How to Use IP Command in Linux [24 Useful Examples]

Narendra K | Last Updated: July 13, 2023 | Read Time: 11 mins | Linux Commands, Networking Commands | 36 Comments

Brief: In this guide, we will discuss some practical examples of the ip command. By the end of this guide, users will be able to perform networking tasks efficiently in Linux from the command line interface.

System administrators often need to perform networking tasks on Linux servers. There are a variety of graphical and command-line tools available in the market. However, most Linux users prefer to use the **ip command** due to its simplicity and rich functionality.

The **ip command** is a new networking command-line utility that is used to assign an IP address to a network interface or configure/update useful network variables on a Linux system.

It is a part of the **iproute2** package and offers several network administration tasks such as bringing up or down network interfaces, assigning and removing IP addresses and routes, managing ARP cache, and much more.

The **ip command** is much similar to the old ifconfig command, but it is greatly more powerful with more functions and capabilities added to it.

[You might also like: Deprecated Linux Networking Commands and Their Replacements]

The ifconfig command has been deprecated and replaced by the **ip command** in all modern Linux distributions. However, the **ifconfig command** is still works and available for most Linux distributions.

[You might also like: ifconfig vs ip: What's Difference and Comparing Network Configuration]

Table of Contents

Note: Please take a configuration file backup before doing any changes.

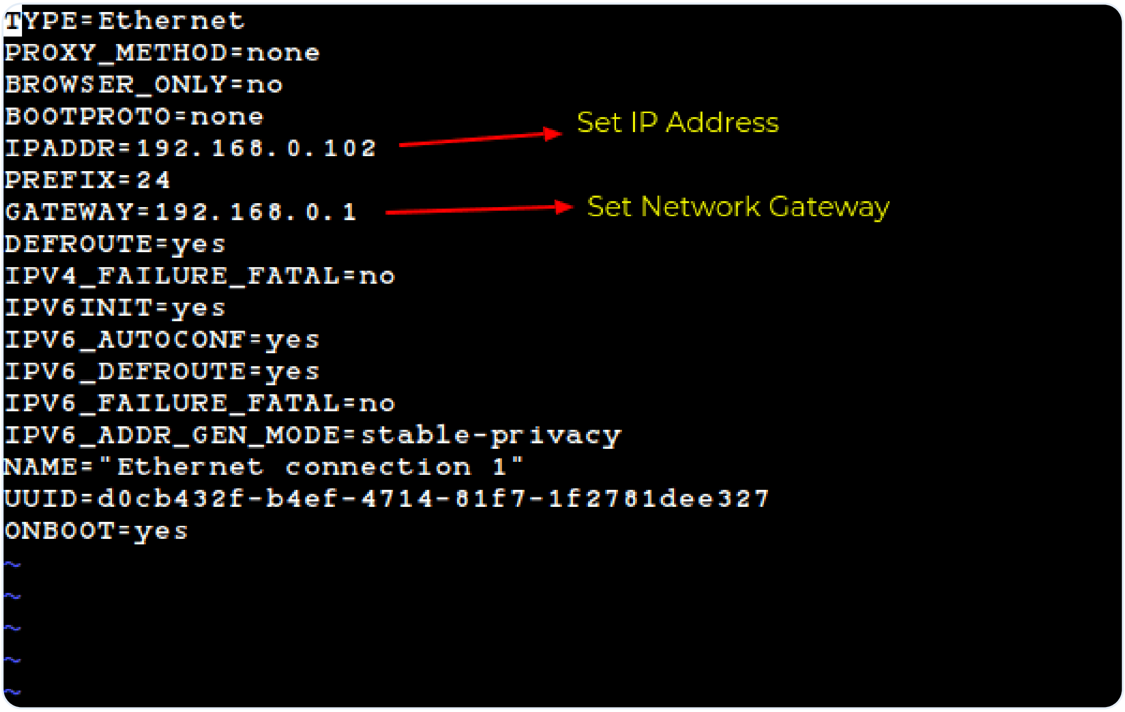
1. Permanently Configure Static IP Address in Linux

To permanently configure a static IP address in Linux, you need to update or edit the network configuration file to assign a static IP address to a system. You must be a superuser with a **su (switch user)** command from the terminal or command prompt.

Set Static IP Address in RHEL Systems

Open and edit the network configuration files for (**eth0** or **eth1**) using your favorite text editor. For example, assigning IP Address to **eth0** interface as follows on RHEL-based distributions.

```
# vi /etc/sysconfig/network-scripts/ifcfg-eth0     [On RHEL/CentOS/Fedora a
```



```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=none
IPADDR=192.168.0.102
PREFIX=24
GATEWAY=192.168.0.1
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME="Ethernet connection 1"
UUID=d0cb432f-b4ef-4714-81f7-1f2781dee327
ONBOOT=yes
~
~
~
~
~
```

Set IP Address

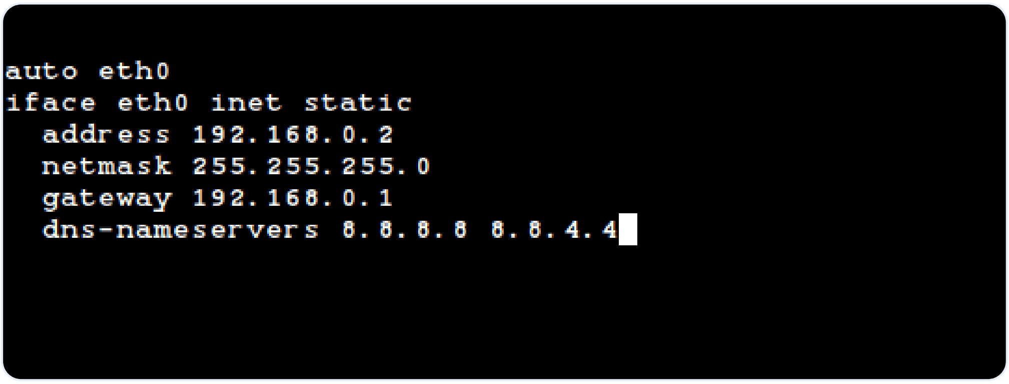
Set Network Gateway

Configure Static IP in RHEL Systems

Set Static IP Address in Debian Systems

To configure the permanent static IP address, you need to modify your network interface configuration file **/etc/network/interfaces** to make permanent changes as shown below for Debian-based distributions.

```
$ sudo nano /etc/network/interfaces     [On Debian, Ubuntu and Mint]
```



```
auto eth0
iface eth0 inet static
  address 192.168.0.2
  netmask 255.255.255.0
  gateway 192.168.0.1
  dns-nameservers 8.8.8.8 8.8.4.4
```

Configure Static IP in Debian Systems

Next, restart network services after entering all the details using the following command.

```
$ sudo systemctl restart networking
```

