

Installation Ubuntu 24.04 Server

Installationsmedium

<https://releases.ubuntu.com/24.04/>

Während der Installation setze ich

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

LVM einrichten

Klassische Aufteilung (Beispiel):

```
/dev/sda1  EFI      /boot/efi  ext4
/dev/sda2  /boot      ext4
/dev/sda3  LVM PV
├─ vg0
│  ├─ lv-swap  swap
│  ├─ lv-root  /
│  ├─ lv-home  /home
│  ├─ lv-var   /var
│  ├─ lv-srv   /srv
│  └─ lv-data  /data
```

```
Storage configuration [ Help ]

FILE SYSTEM SUMMARY

MOUNT POINT    SIZE    TYPE    DEVICE TYPE
[ /             10.000G new ext4 new LVM logical volume ▶ ]
[ /boot         1.000G  new ext4 new partition of local disk ▶ ]
[ /data         52.996G new ext4 new LVM logical volume ▶ ]
[ /srv          20.000G new ext4 new LVM logical volume ▶ ]
[ /var          10.000G new ext4 new LVM logical volume ▶ ]
[ SWAP          6.000G  new swap  new LVM logical volume ▶ ]

AVAILABLE DEVICES

No available devices

[ Create software RAID (md) ▶ ]
[ Create volume group (LVM) ▶ ]

USED DEVICES

DEVICE          TYPE          SIZE
[ vg0 (new)      LVM volume group 98.996G ▶ ]
swap            new, to be formatted as swap 6.000G ▶ ]
root            new, to be formatted as ext4, mounted at / 10.000G ▶ ]
var             new, to be formatted as ext4, mounted at /var 10.000G ▶ ]
srv             new, to be formatted as ext4, mounted at /srv 20.000G ▶ ]
data            new, to be formatted as ext4, mounted at /data 52.996G ▶ ]

[ /dev/xvda      local disk 100.000G ▶ ]
partition 1     new, BIOS grub spacer 1.000M ▶ ]
partition 2     new, to be formatted as ext4, mounted at /boot 1.000G ▶ ]
partition 3     new, PV of LVM volume group vg0 98.997G ▶ ]

[ Done ]
[ Reset ]
[ Back ]
```

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 muss sorgsam gemacht werden**, insbesondere dann, wenn die Deaktivierung via SSH vorgenommen wird. Nach der Deinstallation ist ein Zugriff via IP ggfs. nicht mehr möglich. Es muss auf die Konsole ausgewichen werden!

```
aptitude install ifupdown
```

Netplan-Konfig stilllegen und sichern:

```
mkdir -p /etc/netplan/disabled
mv /etc/netplan/*.yaml /etc/netplan/disabled/
```

Netplan entfernen:

```
apt purge netplan.io
```

Im Bootloader muss / sollte ebenfalls das Laden von netplan unterdrückt werden. Das ist insbesondere wichtig, wenn das Paket auf dem System verbleibt und nicht deinstalliert wird!

[/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. Diese muss nun erstellt werden. Anbei eine Beispiel-Datei.

Der interface-Name wird über den nachfolgenden Befehl ermittelt.

```
ip link

1: lo: <LOOPBACK,UP,LOWER_UP> ...
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> ...
3: enp0s25: <BROADCAST,MULTICAST> ...
```

```
# 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
```

Namensauflösung

Damit diese ebenfalls aus interfaces übernommen werden kann, muss systemd-resolved weiterbetrieben werden. Alternativ kann die resolv.conf manuell erstellt werden!

Variante ohne systemd-resolved:

```
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
```

```
reboot
```

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:0:0:0:0:1

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
}
```

"Mini" Postfix

Der MTA Postfix soll nur dazu dienen Mails zu versenden. So können Informationen, zum Beispiel an den Admin, vom System versendet werden oder Webseiten können mit Ihren Benutzern kommunizieren, wenn beispielsweise ein Passwort zurückgesetzt werden soll.

Zunächst sind die benötigten Pakete zu installieren.

