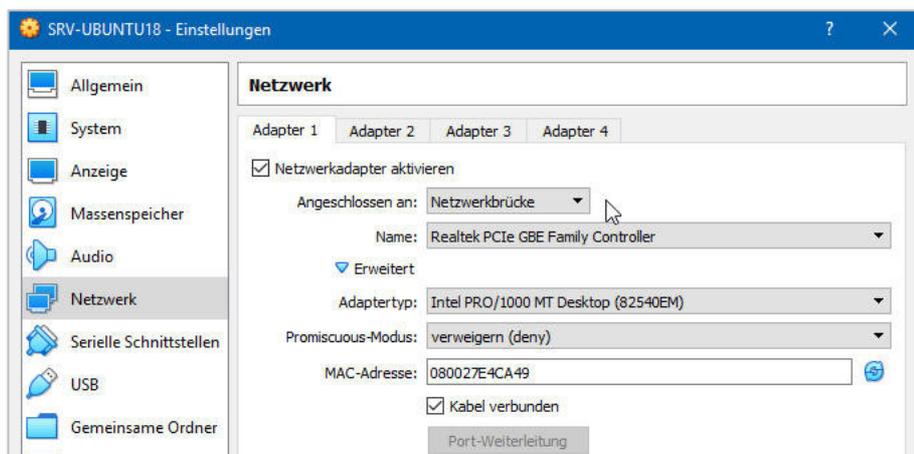


Abbildung 11: Netzwerkeinstellungen

## 2.3 Server-Grundinstallation

Nach dem Bootvorgang startet das Grundsetup. Dieses wird in mehreren Schritten vorgenommen. Die Bedienung der textbasierten Masken wird zur Auswahl mit Tabulator, Cursor-Pfeiltasten und Leertaste (Auswahl-Checkboxen) und der Entertaste (Bestätigung) gesteuert.

- 1) Sprache
- 2) Tastatur
- 3) Server-Installationsart
- 4) Netzwerkverbindungen.  
Die Angaben sind von den örtlichen Gegebenheiten abhängig, z.B. kann die Adresse zunächst mittels DHCP automatisch zugewiesen werden. Später werden die Netzwerkeinstellungen individuell konfiguriert.
- 5) Ein Proxyserver muss angegeben werden, wenn der Ubuntu-Server den Internet-Zugang über diesen "Umweg" Proxy erreichen kann.
- 6) Partitionierung (Gesamtfestplatte oder nur Teilbereich).  
Sind im System mehrere Festplatten eingebaut, muss die gewünschte Platte für das Dateisystem ausgewählt werden.  
Zusammengefasst werden die Partitionierungsdaten angezeigt.  
Abschließend erfolgt eine Warnung, bevor die Partitionierung abgeschlossen wird.  
Während des Partitionierungsprozesses kann der Benutzer angelegt werden.
- 8) Die Installation wird durchgeführt...
- 9) Fertigstellung und Neustart.  
Installationsmedium (VBox-Medium) entfernen.  
USB-Stick oder DVD entfernen bzw. Medium in der VM auswerfen.

Nun erfolgt der Reboot und Ubuntu meldet sich mit dem Anmeldefenster. Zunächst wird ein Schlüssel (Fingerprint) erzeugt, der für den Netzwerk-Fernzugriff erforderlich ist.

Anschließend ist die Anmeldung an der Konsole möglich.

```
Ubuntu 18.04 LTS srv-ubuntu18 tty1
srv-ubuntu18 login: pinguin
Password:
Last login: Wed May  1 09:28:47 UTC 2019 on tty1
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-47-generic x86_64)
```

Abbildung 12: Anmeldung an der Konsole

## 3 Netzwerk konfigurieren

Bevor die Konfiguration der Serverdienste fortgesetzt wird, sollten alle Netzwerkeinstellungen anhand des Netzwerkplans getestet werden. Die Netzwerkeinstellungen werden bei verschiedenen Linux-Distributionen unterschiedlich konfiguriert.

### 3.1 Konfiguration anzeigen

Nach der Neuinstallation des Systems wird die Netzwerkkarte in der Regel mit DHCP-Client

Hierzu wird nach dem Anmelden das Kommando

```
ip address
```

abgesetzt. Damit werden alle Netzwerkadressen des Systems mit ihren Einstellungen angezeigt (siehe Abbildung 13).

```
penguin@srv-ubuntu18:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group def
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
   link/ether 08:00:27:e4:ca:49 brd ff:ff:ff:ff:ff:ff
   inet 192.168.123.160/24 brd 192.168.123.255 scope global dynamic enp0s3
       valid_lft 859822sec preferred_lft 859822sec
   inet6 fe80::a00:27ff:fee4:ca49/64 scope link
       valid_lft forever preferred_lft forever
penguin@srv-ubuntu18:~$ █
```

Abbildung 13: Netzwerkeinstellungen anzeigen

- `lo` ist die Loopbackadresse  
IP4 **127.0.0.1**, IP6 **::1/128**
- **enp0s3** die (neue) Bezeichnung der ersten Netzwerkkarte  
(alte Bezeichnung: **eth0**)

Die Anpassung der Netzwerkkonfiguration erfolgt bei neueren Distributionen über den Netzwerkmanager.

### 3.2 Netzwerkmanager

In Ubuntu18 wird die neue Methode zur Einstellung der Netzwerkkonfiguration angewendet. Sie löst die alte Methode mittels **/etc/network/interfaces** ab. Diese Methode existiert zwar noch, es wird jedoch empfohlen die neuere und einfachere Methode mit dem Netzwerkmanager **netplan** zu verwenden. Die neue Konfiguration arbeitet mit so genannten **yaml**-Dateien.

YAML ist eine Auszeichnungssprache zur Datenserialisierung, angelehnt an XML (ursprünglich) und an die Datenstrukturen in den Sprachen Perl, Python und C. Die Abkürzung ist ein rekursives Akronym für „YAML Ain't Markup Language“ (ursprünglich „Yet Another Markup Language“).

## Statische IP-Adressen mit dem Network-Manager einstellen

Der Netzwerk-Manager benutzt die yaml-Dateien im Verzeichnis `/etc/netplan/` und arbeitet die alle darin gespeicherten Dateien ab (siehe Abbildung 14).

Die Dateien können vom System auch in den Verzeichnissen `/run/netplan` oder `/lib/netplan` gespeichert werden, jedoch der Pfad unter `/etc/netplan/` ist der richtige Ort, um eigene Dateien abzulegen. Netplan ist in der Grundinstallation von Ubuntu im Paket `netplan.io` enthalten oder kann bei Bedarf nachinstalliert werden.

```
sudo apt-get install netplan.io
```

Eigene Konfigurationen können beliebig benannt werden, allerdings muss die Dateierweiterung `.yaml` sein! Werden mehrere Dateien abgespeichert, so liest sie das System in lexikalisch sortierter Reihenfolge. Eine später vorgenommene Einstellung würde eine vorherige, eventuell widersprüchliche überschreiben! Die vom Ubuntu-Setup-Programm abgefragte IP-Konfiguration taucht in der Datei `50-cloud-init.yaml` auf. Den Inhalt der Datei zeigt die folgende Abbildung.

```
penguin@srv-ubuntu18:/etc/netplan$ ls -l
total 4
-rw-r--r-- 1 root root 436 Apr 14 08:25 50-cloud-init.yaml
penguin@srv-ubuntu18:/etc/netplan$ cat 50-cloud-init.yaml
# This file is generated from information provided by
# the datasource. Changes to it will not persist across an instance.
# To disable cloud-init's network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  ethernets:
    enp0s3:
      addresses: []
      dhcp4: true
      optional: true
  version: 2
penguin@srv-ubuntu18:/etc/netplan$ █
```

Abbildung 14: yaml-Datei zur Netzwerkeinstellung

Diese Datei sollte sicherheitshalber kopiert werden, damit für "alle Fälle" eine funktionsfähige Konfiguration schnell wieder eingespielt werden kann.

```
sudo cp 50-cloud-init.yaml 51-srv-ubuntu.yaml
```

Nun wird das Original umbenannt und dadurch von der Benutzung ausgeschlossen. Die eigentliche Netzwerkeinstellung soll ausschließlich mit der neu angelegten Datei erfolgen.

```
sudo mv 50-cloud-init.yaml 50-cloud-init.yaml.orig
```

Jetzt kann die eigene Konfigurationsdatei bearbeitet werden.

```
sudo nano 51-srv-ubuntu.yaml
```

Wie im Beispiel Abbildung 15 dargestellt, werden alle Netzwerkeinstellungen aus Übersichtlichkeitsgründen in einer einzigen Datei verwaltet. Es wäre auch möglich, mehrere `yaml` -Dateien zu erstellen. Diese werden vom Netzwerkmanager hintereinander gelesen und als Konfiguration verwendet.

```

GNU nano 2.9.3                    51-srv-ubuntu.yaml
# This file is generated from information provided by
# the datasource.  Changes to it will not persist across an instance.
# To disable cloud-init's network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  ethernets:
    enp0s3:
      addresses: [192.168.123.160/24,
                  192.168.123.161/24, 192.168.123.162/24,
                  192.168.123.163/24, 192.168.123.164/24]
      dhcp4: false
      gateway4: 192.168.123.1
      nameservers:
        addresses: [8.8.8.8]
      optional: true
  version: 2

^G Get Help      ^O Write Out    ^W Where Is    ^K Cut Text    ^J Justify
^X Exit          ^R Read File    ^\ Replace     ^U Uncut Text  ^T To Spell

```

Abbildung 15: Beispiel mit statischen IP-Adressen

Das Speichern erfolgt mit (Strg)+(o), dann kann **nano** verlassen werden mit (Strg)+(x). Nachdem die Datei gespeichert wurde, sollten zunächst die neuen Einstellungen getestet werden.

```
sudo netplan try
```

Erst wenn alles ok ist, wird die neue Konfiguration ins System übernommen.

```
sudo netplan apply
```

Die Änderungen werden über das Kommando **ip address** noch überprüft. Außerdem ergeben Verbindungstests Sicherheit über die erfolgreiche Netzwerkanbindung im LAN und ins Internet

```

pinguin@srv-ubuntu18:~$ ping 192.168.123.1
PING 192.168.123.1 (192.168.123.1) 56(84) bytes of data.
64 bytes from 192.168.123.1: icmp_seq=1 ttl=64 time=0.726 ms
64 bytes from 192.168.123.1: icmp_seq=2 ttl=64 time=0.947 ms
^C
--- 192.168.123.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.726/0.836/0.947/0.114 ms
pinguin@srv-ubuntu18:~$ ping www.google.de
PING www.google.de (172.217.21.3) 56(84) bytes of data.
64 bytes from fra07s29-in-f3.1e100.net (172.217.21.3): icmp_seq=1 ttl=57 time=9.19 ms
64 bytes from fra07s29-in-f3.1e100.net (172.217.21.3): icmp_seq=2 ttl=57 time=8.54 ms
^C
--- www.google.de ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 8.543/8.866/9.190/0.336 ms
pinguin@srv-ubuntu18:~$

```

Abbildung 16: Verbindungstests

Damit das **traceroute**-Kommando auf Ubuntu verwendet werden kann, muss auf dem Host ein Paket aus den Netzsutilities nachinstalliert werden:

```
sudo apt install inetutils-traceroute
```

Die erfolgreiche Routenverfolgung zeigt ebenso die funktionierende Konfiguration.

```
penguin@srv-ubuntu18:~$ traceroute www.t-online.de
traceroute to www.t-online.de (62.138.238.100), 64 hops max
 1  192.168.123.1  0,807ms  1,510ms  0,426ms
 2  62.155.245.70  18,062ms  4,716ms  4,908ms
 3  62.154.55.222  11,796ms  10,554ms  10,736ms
 4  62.154.55.222  10,558ms  9,755ms  9,778ms
 5  87.186.180.91  11,463ms  10,167ms  10,791ms
 6  62.138.238.100  11,458ms  11,058ms  10,700ms
penguin@srv-ubuntu18:~$
```

Abbildung 17: traceroute unter Ubuntu

### 3.3 Remote-Netzwerkzugriff ermöglichen

Unter Remote-Zugriff versteht man im Allgemeinen das Bedienen eines Computersystems von einem entfernten Platz aus. Dabei kann "nur" die Kommando-Konsole oder auch der gesamte Desktop des Zielrechners auf dem Bedienplatz dargestellt sein.

