

# Configuración básica de redes

Linux es un sistema operativo particularmente orientado a redes. La mayoría de los protocolos de redes modernos se encuentran implementados en él, y la mayoría de los servidores de aplicaciones de red corren hoy día Linux.

Este tema de la certificación versa sobre la configuración básica de red de un sistema Linux, con conexiones de tipo Ethernet y wifi en IPv4 e IPv6.

Para conectar una tarjeta de interfaz de red en una red IP hay que especificar, como mínimo, dos o tres parámetros: una dirección IP, una máscara de red y una pasarela por defecto (excepto si se trata de una red estrictamente local).

## 1. Direcciones IPv4 e IPv6

Existen dos versiones utilizadas del protocolo IP:

- ~ IPv4, la más antigua, con direcciones en 32 bits.
- ~ IPv6, con direcciones en 128 bits.

Un sistema Linux puede trabajar con las dos versiones del protocolo y tener una o varias direcciones para cada uno de los protocolos.

Por otra parte, independientemente de la versión del protocolo IP, puede haber distintas combinaciones entre dirección IP, interfaz de red y sistema:

- ~ Un sistema puede tener una interfaz de red y una dirección IP.
- ~ Un sistema puede tener distintas interfaces de red y distintas direcciones IP.
- ~ Una interfaz de red puede tener una sola dirección IP.
- ~ Una interfaz de red puede tener varias direcciones IP.
- ~ Distintas interfaces de red pueden utilizar solamente una dirección IP.



Tradicionalmente se usa el término dirección IP de host (host address). Este término puede llevar a confusión porque una máquina puede tener distintas interfaces de red, en distintas redes IP, y presentar, por lo tanto, direcciones IP de host diferentes. Se debería decir más bien dirección IP de interfaz de red, aunque, en algunos casos, distintas interfaces de red pueden usar la misma dirección IP.

## 2. Configuración básica de una conexión IPv4

### a. Red/subred

El protocolo IP (*Internet Protocol*) permite conectar distintas redes entre ellas. Para ello, es necesario disponer de equipos (de hardware o software) que estén integrados en distintas redes y sean capaces de transferir un paquete IP de una red a otra (función de enrutamiento).

Las redes IP están organizadas en tres clases, A, B y C, caracterizadas por el tamaño de su identificador de red: 1 byte para la clase A, 2 bytes para la clase B y 3 bytes para la clase C.

Dos redes IP pueden comunicarse entre ellas si están conectadas por, al menos, un router y atravesando, opcionalmente, una serie de redes y de routers intermediarios.

Cuando el número de redes interconectadas comenzó a ser importante, se tuvo que extender la noción de red para permitir que las organizaciones puedan dividir su red en conjuntos interconectados: las subredes. Estas son las reglas que aseguran la comunicación entre las subredes:

- ˘ Las subredes son transparentes para las otras redes, que solamente necesitan conocer el identificador de red y del host de destino en su red para comunicarse con él, sea cual sea su subred.
- ˘ Dos subredes de una misma red no pueden comunicarse entre ellas si no están conectadas por, al menos, un router, atravesando opcionalmente una serie de subredes y de routers intermediarios.

Para identificar las diferentes subredes, se usa una parte del identificador de red del host además de su identificador de red. Por lo tanto, leyendo una dirección IP no se puede saber a qué subred pertenece. Hay que configurar un dato suplementario: el número de bits de la parte de red/subred de la dirección. Este parámetro se llama **máscara de subred** (*subnet mask*).

En la documentación se pueden encontrar distintos términos para máscara de red (*net mask*) o máscara de subred (*subnet mask*). Son equivalentes, el primero es más antiguo y viene de la época en que las subredes se utilizaban muy poco.



Más generalmente, también se pueden agregar distintas redes entre ellas, para organizar una especie de subred, división lógica en redes. En ese caso se habla de direcciones CIDR (*Classless Inter Domain Routing*).

## b. Dirección IP

Se trata de la dirección IP clásica, en 32 bits. Está dividida en dos partes: la dirección de red/subred, seguida de la dirección del host. La dirección de red/subred identifica la red/subred en la que está integrado el host, la dirección de red de host identifica de manera única un elemento de red que está integrado en una red/subred.

El reparto de los 32 bits de la dirección de red entre la parte de red/subred y la parte host es variable y está definido por la máscara de subred.

Se puede especificar una dirección IPv4 usando la sintaxis siguiente:

`w.x.y.z[/NúmbitsMáscara]`

Se trata de cuatro valores enteros en notación decimal separados por un punto (notación decimal punteada), especificando la dirección de red propiamente dicha, seguidos, si fuera necesario, de un carácter / y del número de bits de la parte de red/subred, notación llamada CIDR (*Classless Internet Domain Routing*).

### c. Máscara de subred

La máscara de subred permite determinar la parte de la dirección de red que identifica la red/subred a la que pertenece la dirección de red.

Se puede escribir de dos maneras:

Notación clásica «máscara de subred» (*subnet mask*): todos los bits del identificador red/subred valen 1, los del identificador del host valen 0.

Notación CIDR: se indica el identificador de red/subred, todos los bits del identificador del host en cero, seguido de un / y del número de bits del identificador de red/subred.

#### Ejemplo

Para un identificador de red/subred en `10.1`:

Identificador de red/subred: `10.1.0.0` y máscara de subred: `255.255.0.0`

Notación CIDR: `10.1.0.0/16`

### d. Pasarela por defecto

Si la red/subred no está conectada a otras redes/subredes, no habrá pasarela por defecto. En caso contrario, la pasarela por defecto designa la dirección IP hacia la que se enviarán los paquetes IP dirigidos hacia otra red/subred. Si este parámetro no está especificado, el host no podrá comunicar a través de esta tarjeta de red con otros hosts de otras redes/subredes.

## 3. Configuración básica de una conexión IPv6

### a. Dirección IPv6

La versión 6 del protocolo IP tiene como objetivo aumentar las funcionalidades del protocolo y atenuar algunas de sus limitaciones. El paso a una dirección de 128 bits permite, en particular, responder al riesgo de escasez de direcciones IP en Internet.

## Representación de una dirección IPv6

Una dirección IPv6 tiene un tamaño de 128 bits, es decir 16 bytes. Se representa generalmente bajo la forma de ocho elementos de 2 bytes. El valor de cada elemento está expresado en hexadecimal, y cada elemento está separado por el carácter `:`.

Una simplificación en su escritura consiste en reemplazar una única serie de campos con cero en la dirección de red por dos caracteres de dos puntos seguidos `::`.

### Ejemplo

Observemos la dirección IPv6 de un servidor Google:

**host `www.google.com`**  
**`www.google.com` has address `216.58.215.36`**  
**`www.google.com` has IPv6 address `2a00:1450:4007:80c::2004`**

La dirección IPv6 mostrada corresponde a la dirección de red completa siguiente:

`2a00:1450:4007:080c:0000:0000:0000:2004`

## Estructura de una dirección IPv6

Una dirección IPv6 se compone de tres partes, de izquierda a derecha:

- ˆ Un prefijo, compuesto por 6 bytes.
- ˆ Un identificador de subred, compuesto por 2 bytes.
- ˆ Un identificador de host, compuesto por 8 bytes.

### Ejemplo

Observemos la dirección IPv6 de un servidor Google:

**host `www.google.com`**  
**`www.google.com` has address `216.58.215.36`**  
**`www.google.com` has IPv6 address `2a00:1450:4007:80c::2004`**

La dirección IPv6 presenta la siguiente estructura:

Prefijo: `2a00:1450:4007`

Identificador de subred: `80c`

Identificador de host: `0000:0000:0000:2004`

Los seis primeros bytes constituyen el prefijo de sitio, e identifican la red dentro de su red.

El campo siguiente, de dos 2 bytes, identifica una subred dentro de su red.

Los ocho últimos bytes identifican el host dentro de su subred.

También se usa esta división:

- ✓ Topología pública: está constituida por los seis primeros bytes de la dirección, está definida en relación a una autoridad externa que administra las redes (a menudo se trata del proveedor de acceso a la organización). Una dirección global de red, es decir, enrutable a través de las redes interconectadas, tiene un prefijo de red que empieza por 2 o 3.
- ✓ Topología privada: se trata de los diez últimos bytes de la dirección de red y están definidos por la organización responsable de la red interna, que gestiona la división en subredes (prefijo de subred) que reagrupan los hosts.

### Identificador de host

La parte del host de la dirección de red (llamada también *token* o identificador de interfaz), compuesta por ocho bytes, se puede definir de distintas maneras:

- ✓ Automáticamente a partir de la dirección de red MAC de la interfaz de red que corresponde al host.
- ✓ Manualmente, a partir del plan de direccionamiento IP definido por la red privada.

