



# Network Management

## Network Management

**DE HOGESCHOOL  
MET HET NETWERK**

Hogeschool PXL – Dep. PXL-IT – Elfde-Liniestraat 26 – B-3500 Hasselt  
[www.pxl.be](http://www.pxl.be) - [www.pxl.be/facebook](https://www.facebook.com/pxl.be)



# Intro

- configuratie nic:
  - via GUI: is mogelijk op een desktop  
→ kan problemen geven als simultaan in CLI wordt geconfigureerd
  - via CLI: varieert afhankelijk van distributie  
→ we bekijken Ubuntu



# netplan

- vervangt het vroegere ifup/ifdown
  - Config is nu in een yaml file met volgende algemene opmaak

```
network:  
  Version: 2  
  Renderer: networkd/NetworkManager  
  ethernets:  
    DEVICE_NAME:  
      dhcp4: yes/no  
      addresses: [IP/PREFIX]  
      gateway4: GATEWAY  
      nameservers:  
        addresses: [NAMESERVER, NAMESERVER]  
        search: [LOCAL.DOMAIN]
```



# netplan

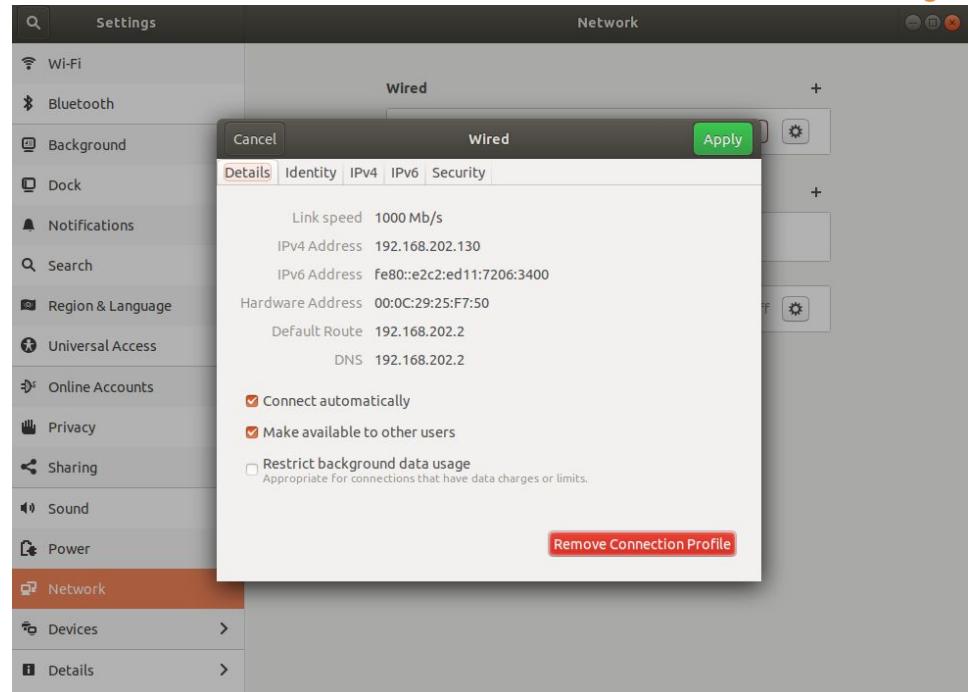
- vervangt het vroegere ifup/ifdown
  - Config in yaml file is afhankelijk van installatie:

```
Install Type Renderer File

Server ISO    systemd-networkd /etc/netplan/01-netcfg.yaml

Cloud Image   systemd-networkd /etc/netplan/50-cloud-init.yaml

Desktop ISO   NetworkManager /etc/netplan/01-network-manager-all.yaml
```



**Op de desktop**  
=> Enkel de Networkmanager als Renderer

```
student@ubuntudesktop01:~$ cd /etc/netplan/
student@ubuntudesktop01:/etc/netplan$ ls
01-network-manager-all.yaml
student@ubuntudesktop01:/etc/netplan$ cat 01-network-manager-all.yaml
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
student@ubuntudesktop01:/etc/netplan$
```

# netplan

- vervangt het vroegere ifup/ifdown
  - Config in yaml file, afhankelijk van installatie:

Install Type	Renderer	File
Server ISO	systemd-networkd	/etc/netplan/01-netcfg.yaml
Cloud Image	systemd-networkd	/etc/netplan/50-cloud-init.yaml
Desktop ISO	NetworkManager	/etc/netplan/01-network-manager-all.yaml

## 1 netwerk kaart:

Renderer : NetworkManager  
 → standaard op Desktop  
 → voor wifi enz. via GUI

## Ethernet (not wifi/bridge)

Naam: ens33 (later meer)  
**Geen vast ip-adres**  
**dhcp4: ip-adres wordt via dhcp**  
**geleverd over ipv4**

**Op de Server**  
**=> IP-adres via DHCP**

```
student@ubuntuserver01:~$ cd /etc/netplan/
student@ubuntuserver01:/etc/netplan$ ls
50-cloud-init.yaml
student@ubuntuserver01:/etc/netplan$ cat 50-cloud-init.yaml
# This file is generated from information provided by
# the datasource. Changes to it will not persist across an instance.
# To disable cloud-init's network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  ethernets:
    ens33:
      addresses: []
      dhcp4: true
      version: 2
student@ubuntuserver01:/etc/netplan$
```

renderer: networkd

mag er ook bij staan



# netplan

Beter voor een server: fixed ip (static)

gateway: alle trafic voor buiten het netwerk wordt naar de gateway (router) gestuurd

Op de Server  
=> IP-adres via DHCP

1 netwerk kaart:

Renderer: networkd

→ geen NetworkManager want  
geen GUI

Ethernet (not wifi/bridge)

Naam: ens33 (later meer)  
vaste instellingen voor  
ip-adres met prefix  
gateway  
nameservers  
search domain  
dhcp4: geen dhcp

```
student@ubuntuserver01:~$ cat 50-cloud-init.yaml
# This file is generated from information provided by
# the datasource. Changes to it will not persist across an instance.
# To disable cloud-init's network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
    version: 2
    renderer: networkd
    ethernets:
        ens33:
            addresses: [172.16.226.137/24]
            gateway4: 172.16.226.2
            nameservers:
                addresses: [172.16.226.2,8.8.8.8,8.8.4.4]
                search: [sysnet.lan]
                # dhcp4: no
                # dhcp6: no
student@ubuntuserver01:~$
```



# netplan

Wijzigingen in /etc/netplan/\*.yaml worden doorgevoerd met

`sudo netplan try` → 120 seconden de tijd om ongedaan te maken  
`sudo netplan apply`

Dit maakt ook de volgende file aan

`/run/systemd/network/<nr>-netplan-<nice>.network`

**STATIC**

```
student@ubuntuserver01:/etc/netplan$ cat /run/systemd/network/10-netplan-ens33.network
[Match]
Name=ens33

[Network]
Address=172.16.226.137/24
Gateway=172.16.226.2
DNS=172.16.226.2
DNS=8.8.8.8
DNS=8.8.4.4
Domains=sysnet.lan
student@ubuntuserver01:/etc/netplan$
```

**DHCP**

```
student@ubuntuserver01:$ cat /run/systemd/network/10-netplan-ens33.network
[Match]
Name=ens33

[Network]
DHCP=ipv4

[DHCP]
UseMTU=true
RouteMetric=100
student@ubuntuserver01:$
```

# netplan

Wijzigingen in /etc/netplan/\*.yaml worden doorgevoerd met  
sudo netplan apply

Dit maakt ook de nodige instellingen in volgende file

/etc/resolv.conf → hier zie je enkel nog de DNS-search  
→ resolving wordt nu gedaan door systemd-resolve  
→ deze instellingen vind je nu terug in  
/run/systemd/resolve/resolv.conf  
OF via systemctl-resolve --status

```
student@ubuntuserver01:/etc/netplan$ systemctl-resolve --status | tail -4
          DNS Servers: 172.16.226.2
                           8.8.8.8
                           8.8.4.4
          DNS Domain: sysnet.lan
student@ubuntuserver01:/etc/netplan$
```



