

	<b>Guía rápida de ayuda configuración IP en Linux</b>	
<b>Redes y Servicios de Comunicaciones</b>		<b>2016-2017</b>

## 1. Introducción

Nada puede substituir a la consulta de un manual o la información de ayuda, pero el siguiente documento es una guía rápida para orientarse en el uso de comandos para la configuración IP en Linux.

## 2. Configuración IP en Linux

### 2.1 Configuración y visualización de direcciones IP

Tradicionalmente se empleaba el comando `ifconfig` y es conveniente estar familiarizado con él, pero nótese que la información que proporciona en kernels Linux actuales puede ser incompleta<sup>1</sup>, por lo que el comando que debe usarse normalmente es el comando `ip` descrito posteriormente:

```
/sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr f8:bc:12:80:b4:9c
          inet addr:163.117.144.103  Bcast:163.117.144.255  Mask:255.255.255.0
          inet6 addr: 2001:720:410:100c::103/64 Scope:Global
          inet6 addr: fe80::fabc:12ff:fe80:b49c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:467 errors:0 dropped:0 overruns:0 frame:0
          TX packets:472 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:54950 (53.6 KiB)  TX bytes:41999 (41.0 KiB)
          Interrupt:9 Base address:0x1f00

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:16436  Metric:1
          RX packets:50 errors:0 dropped:0 overruns:0 frame:0
          TX packets:50 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3480 (3.3 KiB)  TX bytes:3480 (3.3 KiB)
```

El comando `ifconfig`, como puede verse en el ejemplo, informa tanto de las direcciones IPv4 como de las direcciones IPv6 configuradas en las interfaces.

### 2.2 Configuración y visualización de rutas IP

Tradicionalmente se empleaba el comando `route` (pero, por el mismo motivo explicado antes, actualmente debe usarse el comando `ip` descrito posteriormente):

```
/sbin/route
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
163.117.1440.0  *              255.255.255.0  U      0      0      0 eth0
default         163.117.144.2  0.0.0.0        UG     0      0      0 eth0
```

<sup>1</sup> Por ejemplo, si configura varias direcciones IPv4 en una interfaz (lo que puede ocurrir incluso por error), el comando `ifconfig` no las presentará adecuadamente mientras que el comando `ip` sí.

Por ejemplo, el siguiente comando añade una ruta hacia 163.117.168.0/24 empleando como siguiente salto el router 163.117.144.2 accesible a través de la interfaz eth0:

```
route add -net 163.117.168.0 netmask 255.255.255.0 gw 163.117.144.2 dev eth0
```

El comando *route* puede usarse igualmente para consultar o modificar la tabla de forwarding IPv6, pero para ello hay que usar la opción *-6* (route *-6*). Por ejemplo:

```
/sbin/route -6
Kernel IPv6 routing table
Destination                               Next Hop                                Flag  Met  Ref Use If
2001:720:410:100c::/64                    ::                                       U     256  4   610 eth0
fe80::/64                                  ::                                       U     256  0    0 eth0
::/0                                        fe80::215:17ff:fe15:b080              UGDAe 1024 4  2276 eth0
::/0                                        ::                                       !n    -1   1  2875 lo
::1/128                                    ::                                       Un     0   5  72125 lo
2001:720:410:100c::103/128                ::                                       Un     0   5   540 lo
fe80::fab:12ff:fe80:b49c/128              ::                                       Un     0   5   152 lo
ff00::/8                                    ::                                       U     256  4 11624 eth0
```

### 2.3 Configuración y visualización de la caché ARP

Tradicionalmente se empleaba el comando *arp* (pero por el mismo motivo explicado antes, actualmente debe usarse el comando *ip* descrito posteriormente):

```
arp
Address                               HWtype  HWaddress                                Flags Mask                Iface
163.117.144.104                       ether   00:0D:9D:46:F4:29              C                          eth0
```

### 2.4 La herramienta ip

En la actualidad se recomienda utilizar la herramienta *ip* para la gestión y configuración de red en Linux. Aglutina las funcionalidades de *ifconfig*, *route* y *arp* (además de otras adicionales), empleando una sintaxis común, lo que hace más sencilla su utilización. Además, esta herramienta es la única que proporciona información precisa<sup>2</sup>.

```
ip
Usage: ip [ OPTIONS ] OBJECT { COMMAND | help }
where OBJECT := { link | addr | route | rule | neigh | tunnel |
                maddr | mroute | monitor }
OPTIONS := { -V[ersion] | -s[tatistics] | -r[esolve] |
            -f[amily] { inet | inet6 | ipx | dnet | link } | -o[neline] }
```

Mediante *ip* podemos ejecutar un comando (hacer una operación) sobre un objeto (véase la sintaxis arriba). Hay varios objetos posibles, pero los que vamos a usar nosotros son *link* (interfaces de red), *addr* (direcciones IP de interfaces de red), *route* (tablas de forwarding), y *neigh* (caché con direcciones MAC <-> direcciones IP de vecinos, es decir la ARP cache y la Neighbor cache).