### Mapping de una dirección IPv4 en una dirección IPv6

Se puede generar una dirección IPv6 a partir de una dirección IPv4, con el objetivo de facilitar la integración de redes IPv4 en el interior de una IPv6. Esta técnica consiste en crear automáticamente una dirección IPv6, que corresponderá a una dirección IPv4.

Para ello se usa un prefijo particular, seguido de los cuatro bytes de la dirección IPv4:

Los 80 primeros bits de la dirección de red se fijan a 0, los 16 siguientes a 1, lo que nos da:

`::ffff:wx:yz` donde `w`, `x`, `y` y `z` representan el valor en hexadecimal de los 4 bytes de la dirección IPv4.



Para facilitar la lectura de la dirección de red mapeada, a menudo se representan los 4 bytes en notación decimal punteada, es decir: `::ffff:w.x.y.z`

### Dirección de sitio local

Esta dirección permite que las máquinas de una misma red puedan comunicar entre ellas. No se puede usar para comunicar con otras redes conectadas.

Está compuesta por dos partes:

- ✓ El prefijo de sitio local: en 10 bits, vale `FEC0`. Está seguido de 48 bits a cero.
- ✓ El identificador de subred: en 16 bits.
- ✓ El identificador de host: en 54 bits, también está codificado a partir de los 48 bits de la dirección de red MAC de la interfaz de red del host, usando el método EUI-64, descrito a continuación.

### Dirección de enlace local

Esta dirección se genera automáticamente y permite que las máquinas compartan un mismo enlace local para que se comuniquen entre ellas. Esta dirección no se puede utilizar para comunicar con otras redes/subredes interconectadas.

Está constituida por:

- ✓ El prefijo de enlace local: en 10 bits, con un valor `FE80`. Seguido de 64 bits a cero.
- ✓ El identificador de host: en 54 bits, generalmente está codificado a partir de los 48 bits de la dirección de red MAC de la interfaz de red del host, usando el método EUI-64.

## Dirección EUI-64

Una dirección EUI-64 (EUI, *Extended Unique Identifier*), es un método de generación de una dirección IPv6 única dentro de una red local (dirección de enlace local), por la definición de un prefijo común y de un identificador de host en 64 bits.

Para generar la dirección de red, se toma la dirección de red MAC de la interfaz, en 6 bytes:

a.b.c.d.e.f

Se usa el prefijo de enlace local en de 64 bits: FE80:0000:0000:0000

Para el identificador de host en 8 bytes, se toma, de izquierda a derecha:

- ˆ Los tres primeros bytes de la dirección de red MAC: a.b.c
- ˆ Un valor fijo de dos bytes: FF.FE
- ˆ Los tres últimos bytes de la dirección de red MAC: d.e.f

Se obtiene: a.b:c.FF:FE.d:e.f

Finalmente, se le añade 1 al valor del séptimo bit (empezando por la izquierda) del primer byte de la dirección de red MAC (el byte a).

### Ejemplo

Dirección de enlace local de una interfaz de red Ethernet:

Dirección MAC:

00:1b:24:6a:78:14

Dirección IPv6:

fe80::21b:24ff:fe6a:7814

El prefijo es fe80::.

El primer byte de la dirección de red MAC vale 0. Con el séptimo bit (de valor  $2^1$ ), 0 al que se le añade 1, vale 2. Después están los dos bytes siguientes de la dirección de red MAC, lo que nos da 021b:24.



Después vienen los dos bytes convencionales **FF** y **FE**, seguidos de los tres últimos bytes de la dirección de red MAC: **6a**, **78** y **14**.

### Dirección de loopback

La dirección de red de loopback IPv6, reservada para uso interno, presenta todos los bits a cero, excepto el último bit de peso menos significativo, es decir:

**0000:0000:0000:0000:0000:0000:0000:0001**

Se denota de la manera siguiente:

**::1**

### b. Máscara de subred

Una dirección IPv6 contiene un campo específicamente asignado al identificador de subred: los dos bytes que siguen a los 6 bytes del prefijo de red.

El prefijo de red/subred está constituido por 64 bits, excepto en los casos particulares.

Para especificar el número de bits que lo componen, se usa la notación CIDR (*Classless Inter Domain Routing*):

**Identificador/NúmeroBitsID**

#### Ejemplo

*Configuración IPv6 de una interfaz de red:*

**inet6 2a01:e35:2439:1510:c56c:74cd:b67e:eb07/64**

La máscara está especificada con una longitud de prefijo de 64 bits.

### c. Pasarela por defecto

La pasarela por defecto tiene la misma función en IPv6 y en IPv4, determina el nodo de red por el que se deberá de transferir el paquete cuyo destinatario no forme parte de la subred del emisor y para la red/subred que no sea conocida a través de ruta específica declarada en el emisor.

#### Ejemplo

Ruta por defecto de una interfaz de red en IPv6:

```
default via fe80::f6ca:e5ff:fe44:8658
```

La ruta por defecto está asociada a una dirección de enlace local (prefijo en **fe80**) y designa un nodo en la subred, que debe ser capaz de enrutar los paquetes hacia otras redes/subredes. Generalmente se detecta y configura automáticamente.

## 4. Configurar dinámicamente una interfaz de red Ethernet

La configuración de los parámetros mínimos para conectar una interfaz de red Ethernet en una red IP, dirección IP, máscara de subred y, si fuera necesario, pasarela por defecto, se puede hacer usando distintos comandos, capaces de administrar las dos versiones del protocolo, IPv4 e IPv6.

Estos comandos permiten modificar **dinámicamente** la configuración de las tarjetas de interfaz de red, estas modificaciones se perderán al reiniciar el sistema o la configuración de redes.



Algunos de estos comandos, históricamente procedentes del mundo Unix, están considerados como en vía de obsolescencia y no siempre están presentes por defecto en las versiones recientes de las distribuciones. Sin embargo, es preferible conocerlos porque todavía se utilizan mucho en producción y porque la certificación LPIC-2 los tiene en cuenta.

### a. Nombre de una interfaz de red

En las versiones recientes de las distribuciones Linux, se ha hecho una evolución en la denominación de las interfaces de red. Se trata de un nuevo método, llamado ifnames, que permite realizar denominaciones previsibles (*Predictable Network Interface Device Name*).

El antiguo método atribuía nombres genéricos simples (`eth0`, `eth1`, etc.) pero inestables, porque dependían del orden de detección de los periféricos durante el arranque del sistema.

El nuevo método se basa en distintos criterios (la configuración BIOS, el tipo de interfaz, el tipo de bus, la posición en el bus, etc.), para determinar un nombre que sea lo más estable posible. El lado negativo es que este nombre es mucho menos intuitivo, por ejemplo `enp38s0` o `wl01`.

Las reglas de denominación inicial son las siguientes:

Dos caracteres para el tipo de interfaz:

en	Ethernet.
wl	Inalámbrica local (Wi-Fi).
ww	Inalámbrica extendida.

Después se encuentra un conjunto que determina la unicidad del nombre de la interfaz con

respecto a las otras interfaces del mismo tipo. Podemos encontrar:

**o**Núm

Número de la tarjeta con respecto a las otras tarjetas del mismo tipo.

sUbicación [fFunción][dDevId]

Número de ubicación para una interfaz en un bus PCI Express, seguido opcionalmente del número de función y de un número de tarjeta.

xMACadr

Identificador para la dirección de red MAC (por defecto, este tipo de nombre no se utiliza).

[Pdominio]pNúmBussUbicación [fFunción][dDevId]

Para un periférico en un bus PCI.

[Pdominio]pNumBussUbicación [fFunción][uPuerto][.][cConfig][iInterfaz]

Para un periférico en una cadena de hubs USB.



Estos nombres pueden modificarse a través de `udev`, configurando reglas `udev` específicas para la denominación de las interfaces.

## b. El comando ip

Se trata del comando estándar de configuración dinámica de las interfaces de red de tipo

Ethernet, para conexiones en una red IP, versión 4 y versión 6. Dispone de muchos subcomandos para visualizar y configurar los diferentes niveles de atributos (llamados objetos) de una interfaz, según diferentes niveles del protocolo, incluyendo los datos de enrutamiento.



Este comando reemplaza el antiguo comando `ifconfig`, en vía de obsolescencia.