```
aptitude install postfix
```

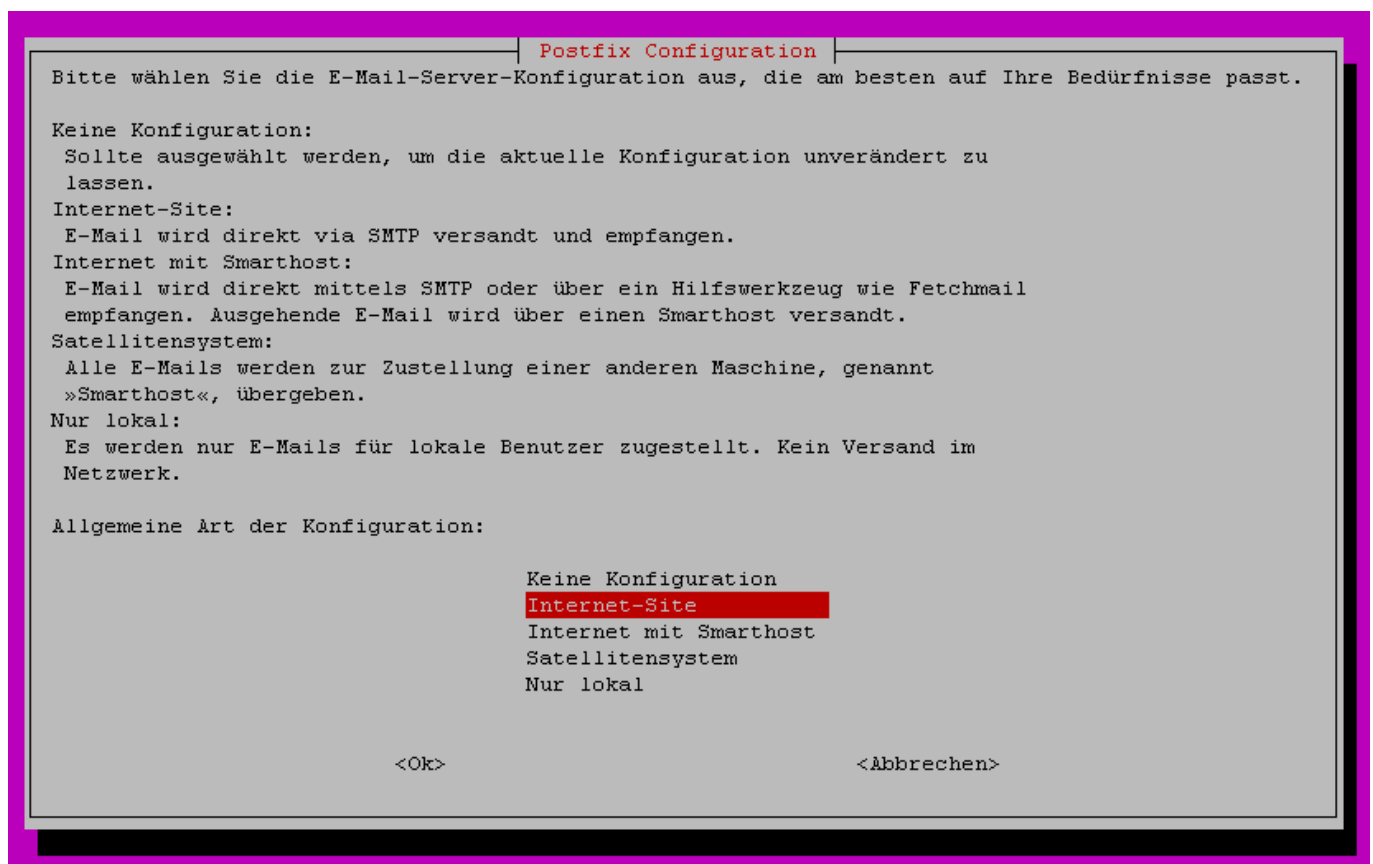
Die folgenden NEUEN Pakete werden zusätzlich installiert:

```
postfix ssl-cert{a}
```

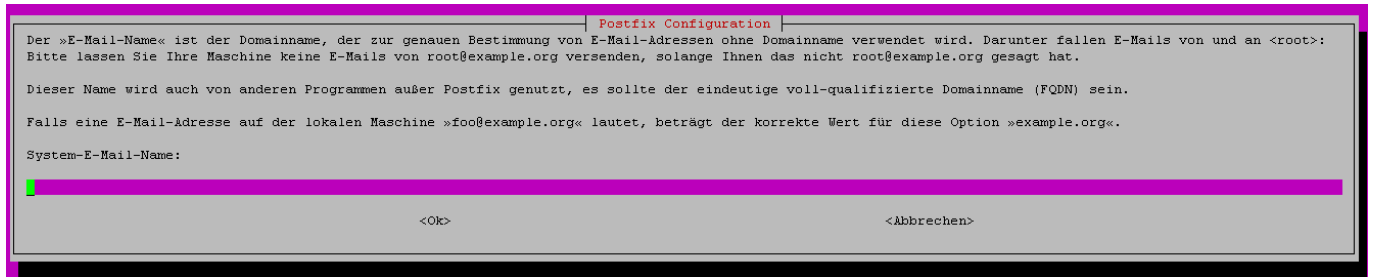
0 Pakete aktualisiert, 2 zusätzlich installiert, 0 werden entfernt und 8 nicht aktualisiert.

1.164 kB an Archiven müssen heruntergeladen werden. Nach dem Entpacken werden 4.141 kB zusätzlich belegt sein.

Möchten Sie fortsetzen? [Y/n/?]



Hier die Default-Maildomäne eintragen:



Folgende Konfigurationsparameter anpassen:

[/etc/postfix/main.cf](#)

```
smtp_generic_maps = hash:/etc/postfix/generic
mydestination = $myhostname, myhostname.mydomain.de, localhost
inet_interfaces = loopback-only
inet_protocols = ipv4
relayhost = [smtp.myprovider.de]
```

[/etc/postfix/generic](#)

```
root@myhostname.mydomain.de      something@mydomain.de
@myhostname.mydomain.de         @mydomain.de
```

[/etc/aliases](#)

```
# See man 5 aliases for format
postmaster:    root
root:          something@mydomain.de
```

Die Konfigurationen anwenden:

```
postmap hash:/etc/postfix/generic
newaliases
service postfix restart
```

Apticron

Installation

```
apt-get update
aptitude install apticron
vi /etc/apticron/apticron.conf
```

Konfiguration

Gegebenenfalls sollte hier die Empfängeradresse angepasst werden:

</etc/apticron/apticron.conf>

```
# apticron.conf
#
# The values set in /etc/apticron/apticron.conf will override the
# settings
# in this file.

#
# Set EMAIL to a space separated list of addresses which will be
# notified of
# impending updates. By default the root account will be notified.
#
EMAIL="root"
[...]
```

Scheduled Task

Wann Apticron ausgeführt wird, kann über Cron angepasst werden:

```
vi /etc/cron.d/apticron
```

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

Permanent link:
https://wikinet.webby.hetzel-netz.de/ubuntu:24-04_server_install?rev=1770201456

Last update: **2026/02/04 11:37**