# network status bekijken

Je kan de huidige instellingen ook bekijken met networkctl

Op de server

```
student@ubuntuserver01:~$ networkctl
IDX LINK           TYPE      OPERATIONAL SETUP
  1 lo             loopback  carrier    unmanaged
  2 ens33          ether     routable   configured

2 links listed.

student@ubuntuserver01:~$ networkctl status ens33
• 2: ens33
      Link File: /lib/systemd/network/99-default.link
      Network File: /run/systemd/network/10-netplan-ens33.network
          Type: ether
          State: routable (configured)
          Path: pci-0000:02:01.0
          Driver: e1000
          Vendor: Intel Corporation
          Model: 82545EM Gigabit Ethernet Controller (Copper) (PRO/1000
HW Address: 00:0c:29:31:8f:3d (VMware, Inc.)
          Address: 172.16.226.137
                     fe80::20c:29ff:fe31:8f3d
          Gateway: 172.16.226.2 (VMware, Inc.)
          DNS: 172.16.226.2
Search Domains: localdomain
student@ubuntuserver01:~$
```



# Benaming netwerkkaarten

```
student@ubuntuserver01:~$ 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: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:9d:eb:4e brd ff:ff:ff:ff:ff:ff
    inet 172.16.226.137/24 brd 172.16.226.255 scope global dynamic ens33
        valid_lft 1030sec preferred_lft 1030sec
    inet6 fe80::20c:29ff:fe9d:eb4e/64 scope link
        valid_lft forever preferred_lft forever
student@ubuntuserver01:~$
```

In de laatste versies van ubuntu zijn de netwerkkaarten niet meer eth0, eth1, eth2, enz. genoemd

**ens33:** is hier de enige netwerkkaart.

→ Zou vroeger dus eth0 geweest zijn



# Benaming netwerkkaarten

Vroeger werden de netwerkkaarten genummerd, in de volgorde dat de kernel deze zag tijdens het booten. Daarom kon het soms zijn dat eth0 en eth1 ineens gewisseld waren!

Nu wordt iedere netwerkkaart tijdens het booten hernoemd naar een eenduidige naam, die steeds hetzelfde is.

en -- ethernet                sl -- serial line IP (slip)                wl -- wlan                ww -- wwan

b<number> -- BCMA bus core number                ccw<name> -- CCW bus group name

o<index>[d<dev\_port>] -- on-board device index number

s<slot>[f<function>][d<dev\_port>] -- hotplug slot index number

x<MAC> -- MAC address

```
$ dmesg | grep -i eth

[    3.050064] e1000 0000:02:01.0 eth0: (PCI:66MHz:32-bit) 00:0c:29:05:a3:e2
[    3.050074] e1000 0000:02:01.0 eth0: Intel(R) PRO/1000 Network Connection
[    3.057410] e1000 0000:02:01.0 ens33: renamed from eth0
```



# Benaming netwerkkaarten

Het hernoemen van de netwerkkaarten naar een naam die steeds hetzelfde blijft is een goed idee. Indien je echter terug wilt naar de oude namen eth0, eth1, enzovoorts, kan je dat op volgende wijze:

```
$ sudo nano /etc/default/grub
```

```
GRUB_CMDLINE_LINUX="net.ifnames=0 biosdevname=0"
```

```
$ sudo grub-mkconfig -o /boot/grub/grub.cfg
```

```
$ sudo nano /etc/network/interfaces
```

```
auto eth0
iface eth0 inet static
    address 192.168.12.12
```

```
$ sudo reboot
```



# ifconfig vs ip

- ifconfig is deprecated
  - ip is het vervangende commando

```
student@ubuntuserver01:~$ ifconfig ens33
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 172.16.226.137 netmask 255.255.255.0 broadcast 172.16.226
           .255
      inet6 fe80::20c:29ff:fe9d:eb4e prefixlen 64 scopeid 0x20<link>
ether 00:0c:29:9d:eb:4e txqueuelen 1000 (Ethernet)
      RX packets 315 bytes 354570 (354.5 KB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 198 bytes 14947 (14.9 KB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

student@ubuntuserver01:~$ ip a s ens33
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel stat
e UP group default qlen 1000
      link/ether 00:0c:29:9d:eb:4e brd ff:ff:ff:ff:ff:ff
      inet 172.16.226.137/24 brd 172.16.226.255 scope global dynamic ens33
          valid_lft 1782sec preferred_lft 1782sec
      inet6 fe80::20c:29ff:fe9d:eb4e/64 scope link
          valid_lft forever preferred_lft forever
student@ubuntuserver01:~$
```