### Sintaxis

ip [Opciones] Objeto SubComando ArgSubComando

### Parámetros principales

<code>-f Proto</code>	Familia de protocolo que se utilizará: inet, inet6...
<code>-4</code>	Modo IPv4.
<code>-6</code>	Modo IPv6.
<code>-br</code>	Vista condensada.
<code>-h</code>	Vista fácil de leer para un humano.
<code>-s</code>	Vista de las estadísticas complementarias.
<code>-s</code>	Vista detallada.
<code>-r</code>	Muestra los nombres de los hosts en lugar de las direcciones.
Objeto	Tipo de objeto.
SubComando	Subcomando que se ejecutará.
ArgSubComando	Argumentos del subcomando.

### Descripción

Las opciones indicadas se aplican a todos los objetos y a sus subcomandos. Un objeto es un atributo de una tarjeta de interfaz de red: dirección de enlace, dirección IP, enrutamiento, etc.

### Ejemplo

El objeto `link` y el comando `show` permiten mostrar la lista de las interfaces de red

conocidas por el sistema, con sus características de nivel de enlace:

#### **ip link show**

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode
DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp38s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
UP mode DEFAULT group default qlen 1000
    link/ether e4:11:5b:50:13:32 brd ff:ff:ff:ff:ff:ff
3: wlo1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
mode DORMANT group default qlen 1000
    link/ether 9c:b7:0d:bb:2b:67 brd ff:ff:ff:ff:ff:ff
```

Este sistema dispone de la pseudo interfaz de loopback `lo`, así como de dos tarjetas de interfaz de red: `enp38s0` (Ethernet) y `wlo1` (Wi-Fi), ambas activas (`state UP`).

### **c. Gestionar las direcciones IP: ip address**

El comando `ip` con el objeto `address` permite configurar dinámicamente la dirección de una tarjeta de interfaz de red.

#### Sintaxis

```
ip [Opciones] [ address|a|addr ] [ SubComando ] [ Arg SubComando ]
```

#### Parámetros principales

SubComando	Subcomando que se ejecutará.
ArgSubComando	Argumentos del subcomando.

#### Descripción

Este subcomando permite administrar (crear, mostrar, modificar y suprimir) una dirección IPv4 o IPv6 asignada a una tarjeta de interfaz de red. Una tarjeta puede tener una dirección IPv4 y/o IPv6. También puede tener varias direcciones en la misma versión del protocolo.

Sin subcomando, muestra la configuración de todas las tarjetas de interfaz de red.

Las direcciones especificadas tienen que seguir la sintaxis usual IPv4 o IPv6:

- w.x.y.z[/LgMáscara]

Notación decimal para las direcciones IPv4 (y si fuera necesario con la notación CIDR para la máscara de subred, si es diferente de la determinada por la clase).

- [0 a FFFF]: [0 a FFFF]: [...] [/LgPrefijo]

Notación hexadecimal, completa o simplificada, para las direcciones IPv6 (opcionalmente seguida de la longitud del prefijo).

Los principales subcomandos y argumentos son:

`show [ dev Interfaz ]`

Muestra la configuración de la tarjeta de interfaz de red `Interfaz`, o de todas las tarjetas de interfaz de red. Es el subcomando por defecto.

`add Dirección dev Interfaz`

Añade la dirección IPv4 o IPv6 `Dirección` a la tarjeta de `interfaz` de red `Interfaz`.

`delete dev Interfaz [ Dirección ]`

Suprime la dirección de red `Dirección` de la tarjeta de interfaz de red `Interfaz`, o, sin dirección, de su primera dirección.



flush dev Interfaz [ Arg ]

Suprime las direcciones de la tarjeta de interfaz de red **Interfaz** , todas ellas o según el criterio **Arg** especificado.

### Ejemplos

Mostrar la información de dirección de todas las tarjetas de interfaz de red:

#### **ip a 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: enp38s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
    link/ether e4:11:5b:50:13:32 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.4/24 brd 192.168.0.255 scope global dynamic noprefixroute enp38s0
        valid_lft 24479sec preferred_lft 24479sec
    inet6 fe80::5508:86a9:e923:75b3/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: wlo1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group
default qlen 1000
    link/ether 9c:b7:0d:bb:2b:67 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.5/24 brd 192.168.0.255 scope global dynamic noprefixroute wlo1
        valid_lft 24481sec preferred_lft 24481sec
    inet6 2a01:e35:2439:1510:be93:7dd3:8725:3060/64 scope global dynamic noprefixroute
        valid_lft 86025sec preferred_lft 86025sec
    inet6 fe80::7f4c:37c9:66fd:4292/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
4: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
group default qlen 1000
    link/ether 52:54:00:a2:5f:41 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
        valid_lft forever preferred_lft forever
5: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel master virbr0 state
```

```
DOWN group default qlen 1000
link/ether 52:54:00:a2:5f:41 brd ff:ff:ff:ff:ff:ff
```

Parece que el sistema operativo tiene varias tarjetas de interfaz. En realidad, dispone de una tarjeta Ethernet `enp38s0` (n° 2) y de una interfaz Wi-Fi `wlo1` (n° 3). Cada una gestiona dos direcciones útiles en IPv4 e IPv6. La interfaz n° 1, `lo`, es una pseudo interfaz de loopback interna. Finalmente, las dos, interfaces n° 4 y 5, `virbr0` y `virbr0-nic` solamente son usadas por programas de virtualización (han sido creadas por la biblioteca `libvirt`).

Para visualizar solamente la interfaz de red `enp38s0`:

```
ip a show enp38s0
2: enp38s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
default qlen 1000
link/ether e4:11:5b:50:13:32 brd ff:ff:ff:ff:ff:ff
inet 192.168.0.4/24 brd 192.168.0.255 scope global dynamic noprefixroute enp38s0
    valid_lft 23875sec preferred_lft 23875sec
inet6 fe80::5508:86a9:e923:75b3/64 scope link noprefixroute
    valid_lft forever preferred_lft forever
```

El comando muestra las características de la tarjeta de interfaz de red, activa (`state UP`), entre ellas:

su dirección MAC: `e4:11:5b:50:13:32`

sy dirección IPv4 CIDR: `192.168.0.4/24`

su dirección IPv6 CIDR: `fe80::5508:86a9:e923:75b3/64` (dirección de enlace local)

Se puede usar el formato de visualización abreviado con la opción `-br`:

```
ip -br a show enp38s0
enp38s0    UP    192.168.0.4/24
fe80::5508:86a9:e923:75b3/64
```

La tarjeta está activa y tiene dos direcciones IP, una en versión 4 y la otra en versión 6.

Definir una nueva dirección IPv4:

Una tarjeta de interfaz de red puede tener distintas direcciones, IPv4 y/o IPv6, y podemos querer añadir la dirección IPv4 `10.1.0.5` a una tarjeta de interfaz de red Ethernet, con una máscara de subred de 16 bits (subred 1 de la red `10`).

Se muestran las interfaces Ethernet disponibles y sus direcciones IPv4:

```
ip -br -4 a
lo          UNKNOWN    127.0.0.1/8
enp38s0     UP          192.168.0.4/24
wlo1       UP          192.168.0.5/24
```

Se añade la dirección de red deseada a la tarjeta Ethernet:

```
ip a add 10.1.0.5/16 dev enp38s0
```

Se hace una comprobación:

```
ip -br -4 a show dev enp38s0
enp38s0     UP          192.168.0.4/24 10.1.0.5/16
```

A partir de ahora, la tarjeta tiene dos direcciones IPv4.

Se comprueba una comunicación hacia una dirección existente en la subred `10.1`:

```
ping 10.1.0.39
PING 10.1.0.39 (10.1.0.39) 56(84) bytes of data.
64 bytes from 10.1.0.39: icmp_seq=1 ttl=64 time=0.401 ms
64 bytes from 10.1.0.39: icmp_seq=2 ttl=64 time=0.192 ms
64 bytes from 10.1.0.39: icmp_seq=3 ttl=64 time=0.228 ms
64 bytes from 10.1.0.39: icmp_seq=4 ttl=64 time=0.220 ms
^C
--- 10.1.0.39 ping statistics ---
```

4 packets transmitted, 4 received, 0% packet loss, time 72ms  
rtt min/avg/max/mdev = 0.192/0.260/0.401/0.083 ms

Una vez hecho el test, se decide suprimir esta dirección:

**ip a del 10.1.0.5/16 dev enp38s0**

Comprobamos las direcciones de la tarjeta de red:

```
ip -br -4 a show dev enp38s0
enp38s0      UP      192.168.0.4/24
```

#### d. Gestionar las direcciones IP: ifconfig

Este comando también permite administrar dinámicamente la configuración de la dirección de una tarjeta de interfaz de red.