[You might also like: How to Configure Network Connection Using 'nmcli' Tool]

2. Temporary Configure Static IP Address in Linux

For temporary network configurations, you can use the **ip command** to assign an IP address to a specific interface (**eth2**) on the fly.

```
# ip addr add 172.19.1.10/24 dev eth2
OR
$ sudo ip addr add 172.19.1.10/24 dev eth2
```

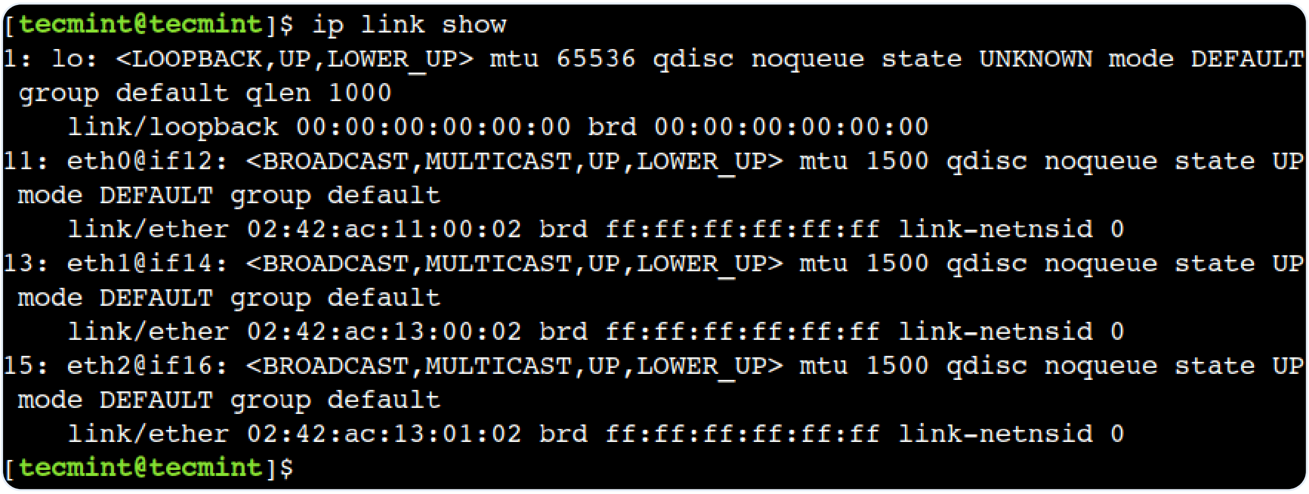
Note: Unfortunately all these settings will be lost after a system restart.

3. How to Display All Network Interfaces

In **ip command**, the link object represents the network interface. We can use the **show** command with it to display all network interfaces.

Now, let's display all network interfaces using the following command:

```
$ ip link show
```



```
[tecmint@tecmint]$ 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
11: eth0@if12: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
 mode DEFAULT group default
    link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
13: eth1@if14: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
 mode DEFAULT group default
    link/ether 02:42:ac:13:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
15: eth2@if16: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
 mode DEFAULT group default
    link/ether 02:42:ac:13:01:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
[tecmint@tecmint]$
```

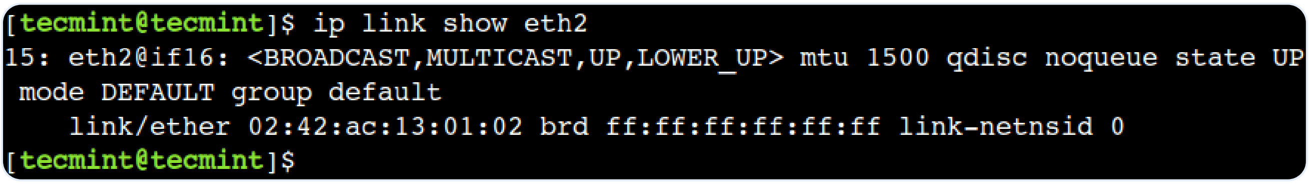
Check IP Addresses of Network Interfaces

The above output shows the details of all network interfaces, such as interface name, flags, status, link address, broadcast address, etc.

4. How to Check an IP Address of a Specific Network Interface

To get the depth information of your individual network interface like IP Address, and MAC Address information, use the following command as shown below.

```
$ ip link show eth2
```



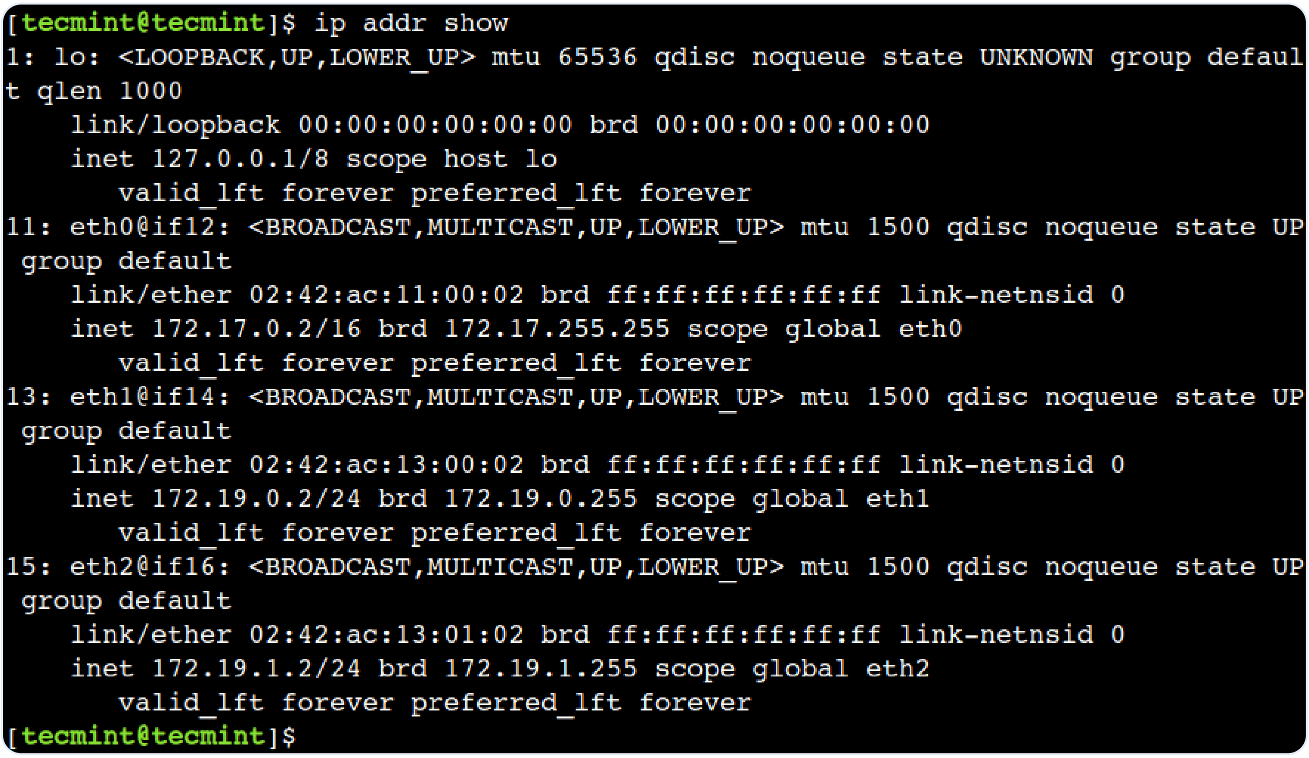
```
[tecmint@tecmint]$ ip link show eth2
15: eth2@if16: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
 mode DEFAULT group default
    link/ether 02:42:ac:13:01:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
[tecmint@tecmint]$
```

