Sistemas instalados por defecto en varias versiones:

```
$sudo systemctl --type=service --all | grep -i "net"
```

Ubuntu 22.04

```
persona@ubuntu22:/etc/NetworkManager/system-connections$ sudo systemctl --type=service --all | grep -i "net"
  netplan-ovs-cleanup.service                 loaded    inactive dead    OpenVSwitch configuration for cleanup
  networkd-dispatcher.service                 loaded    active   running Dispatcher daemon for systemd-networkd
  NetworkManager-wait-online.service          loaded    active   exited  Network Manager Wait Online
  NetworkManager.service                      loaded    active   running Network Manager
  systemd-networkd.service                    loaded    inactive dead    Network Configuration
  systemd-resolved.service                    loaded    active   running Network Name Resolution
  systemd-timesyncd.service                   loaded    active   running Network Time Synchronization
```

Ubuntu 24.04

```
root@persona-VirtualBox:/etc/netplan# systemctl --type=service --all | grep -i "net"
  cloud-init-local.service                 loaded    inactive dead    Initial cloud-init job (pre-networking)
  netplan-ovs-cleanup.service              loaded    inactive dead    OpenVSwitch configuration for cleanup
  networkd-dispatcher.service              loaded    inactive dead    Dispatcher daemon for systemd-networkd
● networking.service                       not-found inactive dead    networking.service
  NetworkManager-wait-online.service       loaded    active   exited  Network Manager Wait Online
  NetworkManager.service                   loaded    active   running Network Manager
  systemd-networkd-wait-online.service     loaded    inactive dead    Wait for Network to be Configured
  systemd-networkd.service                 loaded    inactive dead    Network Configuration
  systemd-resolved.service                 loaded    active   running Network Name Resolution
  systemd-timesyncd.service                loaded    active   running Network Time Synchronization
```

Ubuntu Server 24.04

```
persona@ubuntuserver24:~$ sudo systemctl --type=service --all | grep -i "net"
[sudo] password for persona:
  cloud-init-local.service                 loaded    active   exited  Initial cloud-init job (pre-networking)
  netplan-ovs-cleanup.service              loaded    inactive dead    OpenVSwitch configuration for cleanup
  networkd-dispatcher.service              loaded    inactive dead    Dispatcher daemon for systemd-networkd
● networking.service                       not-found inactive dead    networking.service
● NetworkManager.service                   not-found inactive dead    NetworkManager.service
  systemd-networkd-wait-online.service     loaded    active   exited  Wait for Network to be Configured
  systemd-networkd.service                 loaded    active   running Network Configuration
  systemd-resolved.service                 loaded    active   running Network Name Resolution
  systemd-timesyncd.service                loaded    active   running Network Time Synchronization
```

Debian Desktop 12

```
root@debian12:~# systemctl --type=service --all | grep -i "net"
  networking.service                       loaded    active   exited  Raise network interfaces
  NetworkManager-wait-online.service       loaded    active   exited  Network Manager Wait Online
  NetworkManager.service                   loaded    active   running Network Manager
  systemd-networkd.service                 loaded    inactive dead    Network Configuration
  systemd-timesyncd.service                loaded    active   running Network Time Synchronization
```

Debian Server (minimal) 12

```
root@debianminimo:~# systemctl --type=service --all | grep -i "net"
  networking.service                       loaded    active   exited  Raise network interfaces
● NetworkManager.service                   not-found inactive dead    NetworkManager.service
  systemd-networkd.service                 loaded    inactive dead    Network Configuration
```

LOGS (registros): ver los logs con información de funcionamiento

```
$ journalctl -b
# journalctl -u ssh.service
```

Operación	Command
View log for system services and kernel from the last boot	`journalctl -b --system`
View log for services of the current user from the last boot	`journalctl -b --user`
View job log of "`$unit`" from the last boot	`journalctl -b -u $unit`
View job log of "`$unit`" ("`tail -f`" style) from the last boot	`journalctl -b -u $unit -f`