Este comando ya no está instalado por defecto en las versiones recientes de las distribuciones de tipo Red Hat o Debian, forma parte del paquete `net-tools`.

#### Sintaxis

```
ifconfig [-v | -a | -s] [Interfaz]
```

o

```
ifconfig [-v] Interfaz [IDproto] Opciones | Dirección ...
```

Parámetros principales

<code>-v</code>	Vista detallada.
<code>-a</code>	Todas las interfaces, activas o no.
<code>-s</code>	Vista de las estadísticas de utilización.
<code>Interfaz</code>	Tarjeta de interfaz de red.
<code>IDproto</code>	Tipo de protocolo.
<code>Opciones</code>	Opciones de configuración.
<code>Dirección</code>	Dirección IPv4 o IPv6.

Descripción

En la primera sintaxis, el comando muestra la información de todas las interfaces activas (por defecto) o la de la interfaz especificada.

La segunda sintaxis permite asignar una dirección, IPv4 o IPv6, añadir una nueva, modificar una dirección existente o suprimirla.

Los usos más corrientes son los siguientes:

- Asignar una dirección a la tarjeta Interfaz:

```
ifconfig Interfaz[:Ident] Dirección [netmask Máscara]
```

Para IPv4, la dirección de red debe estar representada en notación decimal punteada. Si la máscara de subred no es la de la clase de la dirección de red, hay que especificarla

con el argumento `netmask Máscara` .

Para IPV6, la dirección de red tiene que estar representada en notación hexadecimal (completa o simplificada) (opcionalmente seguida de la longitud del prefijo).

Si la tarjeta ya tenía una dirección de la misma versión IP, será remplazada por la nueva. Si una instancia de tarjeta ha sido especificada con el sufijo: `Ident` , será creada si no existe, en el caso contrario, su dirección de la misma versión IP será modificada.

~ Añadir una dirección a la tarjeta `Interfaz` :

```
ifconfig Interfaz[:Ident] add Dirección [netmask Máscara]
```

En IPv4, el comando crea una nueva instancia de la interfaz, identificada por el sufijo: `Ident` (por defecto, un valor entero), asociada a la dirección de red especificada.

En IPv6 las direcciones están asociadas directamente a la interfaz.

~ Suprimir la dirección de red `Dirección` de la instancia de tarjeta `Interfaz [ : Ident ]` :

```
ifconfig Interfaz[:Ident] del Dirección
```

## Ejemplos

Mostrar la información de dirección de todas las tarjetas de interfaz de red:

```
ifconfig
enp0s10: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.39 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::21b:24ff:fe6a:7814 prefixlen 64 scopeid 0x20<link>
    inet6 2a01:e35:2439:1510:21b:24ff:fe6a:7814 prefixlen 64 scopeid
0x0<global>
    inet6 2a01:e35:2439:1510:7f:f2b5:c91b:7891 prefixlen 64 scopeid
0x0<global>
```

```

ether 00:1b:24:6a:78:14 txqueuelen 1000 (Ethernet)
RX packets 471 bytes 61908 (60.4 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 280 bytes 33934 (33.1 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

```

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 1000 (Boucle locale)
RX packets 8 bytes 396 (396.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 8 bytes 396 (396.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

El sistema dispone de una tarjeta Ethernet `enp0s10` y de la pseudo interfaz de loopback interna, `lo`.

La tarjeta Ethernet dispone de una dirección IPv4, `192.168.0.39`, con una máscara de subred de 3 bytes. Tiene varias direcciones IPv6, una de enlace local (`scopeid link`) y dos globales (`scopeid global`).

Modificar una dirección IPv4:

Se quiere reemplazar la dirección IPv4 existente de la tarjeta de interfaz de red Ethernet por la dirección de red `192.168.0.38`, con una máscara de subred de 3 bytes.

```
ifconfig enp0s10 192.168.0.38 netmask 255.255.255.0
```

Se comprueba:

```

ifconfig enp0s10
enp0s10: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.0.38 netmask 255.255.255.0 broadcast 192.168.0.255
inet6 2a01:e35:2439:1510:6d68:6c8b:8d6c:b9f7 prefixlen 64 scopeid
0x0<global>

```

```

inet6 fe80::21b:24ff:fe6a:7814 prefixlen 64 scopeid 0x20<link>
inet6 2a01:e35:2439:1510:21b:24ff:fe6a:7814 prefixlen 64 scopeid
0x0<global>
ether 00:1b:24:6a:78:14 txqueuelen 1000 (Ethernet)
RX packets 3242 bytes 352229 (343.9 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 1293 bytes 158636 (154.9 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

La dirección IPv4 existente ha sido reemplazada por la nueva dirección.

Añadir una dirección IPv4:

Se quiere añadir la dirección IPv4 `10.1.0.39` a la tarjeta de interfaz de red Ethernet, con una máscara de subred de 16 bits (subred `1` de la red `10`).

```
ifconfig enp0s10 add 10.1.0.39 netmask 255.255.0.0
```

Comprobamos:

```

ifconfig
enp0s10: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.38 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 2a01:e35:2439:1510:6d68:6c8b:8d6c:b9f7 prefixlen 64 scopeid
0x0<global>
    inet6 fe80::21b:24ff:fe6a:7814 prefixlen 64 scopeid 0x20<link>
    inet6 2a01:e35:2439:1510:dc52:f954:1f3b:77ad prefixlen 64 scopeid
0x0<global>
    inet6 2a01:e35:2439:1510:21b:24ff:fe6a:7814 prefixlen 64 scopeid
0x0<global>
    ether 00:1b:24:6a:78:14 txqueuelen 1000 (Ethernet)
    RX packets 6984 bytes 764441 (746.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2921 bytes 414962 (405.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s10: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

```



```

inet 10.1.0.39 netmask 255.255.255.0 broadcast 192.168.0.255
ether 00:1b:24:6a:78:14 txqueuelen 1000 (Ethernet)
[...]
```

Ahora la tarjeta tiene dos direcciones IPv4. El comando ha creado una nueva instancia de tarjeta, con sufijo:0.

Se hace un test de comunicación hacia una dirección existente en la subred 10.1:

```

ping 10.1.0.4
PING 10.1.0.4 (10.1.0.4) 56(84) bytes of data.
64 bytes from 10.1.0.4: icmp_seq=1 ttl=64 time=0.481 ms
64 bytes from 10.1.0.4: icmp_seq=2 ttl=64 time=0.248 ms
64 bytes from 10.1.0.4: icmp_seq=3 ttl=64 time=0.352 ms
64 bytes from 10.1.0.4: icmp_seq=4 ttl=64 time=0.361 ms
^C
--- 10.1.0.4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 71ms
rtt min/avg/max/mdev = 0.248/0.360/0.481/0.084 ms
```

Una vez que la prueba ha sido efectuada, se decide suprimir esta dirección:

**ifconfig enp0s10 del 10.1.0.39**

Comprobamos:

```

ifconfig
enp0s10: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.38 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 2a01:e35:2439:1510:6d68:6c8b:8d6c:b9f7 prefixlen 64 scopeid
0x0<global>
    inet6 fe80::21b:24ff:fe6a:7814 prefixlen 64 scopeid 0x20<link>
    inet6 2a01:e35:2439:1510:dc52:f954:1f3b:77ad prefixlen 64 scopeid
0x0<global>
    inet6 2a01:e35:2439:1510:21b:24ff:fe6a:7814 prefixlen 64 scopeid
```

```

0x0<global>
  ether 00:1b:24:6a:78:14 txqueuelen 1000 (Ethernet)
  RX packets 7119 bytes 777995 (759.7 KiB)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 2963 bytes 421595 (411.7 KiB)
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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 1000 (Boucle locale)
  RX packets 11 bytes 852 (852.0 B)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 11 bytes 852 (852.0 B)
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

La instancia `:0` ha sido suprimida.

