

Configuración Red Ubuntu 20.04

A partir de Ubuntu 18.04 la manera en que se configura una máquina Ubuntu ha sufrido cambios. En este documento vamos a ver cómo llevar a cabo estas tareas para instalaciones de Ubuntu 18.04 o posteriores.

1. Identificar Interfaces de red Ethernet

El comando **ifconfig** que se usaba para mostrar la información de los interfaces de red ya no está disponible y ha sido reemplazado por el comando **ip**. Veamos la correspondencia entre ambos comandos:

Acción	ifconfig (antiguo)	ip
Mostrar los interfaces de red y su configuración	# ifconfig	\$ ip addr show \$ ip link show
Activar interface de red	# ifconfig eth0 up	# ip link set eth0 up
Desactivar interface de red	# ifconfig eth0 down	# ip link set eth0 down
Establecer dirección IP (temporal)	# ifconfig eth0 192.168.1.1	# ip address add 192.168.1.1 dev eth0

2. Configurar interfaces de red permanentemente

A diferencia de versiones anteriores en las que se podían configurar las interfaces de red mediante el archivo `/etc/network/interfaces`, a partir de Ubuntu 18.04 esta configuración se lleva a cabo mediante **netplan**. Para ello deberemos usar archivos YAML (extensión `.yaml`) ubicados en `/etc/netplan`. Netplan lee estos archivos (da igual el nombre) y genera los archivos de configuración específicos de bajo nivel que son gestionados por el correspondiente proceso de red.

En el archivo se puede especificar qué proceso de red se va a usar:

- En Ubuntu desktop es usual que la red se gestione con la interfaz gráfica de **Network Manager**.
- En Ubuntu server es más usual usar **systemd-networkd** ya que no suelen disponer de interfaz gráfica.

Para indicar qué proceso de red queremos que se use lo podemos indicar en el archivo `.yaml` mediante la clave **renderer** que puede tomar los valores **NetworkManager** o **networkd**. Por defecto, si no se especifica, se toma el valor `networkd`.

A continuación, podemos ver la configuración de netplan tras la instalación de un Ubuntu Server 20.04.

```

profe@ubusersmr00:~$ cat /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
  version: 2

```

Tras la clave **ethernets** aparecen los nombres de las interfaces existentes (enp0s3) y a continuación, para cada interfaz sus características.

a. Asignación de dirección IP dinámica (vía DHCP)

Si queremos que nuestra máquina solicite su dirección IP y el resto de los parámetros de red a un servidor DHCP, en el archivo de configuración netplan deberemos indicarlo con el valor **true** en el campo **dhcp4** correspondiente al interfaz que queremos modificar (enp0s3) como hemos visto en la anterior captura.

b. Asignación de dirección IP estática

Si queremos asignar manualmente los parámetros de red de la máquina, deberemos usar las siguientes claves dentro del interface a modificar (enp0s3):

- **addresses** para indicar la o las direcciones IP estáticas a configurar junto con su máscara de red.
- **gateway4** para indicar la puerta de enlace
- **nameservers** para indicar los parámetros relacionados con las consultas DNS. Dentro de esta clave podemos usar las siguientes claves:
 - **addresses** para indicar la dirección IP de los servidores DNS a los que realizará las consultas DNS.
 - **search** para indicar el sufijo de dominio a adjuntar cuando se pida una consulta DNS que no incorpore el FQDN

En el siguiente ejemplo hemos configurado los siguientes parámetros:

- Dirección IP: 10.0.100.100
- Máscara de red 255.255.255.0 (/24)
- Puerta de enlace 10.0.100.1
- Servidores DNS: 8.8.8.8 y 8.8.4.4

```

# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      addresses: [10.0.100.100/24]
      gateway4: 10.0.100.1
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
  version: 2

```

Una vez terminado de modificar el archivo, los cambios no tienen lugar hasta que se ejecuta el siguiente comando (con permisos de superusuario):

```
# netplan apply
```

Podemos comprobar que la configuración se ha aplicado correctamente con los comandos:

```
$ ip addr show
```

```
profe@ubuserserversmc00:~$ ip addr show
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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:d5:f5:bd brd ff:ff:ff:ff:ff:ff
    inet 10.0.100.100/24 brd 10.0.100.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fed5:f5bd/64 scope link
        valid_lft forever preferred_lft forever
```

```
$ ip route show
```

```
profe@ubuserserversmc00:~$ ip route show
default via 10.0.100.1 dev enp0s3 proto static
10.0.100.0/24 dev enp0s3 proto kernel scope link src 10.0.100.100
```

```
$ systemd-resolve --status
```

```
Link 2 (enp0s3)
    Current Scopes: DNS
DefaultRoute setting: yes
    LLNMR setting: yes
MulticastDNS setting: no
    DNSOverTLS setting: no
    DNSSEC setting: no
    DNSSEC supported: no
    DNS Servers: 8.8.8.8
                8.8.4.4
```