# ifconfig - deprecated

- informatie opvragen en wijzigingen aanbrengen
  - informatie opvragen zonder arguments: alle nic's

```
student@ubuntuServer:~$ ifconfig          Mac address
ens33      Link encap:Ethernet HWaddr 00:0c:29:52:5a:5b           Subnet mask
           IP address inet addr:172.16.110.128 Bcast:172.16.110.255 Mask:255.255.255.0
                     inet6 addr: fe80::20c:29ff:fe52:5a5b/64 Scope:Link
                     UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
                     RX packets:56 errors:0 dropped:0 overruns:0 frame:0
                     TX packets:53 errors:0 dropped:0 overruns:0 carrier:0
                     collisions:0 txqueuelen:1000
                     RX bytes:12834 (12.8 KB) TX bytes:5903 (5.9 KB)
                                         Broadcast address
lo         Link encap:Local Loopback
           inet addr:127.0.0.1 Mask:255.0.0.0
                     inet6 addr: ::1/128 Scope:Host
                     UP LOOPBACK RUNNING MTU:65536 Metric:1
                     RX packets:160 errors:0 dropped:0 overruns:0 frame:0
                     TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
                     collisions:0 txqueuelen:1
                     RX bytes:11840 (11.8 KB) TX bytes:11840 (11.8 KB)
```

# ip address

- informatie opvragen en wijzigingen aanbrengen
  - informatie opvragen zonder arguments: alle nic's

```
student@ubuntuServer:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    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: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 10
  00          Mac address
    link/ether 00:0c:29:52:5a:5b brd ff:ff:ff:ff:ff:ff
    inet 172.16.110.128/24 brd 172.16.110.255 scope global ens33
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe52:5a5b/64 scope link
        valid_lft forever preferred_lft forever
```

IP address      prefix

Link state

Broadcast address



Of ip addr      of ip address show of ip a      of ip a s

# ip address show <nic>

- informatie opvragen en wijzigingen aanbrengen
  - informatie opvragen met arguments: 1 specifieke nic

```
student@ubuntuServer:~$ ip a s dev ens33
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:52:5a:5b brd ff:ff:ff:ff:ff:ff
    inet 172.16.110.128/24 brd 172.16.110.255 scope global ens33
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe52:5a5b/64 scope link
        valid_lft forever preferred_lft forever
```

ip a s ens33      of      ip a s dev ens33



# ip addr show up

```
student@ubuntuServer:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    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: ens33: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast state DOWN group default qlen 1000
    link/ether 00:0c:29:52:5a:5b brd ff:ff:ff:ff:ff:ff
    inet 172.16.110.128/24 brd 172.16.110.255 scope global ens33
        valid_lft forever preferred_lft forever
```

→ Let op: State is down, maar netwerkkaart heeft nog altijd een IP !!!

```
student@ubuntuServer:~$ ip a s up
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    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
```



'ip a s up' toont enkel de actieve linken

('down' bestaat niet als argument)

# ip link up en ip link down

## disablen van een nic

```
ip link set ens33 down
```

## enablen van een nic

```
ip link set ens33 up
```

→ Dit leest de netplan-yaml file opnieuw uit!



# herstarten van de networkservice

sudo netplan apply

OF

sudo systemctl restart NetworkManager (op Desktop)

sudo systemctl restart systemd-networkd (op Server)

configuratie uit yaml-file in /etc/netplan wordt opnieuw toegepast



# setting up IP address

## Tijdelijke wijziging

```
student@ubuntuServer:~$ ip a s ens33 | grep 172
    inet 172.16.110.100/24 brd 172.16.110.255 scope global ens33
student@ubuntuServer:~$ sudo ip addr del 172.16.110.100/24 dev ens33
student@ubuntuServer:~$ ip a s ens33 | grep 172 — geen output=geen ip
student@ubuntuServer:~$ sudo ip addr add 172.16.110.101/24 dev ens33
student@ubuntuServer:~$ ip a s ens33 | grep 172
    inet 172.16.110.101/24 scope global ens33
```

ip addr del 172.16.110.100/24 dev ens 33	om 1 IP van de nic te verwijderen
ip addr flush dev ens33	om alle IPs van de nic te verwijderen
ip addr add 172.16.110.101/24 dev ens33	om 1 IP van de nic toe te voegen