Añadir una dirección IPv6:

Se quiere añadir la dirección IPv6 `fe80::21b:24ff:fe6a:ff01/64` a la tarjeta de interfaz de red Ethernet.

```
ifconfig enp0s10 inet6 add fe80::21b:24ff:fe6a:ff01/64
```

Se comprueba:

```

ifconfig enp0s10
enp0s10: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  inet 192.168.0.38 netmask 255.255.255.0 broadcast 192.168.0.255
  inet6 fe80::21b:24ff:fe6a:ff01 prefixlen 64 scopeid 0x20<link>
  inet6 2a01:e35:2439:1510:6d68:6c8b:8d6c:b9f7 prefixlen 64 scopeid
0x0<global>
  inet6 fe80::21b:24ff:fe6a:7814 prefixlen 64 scopeid 0x20<link>
  inet6 2a01:e35:2439:1510:dc52:f954:1f3b:77ad prefixlen 64 scopeid
0x0<global>
  inet6 2a01:e35:2439:1510:21b:24ff:fe6a:7814 prefixlen 64 scopeid

```

```
0x0<global>
ether 00:1b:24:6a:78:14 txqueuelen 1000 (Ethernet)
RX packets 7460 bytes 817464 (798.3 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 3009 bytes 429647 (419.5 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Se ha añadido la dirección de red directamente a la tarjeta de interfaz de red, sin crear una nueva instancia.

Se comprueba la dirección de red desde otra máquina:

```
ping6 fe80::21b:24ff:fe6a:ff01
PING fe80::21b:24ff:fe6a:ff01(fe80::21b:24ff:fe6a:ff01) 56 data bytes
64 bytes from fe80::21b:24ff:fe6a:ff01%enp38s0: icmp_seq=1 ttl=64 time=0.666 ms
64 bytes from fe80::21b:24ff:fe6a:ff01%enp38s0: icmp_seq=2 ttl=64 time=0.311 ms
64 bytes from fe80::21b:24ff:fe6a:ff01%enp38s0: icmp_seq=3 ttl=64 time=0.230 ms
64 bytes from fe80::21b:24ff:fe6a:ff01%enp38s0: icmp_seq=4 ttl=64 time=0.266 ms
^C
--- fe80::21b:24ff:fe6a:ff01 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 34ms
rtt min/avg/max/mdev = 0.230/0.368/0.666/0.174 ms
```

Se suprime esta dirección:

```
ifconfig enp0s10 inet6 del fe80::21b:24ff:fe6a:ff01/64
```

Se reestablece la dirección IPv4 inicial:

```
ifconfig enp0s10 192.168.0.39 netmask 255.255.255.0
```

Se comprueba la configuración de la tarjeta:

**ifconfig enp0s10**

```

enp0s10: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.39 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 2a01:e35:2439:1510:6d68:6c8b:8d6c:b9f7 prefixlen 64 scopeid
0x0<global>
    inet6 fe80::21b:24ff:fe6a:7814 prefixlen 64 scopeid 0x20<link>
    inet6 2a01:e35:2439:1510:dc52:f954:1f3b:77ad prefixlen 64 scopeid
0x0<global>
    inet6 2a01:e35:2439:1510:21b:24ff:fe6a:7814 prefixlen 64 scopeid
0x0<global>
    ether 00:1b:24:6a:78:14 txqueuelen 1000 (Ethernet)
    RX packets 7669 bytes 837843 (818.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3089 bytes 442869 (432.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

**e. Configurar la pasarela por defecto: ip route**

El comando `ip` con el objeto `route` permite configurar dinámicamente el enrutamiento de las tarjetas de interfaz de red.

Sintaxis

```
ip [ -6 ] [ Opciones ] [ route|r ] [ SubComando ] [ ArgSubComando ]
```

Parámetros principales

<code>-6</code>	Para IPv6.
<code>Opciones</code>	Opciones.
<code>SubComando</code>	Subcomando que se ejecutará.
<code>ArgSubComando</code>	Argumentos del subcomando.

Descripción

Este subcomando permite administrar (crear, mostrar, modificar y suprimir) la tabla de enrutamiento del sistema, en IPv4 y en IPv6.

Sin subcomando, muestra la tabla de enrutamiento IPv4.

Para visualizar o modificar la tabla de enrutamiento IPv6, hay que usar la opción `-6`.

Los principales subcomandos y argumentos para administrar la pasarela por defecto son:

Protocolo IPv4:

```
add default via Dirección [ dev Interfaz ]
```

Añade como pasarela por defecto la dirección de red `Dirección`, especificando opcionalmente qué tarjeta de interfaz de red se debe usar para alcanzarla.

```
show default
```

Muestra la pasarela por defecto.

```
change default via Dirección [ dev Interfaz ]
```

Modifica la pasarela por defecto.

```
delete default
```

Suprime la pasarela por defecto.

Protocolo IPv6:

Los comandos son los mismos, pero hay que especificar la opción `-6`. En general, la pasarela por defecto se detecta automáticamente en el enlace local y no es necesario configurarla.

Ejemplos

Gestión de la pasarela por defecto IPv4:

Vista de la pasarela por defecto:

```
ip r show default
default via 192.168.0.254 dev enp0s10 proto dhcp metric 100
```

Configurar la pasarela por defecto en un sistema dónde no está configurada:

```
ip r show default
```

Ninguna pasarela definida.

Se define la pasarela por defecto:

```
ip r add default via 192.168.0.254
ip r show default
default via 192.168.0.254 dev enp0s10
```

Para modificar la pasarela existente, hay que usar el subcomando `change`:

```
ip r change default via 192.168.0.250
ip r show default
default via 192.168.0.250 dev enp0s10
```

Supresión de la pasarela por defecto:

```
ip r delete default
ip r show default
```

No hay ninguna pasarela definida.

Gestión de la pasarela por defecto IPv6:

Vista de la pasarela por defecto:

```
ip -6 r show default
default via fe80::f6ca:e5ff:fe44:8658 dev enp0s10 proto ra metric 100 pref medium
```

Configurar la pasarela por defecto IPv6, en un sistema donde no está configurada:

```
ip -6 r show default
```

No hay ninguna pasarela definida.

Se define la pasarela por defecto:

```
ip -6 r add default via fe80::f6ca:e5ff:fe44:8658 dev enp0s10
ip -6 r show default
default via fe80::f6ca:e5ff:fe44:8658 dev enp0s10 metric 1024 pref medium
```

Se ha definido la pasarela por defecto (el argumento dev Interfaz es obligatorio en IPv6).

Para modificar la pasarela existente, hay que usar el subcomando `change`:

```
ip -6 r change default via fe80::f6ca:e5ff:fe44:ff01 dev enp0s10
ip -6 r show default
default via fe80::f6ca:e5ff:fe44:ff01 dev enp0s10 metric 1024 pref medium
```

Supresión de la pasarela por defecto:

```
ip -6 r delete default
ip -6 r show default
```

Ya no hay ninguna pasarela IPv6 definida.

## f. Configurar la pasarela por defecto: route

El comando `route` permite administrar dinámicamente la tabla de enrutamiento IPv4 e IPv6.



Este comando ya no está instalado por defecto en las versiones recientes de las distribuciones de tipo Red Hat o Debian, forma parte del paquete `net-tools`.

### Sintaxis

`route [ Opciones ] [ SubComando ] [ ArgSubComando ]`

### Parámetros principales

Opciones	Opciones.
SubComando	Subcomando que se ejecutará
ArgSubComando	Argumentos del subcomando.

### Descripción

Este comando permite administrar (crear, mostrar, modificar y suprimir) la tabla de enrutamiento del sistema, en IPv4 y en IPv6.

Sin subcomando, muestra la tabla de enrutamiento IPv4.

Para visualizar o modificar la tabla de enrutamiento IPv6, hay que usar la opción `-6` o `-A inet6`.



Por defecto, el comando intenta usar los nombres de máquina o los nombres lógicos antes que las direcciones. La opción `-n` fuerza el uso de los valores numéricos (direcciones).

En este caso, la dirección de red correspondiente al destino por defecto es `0.0.0.0 ([::]/0` en IPv6).

Los principales subcomandos y argumentos para administrar la pasarela por defecto son:

Protocolo IPv4:

```
add default gw Dirección
```

Añade como pasarela por defecto la dirección de red `Dirección`.

```
del default
```

Suprime la pasarela por defecto.

Protocolo IPv6:

Los comandos son los mismos, pero hay que especificar la opción `-6` o `-A inet6`. En general, la pasarela por defecto es detectada automáticamente en el enlace local y no necesita ser configurada.

### Ejemplos

*Gestión de la pasarela por defecto IPv4:*

*Vista de la tabla de enrutamiento:*

```
route -n
Kernel IP routing table
Destination  Gateway      Genmask      Indic Metric Ref  Use Iface
0.0.0.0      192.168.0.254 0.0.0.0      UG    0    0    0 enp0s10
169.254.0.0  0.0.0.0      255.255.0.0  U     1000 0    0 enp0s10
192.168.0.0  0.0.0.0      255.255.255.0 U     100 0    0 enp0s10
```

La pasarela por defecto tiene la dirección `192.168.0.254`.