Check the IP Address of the Network Interface

So far, we used the link object shows detailed information about the network interfaces. However, it doesn't show the IP address associated with the network interface. To overcome this limitation, we can use the `addr` object with the ip command.

Let's understand this with an example.

```
$ ip addr show
```



```
[tecmint@tecmint]$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t 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
11: eth0@if12: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
 group default
    link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.17.0.2/16 brd 172.17.255.255 scope global eth0
       valid_lft forever preferred_lft forever
13: eth1@if14: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
 group default
    link/ether 02:42:ac:13:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.19.0.2/24 brd 172.19.0.255 scope global eth1
       valid_lft forever preferred_lft forever
15: eth2@if16: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
 group default
    link/ether 02:42:ac:13:01:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.19.1.2/24 brd 172.19.1.255 scope global eth2
       valid_lft forever preferred_lft forever
[tecmint@tecmint]$
```

Check the IP Address of the Interface

Here, we can see that, now output shows the IP addresses of all network interfaces along with other details.

To display the IP address of the individual network interface, just need to provide the network interface name as an argument to the command.

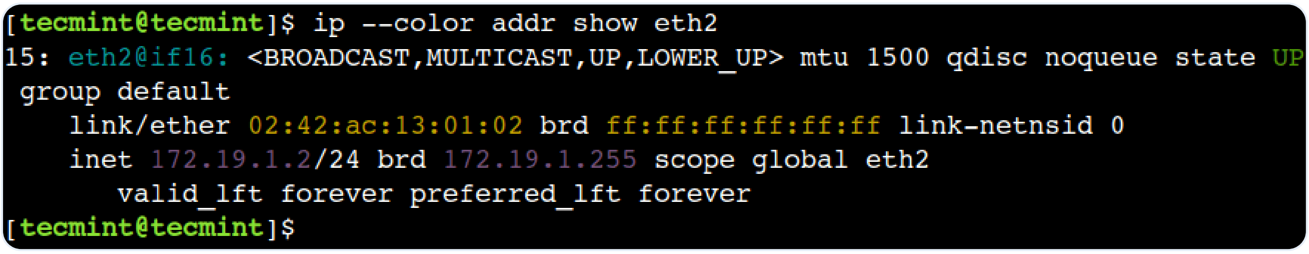
```
$ ip addr show eth2
```

5. How to Display IP Address in Colored Output

The **ip command** shows detailed information about the network objects. However, sometimes we need to take a look at the limited information. In such cases, we can enable the colored output. This option highlights important details in different colors.

Let's use the `--color` option of the command to display the output in different colors:

```
$ ip --color addr show eth2
```



```
[tecmint@tecmint]$ ip --color addr show eth2
15: eth2@if16: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
 group default
    link/ether 02:42:ac:13:01:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.19.1.2/24 brd 172.19.1.255 scope global eth2
       valid_lft forever preferred_lft forever
[tecmint@tecmint]$
```

Show IP Address in Color Output

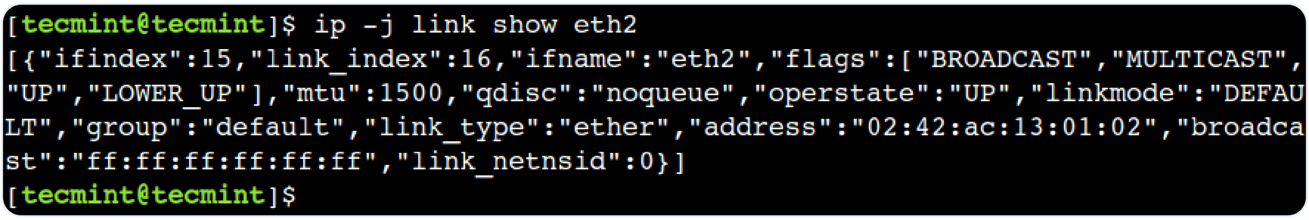
In the above output, we can see that the interface name, ethernet address, and state are highlighted in different colors.

6. How to Display IP Address in JSON Format

In the previous examples, we saw that the **ip command** shows meaningful information. However, it is not an easy task to parse the raw output and extract meaningful information using rudimentary scripts. In such cases, we can instruct the **ip command** to generate the output in a JSON format.

So, let's use the `-j` option with the command to display the same output in a JSON format:

```
$ ip -j link show eth2
```



```
[tecmint@tecmint]$ ip -j link show eth2
[{"ifindex":15,"link_index":16,"ifname":"eth2","flags":["BROADCAST","MULTICAST",
"UP","LOWER_UP"],"mtu":1500,"qdisc":"noqueue","operstate":"UP","linkmode":"DEFAU
LT","group":"default","link_type":"ether","address":"02:42:ac:13:01:02","broadca
st":"ff:ff:ff:ff:ff:ff","link_netnsid":0}]
[tecmint@tecmint]$
```