**ip link up → Verwijdt huidige instellingen en leest de yaml-file opnieuw uit !**

```
student@ubuntuServer:~$ sudo ip link set ens33 down && sudo ip link set ens33 up
student@ubuntuServer:~$ ip a s ens33 | grep 172
    inet 172.16.110.100/24 brd 172.16.110.255 scope global ens33
```



# hostname

## Tijdelijke wijziging

Nieuwe naam zichtbaar bij het starten van een nieuwe shell

```
student@ubuntuServer:~$ sudo hostname nieuwenaam
student@ubuntuServer:~$ bash
student@nieuwenaam:~$ cat /etc/hostname
ubuntuServer
```

/etc/hostname is niet aangepast, dus bij een reboot opnieuw oude naam



hostnames mogen bestaan uit 64 letters, cijfers, (punten en) koppeltekens, maar niet eindigen met een koppelteken.

# hostnamectl

## Permanente wijziging

```
student@ubuntuServer:~$ cat /etc/hostname  
ubuntuServer  
student@ubuntuServer:~$ sudo hostnamectl set-hostname nieuwenaam  
sudo: unable to resolve host nieuwenaam  
student@ubuntuServer:~$ cat /etc/hostname  
nieuwenaam  
student@ubuntuServer:~$ bash  
student@nieuwenaam:~$
```

/etc/hostname is aangepast, dus bij het starten van een nieuwe shell en het herstarten van de PC in de toekomst blijft de nieuwe naam behouden.

ENKEL ALS: in de file /etc/cloud/cloud.cfg => preserve\_hostname: true

Default:

```
student@ubuntuserver01:~$ cat /etc/cloud/cloud.cfg | grep preserve_hostname  
preserve_hostname: false
```

Je kan natuurlijk ook gewoon de file "/etc/hostname" aanpassen met vi of nano.



# /etc/hosts

## Aanpassen van /etc/hosts voor name-resolving (voor sudo)

- sudo doet voor ieder commando een name-resolving voor de hostname
  - Daarom is het belangrijk dat je de file /etc/hosts ook aanpast
    - anders heb je een lange timeout voordat een sudo commando wordt uitgevoerd

```
student@ubuntuServer:~$ cat /etc/hostname
ubuntuServer
student@ubuntuServer:~$ cat /etc/hosts
127.0.0.1      localhost
127.0.1.1      ubuntuServer

# The following lines are desirable for IPv6 capable hosts
::1      localhost ip6-localhost ip6-loopback
ff02::1  ip6-allnodes
ff02::2  ip6-allrouters.
```



# arp - deprecated

Toon alle entries

```
student@ubuntuserver01:~$ arp -a
_gateway (192.168.202.2) at 00:50:56:ec:9d:b4 [ether] on ens33
? (192.168.202.130) at 00:0c:29:25:f7:50 [ether] on ens33
```

```
student@ubuntuserver01:~$ ip a s dev ens33
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
      link/ether 00:0c:29:a8:0c:fb brd ff:ff:ff:ff:ff:ff
      inet 192.168.202.10/24 brd 192.168.202.255 scope global ens33
```

```
student@ubuntuserver01:~$ ping 192.168.202.132
PING 192.168.202.132 (192.168.202.132) 56(84) bytes of data.
64 bytes from 192.168.202.132: icmp_seq=1 ttl=64 time=0.403 ms
```

```
student@ubuntuserver01:~$ arp -a
? (192.168.202.132) at 00:0c:29:76:2f:ad [ether] on ens33
_gateway (192.168.202.2) at 00:50:56:ec:9d:b4 [ether] on ens33
? (192.168.202.130) at 00:0c:29:25:f7:50 [ether] on ens33
```



# arp - deprecated

```
student@ubuntuserver01:~$ arp -a
? (192.168.202.132) at 00:0c:29:76:2f:ad [ether] on ens33
_gateway (192.168.202.2) at 00:50:56:ec:9d:b4 [ether] on ens33
? (192.168.202.130) at 00:0c:29:25:f7:50 [ether] on ens33
```