Se define la pasarela por defecto para un sistema que no presenta ninguna:

**route -n**

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
169.254.0.0	0.0.0.0	255.255.0.0	U	1000	0	0	enp0s10
192.168.0.0	0.0.0.0	255.255.255.0	U	100	0	0	enp0s10

**route add default gw 192.168.0.251**

**route -n**

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	192.168.0.251	0.0.0.0	UG	0	0	0	enp0s10
169.254.0.0	0.0.0.0	255.255.0.0	U	1000	0	0	enp0s10
192.168.0.0	0.0.0.0	255.255.255.0	U	100	0	0	enp0s10

Para modificar la pasarela existente, hay que suprimirla y volverla a crear:

**route del default**

**route add default gw 192.168.0.254**

**route -n**

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	192.168.0.254	0.0.0.0	UG	0	0	0	enp0s10
169.254.0.0	0.0.0.0	255.255.0.0	U	1000	0	0	enp0s10
192.168.0.0	0.0.0.0	255.255.255.0	U	100	0	0	enp0s10

Vista de la pasarela por defecto IPv6:

Vista de la tabla de enrutamiento:

**route -6n**

Kernel IPv6 routing table

Destination	Next Hop	Flag	Met	Ref	Use	If
-------------	----------	------	-----	-----	-----	----

```

::1/128          ::          U  256 2   0 lo
2a01:e35:2439:1510::/64  ::          U  100 1   0 enp0s10
fe80::/64        ::          U  100 1   0 enp0s10
::/0             fe80::f6ca:e5ff:fe44:8658 UG  100 2   0 enp0s10
::1/128          ::          Un  0 5   0 lo
2a01:e35:2439:1510:21b:24ff:fe6a:7814/128::      Un  0 3   0 enp0s10
2a01:e35:2439:1510:61f1:d3:f89f:e47a/128::      Un  0 3   0 enp0s10
fe80::21b:24ff:fe6a:7814/128 ::          Un  0 3   0 enp0s10
ff00::/8         ::          U  256 4   0 enp0s10
::/0             ::          !n  -1 1   0 lo

```

La pasarela por defecto tiene como dirección de enlace local `fe80::f6ca:e5ff:fe44:8658`. Es detectada automáticamente.

## g. ping y ping6

Estos comandos usan el protocolo ICMP (*Internet Control Message Protocol*) y permiten efectuar pruebas elementales de conectividad hacia hosts IPv4 o IPv6 remotos.



En la medida en que las máquinas remotas pueden estar configuradas para no contestar a las solicitudes de echo ICMP, hay que ser prudente en cuanto a las conclusiones que se puedan obtener de la ausencia de respuesta al uso de estos comandos.

### Sintaxis

```
ping[6] [-6] [-c Núm] [-i Intervalo] [-I interfaz] DirecciónIP|NombreDest
```

### Parámetros principales

<code>-6</code>	IPv6.
<code>-v</code>	Vista detallada.
<code>-c Nœm</code>	Número de mensajes que se enviarán.
<code>-i Intervalo</code>	Número de segundos entre dos emisiones.
<code>-I interfaz</code>	Interfaz de red que se utilizará.
<code>-b</code>	Modo broadcast.
<code>DirecciónIP   NombreDest</code>	Dirección IP o nombre del host de destino.

### Descripción

Por defecto, el comando envía una solicitud de echo ICMP por segundo hacia la dirección de red o la máquina especificada. Este comando se termina con [Ctrl] /C y muestra las estadísticas de envío/recepción.

La dirección de red de destino puede ser una dirección de host (*unicast*) o, usando la opción `-b`, una dirección de destino de una subred o de una red (*broadcast*).

El o los hosts de destino configurados para responder a las solicitudes de echo ICMP responden dando su propia dirección IP.



El comando `ping6` es equivalente a `ping -6`.

### Ejemplo

*Solicitud ICMP para un envío hacia una dirección existente:*

```
ping -c 1 192.168.0.39  
PING 192.168.0.39 (192.168.0.39) 56(84) bytes of data.  
64 bytes from 192.168.0.39: icmp_seq=1 ttl=64 time=0.193 ms  
--- 192.168.0.39 ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 0.193/0.193/0.193/0.000 ms
```

*Solicitud ICMP para un envío hacia una dirección inactiva (o que no responde a las solicitudes ICMP):*

```
ping -c 1 192.168.0.100  
PING 192.168.0.100 (192.168.0.100) 56(84) bytes of data.  
From 192.168.0.4 icmp_seq=1 Destination Host Unreachable  
--- 192.168.0.100 ping statistics ---  
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms
```

*Solicitud ICMP para envío hacia un host de otra red/subred:*

```
ping -c 1 www.google.com  
PING www.google.com(par21s11-in-x04.1e100.net (2a00:1450:4007:80c::2004))  
56 data bytes  
64 bytes from par21s11-in-x04.1e100.net (2a00:1450:4007:80c::2004):  
icmp_seq=1 ttl=57 time=10.4 ms  
--- www.google.com ping statistics ---  
1 packets transmitted, 1 received, 0% packet loss, time 0ms  
rtt min/avg/max/mdev = 10.357/10.357/10.357/0.000 ms
```

*Solicitud ICMP para envío, hacia una subred desconocida por parte del router:*

```
ping -c 1 192.168.1.1  
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.  
From 192.168.0.254 icmp_seq=1 Destination Host Unreachable  
--- 192.168.1.1 ping statistics ---  
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms
```

El router responde a la solicitud ICMP con un mensaje de error ICMP: *Destination Host Unreachable*.

Solicitud, en IPv6, hacia un host de otra red/subred:

```
ping6 -c 1 www.centos.org
PING www.centos.org(2001:4de0:aaae::201 (2001:4de0:aaae::201)) 56 data bytes
64 bytes from 2001:4de0:aaae::201 (2001:4de0:aaae::201): icmp_seq=1
ttl=55 time= 14.5 ms
--- www.centos.org ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 14.477/14.477/14.477/0.000 ms
```

Solicitud en broadcast para la subred de la máquina:

```
ping -c1 -b 192.168.0.255
WARNING: pinging broadcast address
PING 192.168.0.255 (192.168.0.255) 56(84) bytes of data.
64 bytes from 192.168.0.25: icmp_seq=1 ttl=64 time=189 ms
--- 192.168.0.255 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 188.906/188.906/188.906/0.000 ms
```

Solamente una máquina ha respondido, aunque hay varias máquinas activas en esa subred.

## h. Resolución de dirección IPv4/MAC usando ARP

El protocolo ARP (*Address Resolution Protocol*) permite establecer la correspondencia entre una dirección IPv4 y una dirección MAC.

Cuando un host IP tiene que enviar una trama a través de una red, tiene que conocer la dirección de red de la capa de enlace de datos de la máquina con la que quiere contactar. Obtendrá esta dirección de la manera siguiente:

- Si el host IP de destino se encuentra en la misma subred que el emisor, este último buscará la relación de dirección IP de destino/dirección MAC en su tabla

ARP. Si la dirección no se encuentra en la tabla, enviará un mensaje ARP en broadcast en la capa de enlace, este mensaje contendrá la dirección IP del destinatario de la trama. La máquina que reconocerá su dirección IP responderá dando su dirección MAC. La máquina emisora actualizará su tabla ARP y emitirá la trama.

Si el host IP de destino no se encuentra en la misma subred que el emisor, este último buscará en su tabla ARP la dirección de red MAC asociada a la dirección IP del router que conducirá la trama a la red/subred donde se encuentra la máquina destino de la trama (o a la pasarela por defecto si no existe ninguna ruta específica en la tabla de enrutamiento). Si la dirección MAC no se encuentra en esta tabla, enviará un mensaje ARP en broadcast en la capa de enlace que contendrá la dirección IP del router/pasarela por defecto. El router responde dando su dirección MAC. La máquina emisora actualizará su tabla ARP y emitirá la trama.

Las entradas de la tabla ARP tienen una duración de vida limitada.

El comando `arp` permite administrar la tabla ARP.



Este comando forma parte del paquete `net-tools`.

### Sintaxis

```
arp [ -vn ] [ -d DirIp | -s DirIp DirMAC ]
```

### Parámetros principales

<code>-n</code>	Utiliza las direcciones en lugar de los nombres de hosts.
<code>-v</code>	Vista detallada.
<code>-d DirIp</code>	Suprime la entrada para la dirección de red <code>DirIp</code> .
<code>-s DirIp DirMAC</code>	Añade o modifica la entrada entre <code>DirIp</code> y <code>DirMAC</code> .

### Descripción

El comando permite visualizar el contenido de la tabla ARP y, opcionalmente, modificarlo. La opción `-n` usa las direcciones IP en lugar de los nombres de hosts, lo que evita que el comando solicite una resolución de nombres de host para cada dirección IP encontrada en la tabla. La opción `-d` permite eliminar una entrada de la tabla, la opción `-s` añade o modifica una entrada.