Show IP Address in JSON Format

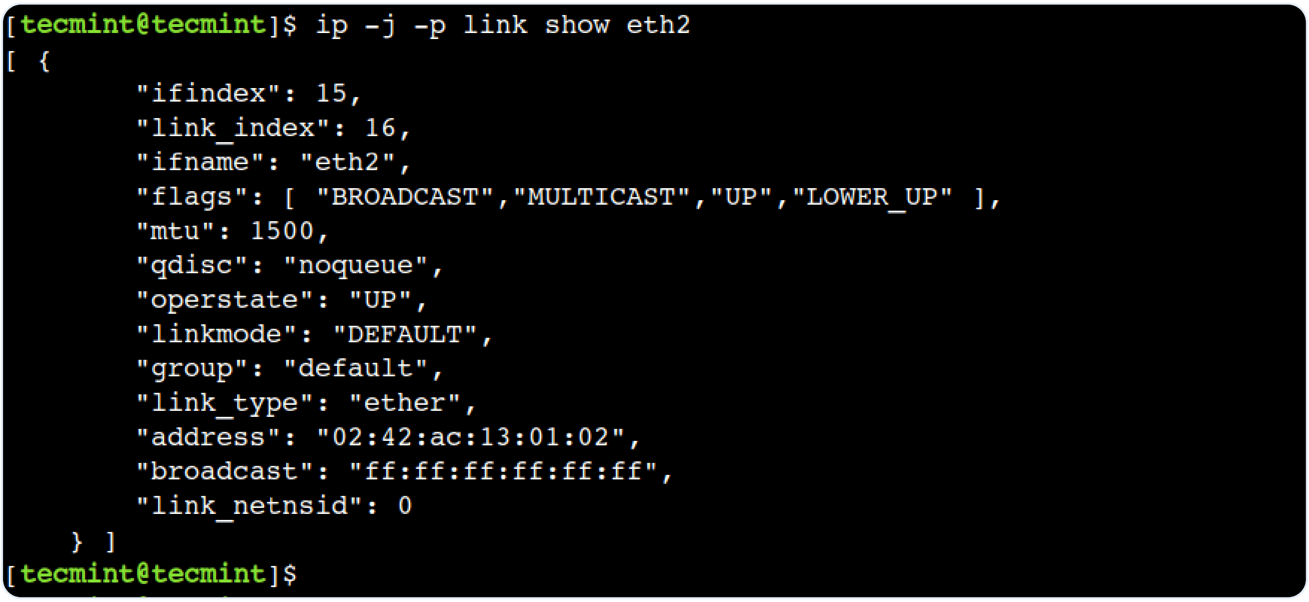
This method comes in handy while doing automation because JSON is a widely accepted format and there are many JSON parser libraries/tools available in various programming languages.

7. How to Make JSON Output More Readable

In the previous example, we used the `-j` option to display output in a JSON format. This default JSON format is compact and space efficient. However, the output is not easy to read due to a lack of indentation.

To overcome this limitation, we can use the `-p` option which makes the output more readable by indenting it. Let's understand this with the below example:

```
$ ip -j -p link show eth2
```



```
[tecmint@tecmint]$ ip -j -p link show eth2
[ {
        "ifindex": 15,
        "link_index": 16,
        "ifname": "eth2",
        "flags": [ "BROADCAST","MULTICAST","UP","LOWER_UP" ],
        "mtu": 1500,
        "qdisc": "noqueue",
        "operstate": "UP",
        "linkmode": "DEFAULT",
        "group": "default",
        "link_type": "ether",
        "address": "02:42:ac:13:01:02",
        "broadcast": "ff:ff:ff:ff:ff:ff",
        "link_netnsid": 0
    } ]
[tecmint@tecmint]$
```

Show IP Address in Readable Format

Here, we can see that the same output is much more readable as compared to the previous examples.

8. How to Remove an IP Address From the Network Interface

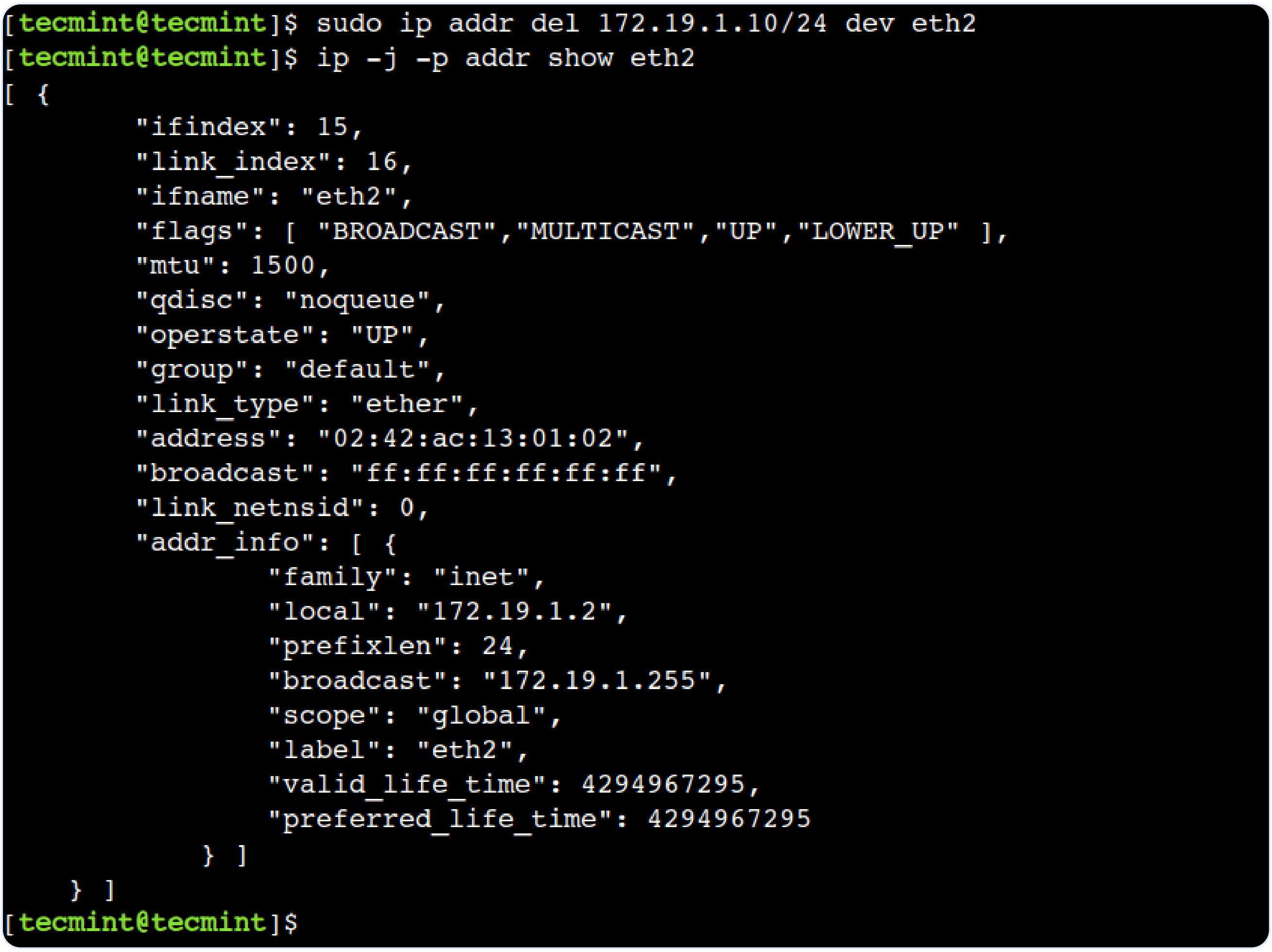
In the previous example, we used the add sub-command to assign an IP address. In a similar way, we can use the **del** sub-command to remove a particular IP address.