Operación	Comando
List all available unit types	`systemctl list-units --type=help`
List all target units in memory	`systemctl list-units --type=target`
	`systemctl list-unit-files`
List all service units in memory	`systemctl list-units --type=service`
List all device units in memory	`systemctl list-units --type=device`
List all mount units in memory	`systemctl list-units --type=mount`
	`systemctl list-unit-files --state=masked | running | active |`
List all socket units in memory	`systemctl list-sockets`
List all timer units in memory	`systemctl list-timers`
List failed units	`systemctl --failed`
Start "`$unit`"	`systemctl start $unit`
Stop "`$unit`"	`systemctl stop $unit`
Reload service-specific configuration	`systemctl reload $unit`
Stop and start all "`$unit`"	`systemctl restart $unit`
Check status of "`$unit\|$PID|device`"	`systemctl status $unit\|$PID|$device`
	`systemctl status --type=service`
	`systemctl --type=service`
Show properties of "`$unit\|$job`"	`systemctl show $unit\|$job`
Reset failed "`$unit`"	`systemctl reset-failed $unit`
List dependency of all unit services	`systemctl list-dependencies --all`
List unit files installed on the system	`systemctl list-unit-files`
Enable "`$unit`" (add symlink)	`systemctl enable $unit`
Disable "`$unit`" (remove symlink)	`systemctl disable $unit`
Check whether a unit is enabled	`systemctl is-enabled unit`
Unmask "`$unit`" (remove symlink to "`/dev/null`")	`systemctl unmask $unit`
Mask "`$unit`" (add symlink to "`/dev/null`")	`systemctl mask $unit`
Get default-target setting	`systemctl get-default`
Set default-target to "`graphical`" (GUI system)	`systemctl set-default graphical`
Set default-target to "`multi-user`" (CLI system)	`systemctl set-default multi-user`
Show job environment	`systemctl show-environment`
Set job environment "`variable`" to "`value`"	`systemctl set-environment variable=value`
Unset job environment "`variable`"	`systemctl unset-environment variable`
Reload all unit files and daemons	`systemctl daemon-reload`
Shut down the system	`systemctl poweroff`
Shut down and reboot the system	`systemctl reboot`
Suspend the system	`systemctl suspend`
Hibernate the system	`systemctl hibernate`

Habilitar/desabilitar y levantar servicios:

```
sudo systemctl disable network-manager
sudo systemctl stop network-manager

sudo systemctl enable systemd-networkd
sudo systemctl start systemd-networkd
sudo systemctl start systemd-resolved

sudo systemctl enable systemd-resolved
sudo systemctl restart systemd-networkd
```

Herramientas

```
lshw -class network
ethtool enp0s3

rgrep wlan0 /etc
find /etc -name '*wlan0*'
echo /sys/class/net/*
udevadm test-builtin net_id /sys/class/net/enp0s1 2>/dev/null
```

```
ss -ltu       ss -tu
ss -tua       ss -s
ss -4         ss -6
ss -p         ss -l

$ ss -at '( dport = :22 or sport = :22 )'
```

```
netstat -putan
```

Configuración de parámetros IPv4 (en azul, lo relativo a configuración de resolución de nombres)

1. Sistema manual

Comandos:
Añadir una dirección IP y una máscara a una interfaz de red:
```
ip addr add 10.102.66.200/24 dev enp0s25
```

Activar (subir) y desactivar (bajar) una interfaz de red:
```
ip link set dev enp0s25 up
ip link set dev enp0s25 down
```

Ver los parámetros IP de una interfaz de red:
```
ip address show dev enp0s25
```

Añadir una ruta a la tabla de enrutamiento:
```
ip route add default via 10.102.66.1
```

Ver la tabla de enrutamiento:
```
ip route show
```

Borrar toda la configuración de la interfaz de red:
```
ip addr flush eth0
```

Propiedades de la interfaz de loopback:
```
ip address show lo
```

Cambiar temporalmente el nombre a una interfaz de red:
```
ip link set dev eth0 down
ip link set dev eth0 name neweth0
ip link set dev neweth0 up
```

2. Sistema clásico sistemas tipo Debian

Debian 12 minimal/Server

Directorios y ficheros:
```
/etc/network/interfaces
/etc/resolv.conf
/etc/hosts
/etc/nsswitch.conf
```

Comandos:
```
ifup enp0s3
ifdown enps03
systemctl restart networking
```

Ejemplo 1: fichero interfaces
```
auto eth0
iface eth0 inet dhcp
```

Ejemplo 2: fichero interfaces
```
iface eth0 inet static
   address 192.168.1.5
   netmask 255.255.255.0
   gateway 192.168.1.254
```