### Ejemplo

Se muestra la tabla ARP:

```
arp -n
Address      HWtype  HWaddress    Flags Mask    Iface
192.168.0.24  ether   dc:a2:66:67:1b:1f C          enp38s0
192.168.0.254 ether   f4:ca:e5:44:86:58 C          enp38s0
```

Se envía una trama a una máquina en la misma subred:

```
ping 192.168.0.39
PING 192.168.0.39 (192.168.0.39) 56(84) bytes of data.
64 bytes from 192.168.0.39: icmp_seq=1 ttl=64 time=0.148 ms
64 bytes from 192.168.0.39: icmp_seq=2 ttl=64 time=0.259 ms
^C
```



```

--- 192.168.0.39 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 16ms
rtt min/avg/max/mdev = 0.148/0.203/0.259/0.057 ms

```

Se vuelve a mostrar la tabla ARP.

```

arp -n

```

Address	HWtype	HWaddress	Flags	Mask	Iface
192.168.0.39	ether	00:1b:24:6a:78:14	C		enp38s0
192.168.0.24	ether	dc:a2:66:67:1b:1f	C		enp38s0
192.168.0.254	ether	f4:ca:e5:44:86:58	C		enp38s0

Hay una nueva entrada en la tabla ARP.

Se modifica la entrada de esta máquina, con una dirección Ethernet inexistente:

```

arp -s 192.168.0.39 00:1b:24:6a:78:01
arp -n

```

Address	HWtype	HWaddress	Flags	Mask	Iface
192.168.0.39	ether	00:1b:24:6a:78:01	CM		enp38s0
192.168.0.24	ether	dc:a2:66:67:1b:1f	C		enp38s0
192.168.0.254	ether	f4:ca:e5:44:86:58	C		enp38s0

Se intenta enviar una trama a esta máquina:

```

ping 192.168.0.39
PING 192.168.0.39 (192.168.0.39) 56(84) bytes of data. ^C
--- 192.168.0.39 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 60ms

```

La máquina ya no es accesible.

Se elimina la entrada de la tabla y se vuelve a empezar:

```

arp -d 192.168.0.39
ping 192.168.0.39
PING 192.168.0.39 (192.168.0.39) 56(84) bytes of data.
64 bytes from 192.168.0.39: icmp_seq=1 ttl=64 time=0.157 ms
64 bytes from 192.168.0.39: icmp_seq=2 ttl=64 time=0.223 ms
64 bytes from 192.168.0.39: icmp_seq=3 ttl=64 time=0.152 ms
^C
--- 192.168.0.39 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 29ms
rtt min/avg/max/mdev = 0.152/0.177/0.223/0.034 ms
arp -n

```

Address	HWtype	HWaddress	Flags	Mask	Iface
192.168.0.39	ether	00:1b:24:6a:78:14	C		enp38s0
192.168.0.24	ether	dc:a2:66:67:1b:1f	C		enp38s0
192.168.0.254	ether	f4:ca:e5:44:86:58	C		enp38s0

La tabla se ha actualizado dinámicamente.

## i. Resolución de dirección IPv6/MAC con NDP

En IPv6, el protocolo NDP (*Neighbor Discovery Protocol*) permite, entre otras cosas, establecer la correspondencia entre una dirección IPv6 y una dirección MAC.

Este protocolo gestiona un conjunto de funcionalidades que cubren las del protocolo ARP y las del protocolo ICMP en IPv4. Permite, en particular, realizar la resolución de una dirección IPv6 con la dirección de red MAC correspondiente.

El comando `ip neigh` permite administrar la tabla de resolución de las direcciones IPv6/MAC.

### Sintaxis

```
ip neigh [ show | add | change | del | flush ] [ dev Interfaz ]
```

### Parámetros principales

[ show   add   change   del   flush ]	Subcomando que se ejecutará.
[ dev Interfaz ]	Interfaz en cuestión.

### Descripción

El subcomando `show` (subcomando por defecto) permite visualizar el contenido de la tabla de resolución de direcciones, en todas las interfaces o en la interfaz especificada. Los subcomandos `add`, `change`, `delete` y `flush` permiten modificarlas.

### Ejemplo

Visualización de la tabla de resolución de todas las interfaces:

```
ip neigh
192.168.0.39 dev enp38s0 lladdr 00:1b:24:6a:78:14 REACHABLE
192.168.0.24 dev enp38s0 lladdr dc:a2:66:67:1b:1f DELAY
192.168.0.254 dev enp38s0 lladdr f4:ca:e5:44:86:58 STALE
fe80::f6ca:e5ff:fe44:8658 dev enp38s0 lladdr f4:ca:e5:44:86:58 router REACHABLE
fe80::f6ca:e5ff:fe44:8658 dev wlo1 lladdr f4:ca:e5:44:86:58 router DELAY
```

Se vacía la tabla de la interfaz `enp38s0`:

```
ip neigh flush dev enp38s0
```

Se visualiza la tabla de resolución de la interfaz `enp38s0`:

```
ip neigh show dev enp38s0 192.168.0.24 lladdr dc:a2:66:67:1b:1f REACHABLE
```

Se envía una trama IPv6 hacia un nodo local:

**ping6 fe80::f6ca:e5ff:fe44:8658**

```

PING fe80::f6ca:e5ff:fe44:8658(fe80::f6ca:e5ff:fe44:8658) 56 data bytes
64 bytes from fe80::f6ca:e5ff:fe44:8658%enp38s0: icmp_seq=1 ttl=64 time=0.887 ms
64 bytes from fe80::f6ca:e5ff:fe44:8658%enp38s0: icmp_seq=2 ttl=64 time=0.346 ms
^C
--- fe80::f6ca:e5ff:fe44:8658 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 2ms

```

Se vuelve a visualizar la tabla de resolución de la interfaz `enp38s0`:

**ip neigh show dev enp38s0**

```

192.168.0.39 lladdr 00:1b:24:6a:78:14 REACHABLE
192.168.0.24 lladdr dc:a2:66:67:1b:1f REACHABLE
fe80::f6ca:e5ff:fe44:8658 lladdr f4:ca:e5:44:86:58 router REACHABLE

```

## 5. Configurar una interfaz de red inalámbrica

Los comandos vistos anteriormente para la gestión de las interfaces de red Ethernet se aplican para las interfaces de red inalámbricas.

Sin embargo, algunos comandos permite administrar las características específicas de estas interfaces.



Para las distribuciones de tipo Red Hat, los comandos específicos para las redes inalámbricas forman parte del paquete opcional `iw`. Los antiguos comandos (`iwconfig`, `iwlist`) están considerados como obsoletos en la versión 8. En la versión 7, se pueden instalar gracias al repositorio EPEL y al paquete `wireless-tools`.

### a. El comando iw

Se trata del comando estándar de configuración dinámica de las interfaces de red inalámbricas, para conexiones en una red IP, versión 4 y versión 6. Presenta muchos subcomandos para visualizar y configurar los diferentes niveles de atributos de una interfaz, según los diferentes niveles del protocolo, incluyendo la información de enrutamiento.



Este comando reemplaza el antiguo comando `iwconfig`, en vía de obsolescencia.

#### Sintaxis

```
iw [ dev | phy(#[ #] ) IdInterfaz SubComando ArgSubComando
```

#### Parámetros principales

<code>dev IdInterfaz</code>	Interfaz que se administrará, usando su archivo especial.
<code>Phy[ # ]</code> <code>IdInterfaz</code>	Interfaz que se administrará, por su nombre o por su número de índice.
<code>SubComando</code>	Subcomando que se ejecutará.
<code>ArgSubComando</code>	Argumentos del subcomando.

#### Descripción

Este comando permite visualizar o modificar los parámetros de una o todas las tarjetas de

interfaz de red inalámbricas.

Se puede especificar una interfaz gracias a su archivo especial (`dev Interfaz`), su nombre (`phy Nombre`) o su índice de interfaz (`phy# Número`). Si el identificador es único, se puede omitir la palabra clave `dev` o `Número`.

Subcomandos principales:

<code>dev</code>	Sin subcomando, muestra todas las interfaces de red inalámbricas.
<code>list</code>	Muestra las características de todas las interfaces de red inalámbricas.
<code>dev phy[#] IdDev info</code>	Muestra las características de la interfaz especificada.
<code>[dev phy[#]]IdDev channels</code>	Muestra los canales disponibles para la interfaz especificada.
<code>[dev] IdDev link</code>	Estado de la conexión actual.
<code>[dev] IdDev scan</code>	Búsqueda de redes disponibles.
<code>[dev phy[#]]IdDev add</code>	Añade una interfaz.