#### Telnet

Telnet (Teletype Network) ist ein sehr bekanntes (altes) Netzwerkprotokoll. Über eine TCP-Verbindung erfolgt ein zeichenorientierter Datenaustausch. Die zugehörigen Dienste heißen Telnet-Server und Telnet-Client. Für den Zugriff muss die Firewall so eingestellt werden, dass der Standardport 23 nicht blockiert wird. Telnet wird in der Regel zur Fernsteuerung von Computern in Form von textbasierten Ein- und Ausgaben eingesetzt.

**ACHTUNG:** Der Telnet-Client baut eine unverschlüsselte Verbindung zu einem Telnet-Server auf. Dabei wird ein notwendiges Kennwort im Klartext übergeben. Die erscheinende Login-Konsole kann volle Adminrechte besitzen, so dass eine Telnet-Verbindung als unsicher gilt und durch eine ssh-Verbindung ersetzt werden sollte!

#### SSH-Dienst

Als Secure-Shell oder SSH werden sowohl ein Netzwerkprotokoll als auch Softwareprogramme bezeichnet, die einen gesicherten Zugriff von einem Client-Computer auf ein Zielsystem gewährleisten. Häufig wird eine entfernte Kommandozeile dargestellt, das bedeutet, dass auf der lokalen Konsole die Ausgaben der entfernten Konsole ausgegeben und lokale Eingaben werden an den Zielcomputer gesendet. Dadurch kann z.B. eine Wartung eines entfernt aufgestellten Servers vorgenommen werden. Die aktuelle Version ist SSH 2. Sie bietet z.B. auch Datenübertragung mit dem Protokoll SFTP. SSH benutzt den Standard-TCP-Port 22.

Das Protokoll wird unter Ubuntu installiert mit

```
sudo apt install ssh
```

Der Status wird abgefragt, um zu testen, ob der Dienst (bereits) läuft.

```
service ssh status
```

Die Ausgabe zeigt in Abbildung 18 zeigt an, dass der ssh-Dienst gestartet wurde und auf Port 22 „hört“.

```

pinguin@srv-ubuntu18:~$ service ssh status
• ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2019-05-01 07:57:59 UTC; 1h 46min ago
  Process: 965 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Main PID: 1110 (sshd)
  Tasks: 1 (limit: 3556)
  CGroup: /system.slice/ssh.service
          └─1110 /usr/sbin/sshd -D

Mai 01 07:57:55 srv-ubuntu18 systemd[1]: Starting OpenBSD Secure Shell server...
Mai 01 07:57:59 srv-ubuntu18 sshd[1110]: Server listening on 0.0.0.0 port 22.
Mai 01 07:57:59 srv-ubuntu18 sshd[1110]: Server listening on :: port 22.
Mai 01 07:57:59 srv-ubuntu18 systemd[1]: Started OpenBSD Secure Shell server.
Mai 01 08:09:53 srv-ubuntu18 sshd[1554]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 t
Mai 01 08:09:55 srv-ubuntu18 sshd[1554]: Failed password for pinguin from 192.168.123.131 port 61084 ssh2
Mai 01 08:09:58 srv-ubuntu18 sshd[1554]: Accepted password for pinguin from 192.168.123.131 port 61084 ssh2
Mai 01 08:09:58 srv-ubuntu18 sshd[1554]: pam_unix(sshd:session): session opened for user pinguin by (uid=0)
pinguin@srv-ubuntu18:~$ █

```

Abbildung 18: Servicestatus ssh

Als Test wird eine SSH-Verbindung über die Kommandozeile eines benachbarten Windows10-Rechners aufgebaut.

```
ssh pinguin@192.168.123.160
```

Durch den in Windows10 enthaltenen SSH-Client wird die Verbindung nach Austausch eines Schlüssels (=Fingerprint) aufgebaut. Der Zugriff auf die Shell des Ubuntu-Servers ist damit möglich (siehe Abbildung 11).

```

C:\>ssh pinguin@192.168.123.160
pinguin@192.168.123.160's password:
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-47-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Wed May  1 09:46:12 UTC 2019

```

```

pinguin@srv-ubuntu18:~$ ls
ergebnis  errorlog  home  protokoll.txt  restore  system
pinguin@srv-ubuntu18:~$ pwd
/home/pinguin
pinguin@srv-ubuntu18:~$

```

Folgende Kommandos sind zur Steuerung des SSH-Dienstes wichtig:

Dienst stoppen und wieder starten

```
sudo service ssh stop
sudo service ssh start
```

Dienst abschalten (auch nach dem Booten)

```
sudo systemctl disable ssh
```

Dienst wieder einschalten (auch nach dem Booten)

```
sudo systemctl enable ssh
```

Statt über das Kommandofenster können weitere und komfortablere Verbindungsmöglichkeiten über die Tools PuTTY, WinSCP und andere aufgebaut werden!

## Fernzugriff über ssh mit PuTTY

Mit **PuTTY** können Verbindungen über Secure Shell, Telnet, Remote login oder serielle Schnittstellen zu einem entfernten Host aufgebaut werden. **PuTTY** dient dabei als Client. Während des Verbindungsaufbaus erfolgt eine Prüfung der Identität des Benutzers. Hierzu wird die bereitgestellte Methode zur Authentifizierung überprüft. **PuTTY** ist frei verfügbar für Windows und Linux.



Die Verbindung wird über das Startfenster von **PuTTY** eingerichtet (siehe Abbildung 12). Da **PuTTY** neben SSH auch Telnet und serielle Verbindungen unterstützt, ist es wichtig, das richtige Protokoll (SSH) und den korrekten Port (22) einzustellen. Zusätzlich kann für die Sitzung das Verhalten des Terminals (Tastatur=Terminal, Fenster=Window) eingestellt und für weitere Verbindungen abgespeichert werden. Vor allem die Zeichensatzeinstellung (Window-Translation) muss häufig angepasst werden.

- Einstellungen vornehmen
- Name Vergeben (Saved Sessions)
- Save

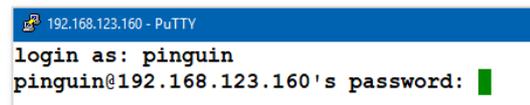
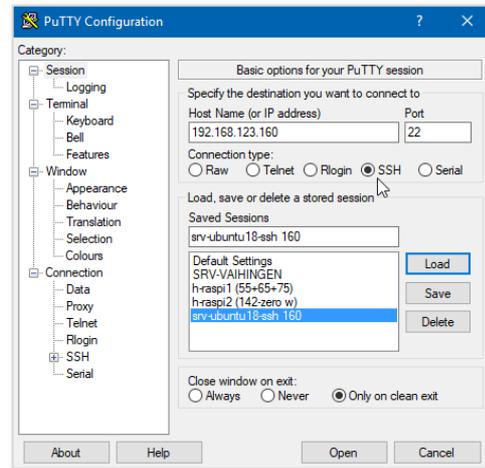


Abbildung 19: ssh-Verbindung mit puTTY

Die Verbindung wird mit "Open" gestartet. Am Ende der SSH-Sitzung wird im Terminal **exit** eingegeben. Danach schließt sich **PuTTY** automatisch.

Auch unter Ubuntu-Desktop kann **PuTTY** nachinstalliert werden:

```
apt-get install putty
```

Das Anwendungsicon kann anschließend über „Anwendungen anzeigen – Alle“ gesucht und zum Starten des Programms verwendet werden.

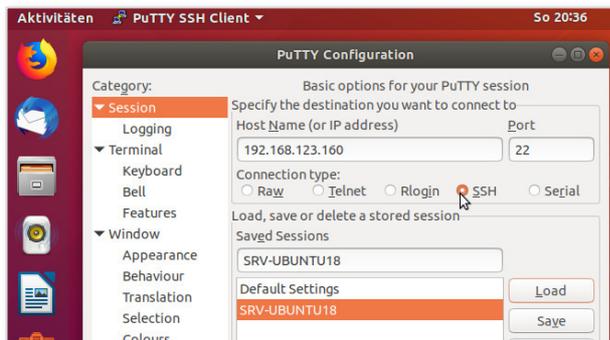


Abbildung 20: PuTTY mit Ubuntu-Desktop



## Fern-Zugriff auf Verzeichnisebene

Soll ein komfortabler Dateiaustausch zwischen Client und Server durchgeführt werden, empfehlen sich Protokolle wie **ftp**, **smb**, **sftp/ssh** oder **webdav** einzusetzen. Der Ubuntu-Dateimanager kann mit diesen Protokollen umgehen. Die Verbindung zum Remoterechner kann in mehreren Fenstern bzw. Registern angezeigt werden.

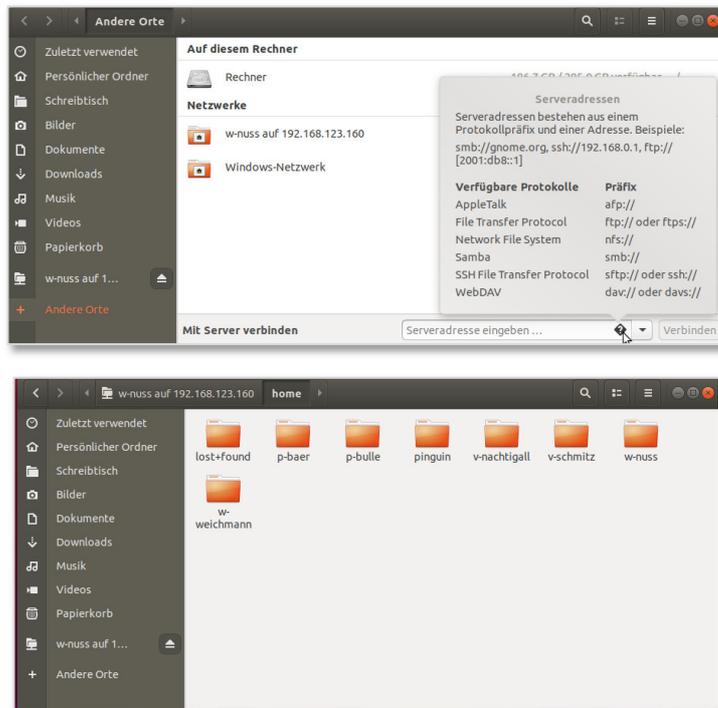


Abbildung 21: sftp-Dateiaustausch

WinSCP bietet von einem Windows-Rechner aus Zugriff auf das Dateisystem des Linux-PC's. Die Anzeige erfolgt in einem geteilten Fenster mit Lokal- und Remotebereich.. Hierzu wird ebenso wie bei PuTTY die verschlüsselte SSH-Verbindung benutzt. Beim Verbindungsaufbau erfolgt wie bei PuTTY ein Schlüsselaustausch („Fingerprint“)