3. systemd-networkd y systemd-resolved

Directorios y ficheros:
```
/etc/systemd/network
/etc/systemd/network/fichero.network
/etc/systemd/network/fichero.network.d/*.conf
/run/systemd/network
/lib/systemd/network
/etc/resolv.conf
/etc/hosts
/etc/nsswitch.conf
/etc/systemd/resolved.conf    configuración global
/etc/systemd/resolved.conf.d/*.conf
/run/systemd/resolved.conf.d/*.conf
/usr/lib/systemd/resolved.conf.d/*.conf
/run/systemd/resolve/stub-resolv.conf
```

Comandos:
```
networkctl
resolvectl


networkctl list
networkctl status
networkctl -a
networkctl -s
networkctl up enp0s3
networkctl down enp0s3
networkctl renew enp0s3
networkctl reconfigure enp0s3
networkctl reload
```

Crear fichero configuración interfaz de red DHCP:
En `/etc/systemd/network/20-dhcp.network`
```
[Match]
Name=enp5s0

[Network]
DHCP=yes
```

Crear fichero configuración interfaz de red estática:
En `/etc/systemd/network/20-enps03.network`
```
[Network]
Address=10.1.10.9/24
Gateway=10.1.10.1
DNS=10.1.10.1
Domains=empresa.net filial.empresa.net
```

Cambiar el nombre de una interfaz por uno personalizado:
`/etc/systemd/network/10-persistent-net.link`
```
[Match]
MACAddress=01:23:45:67:89:ab

[Link]
Name=nuevoNombre
```

RESOLUCIÓN DE NOMBRES

```
resolvectl status
resolvectl flush-caches
resolvectl query www.edu.xunta.gal -i enp0s3
resolvectl statistics
resolvectl dns enps03
resolvectl dns enps03 8.8.4.4 8.8.8.8
resolvectl revert enp0s3
```

Configurar servidores DNS globales
`/etc/systemd/resolved.conf.d/dns_servers.conf`
```
[Resolve]
DNS=192.168.35.1 fd7b:d0bd:7a6e::1
Domains=~.
```

4. NetworkManager

Directorios y ficheros:
```
/etc/NetworkManager
/run/NetworkManager/no-stub-resolv.conf
```

Comandos:
```
nm-connection-editor
nmtui
nmcli
```

Desactivar que NetworkManager gestione resolv.conf:
En `/etc/NetworkManager/NetworkManager.conf`
```
[main]
dns=none
```

Ver la configuración actual:
```
NetworkManager --print-config
```

Comando `nmcli`

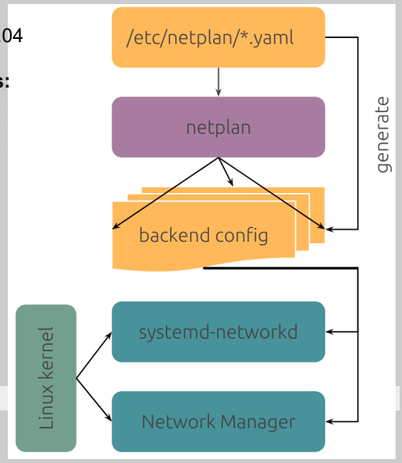
Mostrar estado general de NetworkManager:
```
nmcli general status
nmcli -t -f STATE general
```
Mostrar estado general de NetworkManager:
```
nmcli general status
```
Mostrar todas las conexiones:
```
nmcli connection show
```
Mostrar solo las conexiones activas:
```
nmcli connection show --active
```
Mostrar los dispositivos reconocidos por NetworkManager:
```
nmcli device status
```
Bajar y subir interfaces y/o conexiones:
```
nmcli con up id 'Conexión cableada 1'
nmcli con down id 'Conexión cableada 1'
nmcli dev disconnect enp0s3
nmcli dev connect enp0s3
```
Modifcar el nombre de una interfaz:
```
nmcli connection
```
Listar las propiedades de una conexión concreta:
```
nmcli connection show nombreConexión
```
Listar dispositivos y conexiones gestionadas:
```
nmcli -f NAME,DEVICE,FILENAME connection show
```
Crear una conexión configurada por DHCP:
```
nmcli con add type ethernet
  con-name mi-hogar ifname enp0s3
nmcli con modify mi-hogar mi-hogar ipv4.ignore-auto-dns yes
nmcli con up mi-hogar
```
Crear una conexión con configuración IP estática:
```
nmcli connection add type ethernet
 con-name mi-chollo ifname enp0s3
 ip4 192.168.1.34/24 gw4 192.168.1.1
nmcli con mod mi-chollo ipv4.dns "8.8.8.8 8.8.4.4"
nmcli con up  mi-chollo
```
Añadir más direcciones IP sobre la misma interfaz:
```
nmcli connection modify mi-chollo +ipv4.addresses
"192.168.1.22/24,192.168.1.23/24,192.168.1.24/24"
```
Listar los puntos de acceso WiFi disponibles:
```
nmcli device wifi list
```
Configurar una conexión WiFi, con seguridad WPA2:
```
nmcli con add con-name MiCafe ifname wlp61s0
  type wifi ssid Cafeto
nmcli con modify MiCafe wifi-sec.key-mgmt wpa-psk
nmcli con modify MyCafe wifi-sec.psk SeKret00
nmcli radio wifi on
```
Ver propiedades de la WiFi:
```
nmcli connection show id 'MiCafe'
```