### Ejemplos

Lista de las interfaces de red inalámbricas:

```
iw dev
```

```

phy#0
  Interface wlo1
    ifindex 3
    wdev 0x1
    addr 9c:b7:0d:bb:2b:67
    ssid adri
    type managed
    channel 11 (2462 MHz), width: 20 MHz, center1: 2462 MHz
    txpower 15.00 dBm

```

El sistema dispone de una tarjeta de interfaz de red inalámbrica, `wlo1`, índice `0`, conectada a la red de SSID (Service Set Identifier) `adri`.

Lista de las interfaces inalámbricas, con todas sus características. La vista cubre varias pantallas:

#### **iw list | more:**

Wiphy phy0

```

max # scan SSIDs: 4
max scan IEs length: 2257 bytes
max # sched scan SSIDs: 0
max # match sets: 0
max # scan plans: 1
max scan plan interval: -1
max scan plan iterations: 0
Retry short limit: 7
Retry long limit: 4
Coverage class: 0 (up to 0m)
Device supports RSN-IBSS.
Device supports AP-side u-APSD.
Device supports T-DLS.
Supported Ciphers:
  * WEP40 (00-0f-ac:1)
  * WEP104 (00-0f-ac:5)
  * TKIP (00-0f-ac:2)
  * CCMP-128 (00-0f-ac:4)
  * CCMP-256 (00-0f-ac:10)
  * GCMP-128 (00-0f-ac:8)

```

- \* GCMP-256 (00-0f-ac:9)
- \* CMAC (00-0f-ac:6)
- \* CMAC-256 (00-0f-ac:13)
- \* GMAC-128 (00-0f-ac:11)
- \* GMAC-256 (00-0f-ac:12)

[...]

P2P GO supports CT window setting  
 Driver supports full state transitions **for** AP/GO clients  
 Driver supports a userspace MPM  
 Device supports active monitor (which will ACK incoming frames)  
 Driver/device bandwidth changes during BSS lifetime (AP/GO mode)  
 Device supports configuring vdev MAC-addr **on** create.

Características de una interfaz de red inalámbrica, identificada por su índice (**0**):

#### **iw phy#0 info | head**

Wiphy phy0

max # scan SSIDs: 4  
 max scan IEs length: 2257 bytes  
 max # sched scan SSIDs: 0  
 max # match sets: 0  
 max # scan plans: 1  
 max scan plan interval: -1  
 max scan plan iterations: 0  
 Retry **short** limit: 7  
 Retry **long** limit: 4

Características de una interfaz de red inalámbrica, identificada por su nombre lógico (**phy0**):

#### **iw phy0 info | head**

Wiphy phy0

max # scan SSIDs: 4  
 max scan IEs length: 2257 bytes  
 max # sched scan SSIDs: 0  
 max # match sets: 0  
 max # scan plans: 1



max scan plan interval: -1  
 max scan plan iterations: 0  
 Retry **short** limit: 7  
 Retry **long** limit: 4

Características de una interfaz de red inalámbrica, identificada por su archivo especial (**wlo1**):

#### **iw dev wlo1 info | head**

```
Interfaz wlo1
  ifindex 3
  wdev 0x1
  addr 9c:b7:0d:bb:2b:67
  ssid adri
  type managed
  wiphy 0
  channel 11 (2462 MHz), width: 20 MHz, center1: 2462 MHz
  txpower 15.00 dBm
```

Información sobre la conexión actual:

#### **I w dev wlo1 link**

```
Connected to 00:07:cb:02:18:44 (on wlo1)
  SSID: adri
  freq: 2462
  RX: 23705701 bytes (228929 packets)
  TX: 90468 bytes (912 packets)
  signal: -36 dBm
  tx bitrate: 1.0 MBit/s

  bss flags:  short-slot-time
  dtim period:  2
  beacon int:  96
```

Búsqueda de redes disponibles:

**iw dev wlo1 scan | more**

```

BSS 00:07:cb:02:18:44(on wlo1) -- associated
    TSF: 1055161979067 usec (12d, 05:06:01)
    freq: 2462
    beacon interval: 96 TUs
    capability: ESS Privacy ShortSlotTime (0x0411)
    signal: -36.00 dBm
    last seen: 194 ms ago
    Information elements from Probe Response frame:
    SSID: adri
    Supported rates: 1.0* 2.0* 5.5* 11.0* 22.0 6.0 9.0 12.0
    DS Parameter set: channel 11
    ERP: Barker_Preamble_Mode
    Extended supported rates: 18.0 24.0 36.0 48.0 54.0
    WPA:   * Version: 1
           * Group cipher: TKIP
           * Pairwise ciphers: CCMP TKIP
           * Authentication suites: PSK
BSS 24:95:04:b1:a0:9c(on wlo1)
    TSF: 619761559371 usec (7d, 04:09:21)
    freq: 2412
    beacon interval: 100 TUs
    capability: ESS Privacy ShortSlotTime (0x0411)
    signal: -63.00 dBm
    last seen: 2620 ms ago
    Information elements from Probe Response frame:
    SSID: SFR_A098
    Supported rates: 1.0* 2.0* 5.5* 11.0* 18.0 24.0 36.0 54.0
    DS Parameter set: channel 1
    ERP: Barker_Preamble_Mode
[...]
BSS 92:95:04:b1:a0:9d(on wlo1)
    TSF: 619761561551 usec (7d, 04:09:21)
    freq: 2412
    beacon interval: 100 TUs
    capability: ESS ShortSlotTime (0x0401)
    signal: -64.00 dBm
    last seen: 2618 ms ago
    Information elements from Probe Response frame:
    SSID: SFR WiFi FON

```

Supported rates: 1.0\* 2.0\* 5.5\* 11.0\* 18.0 24.0 36.0 54.0  
 DS Parameter **set**: channel 1  
 ERP: Barker\_Preamble\_Mode  
 [...]

## b. Los comandos iwconfig y iwlist

Estos comandos permiten visualizar y configurar las tarjetas de interfaz de red inalámbricas (**iwconfig**), o listar las redes inalámbricas accesibles (**iwlist**). Aunque son considerados como en vías de obsolescencia, todavía se utilizan en algunos entornos de producción y pueden ser objeto de preguntas en el examen de la certificación.

### Sintaxis

**iwconfig** Interfaz SubComando ArgSubComando

### Parámetros principales

<b>Interfaz</b>	Interfaz que se gestionará, usando su archivo especial.
<b>SubComando</b>	Subcomando que se ejecutará.
<b>ArgSubComando</b>	Argumentos del subcomando.

### Descripción

Este comando permite visualizar o modificar los parámetros de una o todas las tarjetas de interfaz de red inalámbricas. Por defecto, muestra las interfaces de red inalámbricas disponibles. Una interfaz puede ser especificada por su archivo especial.

Los subcomandos permiten configurar distintos parámetros de la tarjeta de interfaz de red.

Ejemplo**iwconfig wlo1**

```
wlo1 IEEE 802.11-DS ESSID:"WIFITEST" 1
Mode:Managed2 Frequency:2.442 GHz3 Access Point: 00:0E:20:7C:E7:C88
Bit Rate:11 Mb/s Tx-Power=20 dBm Sensitivity=0/65535
Retry limit:16 RTS thr:off Fragment thr:off
Power Management:off
Link Quality=100/100 Signal level=-34 dBm Noise level=-90 dBm
Rx invalid nwid:9418 Rx invalid crypt:0 Rx invalid frag:0
Tx excessive retries:0 Invalid misc:54513 Missed beacon:0
```

Sintaxis

```
iwlist [ Interfaz ] scanning
```

Parámetros principales

Interfaz	Interfaz que se gestionará, usando su archivo especial.
----------	---

Descripción

Este comando permite visualizar las redes inalámbricas disponibles. Por defecto, muestra las redes inalámbricas disponibles para todas las interfaces. Se puede especificar una interfaz gracias a su archivo especial.

Ejemplo**iwlist wlo1 scan**

```
Cell 01 - Address: BF:9C:1A:16:B4:32
ESSID:"LABORATORIO"
Mode:Managed
Frequency:2.427 GHz (Channel 4)
Quality:1/5 Signal level:-83 dBm Noise level:-84 dBm
```

Encryption **key:off**

Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 9 Mb/s

18 Mb/s; 36 Mb/s; **54 Mb/s**; 6 Mb/s; 12 Mb/s

24 Mb/s; 48 Mb/s