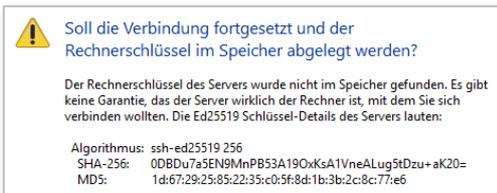
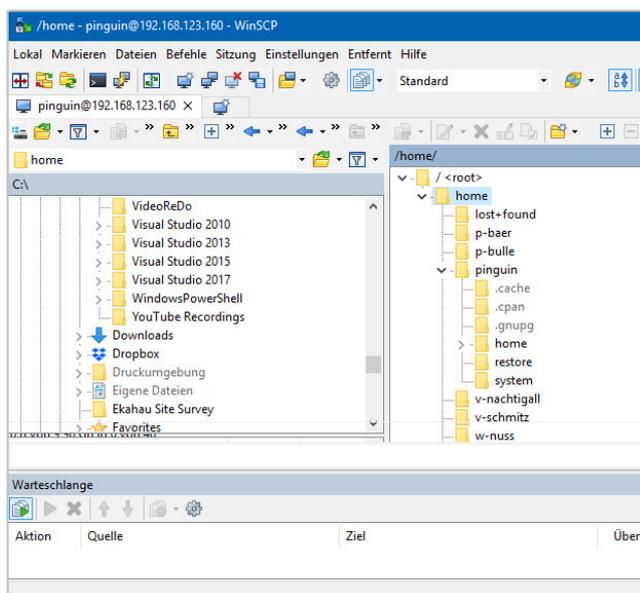


Abbildung 22: WinSCP-Datenaustausch

### 3.4 Root-Zugriff auf Ubuntu

Standardmäßig ist bei Ubuntu 18.04 der User `root` ohne Passwort installiert und kann sich damit nicht am System anmelden. Um an der Konsole oder per Remote auf den Server mit dem Benutzer `root` zuzugreifen, sind folgende Anpassungen nötig:

Passwort setzen

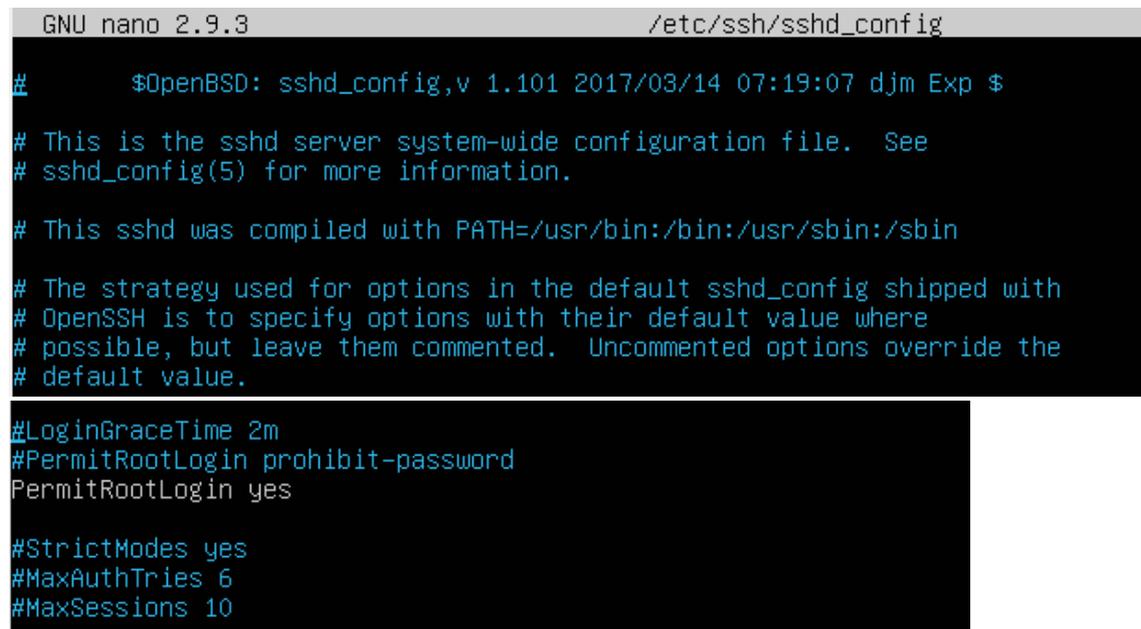
```
sudo passwd root
```

Damit ist zwar die lokale Anmeldung möglich, ein Zugriff über SSH (PuTTY, WinSCP,...) ist jedoch weiterhin aus Sicherheitsgründen gesperrt.



Abbildung 23: Anmeldung mit dem Systemverwalter `root`

Soll auch diese Anmeldung möglich sein, muss in der Datei `/etc/ssh/sshd_config` eine Änderung gemacht werden. Der Eintrag `PermitRootLogin yes` erlaubt den Fernzugriff mit dem Benutzer `root`.



Anschließend wird der `ssh`-Dienst neu gestartet und der Fernzugriff gelingt.

```
sudo service ssh restart
```

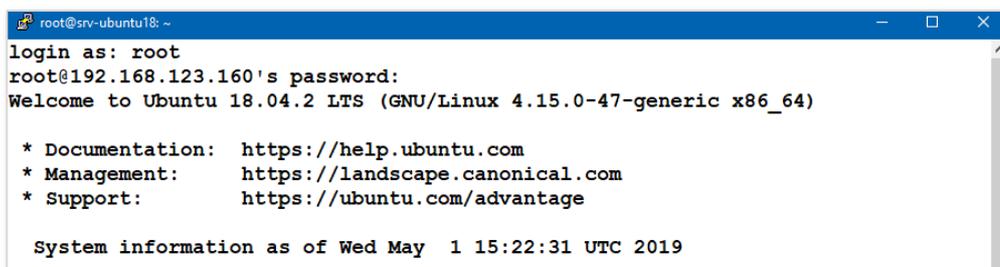


Abbildung 24: `root`-Fernzugriff mit Windows-PuTTY

## 4 Webserver

Ein Webserver überträgt Dokumente an einen Client, der diese z.B. über eine Webbrowser-Software anfordert.



Der Apache HTTP Server ist einer der weltweit meistbenutzten Webserver. Eine Statistik aus 02/2017 zeigt Apache mit 45,78% an der Spitze einer Liste, in der der Microsofts IIS nur 9,22% Markt-anteil besitzt (Quelle: [www.netcraft.com](http://www.netcraft.com)). Er läuft unter der "Apache-Lizenz", eine durch die Free Software Foundation anerkannte Freie-Software-Lizenz. Die erste Version wurde 1995 veröffentlicht. Der Apache-Server kann mit Modulen erweitert werden, die dann Zusatzfunktionen ermöglichen. Beispiele hierfür sind:

- Einbindung von Skriptsprachen (z.B. PHP, Python, Perl)
- Unterstützung von zusätzlichen Protokollen (z.B. WebDAV)

### 4.1 Webserverfunktionen

Der Webserver hat als Hauptaufgabe das Speichern von Inhalten auf einem Festplattenbereich und die Auslieferung der angeforderten Dateien an den Benutzer. Dabei kann es sich um statische Daten (unveränderliche html- oder Mediendateien), aber auch um dynamisch auf Anforderung des Zielsystems aufbereitete Daten handeln.

Die folgenden Abbildungen zeigen beispielhaft die Ordnerstrukturen `C:\xampp\htdocs\` des Apacheservers auf einem Windowssystem (XAMPP) und zum Vergleich die Struktur auf einem Linux-Server des Providers T-Online für ein Kundenkonto unter `/home/www/public_html/`.

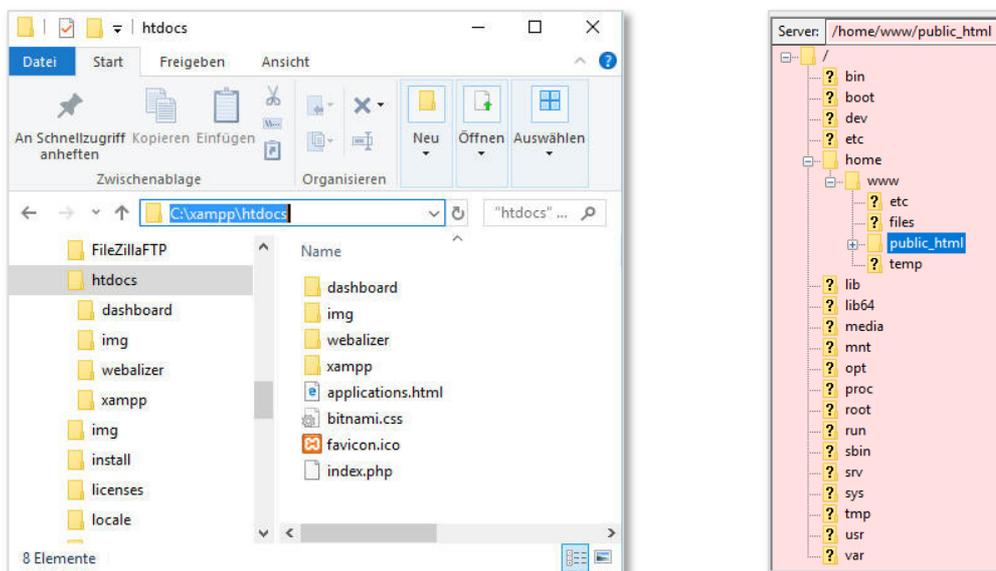


Abbildung 25: Ordnerstruktur für Webserver unter XAMPP und T-Online

Für jede benötigte Datei muss der Webbrowser an den Webserver eine Anfrage senden, d.h. zur Aufbereitung einer umfangreichen Webseite sind u.U. hunderte Anfragen notwendig. Er ist dabei in der Lage, die Inhalte einer Webseite gleichzeitig auf viele verschiedene Rechner auszuliefern. Die Komplexität bestimmt die Performance der Zugriffe. Dynamische Inhalte benötigen mehr Ressourcen als statische.

Als Standarddokument werden in der Regel die Dateinamen `index.html`, `index.php` oder

ähnliche am Webserver eingestellt. Dadurch wird diese Datei, sofern vorhanden, als Startdatei geladen, ohne dass diese der User im Browser extra eingeben muss.

## 4.2 Übertragungsmethoden

Die Übertragungsprotokolle HTTP und HTTPS und die Netzwerkprotokolle IP und TCP werden zur Verbindung der Systeme eingesetzt. Üblicherweise verwendet HTTP den Kommunikationsport 80 und HTTPS den Port 443.

Bei den meisten Webseiten werden die Inhalte erst beim Aufruf aus unterschiedlichen Quellen zusammengestellt. Skriptsprachen wie ASP, PHP oder JSP setzen serverseitig das Dokument aus dem zuvor getrennten Inhalt aus Dateien und Datenbankabfragen dynamisch zusammen.

## 4.3 Installation und Funktionstest

Häufig wird bei der Installation des Webservers unter Linux von einem LAMP-System gesprochen. LAMP ist das Kürzel für Linux-Apache-MySQL-PHP.

Die folgenden Schritte installieren zunächst einen Apache-Server auf Ubuntu 18.04 LTS (Bionic Beaver). Apache 2 ist verfügbar als Ubuntu-Paket, hierfür wird folgende Installation gestartet:

```
apt-get -y install apache2
```

Nachdem die Installation abgeschlossen ist, folgt der Funktionstest. Dazu wird von einem entfernten Client über einen Browser auf die Serveradresse zugegriffen. Dies funktioniert über einen Namen wie z.B. **http://srv-ubuntu18** oder über die IP-Adresse **http://192.168.123.160**.

Die Namensauflösung ist nur dann funktionsfähig, wenn sich im Netzwerk ein Namensserver (DNS) oder eine andere Möglichkeit zur Hostnameauflösung befindet.