La sintaxis de *ip* es muy fácil de usar, el mismo comando aplicado a distintos objetos tiene el mismo efecto, pero sobre los distintos objetos. Por ejemplo “*ip link show*” muestra información sobre las interfaces de red del equipo; “*ip addr show*” muestra la

<sup>2</sup> Los comandos *ifconfig*, *route* y *arp* no deberían usarse con kernels >= linux-2.4). El comando *ip* utiliza *netlink* sockets para interactuar con el kernel, y permite acceder y configurar toda la funcionalidad de la pila IP de un kernel Linux actual.

información del direccionamiento IP de las distintas interfaces de red; “ip route show” muestra la tabla de forwarding del equipo; e “ip neigh show” muestra la tabla caché de vecinos IP con sus respectivas MAC.

Por defecto la herramienta *ip* trabaja con direcciones IPv4 o, si le es posible, con direcciones IPv4 e IPv6 a la vez. Por ejemplo, la salida del comando “*ip addr show*” muestra tanto direcciones IPv4 como IPv6, pero el comando “*ip route show*” solo muestra la tabla de forwarding IPv4. Si se quiere especificar que el comando actúe con direcciones IPv6 hay que añadir “-f inet6” o simplemente “-6” (por ejemplo, “*ip -6 route*” para ver la tabla de forwarding IPv6). Igualmente, si se quiere especificar que solo se trabaje con direcciones IPv4 hay que añadir “-f inet” o simplemente “-4” (aunque normalmente esto no será necesario, porque por defecto ya se trabaja con las direcciones IPv4, con lo que el efecto es explícitamente decir que no queremos actuar con direcciones IPv6).

Algunos trucos para usar la herramienta *ip*:

- Nótese que en *ip*, el comando por defecto es *show*, por eso “*ip addr*” es equivalente a “*ip addr show*”
- Si se abrevia un parámetro y no hay ambigüedad, sigue funcionando. Por ejemplo “*ip ro*” es equivalente a “*ip route*”.
- Si el comando tiene una dirección IPv4 o IPv6 como parámetro, no es necesario poner “-6” o “-4” porque lo interpreta adecuadamente a partir de la dirección.

## 2.4.1 Configuración y visualización de direcciones IP

- Para ver las direcciones asociadas a interfaces en un equipo:

```
ip addr
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: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group
default qlen 1000
    link/ether f8:bc:12:80:b4:9c brd ff:ff:ff:ff:ff:ff
    inet 163.117.144.103/24 brd 163.117.144.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 2001:720:410:100c::103/64 scope global
        valid_lft forever preferred_lft forever
    inet6 fe80::fab:12ff:fe80:b49c/64 scope link
        valid_lft forever preferred_lft forever
```

- Para configurar una nueva dirección IPv4 a una interfaz de red (en este ejemplo, la dirección 192.168.0.103/24 a la interfaz eth1):

```
ip addr add 192.168.0.103/24 dev eth1
```

- Configurar una nueva dirección IPv6 a una interfaz de red (en este ejemplo, la dirección 2001:db8::103/64 a la interfaz eth1):

```
ip -6 addr add 2001:db8::103/64 dev eth1
```

- Para eliminar una dirección se usa el comando *del*, por lo que para quitar las dos direcciones anteriores:

```
ip addr del 192.168.0.103/24 dev eth1
ip -6 addr del 2001:db8::103/64 dev eth1
```

## 2.4.2 Configuración y visualización de rutas IP

- Para ver la tabla de *forwarding* IPv4:

```
ip route
default via 163.117.144.2 dev eth0
163.117.144.0/24 dev eth0 proto kernel scope link src 163.117.144.103
```

- Para ver la table de *forwarding* IPv6:

```
ip -6 route
2001:720:410:100c::/64 dev eth0 proto kernel metric 256
fe80::/64 dev eth0 proto kernel metric 256
default via fe80::215:17ff:fe15:b080 dev eth0 proto ra metric 1024 expires 27sec
hoplimit 64
```

- Para añadir una ruta a la tabla de forwarding IPv4 (en este ejemplo, una ruta hacia 192.168.0.0/16 utilizando como siguiente salto el router 192.168.0.1 accesible a través de la interfaz eth1):

```
ip ro add 192.168.0.0/16 via 192.168.0.1 dev eth1
```

- Para añadir una ruta a la tabla de forwarding IPv6 (en este ejemplo, una ruta hacia 2001:db8::/32 utilizando como siguiente salto el router fe80::2 accesible a través de la interfaz eth1):

```
ip -6 ro add 2001:db8::/32 via fe80::2 dev eth1
```

- Para eliminar una ruta se usa el comando *del*, por lo que para quitar las dos rutas anteriores:

```
ip ro del 192.168.0.0/16 via 192.168.0.1 dev eth1
ip -6 ro del 2001:db8::/32 via fe80::2 dev eth1
```