The following command will remove an assigned IP address from the given interface (**eth2**).

```
# ip addr del 172.19.1.10/24 dev eth2
OR
$ sudo ip addr del 172.19.1.10/24 dev eth2
```

Now, let's verify that the IP address has been removed:

```
$ ip -j -p addr show eth2
```



```
[tecmint@tecmint]$ sudo ip addr del 172.19.1.10/24 dev eth2
[tecmint@tecmint]$ ip -j -p addr show eth2
[ {
        "ifindex": 15,
        "link_index": 16,
        "ifname": "eth2",
        "flags": [ "BROADCAST","MULTICAST","UP","LOWER_UP" ],
        "mtu": 1500,
        "qdisc": "noqueue",
        "operstate": "UP",
        "group": "default",
        "link_type": "ether",
        "address": "02:42:ac:13:01:02",
        "broadcast": "ff:ff:ff:ff:ff:ff",
        "link_netnsid": 0,
        "addr_info": [ {
                "family": "inet",
                "local": "172.19.1.2",
                "prefixlen": 24,
                "broadcast": "172.19.1.255",
                "scope": "global",
                "label": "eth2",
                "valid_life_time": 4294967295,
                "preferred_life_time": 4294967295
            } ]
    } ]
[tecmint@tecmint]$
```

Remove Network IP Address in Linux

In the above output, we can see that the now **eth2** network interface has only one IP address.

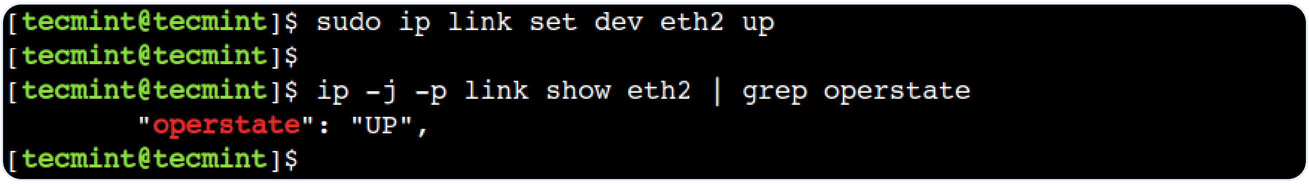
9. How to Enable the Network Interface

The "**up**" flag with interface name (**eth2**) enables a network interface. For example, the following command will activate the **eth2** network interface.

```
# ip link set eth2 up
OR
$ sudo ip link set dev eth2 up
```

Now, let’s check the updated status:

```
$ ip -j -p link show eth2 | grep operstate
```



```
[tecmint@tecmint]$ sudo ip link set dev eth2 up
[tecmint@tecmint]$
[tecmint@tecmint]$ ip -j -p link show eth2 | grep operstate
        "operstate": "UP",
[tecmint@tecmint]$
```

Enable Network Interface in Linux

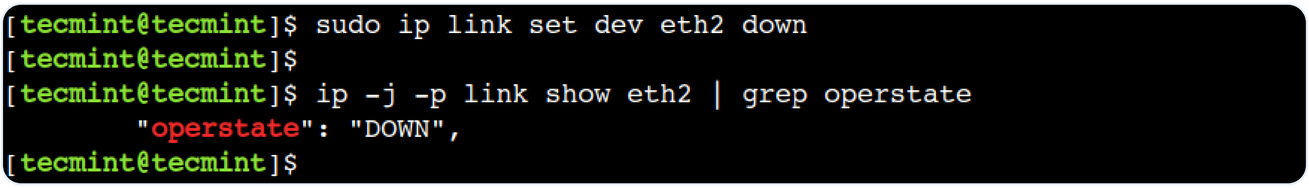
10. How to Disable the Network Interface

The “**down**” flag with interface name (**eth2**) disables a network interface. For example, the following command will De-activates the **eth2** network interface.

```
# ip link set eth2 down
OR
$ sudo ip link set eth2 down
```

Now, let’s check the status of the **eth2** network interface:

```
$ ip -j -p link show eth2 | grep operstate
```



```
[tecmint@tecmint]$ sudo ip link set dev eth2 down
[tecmint@tecmint]$
[tecmint@tecmint]$ ip -j -p link show eth2 | grep operstate
        "operstate": "DOWN",
[tecmint@tecmint]$
```

Disable Network Interface in Linux

The above output shows the modified state of the network interface.

11. How to Flush IP Addresses of Network Interface

In the previous example, we saw how to use a **del** sub-command to remove an IP address. However, sometimes we need to remove all IP addresses of the particular network interface.

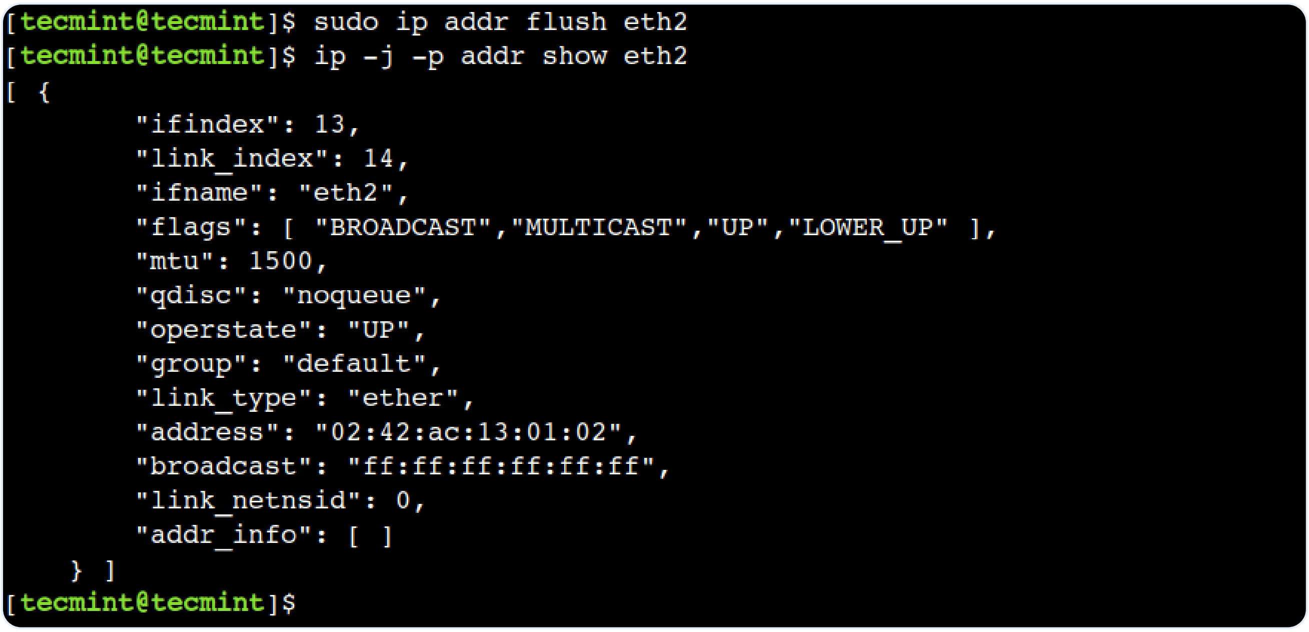
In such cases, we can use the **flush** sub-command.