5. netplan

Ubuntu 22.04 /24.04

Directorios y ficheros:
```
/run/netplan/*.yaml
/etc/netplan/*.yaml
/lib/netplan/*.yaml
```

Comandos:
```
netplan
```



Comando `netplan`
```
netplan try
netplan apply
netplan status
netplan ip leases enp0s3
```

Activar systemd-networkd como gestor /backend de red para netplan
`/etc/netplan/99_config.yaml`
```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp3s0:
      dhcp4: true
```

Activar NetworkManager como gestor /backend de red para netplan
```
network:
  version: 2
  renderer: NetworkManager
```

Configurar interfaz con parámetros IPv4 estáticos:
`/etc/netplan/99_config.yaml`
```
network:
  version: 2
  renderer: networkd
  ethernets:
    eth0:
      addresses:
        - 10.10.10.2/24
      routes:
        - to: default
          via: 10.10.10.1
      nameservers:
          search: [mydomain, otherdomain]
           addresses: [10.10.10.1, 1.1.1.1]
```

Fijar un nombre para una interfaz de red
`/etc/netplan/99_config.yaml`
```
network:
  version: 2
  renderer: networkd
  ethernets:
    eth_lan0:
      dhcp4: true
      match:
        macaddress: 00:11:22:33:44:55
      set-name: eth_lan0
```