Delete entry

```
student@ubuntuserver01:~$ sudo arp -d 192.168.202.132
student@ubuntuserver01:~$ arp -a
_gateway (192.168.202.2) at 00:50:56:ec:9d:b4 [ether] on ens33
? (192.168.202.130) at 00:0c:29:25:f7:50 [ether] on ens33
```

```
student@ubuntuserver01:~$ ping 192.168.202.133
PING 192.168.202.133 (192.168.202.133) 56(84) bytes of data.
From 192.168.202.10 icmp_seq=1 Destination Host Unreachable
```

ping naar onbestaand IP

```
student@ubuntuserver01:~$ arp -a
? (192.168.202.133) at <incomplete> on ens33
_gateway (192.168.202.2) at 00:50:56:ec:9d:b4 [ether] on ens33
? (192.168.202.130) at 00:0c:29:25:f7:50 [ether] on ens33
```



# ip neighbor (arp-table)

IP-NEIGHBOUR(8)

Linux

IP-NEIGHBOUR(8)

## NAME

ip-neighbor - neighbour/arp tables management.

## SYNOPSIS

```
ip [ OPTIONS ] neigh { COMMAND | help }

ip neigh { add | del | change | replace } { ADDR [ lladdr LLADDR ] [
    nud STATE ] | proxy ADDR } [ dev DEV ]

ip neigh { show | flush } [ proxy ] [ to PREFIX ] [ dev DEV ] [ nud
    STATE ]

STATE := { permanent | noarp | stale | reachable | none | incomplete |
    delay | probe | failed }
```

## DESCRIPTION

The **ip neigh** command manipulates neighbour objects that establish bindings between protocol addresses and link layer addresses for hosts sharing the same link. Neighbour entries are organized into tables. The IPv4 neighbour table is also known by another name - the ARP table.



# ip neighbor (of ip n)

Toon alle entries

```
student@ubuntuserver01:~$ ip n show
```

```
192.168.202.133 dev ens33 FAILED
```

```
192.168.202.2 dev ens33 lladdr 00:50:56:ec:9d:b4 STALE
```

```
192.168.202.130 dev ens33 lladdr 00:0c:29:25:f7:50 DELAY
```

192.168.202.133 ping host unreachable

192.168.202.130 Connectie via ssh

```
student@ubuntuserver02:~$ ip a s ens33
```

```
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
```

```
    link/ether 00:0c:29:76:2f:ad brd ff:ff:ff:ff:ff:ff
```

```
    inet 192.168.202.132/24 brd 192.168.202.255 scope global dynamic ens33
```

```
student@ubuntuserver01:~$ ping 192.168.202.132
```

```
PING 192.168.202.132 (192.168.202.132) 56(84) bytes of data.
```

```
64 bytes from 192.168.202.132: icmp_seq=1 ttl=64 time=0.547 ms
```

```
student@ubuntuserver01:~$ ip n show
```

```
192.168.202.132 dev ens33 lladdr 00:0c:29:76:2f:ad STALE
```

```
192.168.202.133 dev ens33 FAILED
```

```
192.168.202.2 dev ens33 lladdr 00:50:56:ec:9d:b4 STALE
```

```
192.168.202.130 dev ens33 lladdr 00:0c:29:25:f7:50 DELAY
```

Delete ent

```
student@ubuntuserver01:~$ sudo ip n del 192.168.202.132 dev ens33
```

```
student@ubuntuserver01:~$ ip n show
```

```
192.168.202.133 dev ens33 FAILED
```

```
192.168.202.2 dev ens33 lladdr 00:50:56:ec:9d:b4 DELAY
```

```
192.168.202.130 dev ens33 lladdr 00:0c:29:25:f7:50 REACHABLE
```



# route - deprecated

```
student@ubuntuserver01:~$ route
Kernel IP routing table
Destination      Gateway          Genmask        Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0        UG     0      0        0 ens33
192.168.202.0   0.0.0.0        255.255.255.0  U       0      0        0 ens33
student@ubuntuserver01:~$ netstat -r
Kernel IP routing table
Destination      Gateway          Genmask        Flags  MSS Window irtt Iface
default         _gateway        0.0.0.0        UG        0 0        0 ens33
192.168.202.0   0.0.0.0        255.255.255.0  U        0 0        0 ens33
```