First, use the **flush** sub-command to remove all the IP addresses of the **eth2** network interface:

```
$ sudo ip addr flush eth2
```

Now, let's check that all IP addresses of the **eth2** network interface have been removed:

```
$ ip -j -p addr show eth2
```



```
[tecmint@tecmint]$ sudo ip addr flush eth2
[tecmint@tecmint]$ ip -j -p addr show eth2
[ {
        "ifindex": 13,
        "link_index": 14,
        "ifname": "eth2",
        "flags": [ "BROADCAST","MULTICAST","UP","LOWER_UP" ],
        "mtu": 1500,
        "qdisc": "noqueue",
        "operstate": "UP",
        "group": "default",
        "link_type": "ether",
        "address": "02:42:ac:13:01:02",
        "broadcast": "ff:ff:ff:ff:ff:ff",
        "link_netnsid": 0,
        "addr_info": [ ]
    } ]
[tecmint@tecmint]$
```

Flush Network IP Addresses in Linux

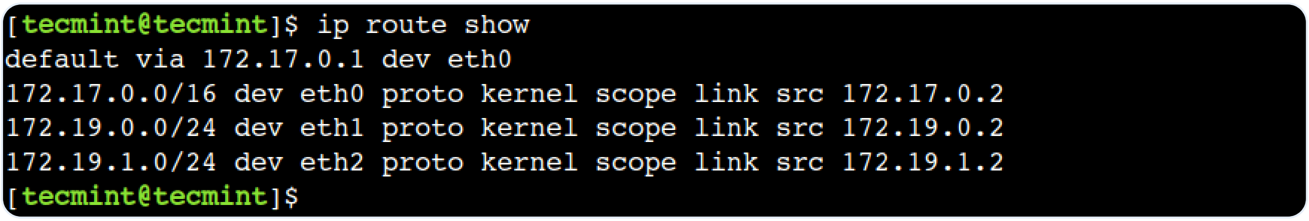
In the above output, the **addr_info** field shows the empty JSON array. This indicates there isn't any IP address associated with the **eth2** network interface.

12. How Do I Check Routing Table

A routing table stores the necessary information to forward a network packet to the correct destination. We can use the **route** object of the ip command to display the routing rules.

Let's use the below command to list the all rules of the routing table:

```
$ ip route show
```



```
[tecmint@tecmint]$ ip route show
default via 172.17.0.1 dev eth0
172.17.0.0/16 dev eth0 proto kernel scope link src 172.17.0.2
172.19.0.0/24 dev eth1 proto kernel scope link src 172.19.0.2
172.19.1.0/24 dev eth2 proto kernel scope link src 172.19.1.2
[tecmint@tecmint]$
```

Check Network Routing Table

In the above output, the first column represents the destination whereas the last column represents the source IP address.

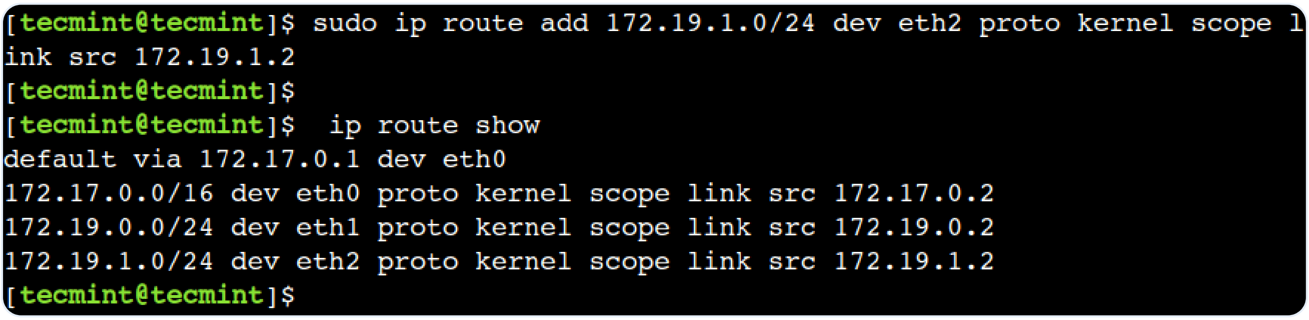
13. How Do I Add New Static Route

Why do you need to add static routes or manual routes, because the traffic must not pass through the default gateway? We need to add static routes to pass traffic from the best way to reach the destination.

```
$ sudo ip route add 172.19.1.0/24 dev eth2 proto kernel scope link src 172.
```

Now, let’s verify that the entry has been added successfully:

```
$ ip route show
```



```
[tecmint@tecmint]$ sudo ip route add 172.19.1.0/24 dev eth2 proto kernel scope l
ink src 172.19.1.2
[tecmint@tecmint]$
[tecmint@tecmint]$  ip route show
default via 172.17.0.1 dev eth0
172.17.0.0/16 dev eth0 proto kernel scope link src 172.17.0.2
172.19.0.0/24 dev eth1 proto kernel scope link src 172.19.0.2
172.19.1.0/24 dev eth2 proto kernel scope link src 172.19.1.2
[tecmint@tecmint]$
```

Add Network Route in Linux

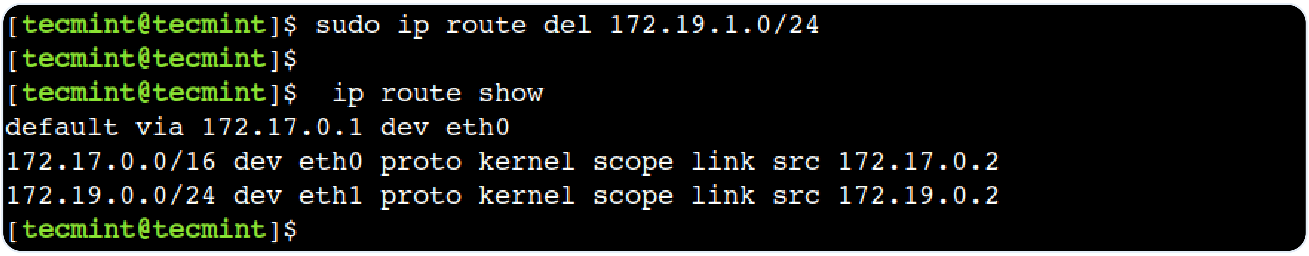
15. How to Remove Static Route

The **del** sub-command removes a particular entry from the routing table. For example, the below command removes the entry of the **eth2** device route:

```
$ sudo ip route del 172.19.1.0/24
```