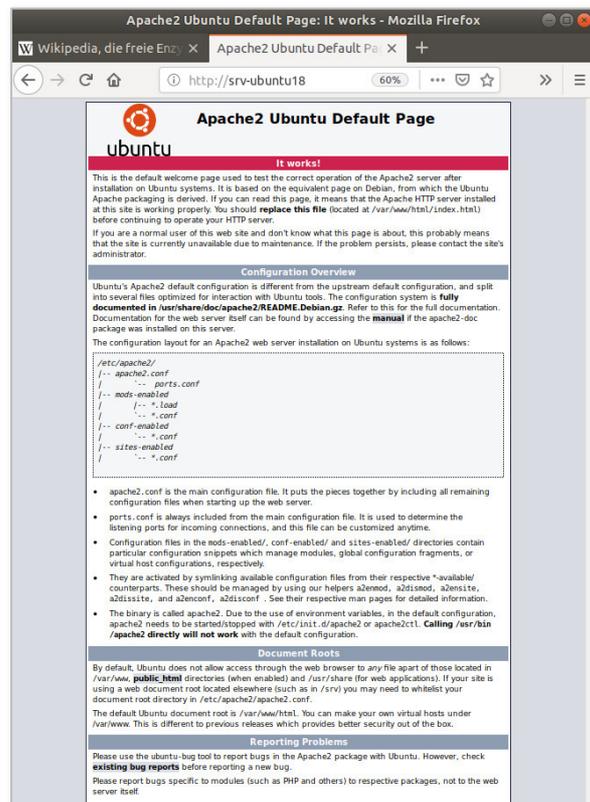


Abbildung 26: Apache-Funktionstest

Der Webserver Apache2 gibt unter Ubuntu 18.04 eine ausführliche Demoseite aus (siehe Abbildung 26), bei anderen Distributionen erscheint teilweise nur eine einfachere Seite mit dem Text "it works!". Die Programmdateien des Webservers befinden sich unter **/usr/bin/apache2/**.

Die Haupt-Konfigurationsdatei des Webservers ist **/etc/apache2/apache2.conf**. In dieser Datei werden alle einzelnen Konfigurationsmodule zusammengeführt. Als Dokumenten-Hauptverzeichnis für eine einzelne Website wird vorgegeben **/var/www/html**, andere Distributionen weichen hier teilweise ab!

## 4.4 Apache starten/stoppen

Zum Starten und Stoppen des Servers können vorgefertigte Scripts verwendet werden. Die Anwendung des Steuerscripts **apachectl** ist die empfohlene Methode, das http-Programm zu starten und zu stoppen (siehe unten). Es setzt verschiedene Variablen, die für die Funktion unter einigen Systemen notwendig sind.

Muss der Server direkt nach einem System-Neustart laufen, sollte ein Aufruf von **apachectl** zu den Startdateien des Systems hinzugefügt werden (üblicherweise ist dies **rc.local** oder eine Datei in einem **rc.n**-Verzeichnis). Dies startet Apache unter root-Rechten.

Apache neustarten

```
systemctl restart apache2.service
```

Apache anhalten

```
systemctl stop apache2.service
```

Apache starten

```
systemctl start apache2.service
```

## 4.5 Apache für VHOSTS vorbereiten

Der Webserver ist auf dem RASPI unter folgendem Verzeichnisbaum gespeichert: **/var/www/**. Darunter wird in der Regel die Startdatei **index.html** abgelegt und vom APACHE dann automatisch abgerufen.

Im Projekt sollen jedoch mehrere Websites auf einem einzigen Server betrieben werden. Hierzu bietet APACHE mit so genannten virtuellen Hosts (VHOSTS) mehrere Möglichkeiten:

- namensbasierte Virtuelle Hosts
- IP-Basierte Virtuelle Hosts

Im folgenden Abschnitt wird die Vorgehensweise zum Konfigurieren mehrerer IP-basierter VHOSTS beschrieben.

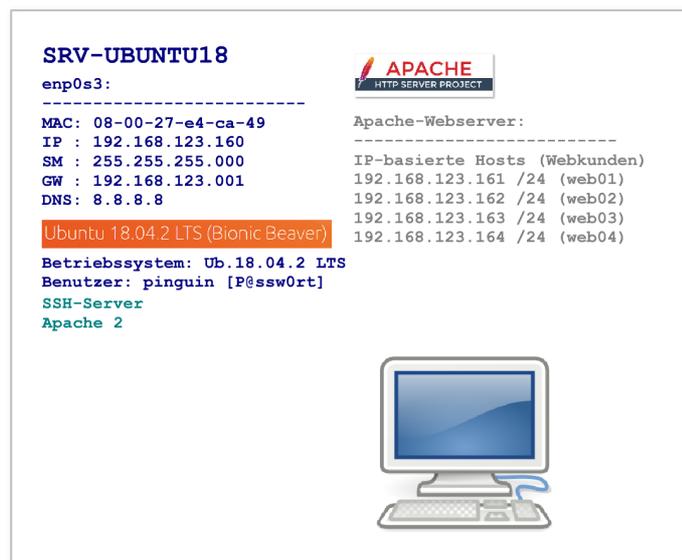
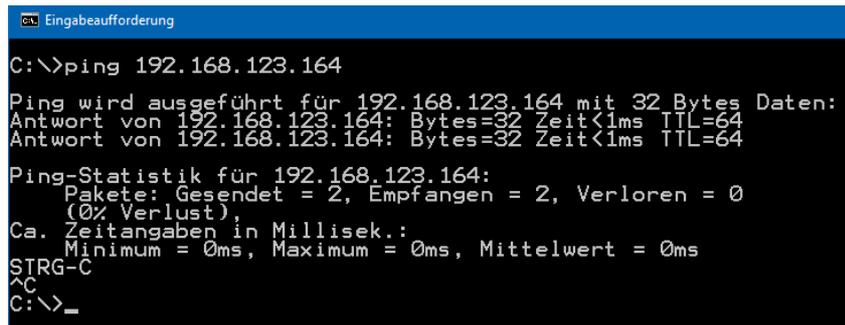


Abbildung 27: zusätzliche IP-Adressen planen

Für jeden VHOST muss eine weitere IP-Adresse an die Netzwerkkarte gebunden werden. Dies ist bei IPv6 kein Problem, da hier genügend Adressen zur Verfügung stehen. Innerhalb eines privaten Netzwerks können auch nahezu beliebig viele IP-Adressen der Version 4 verwendet werden. Im Endzustand ist soll der Server für vier Kunden getrennt unter den IP-Adressen **192.168.123.161** bis **192.168.123.164** erreichbar. Das Anlegen mit der Methode des Netzwerkmanagers mit Hilfe der **yaml**-Datei ist unter Abschnitt 3.2 (Seite13) ausführlich beschrieben.

Die Funktion der zusätzlichen IP-Adressen wird durch Anpingen von den Clients aus getestet.



```

C:\>ping 192.168.123.164

Ping wird ausgeführt für 192.168.123.164 mit 32 Bytes Daten:
Antwort von 192.168.123.164: Bytes=32 Zeit<1ms TTL=64
Antwort von 192.168.123.164: Bytes=32 Zeit<1ms TTL=64

Ping-Statistik für 192.168.123.164:
    Pakete: Gesendet = 2, Empfangen = 2, Verloren = 0
    (0% Verlust),
    Ca. Zeitangaben in Millisek.:
    Minimum = 0ms, Maximum = 0ms, Mittelwert = 0ms
STRG-C
^C
C:\>_
  
```

Abbildung 28: ping-Test vom Windows Client aus

#### 4.6 Planung der Web-User (Website-Kunden)

Zur Vorarbeit für kundentrennte Websites (VHOSTS) müssen separate Speicherbereiche auf der Serverfestplatte eingerichtet werden. Dies kann direkt an der Kommandozeile des Servers oder über den SSH-Zugriff mit root-Rechten über PuTTY, WinSCP o.ä. erfolgen.

Damit später mit getrennten Kunden (Usern) von deren Clients auch per Upload und SCP-Zugriff gearbeitet werden kann, werden bereits jetzt die Vorbereitungen auf Seiten der Linux-Benutzerverwaltung getroffen.

- Benutzer unter Linux anlegen mit leerem Homeverzeichnis (hierüber erfolgt später der ftp und SCP-Zugriff)
- Im Homeverzeichnis jedes Users Unterordner html (für den Apachezugriff) und files (für zusätzliche Dateien) erzeugen.

Im folgenden Abschnitt sind die einzelnen Arbeitsschritte zum Anlegen der User und deren Web-Speicherplatz dargestellt.

#### 4.7 Web-User erstellen

Alle Web-User (Kunden) werden nun über die Kommandozeile neu erstellt.

- Wechseln ins Apache-Webverzeichnis (Document-root)

```
cd /var/www
```

- Leeres Verzeichnis **/home/ordnerleer** anlegen (dieser wird unten für das User-Homeverzeichnis benötigt)

```
sudo mkdir /home/ordnerleer
```

- Webkunden als LINUX-User anlegen mit leerem Homeverzeichnis (**-k**), nicht unter **/home (-d)**, sondern unter **/var/www**. Anschließend Passwort setzen.

```
sudo useradd -d /var/www/web01 -m -k /home/ordnerleer webkunde01
```

```
sudo passwd webkunde01
```

...

```
sudo useradd -d /var/www/web04 -m -k /home/ordnerleer webkunde04
```

```
sudo passwd webkunde04
```

Nun werden die Verzeichnisse für das Startdokument jedes Kunden im Verzeichnisbaum erzeugt. Zusätzlich ein weiteres Verzeichnis für Zugangsdaten außerhalb des so genannten DocumentRoot.

```
sudo mkdir /var/www/web01/html
sudo mkdir /var/www/web01/files
...
sudo mkdir /var/www/web04/html
sudo mkdir /var/www/web04/files
```

Anschließend wird der Login der neuen User getestet (hier im Beispiel **webkunde04**):

```
login as: webkunde04
webkunde04@192.168.123.160's password:
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-47
```

Außerdem sollte der Webkunde Dateien in den Verzeichnissen **html** und **files** neu erstellen können. Hier werden später die Dateien des Webauftritts gespeichert.

```
$ pwd
/var/www/web04
$ ls
files html
$ cd html
$ touch testdatei.txt
touch: cannot touch 'testdatei.txt': Permission denied
$
```

Der Login funktioniert zwar, der User befindet sich in seinem jeweiligen Homeverzeichnis, ein Anlegen von Dateien im Ordner **html** schlägt jedoch fehl!

Grund für die Fehlfunktion ist, dass die Ordner mit der Systemverwalterberechtigung **sudo** angelegt wurden und dadurch nun Eigentümer von **root** sind. Damit die Benutzer auch unter seinem jeweiligen Verzeichnis schreiben darf, werden die Rechte entsprechend gesetzt. Dies muss für alle angelegten Webverzeichnisse durchgeführt werden.

```
sudo chown webkunde01 web01 -R
...
sudo chown webkunde02 web02 -R
```

## 4.8 VHOSTS für den Apache-Server konfigurieren

Damit der Apache-Webserver auch die getrennten Sites verwalten kann, müssen Eintragungen in eine Konfigurationsdatei gemacht werden.

- Neue Datei erstellen

```
touch /etc/apache2/sites-available/160-vhosts.conf
```

- VHOSTS eintragen

```
nano /etc/apache2/sites-available/160-vhosts.conf
```

Das Beispiel zeigt die Erstellung von vier virtuellen Hosts. Als Standardserver ist Apache unter der Haupt-IP-Adresse des Hosts **192.168.123.160** erreichbar. Als DocumentRoot ist **/var/www/html** eingestellt.

Nun werden VHOSTs hinzugefügt.

```

/etc/apache2/sites-available/160-vhosts.conf
<VirtualHost 192.168.123.161:80>
    DocumentRoot "/var/www/web01/html"
    ServerName web01
</VirtualHost>
<VirtualHost 192.168.123.162:80>
    DocumentRoot "/var/www/web02/html"
    ServerName web02
</VirtualHost>
<VirtualHost 192.168.123.163:80>
    DocumentRoot "/var/www/web03/html"
    ServerName web03
</VirtualHost>
<VirtualHost 192.168.123.164:80>
    DocumentRoot "/var/www/web04/html"
    ServerName web04
    <Directory "/var/www/web04/html">
        allow from all
        Options None
        Require all granted
    </Directory>
</VirtualHost>
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text

```

Abbildung 29: VHOSTs konfigurieren

Anmerkung: Bei Host 164 wurden beispielhaft zusätzliche Rechte für das Webverzeichnis vergeben.

