THE GEEK DIARY

CONCEPTS / BASICS / HOWTO

ip Command Examples to Manage Networking in Linux

By admin (https://www.thegeekdiary.com/author/sandeep_patil/)

You can use the ip command to display the status of an interface, configure network properties, or for debugging or tuning the network. The ip command replaces the **ifconfig** command, which is deprecated. The syntax of the ip utility follows:

```
# ip [OPTIONS] OBJECT {COMMAND | help}
```

ip command provides a number of **OBJECT** arguments, such as:
– **link**: Network device
– **address (or addr)**: IPv4 or IPv6 address on a device
– **route**: Routing table entry

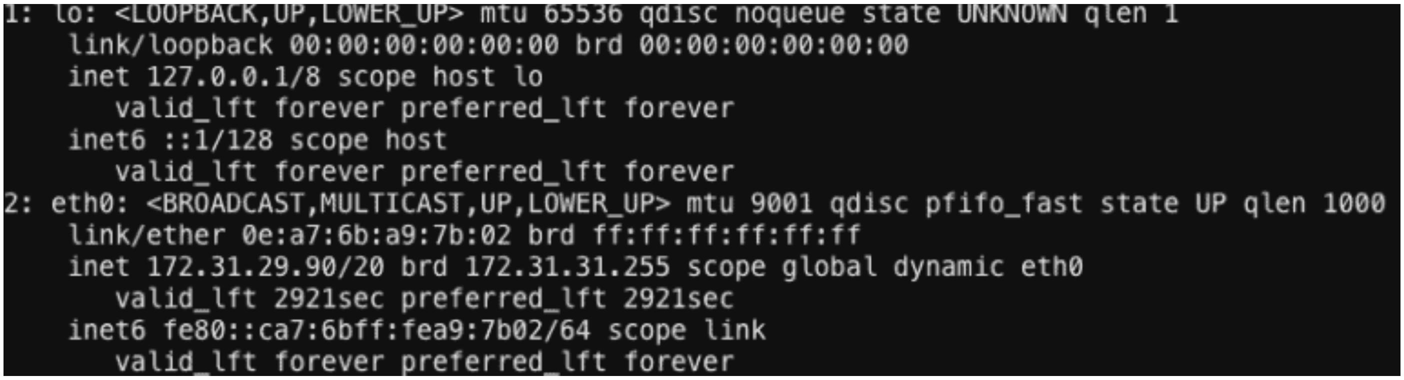
It also provides a number of **COMMANDS** for each OBJECT, such as:
– **add, change, del, show, more**

Below are few of the most commonly used ip command examples in Linux.

The ip addr sub-command

1. Show current network configuration

Use the **ip addr** object to show and manage IPv4 or IPv6 address on a device. The following example shows IP status for all active devices. The show command is the default.



```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN 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 9001 qdisc pfifo_fast state UP qlen 1000
    link/ether 0e:a7:6b:a9:7b:02 brd ff:ff:ff:ff:ff:ff
    inet 172.31.29.90/20 brd 172.31.31.255 scope global dynamic eth0
       valid_lft 2921sec preferred_lft 2921sec
    inet6 fe80::ca7:6bff:fea9:7b02/64 scope link
       valid_lft forever preferred_lft forever
```

2. Assign IPv4 Address to a network interface

The following example uses the add argument to add the IPv4 address 172.31.29.90/20 to the eth0 interface. The show argument is given afterwards to display the result. The show argument is given afterwards to display the result. This example assumes the interface already has 172.31.124.62/20 assigned to it.

```
# ip addr add 172.31.29.90/20 dev eth0
```

```
# ip addr show eth0
2: eth0: [BROADCAST,MULTICAST,UP,LOWER_UP] mtu 9001 qdisc pfifo_fast state UP
    link/ether 06:e6:41:84:10:48 brd ff:ff:ff:ff:ff:ff
    inet 172.31.124.62/20 brd 172.31.127.255 scope global dynamic eth0
       valid_lft 3311sec preferred_lft 3311sec
    inet 172.31.29.90/20 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::4e6:41ff:fe84:1048/64 scope link
       valid_lft forever preferred_lft forever
```

3. Delete IPv4 Address from an interface

Use the del argument to delete the IPv4 address. Example:

```
# ip addr show eth0
2: eth0: [BROADCAST,MULTICAST,UP,LOWER_UP] mtu 9001 qdisc pfifo_fast state UP
    link/ether 06:e6:41:84:10:48 brd ff:ff:ff:ff:ff:ff
    inet 172.31.124.62/20 brd 172.31.127.255 scope global dynamic eth0
       valid_lft 3248sec preferred_lft 3248sec
    inet6 fe80::4e6:41ff:fe84:1048/64 scope link
       valid_lft forever preferred_lft forever
```

4. Flush all IPv4 Addresses assigned to an Interface

Use the flush argument to remove all the IPv4 addresses assigned to an interface. Example:

```
# ip addr flush dev eth0
```

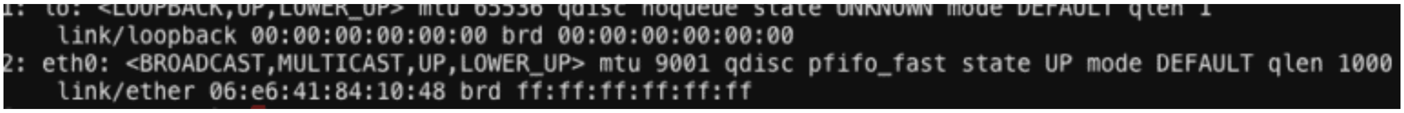
Note : Any settings that you configure for network interfaces using ip do not persist across system reboots. To make the changes permanent, set the properties in the /etc/sysconfig /network-scripts/ifcfg-[interface] file.

