





# Git Server

Diese Anleitung beschreibt wie man ein Git-Server auf einem Ubuntu (Debian) Linux System einrichtet und installiert. Alle Kommandos oder Befehle werden im Terminal eingegeben und es wird root Rechte vorausgesetzt.

## Bestehende IP Adresse abfragen

```
# Netzwerkkarten anzeigen (Ubuntu)
ifconfig -a
```

Ausgabe Beispiel Ubuntu

```
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 0 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

venet0: flags=211<UP,BROADCAST,POINTOPOINT,RUNNING,NOARP> mtu 1500
    inet 127.0.0.2 netmask 255.255.255.255 broadcast 0.0.0.0
    destination 127.0.0.2
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 0
    (UNSPEC)
    RX packets 3591649 bytes 4930223690 (4.5 GiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1924771 bytes 166284908 (158.5 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

venet0:0: flags=211<UP,BROADCAST,POINTOPOINT,RUNNING,NOARP> mtu 1500
    inet 185.245.96.84 netmask 255.255.255.255 broadcast 185.245.96.84
    destination 185.245.96.84
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 0
    (UNSPEC)
```

```
# Netzwerkkarten anzeigen (Debian)
ip add
```

Ausgabe Beispiel Debian

```
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: enp2s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq master
vibr0 state UP group default qlen 1000
    link/ether b8:83:03:54:8d:54 brd ff:ff:ff:ff:ff:ff
3: enp2s0f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
group default qlen 1000
    link/ether b8:83:03:54:8d:55 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::ba83:3ff:fe54:8d55/64 scope link
        valid_lft forever preferred_lft forever
4: vibr0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
group default qlen 1000
    link/ether b8:83:03:54:8d:54 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.167/24 brd 192.168.1.255 scope global vibr0
        valid_lft forever preferred_lft forever
    inet6 fe80::ba83:3ff:fe54:8d54/64 scope link
        valid_lft forever preferred_lft forever
```

## Static IP Adresse einrichten

Ein statische IP Adresse lässt sich auf verschiedene Weise einrichten.

1. Festlegen der IP Adresse auf dem DHCP Server mittels MAC Adresse. [Beispiel](#)
2. Einstellen der Adresse direkt auf der Netzwerkkarte. [Beispiel](#)

```
# Einstellen der IP Adresse über die Interface Konfigurationsdatei.
# Zu finden unter /etc/network
cd /etc/network
# Datei editieren, nano ist der texteditor
nano interfaces
# In diese Datei lässt sich nun der Netzwerkkarte eine IP Adresse zuordnen.

# Hier ein Beispiel, Ip adresse wird vom DHCP gesetzt
auto enp0s3
allow-hotplug enp0s3
iface enp0s3 inet dhcp

# Ändern wir nun als statische Adresse ein.
auto enp0s3
allow-hotplug enp0s3
iface enp0s3 inet static
    address 192.168.250.99
    netmask 255.255.255.0
    network 192.168.1.1
    broadcast 192.168.255.255
    gateway 192.168.1.1
```

```
# Die Netzwerkkarte erhält die IP 192.168.250.99

# Jetzt muss das die Konfiguration durch Neustarten des Netzwerkdienstes
eingeladen werden.
systemctl restart networking
```

Eine Anleitung für die Netzwerkeinstellungen unter Debian 10.

## Installieren unter Debian

Die Einrichtungsanleitung wurde aus der Originalanleitung von [git-scm.com](https://git-scm.com) entnommen und auf Debian angepasst.

### User 'git' hinzufügen

Bevor wir mit der Installation beginnen können **müssen noch die User**, die auf den Git-Server Zugriff haben sollen, **eingrichtet werden**.

```
# User für Git-Server einrichten. Über SSH

# Der neue user heisst 'git' ink Passwort
adduser git

# Die 'git' Identität annehmen, git Userpasswort muss eingegeben werden.
su git

# In User Verzeichnis wechseln
cd

# Ausgeben in welchem Ordner man sich befindet.
# Sollte /home/git anzeigen.
pwd

# Ordner für die Autorisierungs-Keys erstellen.
# In diesen werden die Public-Keys der User abgespeichert, die Zugriff auf
den Git-Server haben sollen.
mkdir .ssh

# Ordner Zugriffsberechtigungen setzen
chmod 700 .ssh

# Datei erstellen in der die Keys gespeichert werden.
touch .ssh/authorized_keys

# Rechte der Datei so setzen, dass nur der Admin (root) diese ändern kann
chmod 600 .ssh/authorized_keys
```



Im Ordner `.ssh` der sich nun unter `/home/git` befindet, befindet sich die Datei `authorized_keys`. In diese können dann beliebig viele Public-Keys anderer User angefügt werden. Auf diesem Weg können sich auch andere Benutzer am Git-Server anmelden. Vorausgesetzt diese kennen das Login Passwort des Users 'git'. Folgendes Beispiel ergänzt die Public-Keys der Benutzer john, josie und jessica in der Datei `authorized_keys`. (cat Befehl)

```
cat /tmp/id_rsa.john.pub >> ~/.ssh/authorized_keys
cat /tmp/id_rsa.josie.pub >> ~/.ssh/authorized_keys
cat /tmp/id_rsa.jessica.pub >> ~/.ssh/authorized_keys
```

## Server Zugriff ohne Login

From:  
<https://jmz-elektronik.ch/dokuwiki/> - Bücher & Dokumente

Permanent link:  
<https://jmz-elektronik.ch/dokuwiki/doku.php?id=start:linux:git-server:installieren&rev=1617037132>

Last update: 2021/03/29 18:58