Now, let's verify that the entry has been removed successfully:

```
$ ip route show
```



```
[tecmint@tecmint]$ sudo ip route del 172.19.1.0/24
[tecmint@tecmint]$
[tecmint@tecmint]$  ip route show
default via 172.17.0.1 dev eth0
172.17.0.0/16 dev eth0 proto kernel scope link src 172.17.0.2
172.19.0.0/24 dev eth1 proto kernel scope link src 172.19.0.2
[tecmint@tecmint]$
```

Remove Network Route in Linux

16. How Do I Add Permanent Static Routes

All the above routes will be lost after a system restart. To add a permanent static route, edit file **/etc/sysconfig/network-scripts/route-eth2** (We are storing static route for (**eth2**). By default, the **route-eth2** file will not be there and need to be created.

Set Permanent Route in RHEL Systems

```
# vi /etc/sysconfig/network-scripts/route-eth2
```

and add the following lines and save and exit.

```
172.19.1.0/24 via 172.19.1.2 dev eth2
```

Set Permanent Route in Debian Systems

Open the file **/etc/network/interfaces** and at the end add the persistence static routes. IP Addresses may differ in your environment.

```
$ sudo vi /etc/network/interfaces
```

```
auto eth2
iface eth2 inet static
address 172.19.50.2
netmask 255.255.255.0
gateway 172.19.50.100
#########{Static Route}###########
up ip route add 172.19.1.0/24 via 172.19.1.2 dev eth2
```

Next, restart network services after entering all the details using the following command.

```
$ sudo systemctl restart networking
```

17. How Do I Add the Default Gateway

In networking, the default gateway plays an important role. It gets used when the routing table doesn't contain any information about the destination.

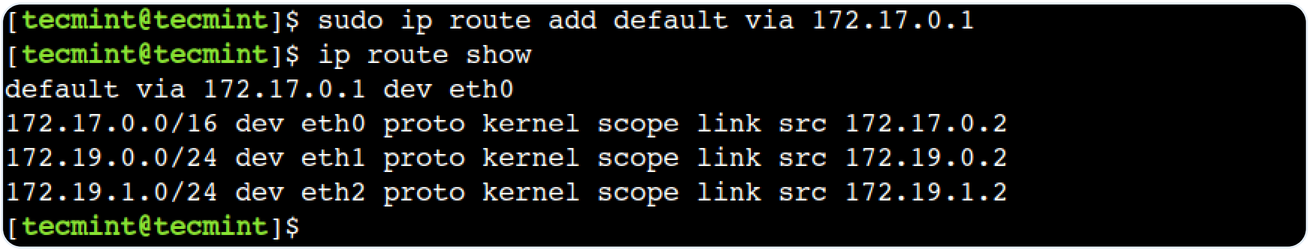
The default gateway can be specified globally or for interface-specific config files. The advantage of the default gateway is that we have more than one NIC present in the system. You can add the default gateway on the fly as shown below the command.

First, let's add an **eth0** network interface as a default gateway:

```
$ sudo ip route add default via 172.17.0.1
```

Now, let's verify the default gateway setting using the following command:

```
$ ip route show
```



```
[tecmint@tecmint]$ sudo ip route add default via 172.17.0.1
[tecmint@tecmint]$ ip route show
default via 172.17.0.1 dev eth0
172.17.0.0/16 dev eth0 proto kernel scope link src 172.17.0.2
172.19.0.0/24 dev eth1 proto kernel scope link src 172.19.0.2
172.19.1.0/24 dev eth2 proto kernel scope link src 172.19.1.2
[tecmint@tecmint]$
```

Add Network Gateway in Linux

Please note that we have executed this command on a test machine. Be careful while using this command in the production environment.

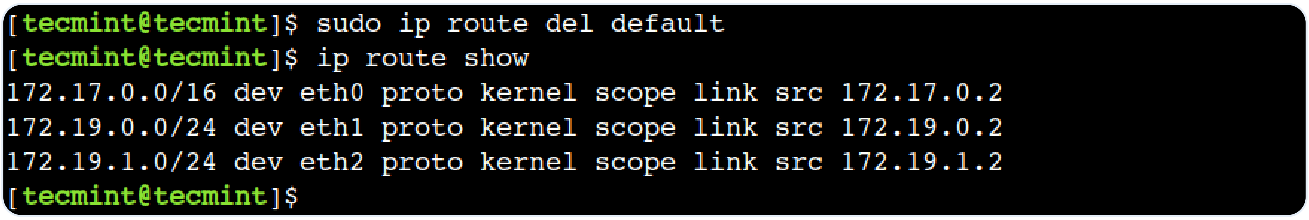
18. How to Remove a Default Gateway

We can use the following command to remove the default gateway:

```
$ sudo ip route del default
```

Now, let’s list the routing table to verify the default gateway has been removed:

```
$ ip route show
```



```
[tecmint@tecmint]$ sudo ip route del default
[tecmint@tecmint]$ ip route show
172.17.0.0/16 dev eth0 proto kernel scope link src 172.17.0.2
172.19.0.0/24 dev eth1 proto kernel scope link src 172.19.0.2
172.19.1.0/24 dev eth2 proto kernel scope link src 172.19.1.2
[tecmint@tecmint]$
```

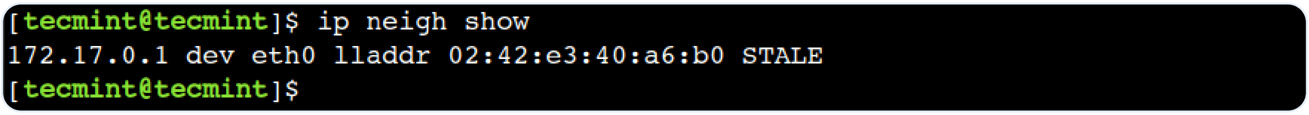
Remove Network Gateway in Linux

19. How to Display ARP Cache

ARP stands for the Address Resolution Protocol, which is used to find the MAC address associated with the particular IP address.

We can use the neigh object with the ip command to display the ARP cache:

```
$ ip neigh show
```



```
[tecmint@tecmint]$ ip neigh show
172.17.0.1 dev eth0 lladdr 02:42:e3:40:a6:b0 STALE
[tecmint@tecmint]$
```

Check ARP Cache in Linux

In the above command, the neigh represents neighboring objects.