- Neueintragungen aktivieren

Die erstellte Datei wird dem Apacheserver bekannt gemacht, anschließend erfolgt ein Neustart des Dienstes.

```
sudo a2ensite 160-vhosts.conf
systemctl reload apache2
```

## 4.9 Websites testen

Das Testen der VHOST-Webseiten geschieht durch Erzeugen einer Startdatei `index.html`-Datei im jeweiligen DefaultRoot-Verzeichnis (z.B. `/var/www/web04/html/`). Hierzu wird ein einfacher Texteditor verwendet. Für jede Website muss zum Funktionstest je eine Startseite erstellt werden.

```
sudo nano /var/www/web04/html/index.html
```

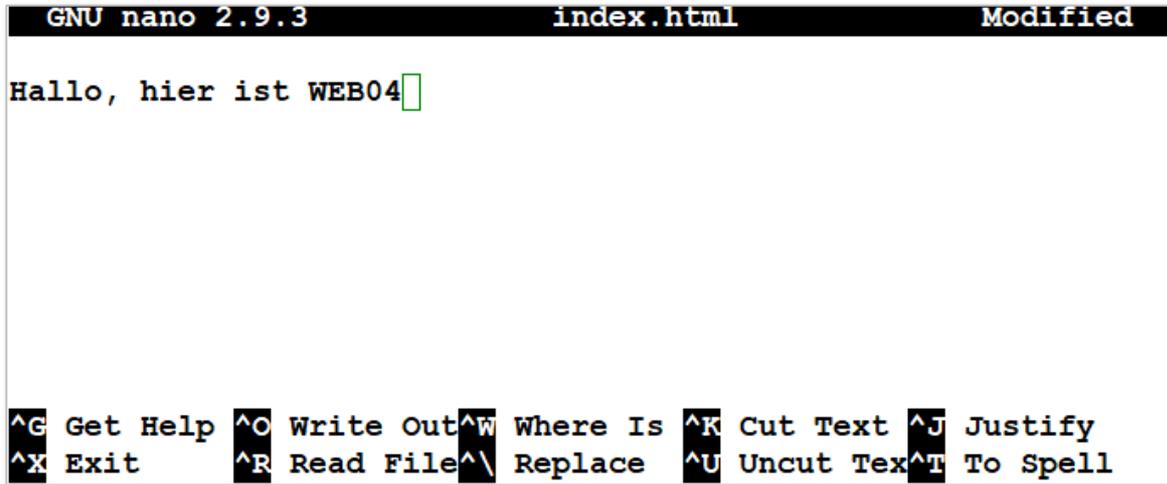


Abbildung 30: Startseite

Die Startseite wird von einem entfernten Host per URL erfolgreich aufgerufen.

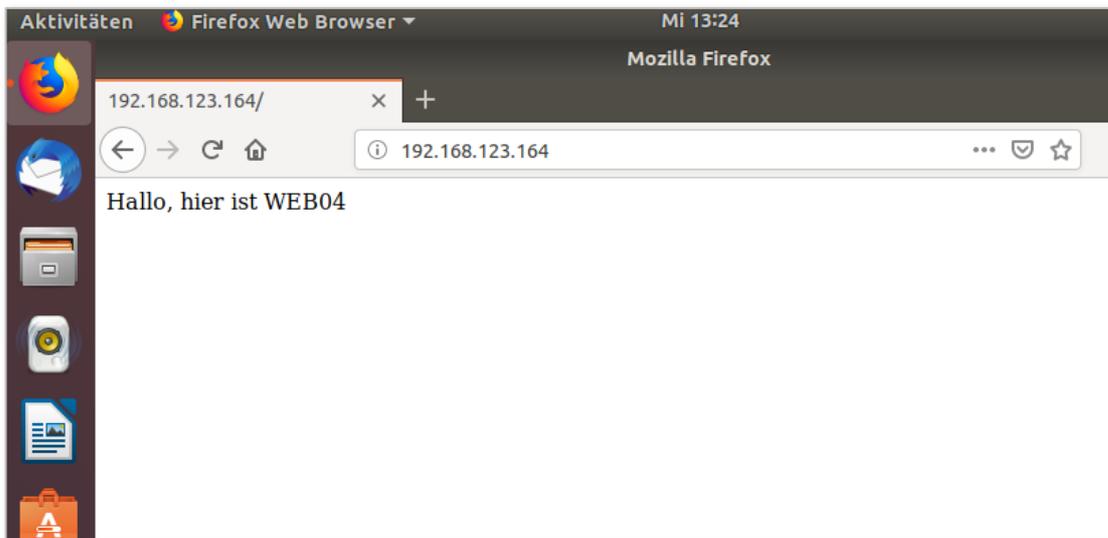


Abbildung 31: Test der Webseite

## 4.10 Funktionsweise der Apache-Konfiguration

Die Hauptdatei **apache2.conf** wurde früher für die komplette Konfiguration eingesetzt. Heute werden viele Funktionen in einzelnen weiteren Textdateien und hier entsprechend eingebettet. Unter der Debian-Distribution wird die Datei **apache2.conf** folgendermaßen ausgeliefert (Teilausschnitte):

Beschreibung:

```

1  # This is the main Apache server configuration file. It contains the
2  # configuration directives that give the server its instructions.
3  # See http://httpd.apache.org/docs/2.4/ for detailed information about
4  # the directives and /usr/share/doc/apache2/README.Debian about Debian specific
5  # hints.
6  #
7  #
8  # Summary of how the Apache 2 configuration works in Debian:
9  # The Apache 2 web server configuration in Debian is quite different to
10 # upstream's suggested way to configure the web server. This is because Debian's
11 # default Apache2 installation attempts to make adding and removing modules,
12 # virtual hosts, and extra configuration directives as flexible as possible, in
13 # order to make automating the changes and administering the server as easy as
14 # possible.
15
16 # It is split into several files forming the configuration hierarchy outlined
17 # below, all located in the /etc/apache2/ directory:
18 #
19 # /etc/apache2/
20 # |-- apache2.conf
21 # |  |-- ports.conf
22 # |-- mods-enabled
23 # | |-- *.load
24 # | |-- *.conf
25 # |-- conf-enabled
26 # | |-- *.conf
27 # |-- sites-enabled
28 # |-- *.conf

```

Sicherheitsmodell: Apache2 darf nur Dateien unterhalb der Verzeichnisse **/var/www/** bzw. **/usr/share/** über den Dienst HTTPD ausliefern.

```

153 # Sets the default security model of the Apache2 HTTPD server. It does
154 # not allow access to the root filesystem outside of /usr/share and /var/www.
155 # The former is used by web applications packaged in Debian,
156 # the latter may be used for local directories served by the web server. If
157 # your system is serving content from a sub-directory in /srv you must allow
158 # access here, or in any related virtual host.
159 <Directory />
160     Options FollowSymLinks
161     AllowOverride None
162     Require all denied
163 </Directory>
164
165 <Directory /usr/share>
166     AllowOverride None
167     Require all granted
168 </Directory>
169
170 <Directory /var/www/>
171     Options Indexes FollowSymLinks
172     AllowOverride None
173     Require all granted
174 </Directory>

```

Einbetten (Include) der VHOSTs-Definitionen (Zeile 225):

```
218 # Include of directories ignores editors' and dpkg's backup files,  
219 # see README.Debian for details.  
220  
221 # Include generic snippets of statements  
222 IncludeOptional conf-enabled/*.conf  
223  
224 # Include the virtual host configurations:  
225 IncludeOptional sites-enabled/*.conf  
226  
227 # vim: syntax=apache ts=4 sw=4 sts=4 sr noet  
228
```

Für jeden VHOST wird je eine Konfigurationsdatei im Ordner **/etc/apache2/sites-available/** angelegt. Nach dem Ändern der Dateien muss der Apache-Dienst neu gestartet werden.

## 5 Scriptsprache PHP

Die Skriptsprache PHP („PHP: Hypertext Preprocessor“, ursprünglich „Personal Home Page Tools“) ist an die höhere Programmiersprache C und an Perl angelehnt. Sie dient vor allem zur Erstellung dynamischer Webseiten. PHP wird als freie Software unter der PHP-Lizenz verbreitet. Die Sprache zeichnet sich durch die Unterstützung von Datenbanken und durch viele Funktionsbibliotheken aus. Sie kann direkt in HTML eingebettet werden.



PHP unterscheidet sich von clientseitigen Sprachen wie Javascript dadurch, dass der Code auf dem Server ausgeführt wird und dort HTML-Ausgaben generiert, die an den Client gesendet werden. Die Abfolge eines Seitenabrufs vom Client aus ist in Abbildung 34 dargestellt. Der Client erhält nur das Ergebnis der Skriptausführung, ohne dass es möglich ist herauszufinden, wie der eigentliche Code aussieht.

Im vorliegenden Projekt soll PHP in Zusammenarbeit mit dem Webserver Apache konfiguriert werden. Dadurch wird die Ausführung einer Vielzahl von Webanwendungen auf dem Server möglich.

### 5.1 PHP Versionsgeschichte

PHP wurde 1995 von Rasmus Lerdorf entwickelt und wurde bis Anfang 2013 auf etwa 244 Millionen Websites eingesetzt, Ende 2015 von 82 % aller Websites als serverseitige Programmiersprache verwendet. Sie ist die am häufigsten verwendete Programmiersprache zum Erstellen von Websites. Zudem ist sie bei den meisten Webhostern vorinstalliert.

- 1995 Entwicklung als „Personal Home Page“
- PHP2: Weiterentwicklung in C-ähnlicher Syntax
- PHP3: Neu codiert und weiterentwickelt 1997
- PHP4: Ausführungsgeschwindigkeit verbessert, Sicherheit der globalen Variablen verbessert  
Unterstützung für viele Webserver, Session Management  
PHP löste den damaligen de facto Standard Perl ab
- PHP 5 (veröffentlicht 2005): Diese Version hatte ein verbessertes Objektmodell, objektorientierte Anwendungen können effizienter ausgeführt werden.
- PHP 7: Die Versionsnummer PHP 6 wurde übersprungen, da die Weiterentwicklung vor der Veröffentlichung eingestellt wurde. PHP 7.1 wurde im Dezember 2016 veröffentlicht, aktuelle Version ist PHP 7.2.9 (Stand 16.8.2018)

### 5.2 PHP7 auf dem Server installieren

Bevor die Installation von PHP gestartet wird, sollt das System auf den neusten Stand gebracht werden.

```
sudo apt update
sudo apt upgrade
```

Anschließend wird die Softwarequelle verändert, da PHP 7.3. (noch) nicht in den aktuellen Ubuntu-Quellen vorhanden ist und geupdatet.

```
sudo add-apt-repository ppa:ondrej/php
sudo apt-get update
```

PHP 7.3 wird installiert.

```
sudo apt-get install php7.3
```

Die installierte Version muss geprüft werden.

```
php -v
```

```
penguin@srv-ubuntu18:~$ php -v
PHP 7.3.4-1+ubuntu18.04.1+deb.sury.org+3 (cli) (built: Apr 10 2019 10:51:11) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.3.4, Copyright (c) 1998-2018 Zend Technologies
    with Zend OPcache v7.3.4-1+ubuntu18.04.1+deb.sury.org+3, Copyright (c) 1999-2018,
penguin@srv-ubuntu18:~$ █
```

Abbildung 32: php-Versionstest

### 5.3 PHP-Erweiterungen installieren

Die PHP-Erweiterungen werden je nach Bedarf auf folgende Art installiert, z.B.:

```
sudo apt install php7.3-cli php7.3-json php7.3-pdo php7.3-mysql
php7.3-zip php7.3-gd php7.3-curl php7.3-xml php7.3-bcmath
```

Nun werden die installierten PHP-Erweiterungen geprüft:

```
apt policy php7.3-cli
```

```
penguin@srv-ubuntu18:~$ apt policy php7.3-cli
php7.3-cli:
  Installiert:          7.3.4-1+ubuntu18.04.1+deb.sury.org+3
  Installationskandidat: 7.3.4-1+ubuntu18.04.1+deb.sury.org+3
  Versionstabelle:
  *** 7.3.4-1+ubuntu18.04.1+deb.sury.org+3 500
      500 http://ppa.launchpad.net/ondrej/php/ubuntu bionic/main amd64 Package
      100 /var/lib/dpkg/status
penguin@srv-ubuntu18:~$ █
```

Abbildung 33: php-Erweiterungen

Ein Neustart des Webservers schließt die Installation ab.

```
sudo systemctl restart apache2
```

### 5.4 PHP-Funktionstest

PHP ist eine Skriptsprache, die auf dem Webserver ausgeführt wird. Deshalb können PHP-Dateien nicht auf dem lokalen Rechner gestartet werden und müssen immer im Serververzeichnis gespeichert sein.

