

Installation Ubuntu 22.04 Server

Installationsmedium

<https://releases.ubuntu.com/22.04/>

Während der Installation setze ich

- die Locale auf de_de
- das Keyboard-Layout auf „German QWERTZ“
- Hostname

LVM einrichten



Handwerkszeug installieren

aptitude

```
# apt-get install aptitude
```

VIMnox

```
# aptitude install vim-nox
```

Midnight Commander

```
# aptitude install mc
```

Net-Tools (ifconfig, etc.)

```
# aptitude install net-tools
```

Timezone

Aktuell eingestellte Zeitzone:

```
# timedatectl
      Local time: Sun 2020-10-11 11:00:01 UTC
      Universal time: Sun 2020-10-11 11:00:01 UTC
      RTC time: Sun 2020-10-11 11:00:02
      Time zone: Etc/UTC (UTC, +0000)
System clock synchronized: yes
      NTP service: active
      RTC in local TZ: no
```

Zeitzone umstellen:

```
# timedatectl list-timezones|grep Berlin
Europe/Berlin
# timedatectl set-timezone Europe/Berlin
# timedatectl
      Local time: Sun 2020-10-11 13:02:31 CEST
      Universal time: Sun 2020-10-11 11:02:31 UTC
      RTC time: Sun 2020-10-11 11:02:32
      Time zone: Europe/Berlin (CEST, +0200)
System clock synchronized: yes
      NTP service: active
      RTC in local TZ: no
```

NTP Client

</etc/systemd/timesyncd.conf>

```
# This file is part of systemd.
```

```
#
# systemd is free software; you can redistribute it and/or modify it
# under the terms of the GNU Lesser General Public License as
# published by
# the Free Software Foundation; either version 2.1 of the License, or
# (at your option) any later version.
#
# Entries in this file show the compile time defaults.
# You can change settings by editing this file.
# Defaults can be restored by simply deleting this file.
#
# See timesyncd.conf(5) for details.

[Time]
NTP=ptbtime1.ptb.de
FallbackNTP=ptbtime3.ptb.de ptbtime2.ptb.de
```

Momentane Systemzeit ansehen:

```
timedatectl
```

```
          Local time: So 2018-11-25 11:26:59 CET
          Universal time: So 2018-11-25 10:26:59 UTC
          RTC time: So 2018-11-25 10:27:00
          Time zone: Europe/Berlin (CET, +0100)
System clock synchronized: yes
systemd-timesyncd.service active: yes
          RTC in local TZ: no
```

```
systemctl restart systemd-timesyncd
systemctl status systemd-timesyncd
● systemd-timesyncd.service - Network Time Synchronization
   Loaded: loaded (/lib/systemd/system/systemd-timesyncd.service; enabled;
   vendor preset: enabled)
   Active: active (running) since Sun 2018-11-25 11:29:00 CET; 1s ago
     Docs: man:systemd-timesyncd.service(8)
  Main PID: 16475 (systemd-timesyn)
   Status: "Synchronized to time server 192.53.103.108:123
(ptbtime1.ptb.de)."
```

```
Tasks: 2 (limit: 2319)
  CGroup: /system.slice/systemd-timesyncd.service
          └─16475 /lib/systemd/systemd-timesyncd
```

```
Nov 25 11:29:00 backup systemd[1]: Starting Network Time Synchronization...
Nov 25 11:29:00 backup systemd[1]: Started Network Time Synchronization.
Nov 25 11:29:01 backup systemd-timesyncd[16475]: Synchronized to time server
192.53.103.108:123 (ptbtime1.ptb.de).
```

Reaktivierung von ifupdown

Um netplan.io zu deaktivieren, muss lediglich das Paket ifupdown installiert werden. **Die Deinstallation von netplan.io ist nicht empfehlenswert**, insbesondere dann nicht, wenn die Deaktivierung via SSH vorgenommen wird. Nach der Deinstallation ist ein Zugriff via IP nicht mehr möglich. Es muss auf die Konsole ausgewichen werden!

```
aptitude install ifupdown
```

Im Bootloader muss ebenfalls das Laden von netplan unterdrückt werden:

[/etc/default/grub](#)

```
[...]  
GRUB_CMDLINE_LINUX="netcfg/do_not_use_netplan=true"
```

```
update-grub
```

Um das klassische Verhalten von ifupdown wiederherzustellen, muss ebenfalls systemd-networkd ausgeschaltet werden. Dies geschieht folgendermaßen:

```
systemctl disable systemd-networkd.service  
systemctl mask systemd-networkd.service  
systemctl stop systemd-networkd.service
```

Die Netzwerkkonfiguration sollte nun komplett aus der interfaces-Datei übernommen werden. Eine Ausnahme stellen die DNS-Server dar. Damit diese ebenfalls aus interfaces übernommen werden, muss systemd-resolved ausgeschaltet und resolvconf aktiviert werden!

```
aptitude install resolvconf
```

```
systemctl disable systemd-resolved.service  
systemctl stop systemd-resolved.service  
systemctl mask systemd-resolved.service
```

```
systemctl disable systemd-networkd-wait-online.service  
systemctl stop systemd-networkd-wait-online.service  
systemctl mask systemd-networkd-wait-online.service
```

Jetzt die `/etc/network/interfaces` final editieren, sonst kein Zugriff mehr!

