

Einrichtung eines Linux Ubuntu Servers

Erik Meyer, Robin Gerwe

Apache Webserver installieren

Zuerst die Pakete updaten.

```
sudo apt update
```

Dann kann das Apache-Paket installiert werden.

```
sudo apt install apache2
```

Prüfen um zu prüfen ob der Server nun funktioniert geben wir unsere IP-Adresse in den Browser ein. Diese können wir durch diesen Command bekommen:

```
hostname -I
```

Wenn alles geklappt hat sollte unter dieser IP-Adresse die Standardseitenvorlage von Apache angezeigt werden **1**.

Die Standardseite von Apache **index.html** ist in dem Verzeichnis **/var/www/html/** zu finden. Mit einem Editor wie **nano** kann die Apachestandardseite bearbeitet werden.

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

[Install Apache Webserver Ubuntu 20.04](#)

Informationsseite über das installierte System hinzufügen

Samba installieren

Zuerst alle Pakete updaten.

```
sudo apt update
```

Dann das Samba-Package installieren

```
sudo apt install samba
```


Samba readonly/readwrite Freigabe

Lokale Benutzer anlegen

[Samba Users](#)

Benutzer

```
sudo adduser benutzer
```



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in [/usr/share/doc/apache2/README.Debian.gz](#)**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```

/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.Load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf

```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Document Roots

By default, Ubuntu does not allow access through the web browser to *any* file apart of those located in `/var/www`, **public_html** directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Ubuntu document root is `/var/www/html`. You can make your own virtual hosts under `/var/www`. This is different to previous releases which provides better security out of the box.

Reporting Problems

Please use the `ubuntu-bug` tool to report bugs in the Apache2 package with Ubuntu. However, check **existing bug reports** before reporting a new bug.

Please report bugs specific to modules (such as PHP and others) to respective packages, not to the web server itself.

Abbildung 1: Die Standardseite von Apache

Fernzugriff

```
sudo adduser fernzugriff
```

```
sudo usermod -aG sudo fernzugriff
```

SSH-Dienst für Fernzugriff einrichten

1 2 (Linuxize, 2016)

Die Packages aktualisieren.

```
sudo apt update
```

```
sudo apt upgrade
```

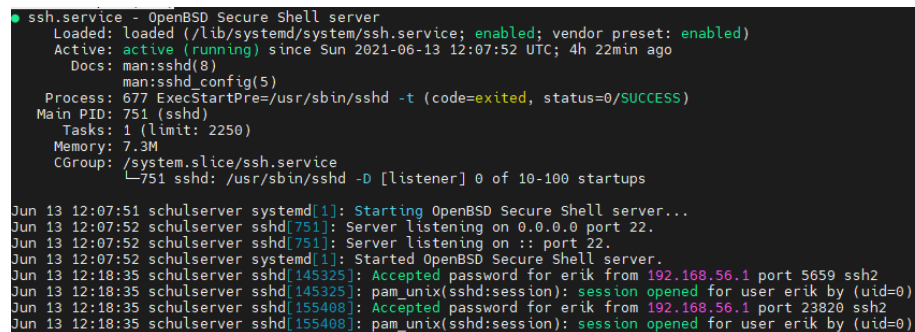
Dann installieren wir das SSH-Paket.

```
sudo apt install ssh
```

Mit `systemctl` können wir uns den Status anzeigen lassen.

```
sudo systemctl status ssh
```

Wenn alles geklappt hat sollte es ungefähr so aussehen 2:



```
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2021-06-13 12:07:52 UTC; 4h 22min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 677 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
    Main PID: 751 (sshd)
      Tasks: 1 (limit: 2250)
     Memory: 7.3M
    CGroup: /system.slice/ssh.service
            └─751 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Jun 13 12:07:51 schulserver systemd[1]: Starting OpenBSD Secure Shell server...
Jun 13 12:07:52 schulserver sshd[751]: Server listening on 0.0.0.0 port 22.
Jun 13 12:07:52 schulserver sshd[751]: Server listening on :: port 22.
Jun 13 12:07:52 schulserver systemd[1]: Started OpenBSD Secure Shell server.
Jun 13 12:18:35 schulserver sshd[145325]: Accepted password for erik from 192.168.56.1 port 5659 ssh2
Jun 13 12:18:35 schulserver sshd[145325]: pam_unix(sshd:session): session opened for user erik by (uid=0)
Jun 13 12:18:35 schulserver sshd[155408]: Accepted password for erik from 192.168.56.1 port 23820 ssh2
Jun 13 12:18:35 schulserver sshd[155408]: pam_unix(sshd:session): session opened for user erik by (uid=0)
```

Abbildung 2: Ausgabe von ‘sudo systemctl status ssh’

Netzwerkconfiguration mit einer statischen IP-Adresse im lokalen Subnetz

(Leseberg, 2020)

Server runterfahren

Netzwerk Adapter in Virtual Box einstellen

In VirtualBox bei dem server auf Ändern Klicken danach auf den Netzwerk Tab gehen.

1. Adapter 1 auf NAT einstellen, dieser ist dafür zuständig, dass der Server Internet hat.
2. Adapter 2 auf Host-only Adapter stellen. Dieser ist später für die statische Ip zuständig.

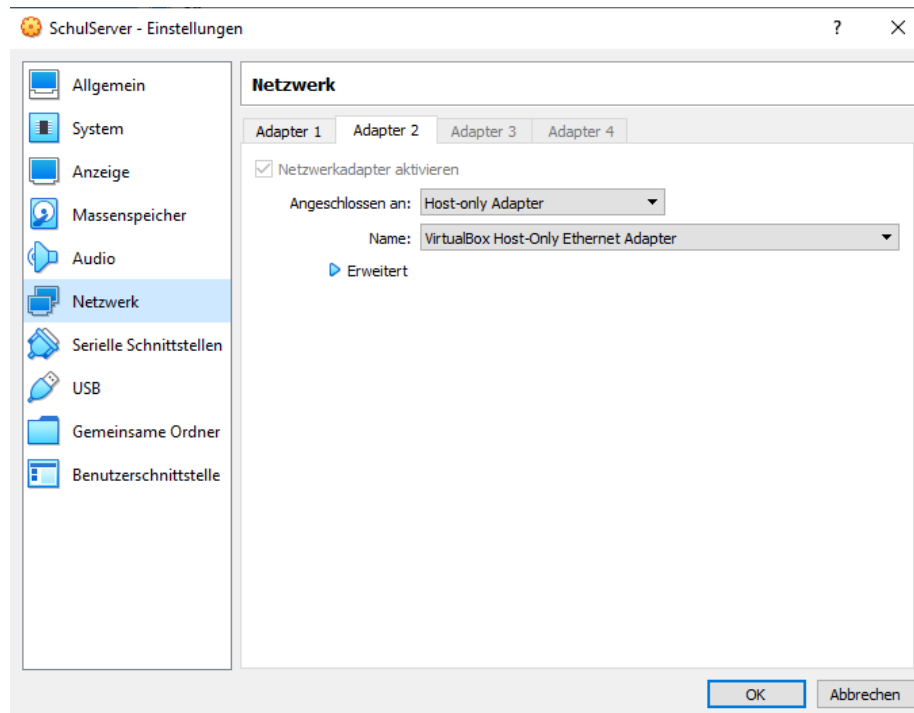


Abbildung 3: Adaptereinstellungen

Server Starten

Mit **ip a** testen, ob beide Adapter erfolgreich übernommen wurden.

In das Verzeichnis **cd /etc/netplan** wechseln.

Die .yaml Datei in diesem Ordner editieren **nano 00-installer-config.yaml**

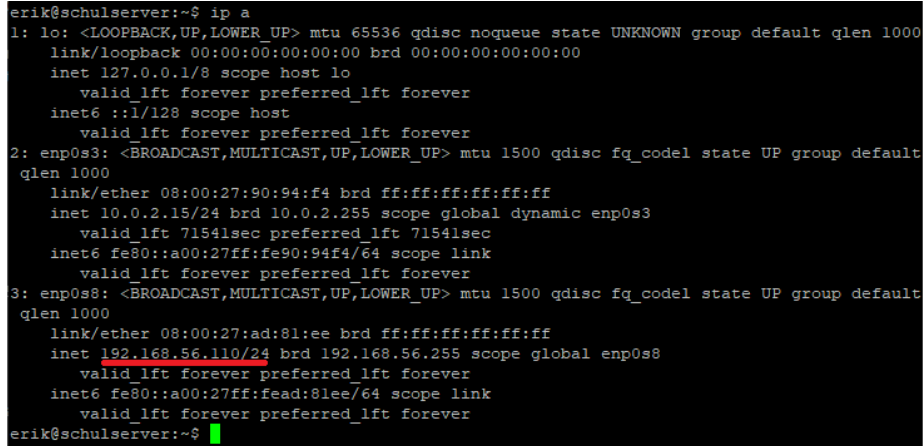
Den neuen Host-only Adapter in der Datei hinzufügen, diesem eine Ip Adresse zuweisen und speichern. (strg + s)