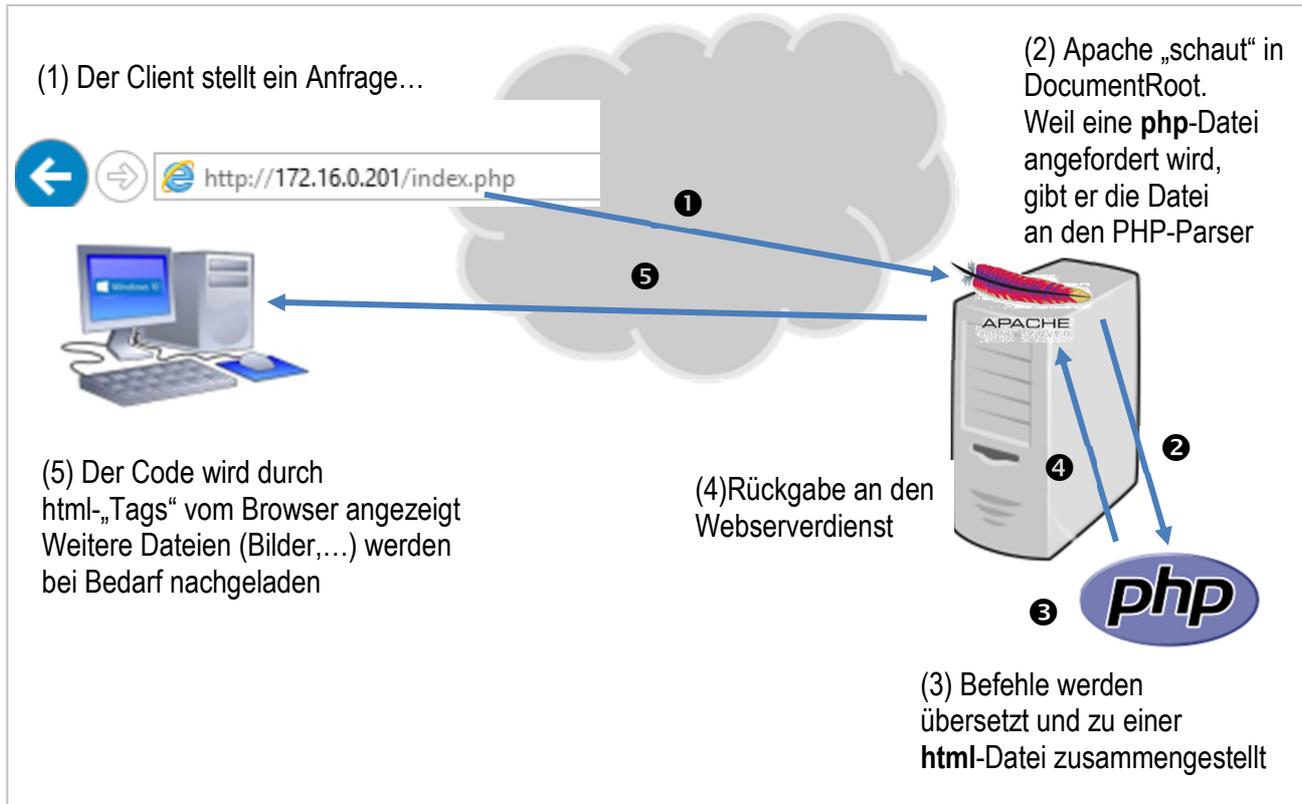


Abbildung 34: Abruf einer PHP-Datei

Im DocumentRoot des VHOSTs soll nun ein PHP-Dokument auf Funktionsfähigkeit getestet werden. Hierzu wird mit einem Texteditor eine php-Datei angelegt und mit der typischen „Hallo-Welt“-Ausgabe programmiert. Zusätzlich erfolgt die Anzeige der Servereinstellungen über die Funktion `phpinfo()`. Der Aufruf erfolgt wieder von einem Host im Netzwerk über einen beliebigen Browser.

Wenn die Seite erscheint, ist das PHP-Paket korrekt installiert. Hierzu muss an die http-Webadresse der Dateiname angehängt werden, sofern die Datei nicht `index.php` benannt wurde.

```

/var/www/web04/html/testseite.php
<?php
echo "Hallo Welt -- ";
echo "Dies ist meine erste PHP-Seite";
phpinfo();
    
```

Abbildung 35: PHP-Quellcode



Abbildung 36: Ausgabe der php-Seite

## 6 Datenbankserver installieren

Im Moment sind zwei weit verbreitete MySQL-Datenbanksysteme verfügbar, der klassische „MySQL-Server“, der seit 2010 auch als proprietäre Lizenz von der Firma Oracle vertrieben wird und dem Ableger mit dem Namen MariaDB. Das Datenbanksystem MariaDB wird vom ursprünglichen MySQL-Entwickler Monty Widenius weiterentwickelt.

### 6.1 Vergleich MySQL vs. MariaDB

MySQL ist eines der weltweit meist verbreitetsten Datenbankverwaltungssysteme vom Typ "relationale Datenbank". Die Datenbank bildet die Voraussetzung für viele dynamische Webseiten. MySQL ist als Open-Source-Software und als kommerzielle Enterprise-Version für verschiedene Betriebssysteme erhältlich. Das System MySQL wurde seit 1994 vom schwedischen Unternehmen MySQL AB entwickelt, 2008 von Sun Microsystems übernommen, das dann 2010 von Oracle gekauft wurde. Der Name MySQL deutet auf den Vornamen My der Tochter des Mitbegründers Michael Widenius und das Kürzel der Abfragesprache "Structured Query Language" hin.

Seit der Übernahme vom MySQL von Sun durch Oracle entwickelten sich die Open-Source- und die kommerzielle Variante immer mehr auseinander. Viele Linux-Distributoren ersetzen seit 2013 MySQL durch die Parallelentwicklung MariaDB.

MariaDB ist durch eine Abspaltung ("Fork") aus MySQL entstanden. Das Projekt wurde von MySQLs früherem Hauptentwickler Michael Widenius eingeleitet. Da Oracle die Markenrechte an MySQL hält, wurde der Name MariaDB kreiert, der auf Widenius' jüngere Tochter Maria hinweist.

Der MariaDB Server bildet die Kernkomponente der MariaDB-Umgebung. Es wird hier nicht zwischen einer kommerziellen und Open-Source-Variante unterschieden, es gibt ausschließlich die freie GPL-Variante. Auch ohne Supportvertrag darf der Serverdienst für MariaDB ohne Einschränkung genutzt werden. Er entspricht dem Funktionsumfang der Version MySQL Enterprise.

## 6.2 MariaDB-Installation

Zunächst muss der MariaDB-Server und -Client installiert werden.

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

Nach der Installation wird das root-Passwort für den Datenbankserver gesetzt und weitere Sicherheitseinstellungen durchgeführt.

```
sudo mysql_secure_installation
```

```

→ root@srv-ubuntu18:~# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user.  If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

→ Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

→ You already have a root password set, so you can safely answer 'n'.

→ Change the root password? [Y/n] Y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.

→ Remove anonymous users? [Y/n] y
... Success!

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

→ Disallow root login remotely? [Y/n] n
... skipping.

By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.

→ Remove test database and access to it? [Y/n] y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

→ Reload privilege tables now? [Y/n] Y
... Success!

Cleaning up...

All done!  If you've completed all of the above steps, your MariaDB
installation should now be secure.

→ Thanks for using MariaDB!
root@srv-ubuntu18:~# █

```

Soll das Passwort nachträglich verändert werden, kann dies an der Kommandozeile durchgeführt werden.

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

### 6.3 MariaDB-Funktionstest

Zur Anmeldung am Datenbanksystem wird die Kommandozeile benutzt.

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

```
root@srv-ubuntu18:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 48
Server version: 10.1.38-MariaDB-0ubuntu0.18.04.1 Ubuntu 18.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> quit
Bye
root@srv-ubuntu18:~# █
```

## 6.4 Module nachinstallieren

Im folgenden Abschnitt werden mehrere Module installiert, unter anderem die PHP-Unterstützung für MariaDB (**php7.3-mysql**). Um spezielle Module zu finden wird der **apt-cache** Suchbefehl angewandt.

```
sudo apt-cache search php7.3
```

Oft fehlt der Versionsname bei den Paketen, so dass auch so gesucht werden muss:

```
sudo apt-cache search php-
```

Für das MySQL-Projekt werden nun folgende Module nachinstalliert, die eine Vielzahl zusätzlicher Funktionen bereitstellen.

```
sudo apt-get -y install php7.3-mysql php7.3-curl php7.3-gd php7.3-intl
php-pear php-imagick php7.3-imap php-memcache php7.3-pspell
php7.3-recode php7.3-sqlite3 php7.3-tidy php7.3-xmlrpc
php7.3-xsl php7.3-mbstring php-gettext
```

Ein Neustart des Webservers schließt die Installation ab. Ein nochmaliger Blick auf die **phpinfo()**-Funktion der Testseite zeigt, dass nun MySQL/MariaDB-Unterstützung aktiv ist.

Directive	Local Value	Master Value
memcache.redundancy	1	1
memcache.session_redundancy	2	2

mysqli	
mysqli Support	enabled
Client API library version	mysqli 5.0.12-dev - 20150407 - \$Id: 7cc7cc96e675f6d72e5cf0f267f48e167c2abb23 \$
Active Persistent Links	0
Inactive Persistent Links	0
Active Links	0

Directive	Local Value	Master Value
mysqli.allow_local_infile	Off	Off
mysqli.allow_persistent	On	On
mysqli.default_host	no value	no value
mysqli.default_port	3306	3306
mysqli.default_pw	no value	no value
mysqli.default_socket	no value	no value
mysqli.default_user	no value	no value
mysqli.max_links	Unlimited	Unlimited
mysqli.max_persistent	Unlimited	Unlimited
mysqli.reconnect	Off	Off
mysqli.rollback_on_cached_plink	Off	Off

mysqli	
mysqli	enabled
Version	mysqli 5.0.12-dev - 20150407 - \$Id: 7cc7cc96e675f6d72e5cf0f267f48e167c2abb23 \$
Compression	supported
core SSL	supported
extended SSL	supported
Command buffer size	4096

Abbildung 37: MariaDB mit MySQL auf der Testseite

Als Abschluss wird dringend empfohlen, noch eine Cachefunktion („Opcache“) für PHP zu installieren. Dadurch wird die Ausführung des Codes optimiert.

```
sudo apt-get -y install php7.3-opcache php-apcu
```

Nochmals ein Neustart des Apachen und damit ist die PHP-Implementierung abgeschlossen.

```
sudo systemctl restart apache2
```

## 7 PHP MyAdmin installieren

Zur Administration von MySQL-Datenbanken wird seit 1998 das webbasierte Frontend **phpMyAdmin** entwickelt. Es handelt sich um eine freie Webanwendung. Die Software ist in PHP implementiert. Viele SQL-Funktionen können per Mausklick im Browser ausgeführt werden, z.B. Datensätze auflisten, Tabellen anlegen/löschen, Spalten hinzufügen, Datenbanken anlegen/löschen und Benutzer verwalten.