Netzwerkkonfiguration

Beispiel:

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface

auto eth0
iface eth0 inet6 static
    address 1b2c:3d4e:0:0:0:0:0:123
    netmask 64
    dns-nameservers 2620:fe::fe 2606:4700:4700::1111
    pre-up echo 0 > /proc/sys/net/ipv6/conf/eth0/autoconf
    pre-up echo 0 > /proc/sys/net/ipv6/conf/eth0/accept_ra
    post-up /sbin/ip -6 route add default via 1b2c:3d4e:0:0:0:0:0:1

iface eth0 inet static
    address 174.255.120.12
    netmask 255.255.255.0
    network 174.255.120.0
    broadcast 174.255.120.255
    gateway 174.255.120.1
    dns-nameservers 9.9.9.9 1.1.1.1

auto eth0:smtp
iface eth0:smtp inet6 static
    address 1b2c:3d4e:0:0:0:0:0:124
    netmask 64

iface eth0:smtp inet static
    address 174.255.120.110
    netmask 255.255.255.0
    broadcast 174.255.120.255
```

```
reboot
```

Firewall

Installation

Die Pakete „iptables-persistent“ und „netfilter-persistent“ stehen in direkter Abhängigkeit und müssen daher beide installiert werden.

```
apt-get update
aptitude install iptables-persistent netfilter-persistent
```

Ubuntu kommt von Hause aus mit dem Paket ufw, ebenfalls eine auf iptables-basierende Firewall. Den Job übernimmt nun netfilter-persistent, daher deinstalliere ich es:

```
aptitude purge ufw
```

Konfiguration / Regelwerk

Um ein Regelwerk zu kreieren, empfehle ich, ein Bash-Skript mit iptables-Befehlen zu schreiben. Sobald dieses ausgeführt worden ist, muss das Regelwerk abgespeichert werden. Dies geschieht mit folgendem Befehl:

```
netfilter-persistent save
```

Netfilter erstellt nun unter `/etc/iptables` zwei Dateien, `rules.v4` und `rules.v6`. Die Dateien `add-blocked.ips` sowie `blocked.ips` stammen von einem eigenen Erweiterungskript, mit dem sich IP-Adressen einfach einer Sperrliste hinzufügen lassen. Darauf werde ich hier nicht weiter eingehen.

```
ll /etc/iptables/  
insgesamt 24  
drwxr-xr-x  2 root root 4096 Feb  7 23:47 ./  
drwxr-xr-x 99 root root 4096 Feb  7 23:18 ../  
-rwxr-xr-x  1 root root  742 Feb  7 23:43 add-blocked.ips*  
-rw-r--r--  1 root root    0 Feb  7 23:18 blocked.ips  
-rw-r----- 1 root root 4189 Feb  7 23:46 rules.v4  
-rw-r----- 1 root root  183 Feb  7 23:46 rules.v6
```

Die Firewall sollte nun bereits einsatzfähig sein.

Logfile

Dummerweise schreibt iptables das syslog voll, welches somit unübersichtlich wird. Mit Hilfe des `rsyslogd` leite ich die Ausgaben in eine eigene Datei um:

```
vi /etc/rsyslog.d/25-iptables.conf
```

Damit dieser Weg funktioniert, habe ich mittels des Parameters `-log-prefix` von iptables der Ausgabe das Präfix „IPT:“ hinzugefügt. Das können wir uns als Filter zur Nutze machen.

[/etc/rsyslog.d/25-iptables.conf](#)

```
:msg,contains,"IPT:" -/var/log/iptables.log  
& ~
```

Beim ersten Mal muss die Datei erstellt werden und mit Rechten für den `rsyslogd` versehen werden.

```
touch /var/log/iptables.log  
chown syslog.adm /var/log/iptables.log
```

Die Änderungen werden erst nach einem Dienstneustart übernommen.

```
service rsyslog restart
```

Das Logfile wird schnell groß und sollter daher rotiert werden:

[/etc/logrotate.d/iptables](#)

```
/var/log/iptables.log
{
    rotate 7
    daily
    missingok
    notifempty
    delaycompress
    compress
    create 640 syslog adm
    sharedscripts
}
```

From:

<https://wikinet.webby.hetzel-netz.de/> - **Sebastians IT-Wiki**

Permanent link:

https://wikinet.webby.hetzel-netz.de/ubuntu:22-04_server_install?rev=1667670665

Last update: **2022/11/05 18:51**