```
network:
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: yes
    enp0s8:
```

```
addresses: [192.168.56.110/24]
dhcp4: false
version: 2
```

Den Netplan mit **sudo netplan apply** anwenden

Mit **ip a** prüfen, ob die statische Ip übernommen wurde.



```
erik@schulserver:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    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 group default qlen 1000
    link/ether 08:00:27:90:94:f4 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 7154lsec preferred_lft 7154lsec
    inet6 fe80::a00:27ff:fe90:94f4/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ad:81:ee brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.110/24 brd 192.168.56.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fead:81ee/64 scope link
        valid_lft forever preferred_lft forever
erik@schulserver:~$
```

Abbildung 4: ip a output

Nextcloud als Filehosting-Lösung in der Private Cloud

(Richter, 2021)

In das root Verzeichnes der webserver wechseln **cd /var/www/html**

Die neueste Nextcloud Version herunterladen

wget https://download.nextcloud.com/server/releases/latest.tar.bz2

Die Datei entpacken **tar xfvj latest.tar.bz2** und mit **rm latest.tar.bz2** löschen

a2enmod rewrite

systemctl restart apache2

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

Mit einem Browser zu {server ip}/nextcloud verbinden. Ein admin account anlegen und bei der Datenbank sqlite auswählen.

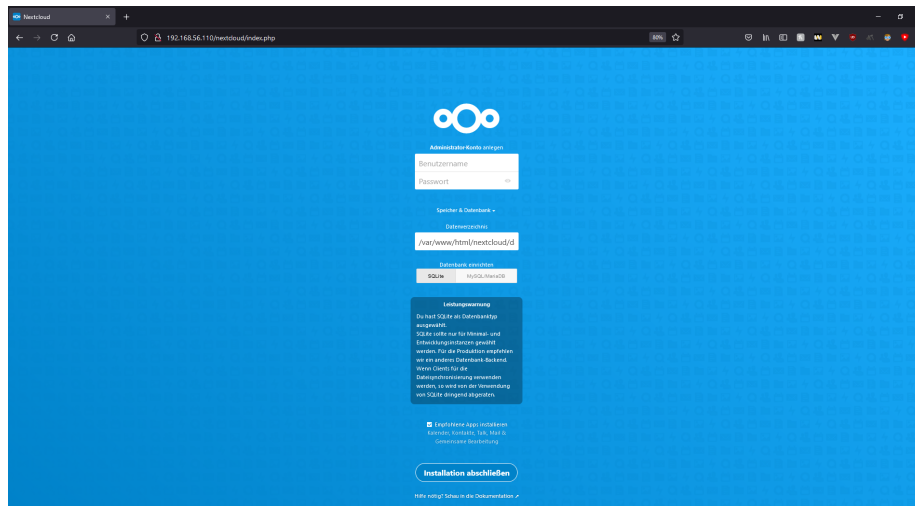


Abbildung 5: Nextcloud startseite

Anschließend müssen noch die in Nextcloud angezeigten Fehler behoben werden.

.htaccess aktivieren **nano /etc/apache2/sites-available/000-default.conf** und folgenden Text hinzufügen und speichern (strg + s)

```
<Directory /var/www/html>
    AllowOverride All
</Directory>
```

php script memory limit erhöhen