Desinstalar netplan:
```
sudo apt remove --purge netplan.io
```

Anexo I: Más ejemplos /etc/network/intefaces

https://www.cyberciti.biz/faq/setting-up-an-network-interfaces-file/

```
# We always want the loopback interface.
# auto lo
# iface lo inet loopback

# An example ethernet card setup: (broadcast and gateway are optional)
# auto eth0
# iface eth0 inet static
#     address 192.168.0.42
#     network 192.168.0.0
#     netmask 255.255.255.0
#     broadcast 192.168.0.255
#     gateway 192.168.0.1

# A more complicated ethernet setup, with a less common netmask,
# and a downright weird broadcast address: (the "up" lines are executed
# verbatim when the interface is brought up, the "down" lines when it's
# brought down)
# auto eth0
# iface eth0 inet static
#     address 192.168.1.42
#     network 192.168.1.0
#     netmask 255.255.255.128
#     broadcast 192.168.1.0
#     up route add -net 192.168.1.128 netmask 255.255.255.128 gw 192.168.1.2
#     up route add default gw 192.168.1.200
#     down route del default gw 192.168.1.200
#     down route del -net 192.168.1.128 netmask 255.255.255.128 gw 192.168.1.2

# A more complicated ethernet setup with a single ethernet card with
# two interfaces.
# Note: This happens to work since ifconfig handles it that way, not because
# ifup/down handles the ':' any differently.
# Warning: There is a known bug if you do this, since the state will not
# be properly defined if you try to 'ifdown eth0' when both interfaces
# are up. The ifconfig program will not remove eth0 but it will be
# removed from the interfaces state so you will see it up until you execute:
# 'ifdown eth0:1 ; ifup eth0; ifdown eth0'
# BTW, this is "bug" #193679 (it's not really a bug, it's more of a
# limitation)
# auto eth0 eth0:1
# iface eth0 inet static
#     address 192.168.0.100
#     network 192.168.0.0
#     netmask 255.255.255.0
#     broadcast 192.168.0.255
#     gateway 192.168.0.1
# iface eth0:1 inet static
#     address 192.168.0.200
#     network 192.168.0.0
#     netmask 255.255.255.0

# "pre-up" and "post-down" commands are also available. In addition, the
# exit status of these commands are checked, and if any fail, configuration
# (or deconfiguration) is aborted. So:
# auto eth0
# iface eth0 inet dhcp
#     pre-up [ -f /etc/network/local-network-ok ]
# will allow you to only have eth0 brought up when the file
# /etc/network/local-network-ok exists.

# Two ethernet interfaces, one connected to a trusted LAN, the other to
# the untrusted Internet. If their MAC addresses get swapped (because an
# updated kernel uses a different order when probing for network cards,
# say), then they don't get brought up at all.
# auto eth0 eth1
# iface eth0 inet static
#     address 192.168.42.1
#     netmask 255.255.255.0
#     pre-up /path/to/check-mac-address.sh eth0 11:22:33:44:55:66
#     pre-up /usr/local/sbin/enable-masq
# iface eth1 inet dhcp
#     pre-up /path/to/check-mac-address.sh eth1 AA:BB:CC:DD:EE:FF
#     pre-up /usr/local/sbin/firewall
```

Firewall

Directorios y ficheros:

```
/etc/services

/etc/default/ufw
/etc/ufw/ufw.conf
/etc/ufw/sysctl.conf
/etc/ufw/before.rules
/etc/ufw/after.rules
/etc/ufw/user.rules
```

Comandos:

```
ufw
gufw
iptables


iptables -L
iptables -L -nat
iptables -L -mangle
```

sysctl.conf puede se necesario activar (descomentar para ello):

```
net/ipv4/ip_forward=1      = habilitar el reenvío IPv4
net/ipv6/conf/default/forwarding=1
net/ipv6/conf/all/forwarding=1

ufw disable
ufw enable
ufw reload

ufw status
ufw status numbered

ufw allow in on enp0s3 from 192.168.1.30 to any
  port 22 proto tcp
ufw allow proto tcp from any to any port 22
ufw allow 53
ufw allow 25/tcp
ufw allow smtp     smtp es traducido a un número usando /etc/services
ufw allow in http
ufw reject out smtp
ufw reject telnet comment 'telnet is unencrypted'
ufw deny proto tcp to any port 80
ufw deny proto tcp from 10.0.0.0/8 to 192.168.0.1
   port 25
ufw allow proto tcp from any to any port 80,443,8080:8090
   comment 'web app'
ufw limit ssh/tcp
ufw deny 80/tcp
ufw delete deny 80/tcp
ufw delete 3
```

Logging:

```
ufw allow log 22/tcp
ufw loggin off
ufw loggin full
ufw loggin low
```

Reglas para controlar qué tráfico enrutar:

```
ufw route allow in on enp0s3 out on enp0s8
ufw route allow in on enp0s3 out on enp0s8
   to 12.34.45.67 port 80 proto tcp
```

Activar enmascaramiento IP (IP Masquerading):

Activar reenvío IP (convertir equipo en router):
En `/etc/ufw/sysctl.conf` descomentar:
`net.ipv4.ip_forward=1`

En `/etc/ufw/before.rules`, después de la sección **`*filter`**:

```
*nat
:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -s 10.0.0.0/8 -o eth0 -j MASQUERADE
COMMIT
```

En `/etc/default/ufw` activar (descomentar):

```
DEFAULT_FORWARD_POLICY="ACCEPT"
```

Permitir el paso de tráfico desde la intranet a internet:

```
ufw route allow in on enp0s3 out on enp0s8 from 192.168.1.0/24
```

Redireccionar un puerto (abrir un puerto):

```
ufw route allow in on enp0s8 to 192.168.1.4 port 80 proto tcp
```