Tijdelijk wijzigingen routering: (vb. default gateway)

```
sudo route add default gw 192.168.202.xx  
(man route)
```



# ip route

IP-ROUTE (8)

Linux

IP-ROUTE (8)

## NAME

ip-route - routing table management

## SYNOPSIS

```
ip [ ip-OPTIONS ] route { COMMAND | help }

ip route { list | flush } SELECTOR

ip route get ADDRESS [ from ADDRESS iif STRING ] [ oif STRING ] [ tos
TOS ]

ip route { add | del | change | append | replace } ROUTE

SELECTOR := [ root PREFIX ] [ match PREFIX ] [ exact PREFIX ] [ table
TABLE_ID ] [ proto RTPROTO ] [ type TYPE ] [ scope SCOPE ]
```

## DESCRIPTION

**ip route** is used to manipulate entries in the kernel routing tables.

### Route types:

**unicast** - the route entry describes real paths to the destinations covered by the route prefix



# ip route (of ip r)

```
student@ubuntuserver01:~$ ip route
default via 192.168.202.2 dev ens33 proto static
192.168.202.0/24 dev ens33 proto kernel scope link src 192.168.202.10
```

Of ip r show of ip r list

Tijdelijk wijzigingen routering:

vb. default gateway

eventueel eerst de foutieve verwijderen

sudo ip route del default

nadien nieuwe toevoegen

sudo ip route add default via 192.168.14.xx



# ping

Met ping wordt vaak de TCP/IP configuratie getest.  
(ook traceroute, dig)

```
student@ubuntuserver01:~$ ping 192.168.202.132
PING 192.168.202.132 (192.168.202.132) 56(84) bytes of data.
64 bytes from 192.168.202.132: icmp_seq=1 ttl=64 time=0.711 ms
64 bytes from 192.168.202.132: icmp_seq=2 ttl=64 time=0.830 ms
64 bytes from 192.168.202.132: icmp_seq=3 ttl=64 time=0.871 ms
Ctrl-C ^C
--- 192.168.202.132 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2030ms
rtt min/avg/max/mdev = 0.711/0.804/0.871/0.067 ms
```

```
ping -c5 192.168.202.132
```



# VMware: NAT-instelling in Workstation

The screenshot shows the VMware Workstation interface with the following details:

- Library:** Shows various virtual machines and shared VMs.
- Virtual Machine Details:** For the selected VM "OS\_Ess\_1J\_UbServ1804".
- Devices:** Configuration for Memory (1 GB), Processors (1), Hard Disk (SCSI, 20 GB, using file C:\Use...), Network Adapter (NAT), USB Controller (Present), Sound Card (Auto detect), Printer (Present), and Display (Auto detect).
- Description:** A field to enter a description of the virtual machine.
- Virtual Network Editor:** A dialog box showing network adapter settings:

Name	Type	External Connection	Host Connection	DHCP	Subnet Address
VMnet1	Host-only	-	-	Enabled	192.168.152.0
VMnet5	Custom	-	-	-	192.168.220.0
VMnet8	NAT	NAT	Connected	Enabled	172.16.226.0

- VMnet Information:** Options for network connection type:
  - Bridged (connect VMs directly to the external network)  
Bridged to: [dropdown] Automatic Settings...
  - NAT (shared host's IP address with VMs)  
NAT Settings...
  - Host-only (connect VMs internally in a private network)
- Checkboxes:** Connect a host virtual adapter to this network (Host virtual adapter name: VMware Network Adapter VMnet8) and Use local DHCP service to distribute IP address to VMs.
- Buttons:** DHCP Settings..., OK, Cancel, Apply, Help.
- Warning:** Administrator privileges are required to modify the network configuration.
- Bottom:** To direct input to this VM, click inside or press Ctrl+G.



