

 <div data-bbox="395 197 579 302"> Universidad Carlos III de Madrid </div>	Linux IP configuration - Quick Guide
Redes y Servicios de Comunicaciones	2023-2024

1. Introduction

Nothing can substitute for consulting a manual or help information, but the following document is a quick guide to orient yourself in the use of commands for IP configuration in Linux.

2. Linux IP Configuration

2.1 IP address configuration and visualization

Traditionally the `ifconfig` command was used and it is convenient to be familiar with it, but note that the information it provides in current Linux kernels may be incomplete, so the command that should normally be used is the `ip` command described below:

#> 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)
```

The `ifconfig` command, as shown in the example, reports both IPv4 and IPv6 addresses configured on the interfaces.

¹ For example, if you configure multiple IPv4 addresses on an interface (which can happen even by mistake), the `ifconfig` command will not present them properly while the `ip` command will.

² We use #> to represent the system prompt.

2.2 Configuration and display of IP routes

Traditionally, the route command was used (but, for the same reason explained above, the ip command described below should now be used):

```
#> route
```

Kernel IP routing table

<i>Destination</i>	<i>Gateway</i>	<i>Genmask</i>	<i>Flags</i>	<i>Metric</i>	<i>Ref</i>	<i>Use</i>	<i>Iface</i>
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

For example, the following command adds a route to 163.117.168.0/24 using as a next hop the router 163.117.144.2 accessible through the eth0 interface:

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

The route command can also be used to query or modify the IPv6 forwarding table, but the -6 option (route -6) must be used. For example:

```
#> /sbin/route -6
```

Kernel IPv6 routing table

<i>Destination</i>	<i>Next Hop</i>	<i>Flag</i>	<i>Met</i>	<i>Ref</i>	<i>Use</i>	<i>If</i>
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
::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 Configuration and display of the ARP cache

Traditionally, the arp command was used (but for the same reason explained above, the ip command described below should now be used):

```
#> arp
```

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

2.4 ip Tool

It is currently recommended to use the ip tool for network management and configuration in Linux. It combines the functionalities of ifconfig, route and arp (plus additional ones), using a common syntax, which makes it easier to use. In addition, this tool is the only one that provides accurate information ¹.

```
#> 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] }

Using ip we can execute a command (perform an operation) on an object (see syntax above). There are several possible objects, but the ones we are going to use are link (network interfaces), addr (IP addresses of network interfaces), route (forwarding tables), and neigh (cache with MAC addresses <-> IP addresses of neighbors, i.e. the ARP cache and the Neighbor cache).

The syntax of ip is very easy to use, the same command applied to different objects has the same effect, but on different objects. For example "ip link show" shows information about the network interfaces of the device; "ip addr show" shows the IP addressing information of the different network interfaces; "ip route show" shows the forwarding table of the device; and "ip neigh show" shows the IP neighbor cache table with their respective MACs.

By default the ip tool works with IPv4 addresses or, if possible, with IPv4 and IPv6 addresses at the same time. For example, the output of the "ip addr show" command shows both IPv4 and IPv6 addresses, but the "ip route show" command only shows the IPv4 forwarding table. If you want to specify that the command works with IPv6 addresses, add "-f inet6" or simply "-6" (e.g. "ip -6 route" to see the IPv6 forwarding table). Likewise, if you want to specify that it

¹ The ifconfig, route and arp commands should not be used with kernels >= linux-2.4). The ip command uses netlink sockets to interact with the kernel, and allows you to access and configure the full functionality of the IP stack of a current Linux kernel..

only works with IPv4 addresses, you must add "-f inet" or simply "-4" (although normally this is not necessary, because by default it already works with IPv4 addresses, so the effect is explicitly to say that you do not want to work with IPv6 addresses).

Some tips for using the ip tool:

- Note that in ip, the default command is show, so "ip addr" is equivalent to "ip addr show".
- If a parameter is abbreviated and there is no ambiguity, it still works. For example "ip ro" is equivalent to "ip route".
- If the command has an IPv4 or IPv6 address as a parameter, there is no need to put "-6" or "-4" because it interprets it properly from the address.

2.4.1 IP address configuration and display

- To view the addresses associated with interfaces on a computer:

```
#> 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::fabc:12ff:fe80:b49c/64 scope link
    valid_lft forever preferred_lft forever
```

- To configure a new IPv4 address to a network interface (in this example, the address 192.168.0.103 of a subnet with prefix /24 to interface eth1):

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

- Configure a new IPv6 address to a network interface (in this example, the address 2001:db8::103/64 to the eth1 interface):

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

- To remove an address the del command is used, so that, to remove the two addresses above:

```
#> ip addr del 192.168.0.103/24 dev eth1
```

```
#> ip -6 addr del 2001:db8::103/64 dev eth1
```

2.4.2 Configuration and display of IP routes

- To view the IPv4 forwarding table:

```
#> 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
```

- To view the IPv6 forwarding table::

```
#> 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
```

- To add a route to the IPv4 forwarding table (in this example, a route to 192.168.0.0/16 using as a next hop the router 192.168.0.1 accessible through interface eth1):

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

- To add a route to the IPv6 forwarding table (in this example, a route to 2001:db8::/32 using as a next hop the router fe80::2 reachable through interface eth1):

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

- To remove a route the del command is used, so to remove the two routes above:

```
#> 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 Viewing IP neighbors

To view the cache with MAC addresses <-> IP addresses of neighbors:

```
#> 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
```

Displays both the ARP cache (IPv4) and the Neighbor Cache (IPv6). If you want to see only one of them, indicate: "ip -4 neigh" or "ip -6 neigh".

2.4.4 Other configurations

The ip tool allows access to the configuration of other parameters (other objects in ip tool terminology), not used in this course, but which may be useful in the future:

- ◆ tunnels,
- ◆ multicast addresses
- ◆ multicast routes
- ◆ etc ...

2.4.5 Getting help

You can get help by consulting the man pages: man ip / man 8 ip

You can also add the word 'help' to the end of the command that performs the action we want to get help on:

```
#> 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 Quick Guide to Survival Commands

Examples of commands (many explained above):

- Configure a new IPv4 address to a network interface (in this example, the address 192.168.0.103/24 to the eth1 interface):

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

- Configure a new IPv6 address to a network interface (in this example, the address 2001:db8::103/64 to the eth1 interface):

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

- Add a route to the IPv4 forwarding table (in this example, a route to 192.168.0.0/16 using as a next hop the router 192.168.0.1 accessible through interface eth1):

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

- Add a route to the IPv6 forwarding table (in this example, a route to 2001:db8::/32 using as next hop the router fe80::2 accessible through interface eth1):

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

- All of the above add (address or route) commands can be used to delete by simply substituting "add" for "del"
- Delete an entry in the ARP table (in this example, the one corresponding to the IP address 192.168.0.1 accessible through the eth1 interface):

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

- Delete an entry from the Neighbor Cache table (in this example, the one corresponding to the IP address 2001:db8::115 accessible through the eth1 interface):

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

- Delete all ARP and Neighbor Cache entries for addresses accessible through an interface (in this example, interface eth1):

```
ip neigh flush dev eth1
```

Deletes both ARP cache and Neighbor Cache entries. If you only want to delete one of them you can use the options "-4" or "-6".

- Enable an interface (in this example, the eth1 interface):

```
ip link set eth1 up
```