**phpMyAdmin** ist unter der GNU General Public License lizenziert und ist auch in vielen Linux-Distributionen enthalten. Es ist sehr weit verbreitet und wird unter anderem von großen Webhosting-Providern verwendet z.B. T-Online. Schon im Jahr 2000 gehörte phpMyAdmin zu den populärsten PHP-Anwendungen und MySQL-Administrationswerkzeugen und hatte eine große Nutzer- und Entwicklergemeinde.

Zur Installation von **phpMyAdmin** gibt es verschiedene Vorgehensweisen. Prinzipiell wird das Gesamt-paket als ZIP- oder TAR-Paket von der Seite [www.phpmyadmin.net](http://www.phpmyadmin.net) heruntergeladen und entpackt einfach in ein Unterverzeichnis des Webserver kopiert. Eine gute Dokumentation (engl.) findet sich unter <https://docs.phpmyadmin.net/en/latest/index.html>. Speziell auf die Betriebssysteme abgestimmte Vorgehensweisen, die das System gegen Angriffe sicherer machen, werden unten beschrieben.

### 7.1 Installation auf dem Ubuntu-Server

Zunächst wird **phpMyAdmin** von den Standard-Ubuntu-Repositories installiert. Dies geschieht durch Updaten des Server-Paketverzeichnisses und das Herunterladen über das apt-Paketsystem.

```
sudo apt update
sudo apt install phpmyadmin php-
mbstring php-gettext
```

Das Installationsprogramm stellt einige Fragen zur Konfiguration (siehe rechts).

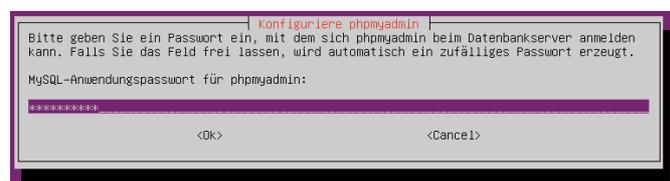
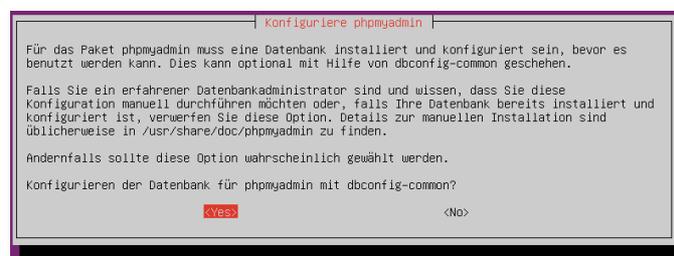
Der Installationsprozess fügt die Apache-Konfigurationsdatei für phpMyAdmin in das **/etc/apache2/conf-enabled/** - Verzeichnis ein, von wo es automatisch gelesen wird.

Nun muss explizit die „mbstring PHP extension“ hinzugefügt werden, Sie wird für die Behandlung verschiedener nationaler und internationaler Zeichensätze benötigt.

```
sudo phpenmod mbstring
```

Anschließend wird Apache neu gestartet

```
sudo systemctl restart apache2
```



## 7.2 Administrator für den Remotezugriff erstellen

PhpMyAdmin ist nun installiert und eingerichtet. Bevor man sich jedoch einloggen und sich mit der MySQL-Datenbank verbinden kann, müssen die MySQL-User Rechte zur Zusammenarbeit mit phpMyAdmin erhalten.

Bei der Installation von phpMyAdmin auf dem Server wird automatisch eine Datenbank mit dem Namen **phpmyadmin** angelegt. Es wird empfohlen sich an phpMyAdmin nicht mit dem MySQL-root, sondern einem anderen MySQL-User anzumelden. In Ubuntu-Systemen, die MySQL ab Version 5.7 installiert haben, wird bezüglich des root-Zugriffs eine erhöhte Sicherheitsstufe aktiviert. MariaDB und MySQL aktivieren ein Plugin mit dem Namen „UNIX\_SOCKET“ für den root-User. Dieses Plugin verhindert, dass sich root an phpMyAdmin anmelden kann und dass TCP-Verbindungen zu MySQL mit dem root-User funktionieren (siehe Abbildung 39).

Zur Lösung des Problems wird nun ein neuer MySQL-User mit dem Namen „**administrator**“ erstellt, der dieselben Rechte wie root erhalten soll.

```
$ sudo mysql -u root
```



Abbildung 38: root Zugriff gesperrt

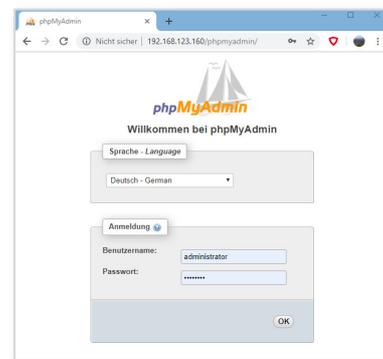


Abbildung 39: Zugriff möglich

Die folgende Grafik zeigt den Einstellungsprozess.

```
penguin@srv-ubuntu18:~$ sudo mysql -u root
[sudo] password for penguin:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 44
Server version: 10.1.38-MariaDB-0ubuntu0.18.04.1 Ubuntu 18.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE USER 'administrator'@'localhost' IDENTIFIED BY 'Passwort';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON *.* TO 'administrator'@'localhost' WITH GRANT OPTION;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

MariaDB [(none)]> exit
Bye
penguin@srv-ubuntu18:~$
```

Abbildung 40: mysql-Grundeinstellung

Nun ist die Anmeldung an phpMyAdmin auch von einem entfernten Rechner mit einem beliebigen Browser möglich.

### 7.3 Funktionsüberblick phpMyAdmin

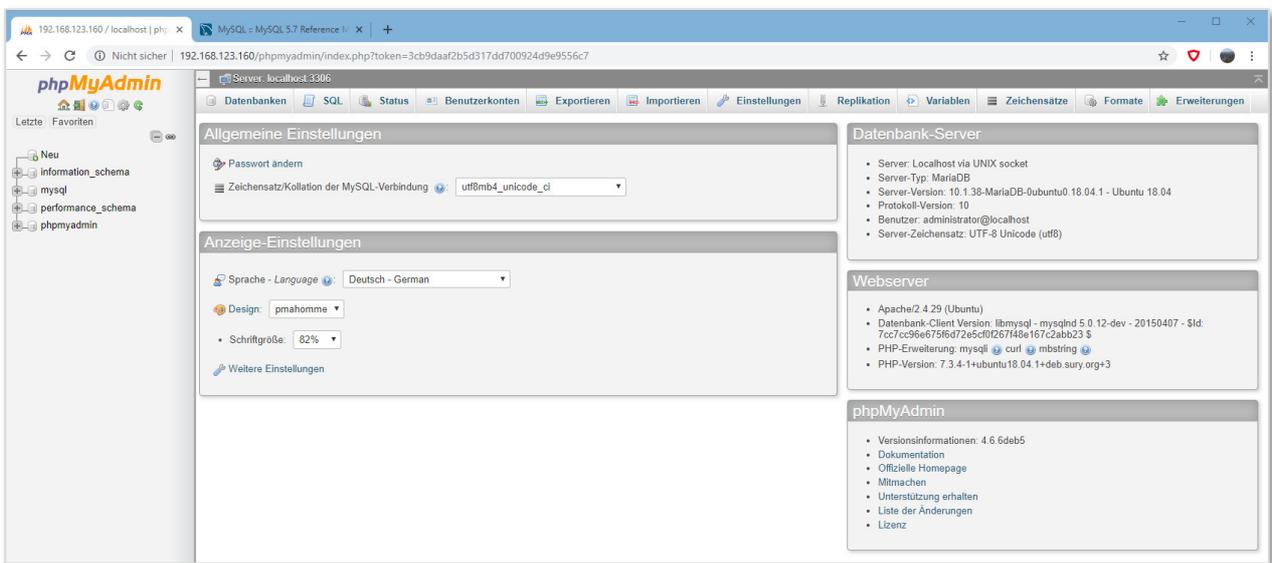
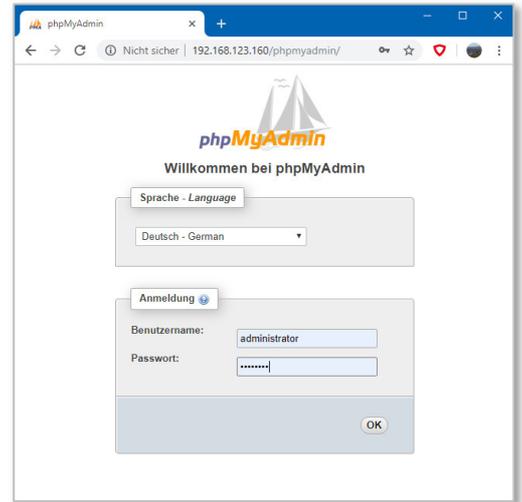
Das Webfrontend **phpmyadmin** dient als Programm zum Bearbeiten der Datenbanken sowohl aus Benutzersicht als auch aus Datenbankadministrator- / Programmiersicht.

#### Anmeldung

Die Anmeldung an der Datenbank erfolgt über das Frontend von **phpmyadmin** durch Aufruf der Startseite, die auf dem Webserver im Unterverzeichnis **...html/phpmyadmin** abgelegt ist.

```
http://webserver.local/phpmyadmin
```

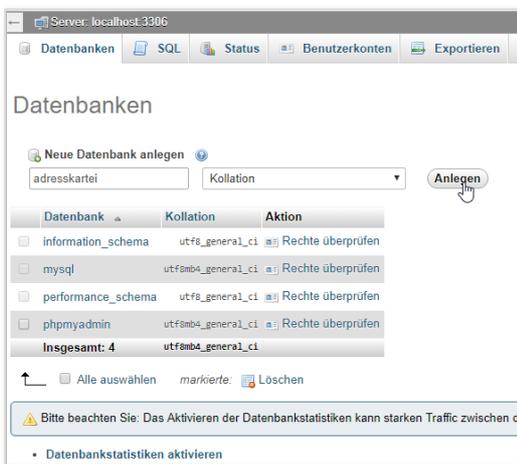
Das Startfenster zeigt das Objektpanel links, eine Leiste mit Menüschaltflächen oben und das Hauptpanel zentral.



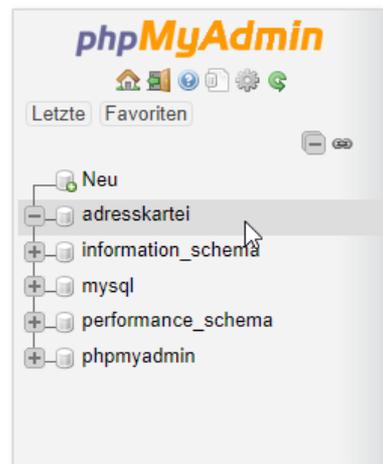
In den folgenden Abschnitten werden die wichtigsten Funktionen zum Anlegen einer Datenbank in Stichworten aufgelistet und durch Screenshots dargestellt.

#### Datenbank neu anlegen

Datenbanken... - Neue Datenbank anlegen – Name der Datenbank...