```
sudo nano /etc/php/7.4/apache2/php.ini
```

Variable memory_limit in der Datei suchen und auf 512 setzen

```
sudo service apache2 restart
```

Danach sollten alle Fehler behoben sein. Und Nextcloud kann genutzt werden.

Firewall-Regeln mit nftables

(Kessel, 2020)

```
sudo apt install nftables -y
```

```
sudo nano /etc/nftables.conf
```

Den Inhalt mit folgendem Text ersetzen und speichern (strg + s)

```
#!/usr/sbin/nft -f
# vim: ft=nft
```

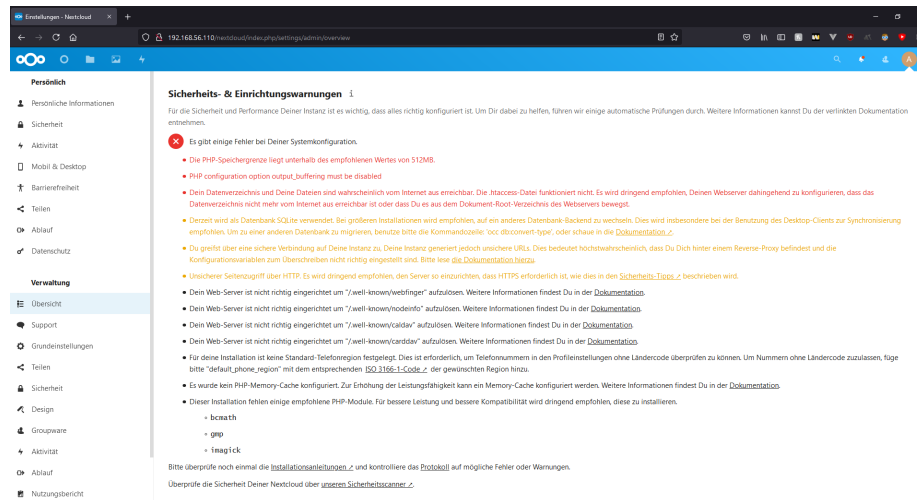


Abbildung 6: Nextcloud Fehler

```
flush ruleset
```

```
define tcp_services = { ssh, 80, 443 }
```

```
table inet filter {
    chain input {
        type filter hook input priority 0;
        policy drop;

        iif lo accept;

        ct state established, related accept;
        ct state invalid drop;

        # accept incoming connections on these ports
        tcp dport $tcp_services accept;

        ip6 nexthdr icmpv6 icmpv6 type {
            nd-neighbor-solicit, nd-router-advert, nd-neighbor-advert
        } accept

    chain forward {
        type filter hook forward priority 0;

        policy drop;
```



```

    }

    chain output {
        type filter hook output priority 0;
        policy accept;
    }
}

```

Config mit folgenden Befehl laden

```
sudo systemctl restart nftables && systemctl status nftables && nft list ruleset
```

Jetzt kann man den Server nur noch auf dem SSH Port (22) und Webserver Port (80,443) erreichen.

Quellenverzeichnis

1 2

Kessel, J. van, 2020. *Setting up nftables Firewall* — *Cryptsus Blog*. [online] Verfügbar unter: <<https://cryptsus.com/blog/setting-up-nftables-firewall.html>> [Zugegriffen 1 Juni 2020].

Leseberg, E., 2020. *VirtualBox with Static IP on Ubuntu Server Guest*. [online] Verfügbar unter: <<https://ernieleseberg.com/virtualbox-with-static-ip-on-ubuntu-server-guest/>> [Zugegriffen 11 Apr. 2020].

Linuxize, 2016. *How to Enable SSH on Ubuntu 20.04* / *Linuxize*. [online] Verfügbar unter: <<https://linuxize.com/post/how-to-enable-ssh-on-ubuntu-20-04/>>.

Richter, B., 2021. *Anleitung: Nextcloud unter Linux installieren* - *Bennet Richter*. [online] Verfügbar unter: <<https://www.bennetrichter.de/anleitungen/nextcloud-linux/>> [Zugegriffen 2 Apr. 2021].