# VMware: NAT-instellingen op de host

interface VMnet8

NAT router en DNS Server  
ip = 172.16.226.**2**

DHCP Server  
ip = 172.16.226.**254**

```
PS C:\Users\gertv> Get-NetIPConfiguration *VMnet8

InterfaceAlias      : VMware Network Adapter VMnet8
InterfaceIndex       : 15
InterfaceDescription : VMware Virtual Ethernet Adapter for VMnet8
IPv4Address          : 172.16.226.1
IPv6DefaultGateway   :
IPv4DefaultGateway   :
DNSServer            : fec0:0:0:ffff::1
                         fec0:0:0:ffff::2
                         fec0:0:0:ffff::3
```

```
PS C:\Users\gertv>
```



# VMware: NAT-instelling in de VM

**Desktop:** Een lease wordt aangevraagd door de NetworkManager bij het opstarten of herstarten van het netwerk van deze netwerkkaart.

```
student@ubuntudesktop01:~$ tail -16 /var/lib/NetworkManager/dhclient-e6c4d070-9d  
3e-3144-837c-92a3594e9379-ens33.lease  
lease {  
    interface "ens33";  
    fixed-address 172.16.226.148;  
    option subnet-mask 255.255.255.0;  
    option routers 172.16.226.2;  
    option dhcp-lease-time 1800;  
    option dhcp-message-type 5;  
    option domain-name-servers 172.16.226.2;  
    option dhcp-server-identifier 172.16.226.254;  
    option broadcast-address 172.16.226.255;  
    option netbios-name-servers 172.16.226.2;  
    option domain-name "localdomain";  
    renew 3 2018/10/03 18:31:20;  
    rebind 3 2018/10/03 18:43:52;  
    expire 3 2018/10/03 18:47:37;  
}  
student@ubuntudesktop01:~$
```



# VMware: NAT-instelling in de VM

**Server:** Een lease wordt aangevraagd door de networkd bij het opstarten of herstarten van het netwerk van deze netwerkkaart.

```
student@ubuntuserver01:~$ cat /var/run/systemd/netif/leases/Z
# This is private data. Do not parse.
ADDRESS=172.16.226.137
NETMASK=255.255.255.0
ROUTER=172.16.226.2
SERVER_ADDRESS=172.16.226.254
NEXT_SERVER=172.16.226.254
T1=900
T2=1575
LIFETIME=1800
DNS=172.16.226.2
DOMAINNAME=localdomain
CLIENTID=ff2b9434c100020000ab11e422c0dcfce95e73
student@ubuntuserver01:~$
```

```
student@ubuntuserver01:~$ networkctl
IDX LINK           TYPE      OPERATIONAL SETUP
  1 lo             loopback  carrier    unmanaged
  2 ens33          ether     routable  configured

2 links listed.
student@ubuntuserver01:~$
```



# VMware: NAT-instelling in de VM

## DNS en GATEWAY

```
student@ubserv: $ networkctl status
•      State: routable
          Address: 192.168.87.128 on ens33
                      fe80::20c:29ff:fe83:7dfc on ens33
          Gateway: 192.168.87.2 (VMware, Inc.) on ens33
          DNS: 8.8.8.8
                  8.8.4.4
Search Domains: localdomain
```



# ifconfig vs ip

## COMPARING NET-TOOLS VS. IPROUTE PACKAGE COMMANDS

### NET-TOOLS COMMANDS

```
arp -a  
arp -v  
arp -s 192.168.1.1 1:2:3:4:5:6  
arp -i eth1 -d 192.168.1.1  
ifconfig -a  
ifconfig eth0 down  
ifconfig eth0 up  
ifconfig eth0 192.168.1.1  
ifconfig eth0 netmask 255.255.255.0  
ifconfig eth0 mtu 9000  
ifconfig eth0:0 192.168.1.2  
netstat  
netstat -neopa  
netstat -g  
route  
route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0  
route add default gw 192.168.1.1
```

### IPROUTE COMMANDS

```
ip neigh  
ip -s neigh  
ip neigh add 192.168.1.1 lladdr 1:2:3:4:5:6 dev eth1  
ip neigh del 192.168.1.1 dev eth1  
ip addr  
ip link set eth0 down  
ip link set eth0 up  
ip addr add 192.168.1.1/24 dev eth0  
ip addr add 192.168.1.1/24 dev eth0  
ip link set eth0 mtu 9000  
ip addr add 192.168.1.2/24 dev eth0  
ss  
ss -neopa  
ip maddr  
ip route  
ip route add 192.168.1.0/24 dev eth0  
ip route add default via 192.168.1.1
```