The ip link Sub-command

Use the ip link object to show and manage the state of network interface devices on the system. Below are some of the common usage of the ip link sub-command.

1. Check the status of network devices

The following example shows the status of all active devices. The show argument is the default. Notice that the output is similar to that of the ip addr command, but without the IP address information.



```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast state UP mode DEFAULT qlen 1000
    link/ether 06:e6:41:84:10:48 brd ff:ff:ff:ff:ff:ff
```

2. Up/Down the network interface

Use the set argument to change device attributes. The **up** and **down** arguments change the state of the device. The following example brings the eth0 interface down and then back up. The show argument displays the results of the set argument.

```
# ip link set eth0 down
```

```
# ip link show eth0
2: eth0: [BROADCAST,MULTICAST] mtu 9001 qdisc pfifo_fast state DOWN mode DEFAU
```

```
# ip link set eth0 up
```

```
# ip link show eth0
2: eth0: [BROADCAST,MULTICAST,UP,LOWER_UP] mtu 9001 qdisc pfifo_fast state UP
    link/ether 06:e6:41:84:10:48 brd ff:ff:ff:ff:ff:ff
```

3. Change the MTU attribute

The following example uses the set argument to change the MTU attribute to 1000:

```
# ip link set eth0 mtu 1000
```

ip neigh sub-command to administer ARP table

ARP resolves an IP address to the MAC address. The MAC address is a 48-bit physical hardware address, which is burned into the network interface card (NIC). Network applications use the IP address to communicate with another device but the MAC address is needed to ensure network packets are delivered.

Use the ‘**ip neigh**‘ sub-command to display the ARP table, to delete an ARP entry, or to add an entry to the table. The ip neigh sub-command replaces the arp command, which is deprecated. The ARP table is also known by another name, the **IP neighbor table**.

The ip neigh object commands are summarized as follows:

Sub-command	**Description and Task**
ip neigh add	Add a new neighbor entry.
ip neigh change	Change an existing entry.
ip neigh replace	Add a new entry or change an existing entry.
ip neigh delete	Delete a neighbor entry.
ip neigh show	List neighbor entries.
ip neigh flush	Flush neighbor tables.

1. Display ARP table

The following example displays the ARP table. The show command is the default.

2. Clear all the ARP table entries

The following example clears all entries in the ARP table with verbosity:

```
# ip -s -s neigh flush all
172.31.112.1 dev eth0 lladdr 06:75:25:15:b5:ba ref 1 used 33/0/29 probes 1 REA
*** Round 1, deleting 1 entries ***
*** Flush is complete after 1 round ***
```

3. Remove ARP table entries for a specific interface only

The following example removes entries in the ARP table on device eth0:

```
# ip neigh flush dev eth0
```

ip route sub-command

The ip route utility displays or manipulates the IP routing table. Its primary use is to set up static routes to specific hosts or networks through a network interface.

1. Displaying the Routing Table

Use the ip route command to display the routing table. Example:

```
# ip route
default via 172.31.112.1 dev eth0 proto static metric 100
172.31.112.0/20 dev eth0 proto kernel scope link src 172.31.124.62 metric 100
```

You can also use the old good ‘netstat -r’ command to display the routing table.

```
172.31.112.0    0.0.0.0         255.255.240.0   U         0 0          0 eth0
```

2. Adding a Route

Use the ip route add command to add a static route. The following example adds a default route, which is used if no other route matches. All network packages using this route are “gatewayed” through the 172.31.112.1 IP address:

```
# ip route add default via 172.31.112.1 dev eth0 proto static
```

The following example adds a static route to a host address via a specific network interface.

```
# ip route add 192.0.2.1 via 10.150.36.2 dev eth0
```

3. Delete a route

Use the ip route delete command to delete an entry from the routing table, for example:

```
# ip route delete default via 172.31.112.1
# ip route delete 192.0.2.1
```

4. Configuring Permanent Static Routes

Any changes that you make to the routing table by using ip route do not persist across system reboots. To make static routes permanent, configure them for each interface. Static route configuration is stored in a **/etc/sysconfig/network-scripts/route-interface** file. For example, static routes for the eth0 interface would be stored in the **/etc/sysconfig/network-scripts/route-eth0** file.

The route-interface file has two formats:

```
x.x.x.x/x via x.x.x.x dev interface
```

Use the term default to create a default gateway, for example:

```
default via x.x.x.x dev interface
```

The following example creates a static route to the 192.168.2.0/24 subnet through an eth0 interface (10.10.10.1):

```
# cat /etc/sysconfig/network-scripts/route-eth0 198.168.2.0/24 via 10.10.10.1
```

You can also use the **network/netmask** directives format for route-interface files. The format is as follows:

```
ADDRESS0=X.X.X.X NETMASK0=X.X.X.X GATEWAY0=X.X.X.X
```

The following example shows use of the IP command arguments to define the same entry:

```
ADDRESS0=198.168.2.0
NETMASK0=255.255.255.0
GATEWAY0=10.10.10.1
```

Start at **0** (as shown) and increment by one for each additional static route.