

# Netzwerk

LAN, WLAN, Routing, etc.

- [Chrony installieren und konfigurieren](#)
- [iproute2 \(ip\) Befehle](#)
- [Name einer Netzwerkkomponente ändern](#)
- [NetworkManager - IPv6 deaktivieren mit nmcli](#)
- [WLAN Konfiguration mit wpa\\_supplicant \(veraltet\)](#)
- [Linux - Dienste erst nach Zeitsynchronisation starten](#)

# Chrony installieren und konfigurieren

## Chrony installieren

Chrony ist eine Implementierung des Network Time Protokolls NTP und damit eine Alternative zu **ntpd** und **systemd-timesyncd**.

```
apt install chrony
```

ntp, ntpsec oder systemd-timesyncd werden durch die Installation von Chrony entfernt.

## Chrony konfigurieren

Chrony verwendet bei Debian / RaspiOS standardmäßig den NTP-Pool von ntp.org.

Wenn ein eigener Zeitserver verwendet werden soll muss die Zeile in der `/etc/chrony/chrony.conf` auskommentiert werden:

```
# pool 2.debian.pool.ntp.org iburst
```

Dann eine Datei im Verzeichnis **/etc/chrony/sources.d** erstellen:

```
echo 'server <IP NTP-SERVER> iburst' > /etc/chrony/sources.d/local-ntp-server.sources
```

Chrony neu starten:

```
systemctl restart chrony
```

## Chrony Befehle

Hilfe anzeigen:

```
chronyc help
```

Informationen zur Systemzeit:

chronyc tracking

Verwendete NTP-Server:

chronyc sources

Aktivität anzeigen:

chronyc activity

# iproute2 (ip) Befehle

Die wichtigsten iproute2 (ip) Befehle

## Show informations

|                          |                                                               |                         |
|--------------------------|---------------------------------------------------------------|-------------------------|
| Show link status         | <b><i>ip link show</i></b>                                    | <b><i>ip l</i></b>      |
| Show link status & stats | <b><i>ip -statistics link show</i></b>                        | <b><i>ip -s l</i></b>   |
| Show IP address          | <b><i>ip addr show</i></b>                                    | <b><i>ip a</i></b>      |
| Show ARP cache           | <b><i>ip neigh show</i></b>                                   | <b><i>ip n</i></b>      |
| Show connections         | <b><i>ss --tcp --all --processes --extended --numeric</i></b> | <b><i>ss -tapen</i></b> |
| Show routing table       | <b><i>ip route show</i></b>                                   | <b><i>ip r</i></b>      |

## Set interface

|                 |                                              |                                            |
|-----------------|----------------------------------------------|--------------------------------------------|
| Set IP address  | <b><i>ip addr add IP/NETMASK dev DEV</i></b> | <b><i>ip a a IP/NETMASK dev DEVICE</i></b> |
| Set device up   | <b><i>ip link set DEV up</i></b>             |                                            |
| Set device down | <b><i>ip link set DEV down</i></b>           |                                            |

## Add VLANs

|                  |                                                                  |
|------------------|------------------------------------------------------------------|
| Add VLAN to link | <b><i>ip link add link DEV name DEV.VID type vlan id VID</i></b> |
|------------------|------------------------------------------------------------------|

## Delete device addresses

|                   |                                              |                                            |
|-------------------|----------------------------------------------|--------------------------------------------|
| Delete IP address | <b><i>ip addr del IP/NETMASK dev DEV</i></b> | <b><i>ip a d IP/NETMASK dev DEVICE</i></b> |
|-------------------|----------------------------------------------|--------------------------------------------|

|                |                                     |                                 |
|----------------|-------------------------------------|---------------------------------|
| Delete all IPs | <i><b>ip addr flush dev DEV</b></i> | <i><b>ip a f dev DEVICE</b></i> |
|----------------|-------------------------------------|---------------------------------|

## Set gateway

|                      |                                           |                                     |
|----------------------|-------------------------------------------|-------------------------------------|
| Set standard gateway | <i><b>ip route add default via IP</b></i> | <i><b>ip r a default via IP</b></i> |
|----------------------|-------------------------------------------|-------------------------------------|

# Name einer Netzwerkkomponente ändern

Die Bezeichnung einer Netzwerkkomponente mit Hilfe eines **systemd.link** Files unveränderlich festlegen.

## Bezeichnung und MAC Adresse der Netzwerkkomponenten auslesen

```
ip a
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    .
    .
    .
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 8c:6a:00:ad:bd:f0 brd ff:ff:ff:ff:ff:ff
```

Da die MAC Adresse des Netzwerkgeräts immer gleich bleibt macht es Sinn diese als Match für den Link zu verwenden.

## Bezeichnung des Netzwerkgeräts ändern

In diesem Beispiel wird die Bezeichnung von **eth0** auf **ethlan0** geändert.

Im Verzeichnis **/etc/systemd/network** eine Datei mit folgendem Inhalt anlegen:

```
nano /etc/systemd/network/10-ethlan0.link
```

```
[Match]
MACAddress=8c:6a:00:ad:bd:f0
```

[Link]

Name=ethlan0

Danach müssen noch die Bezeichnungen in der ***/etc/network/interfaces*** geändert werden.

Die Änderungen werden erst nach einem Neustart übernommen!

# NetworkManager - IPv6 deaktivieren mit nmcli

## IPv6 Deaktivieren

Anzeigen der Netzwerkverbindungen:

```
nmcli connection show
```

IPv6 mit **modify** deaktivieren:

```
nmcli connection modify "CONNECTIONNAME" ipv6.method "disabled"
```

Aktivieren der Einstellung:

```
nmcli connection up "CONNECTIONNAME"
```

# WLAN Konfiguration mit wpa\_supplicant (veraltet)

## Beispiel einer wpa\_supplicant Datei

```
ctrl_interface=DIR=/var/run/wpa_supplicant  
update_config=1  
country=AT
```

```
network={  
    ssid="mywifi"  
    psk=fda1a0139acf704ca9dc8c8d40d301aceb13202fea922373e43c505a3a92705c  
    scan_ssid=1  
}
```

# Linux - Dienste erst nach Zeitsynchronisation starten

Wenn es notwendig ist, einen Dienst nach einem Neustart erst nach der Zeitsynchronisation zu starten, kann das mit dem SNTP (Simple Network Time Protocol) Client von systemd erreicht werden.

Falls ein anderer NTP-Server installiert ist diesen löschen

```
apt purge chrony ntp
```

Den NTP-Server von systemd aktivieren:

```
systemctl enable systemd-timesyncd
```

Den Dienst zum verzögerten Starten anderer Dienste aktivieren:

```
systemctl enable systemd-time-wait-sync
```

In der zu verzögernden Service-Unit folgendes eintragen:

```
systemctl edit foo.service
```

```
[Unit]
After=time-sync.target
Wants=time-sync.target
```