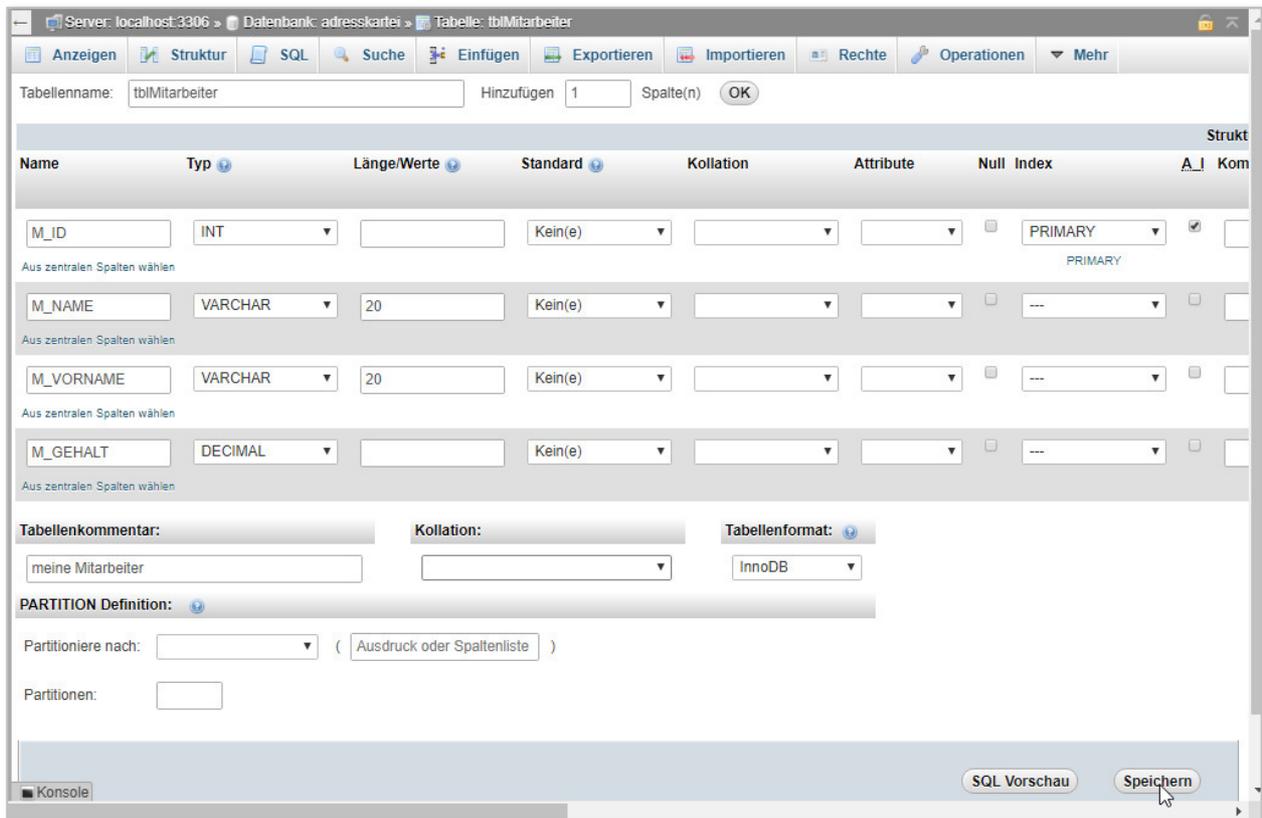
... ..



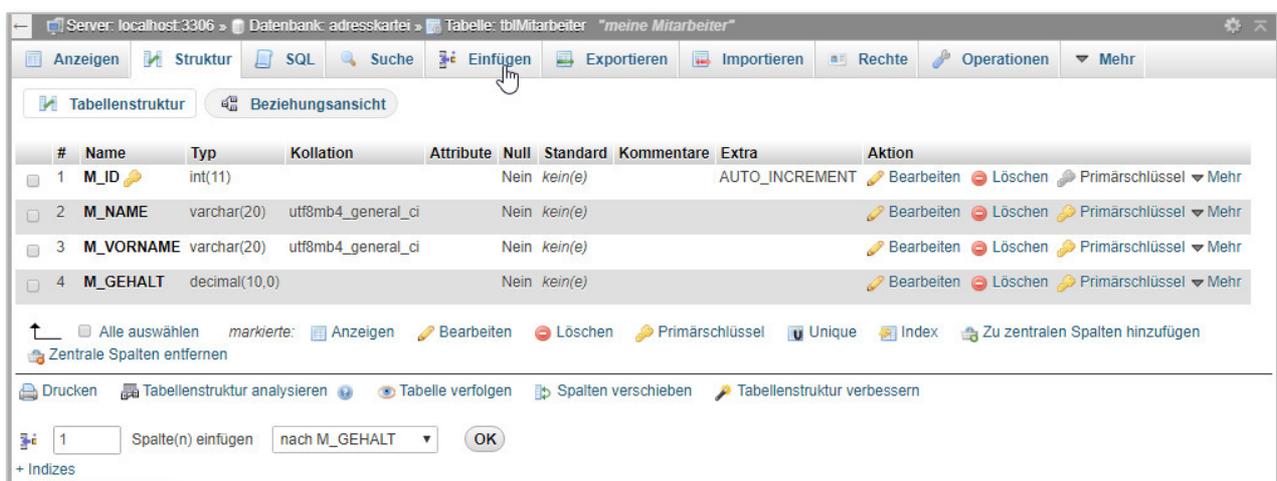
## Tabellen

Eine Datenbank besteht aus einer oder mehreren Tabellen, die mit dem Webfrontend angelegt und verwaltet werden können. Die Tabelle enthält mehrere Spalten (=Felder). Jedes Feld ist mit einem Datentyp und weiteren Einstellungen versehen.

Im folgenden Beispiel werden die Felder M\_ID, M\_NAME, M\_VORNAME und M\_GEHALT angelegt.



Wird nach dem Speichern die Struktur nochmals betrachtet, zeigt sich folgende Darstellung:



## Daten eingeben

Nach Erstellen der Tabellen können Daten erfasst werden.

The screenshot shows the PHP MyAdmin interface for the 'meine Mitarbeiter' table. The table structure is defined as follows:

Spalte	Typ	Funktion	Null	Wert
M_ID	int(11)			
M_NAME	varchar(20)			Pinguin
M_VORNAME	varchar(20)			Tux
M_GEHALT	decimal(10,0)			24380

Below the structure, there is an 'Ignorieren' checkbox and a second data entry form with the following values:

Spalte	Typ	Funktion	Null	Wert
M_ID	int(11)			
M_NAME	varchar(20)			Muster
M_VORNAME	varchar(20)			Lilly
M_GEHALT	decimal(10,0)			23501

## Daten abfragen

Die Suche nach bestehenden Daten erfolgt über die Eingabe von Abfragekriterien.

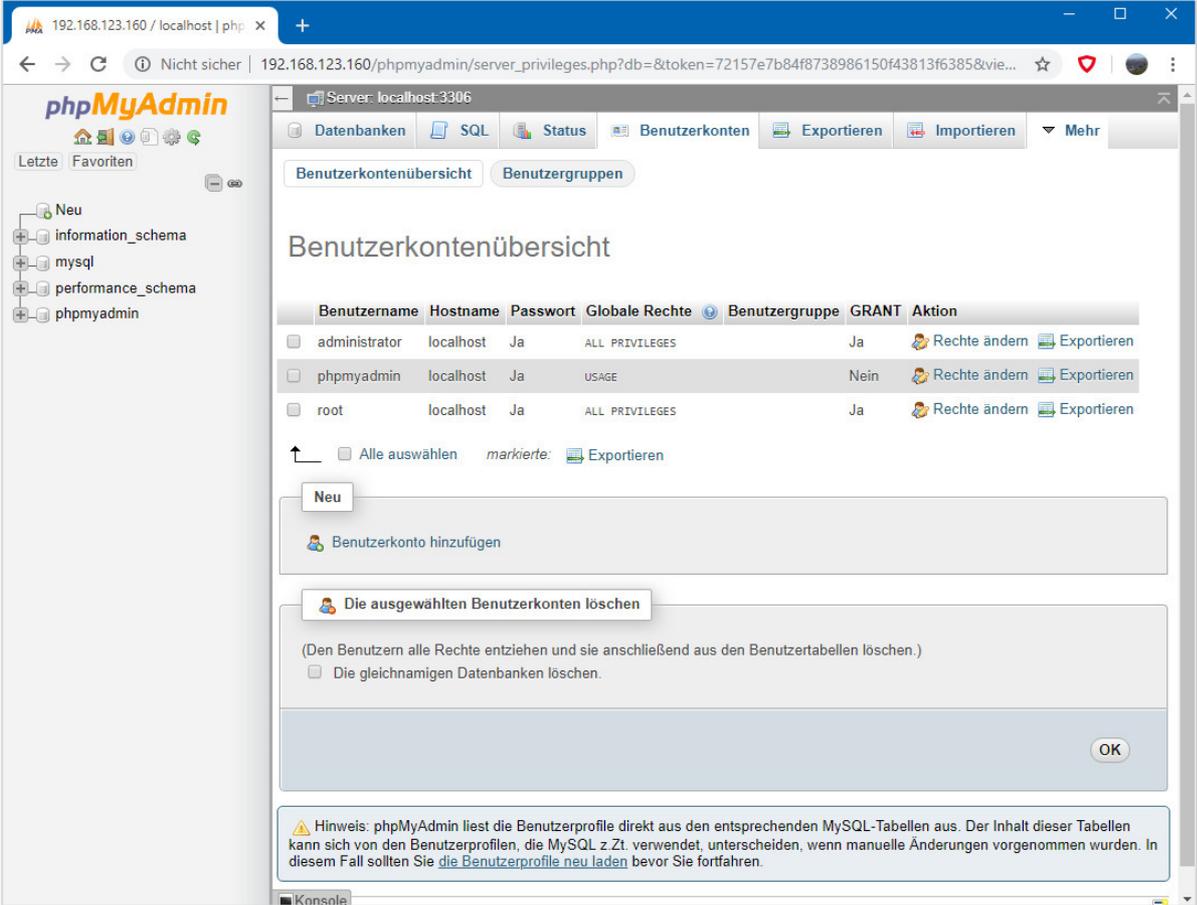
The screenshot shows the search interface in PHP MyAdmin. The search criteria are defined as follows:

Spalte	Typ	Kollation	Operator	Wert
M_ID	int(11)		=	
M_NAME	varchar(20)	utf8mb4_general_ci	LIKE	Pinguin%
M_VORNAME	varchar(20)	utf8mb4_general_ci	LIKE	
M_GEHALT	decimal(10,0)		=	

Below the search criteria, there is an '+ Optionen' link and an 'OK' button.

## Benutzerkonten

Der Zugriff auf die Entwicklung der Datenbank und die Datenerfassung wird über ein Benutzerkonzept geregelt. Standardbenutzer sind **root**, **phpmyadmin** und **administrator**.



The screenshot shows the phpMyAdmin interface for a MySQL server on localhost:3306. The 'Benutzerkonten' (User Accounts) section is active, displaying a table of user accounts. The table has columns for 'Benutzername', 'Hostname', 'Passwort', 'Globale Rechte', 'Benutzergruppe', 'GRANT', and 'Aktion'. Three users are listed: 'administrator', 'phpmyadmin', and 'root'. Below the table, there are options to 'Neu' (New) and 'Die ausgewählten Benutzerkonten löschen' (Delete selected user accounts). A warning message at the bottom states: 'Hinweis: phpMyAdmin liest die Benutzerprofile direkt aus den entsprechenden MySQL-Tabellen aus. Der Inhalt dieser Tabellen kann sich von den Benutzerprofilen, die MySQL z. Zt. verwendet, unterscheiden, wenn manuelle Änderungen vorgenommen wurden. In diesem Fall sollten Sie die Benutzerprofile neu laden bevor Sie fortfahren.'

Benutzername	Hostname	Passwort	Globale Rechte	Benutzergruppe	GRANT	Aktion
administrator	localhost	Ja	ALL PRIVILEGES		Ja	Rechte ändern Exportieren
phpmyadmin	localhost	Ja	USAGE		Nein	Rechte ändern Exportieren
root	localhost	Ja	ALL PRIVILEGES		Ja	Rechte ändern Exportieren

...

...