### 2.4.3 Visualización de vecinos IP

Para ver la caché con direcciones MAC <--> direcciones IP de vecinos:

```
ip neigh
163.117.144.2 dev eth0 lladdr 00:15:17:15:b0:80 REACHABLE
fe80::215:17ff:fe15:b080 dev eth0 lladdr 00:15:17:15:b0:80 router STALE
2001:720:410:100c::129 dev eth0 lladdr b8:ac:6f:8f:f7:dc STALE
```

Muestra tanto la *ARP cache* (IPv4) como la *Neighbor Cache* (IPv6). Si se quiere ver solo una de ellas, indicar: “*ip -4 neigh*” o “*ip -6 neigh*”.

### 2.4.4 Otras configuraciones

La herramienta *ip* permite acceder a la configuración de otros parámetros (otros objetos en terminología de la herramienta *ip*), no empleados en esta asignatura, pero que le pueden resultar útiles en el futuro:

- ◆ túneles,
- ◆ direcciones multicast
- ◆ rutas multicast
- ◆ etc ...

### 2.4.5 Obtención de ayuda

Puede obtener ayuda consultando las páginas de manual: `man ip / man 8 ip`  
También puede añadir la palabra ‘help’ al final del comando que realiza la acción sobre la que queremos obtener ayuda:

```
ip addr help
Usage: ip addr {add|change|replace} IFADDR dev STRING [ LIFETIME ]
                                     [ CONFFLAG-LIST ]

ip addr del IFADDR dev STRING [mngtmpaddr]
ip addr {show|save|flush} [ dev STRING ] [ scope SCOPE-ID ]
                                     [ to PREFIX ] [ FLAG-LIST ] [ label PATTERN ] [up]

ip addr {showdump|restore}
IFADDR := PREFIX | ADDR peer PREFIX
         [ broadcast ADDR ] [ anycast ADDR ]
         [ label STRING ] [ scope SCOPE-ID ]
SCOPE-ID := [ host | link | global | NUMBER ]
FLAG-LIST := [ FLAG-LIST ] FLAG
FLAG := [ permanent | dynamic | secondary | primary |
         tentative | deprecated | dadfailed | temporary |
         CONFFLAG-LIST ]
CONFFLAG-LIST := [ CONFFLAG-LIST ] CONFFLAG
CONFFLAG := [ home | nodad | mngtmpaddr | noprefixroute ]
LIFETIME := [ valid_lft LFT ] [ preferred_lft LFT ]
LFT := forever | SECONDS
```

### 2.4.6 Guía rápida de comandos de supervivencia

Ejemplos de comandos (muchos explicados antes):

- Configurar una nueva dirección IPv4 a una interfaz de red (en este ejemplo, la dirección 192.168.0.103/24 a la interfaz eth1):

```
ip addr add 192.168.0.103/24 dev eth1
```

- Configurar una nueva dirección IPv6 a una interfaz de red (en este ejemplo, la dirección 2001:db8::103/64 a la interfaz `eth1`):

```
ip -6 addr add 2001:db8::103/64 dev eth1
```

- Añadir una ruta a la tabla de forwarding IPv4 (en este ejemplo, una ruta hacia 192.168.0.0/16 utilizando como siguiente salto el router 192.168.0.1 accesible a través de la interfaz `eth1`):

```
ip ro add 192.168.0.0/16 via 192.168.0.1 dev eth1
```

- Añadir una ruta a la tabla de forwarding IPv6 (en este ejemplo, una ruta hacia 2001:db8::/32 utilizando como siguiente salto el router fe80::2 accesible a través de la interfaz `eth1`):

```
ip -6 ro add 2001:db8::/32 via fe80::2 dev eth1
```

- Todos los comandos anteriores para añadir (dirección o ruta), se pueden usar para borrar simplemente substituyendo “*add*” por “*del*”
- Borrar una entrada de la tabla ARP (en este ejemplo, la correspondiente a la dirección IP 192.168.0.1 accesible a través de la interfaz `eth1`):

```
ip neigh del 192.168.0.1 dev eth1
```

- Borrar una entrada de la tabla Neighbor Cache (en este ejemplo, la correspondiente a la dirección IP 2001:db8::115 accesible a través de la interfaz `eth1`):

```
ip -6 neigh del 2001:db8::115 dev eth1
```

- Borrar todas las entradas ARP y Neighbor Cache de direcciones accesibles a través de una interfaz (en este ejemplo, la interfaz `eth1`):

```
ip neigh flush dev eth1
```

Borra tanto las entradas de la ARP cache como de la Neighbor Cache. Si solo se quiere borrar una de ellas se puede hacer uso de las opciones “-4” o “-6”.

- Habilitar una interfaz (en este ejemplo, la interfaz `eth1`):

```
ip link set eth1 up
```