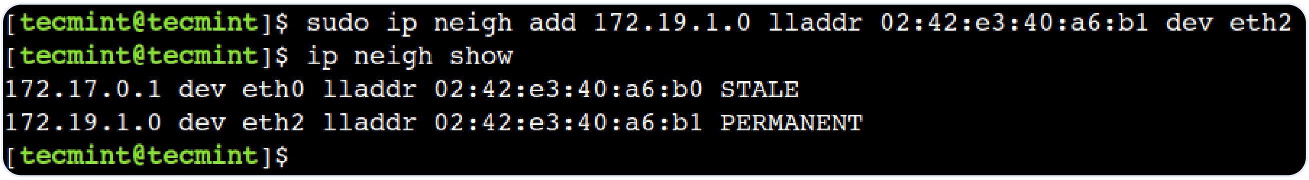
20. How to Add an ARP Entry

To create a new ARP entry, we can use the add sub-command with the **neigh** object.

```
$ sudo ip neigh add 172.19.1.0 lladdr 02:42:e3:40:a6:b1 dev eth2
```

Now, let’s list the ARP cache entries:

```
$ ip neigh show
```



```
[tecmint@tecmint]$ sudo ip neigh add 172.19.1.0 lladdr 02:42:e3:40:a6:b1 dev eth2
[tecmint@tecmint]$ ip neigh show
172.17.0.1 dev eth0 lladdr 02:42:e3:40:a6:b0 STALE
172.19.1.0 dev eth2 lladdr 02:42:e3:40:a6:b1 PERMANENT
[tecmint@tecmint]$
```

Add ARP Cache in Linux

In the above output, we can see the new entry for the **eth2** network interface.

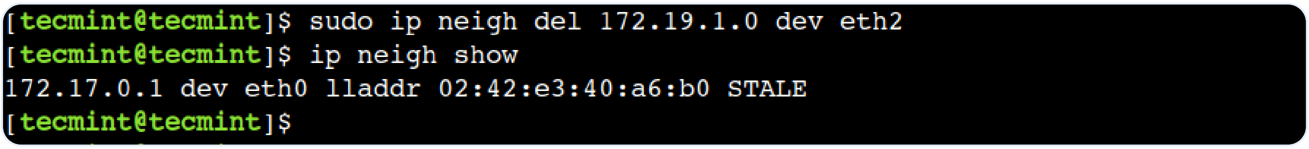
21. How to Remove an ARP Entry

Like other network objects, we can use the **del** sub-command to remove the ARP entry. For example, the below command removes the ARP entry of the **eth2** network interface:

```
$ sudo ip neigh del 172.19.1.0 dev eth2
```

Now, let’s verify that the entry has been removed by listing the ARP cache:

```
$ ip neigh show
```



```
[tecmint@tecmint]$ sudo ip neigh del 172.19.1.0 dev eth2
[tecmint@tecmint]$ ip neigh show
172.17.0.1 dev eth0 lladdr 02:42:e3:40:a6:b0 STALE
[tecmint@tecmint]$
```

Remove ARP Cache in Linux

22. How to Flush the ARP Entries

We can use the **flush** sub-command to remove multiple ARP entries. To understand this, first, add a few ARP entries with the STALE state:

```
$ sudo ip neigh add 172.19.1.0 lladdr 02:42:e3:40:a6:b1 dev eth2 nud stale
$ sudo ip neigh add 172.19.2.0 lladdr 02:42:e3:40:a6:b2 dev eth2 nud stale
$ sudo ip neigh add 172.19.3.0 lladdr 02:42:e3:40:a6:b3 dev eth2 nud stale
```

Next, verify that the new entries have been added successfully:

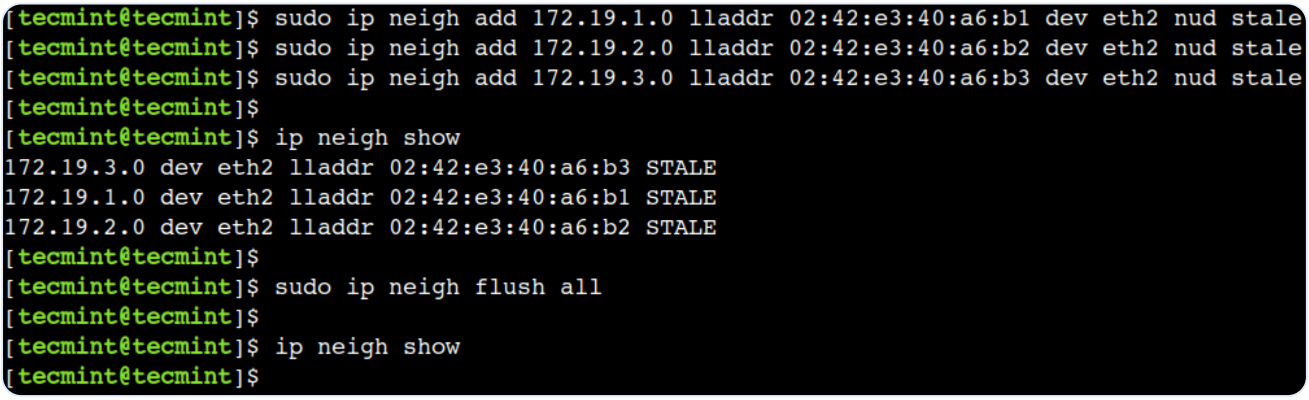
```
$ ip neigh show
```

Then, flush all the entries using the below command:

```
$ sudo ip neigh flush all
```

Finally, verify that all the entries have been removed:

```
$ ip neigh show
```



```
[tecmint@tecmint]$ sudo ip neigh add 172.19.1.0 lladdr 02:42:e3:40:a6:b1 dev eth2 nud stale
[tecmint@tecmint]$ sudo ip neigh add 172.19.2.0 lladdr 02:42:e3:40:a6:b2 dev eth2 nud stale
[tecmint@tecmint]$ sudo ip neigh add 172.19.3.0 lladdr 02:42:e3:40:a6:b3 dev eth2 nud stale
[tecmint@tecmint]$
[tecmint@tecmint]$ ip neigh show
172.19.3.0 dev eth2 lladdr 02:42:e3:40:a6:b3 STALE
172.19.1.0 dev eth2 lladdr 02:42:e3:40:a6:b1 STALE
172.19.2.0 dev eth2 lladdr 02:42:e3:40:a6:b2 STALE
[tecmint@tecmint]$
[tecmint@tecmint]$ sudo ip neigh flush all
[tecmint@tecmint]$
[tecmint@tecmint]$ ip neigh show
[tecmint@tecmint]$
```

Flush ARP in Linux

23. How to Set MTU for Network Interface

MTU stands for **Maximum Transmission Unit**, which represents the largest packet size that can be transmitted in a single transaction. We can manipulate the MTU size as per our performance requirements.

First, let's find the **MTU** of the **eth2** network interface:

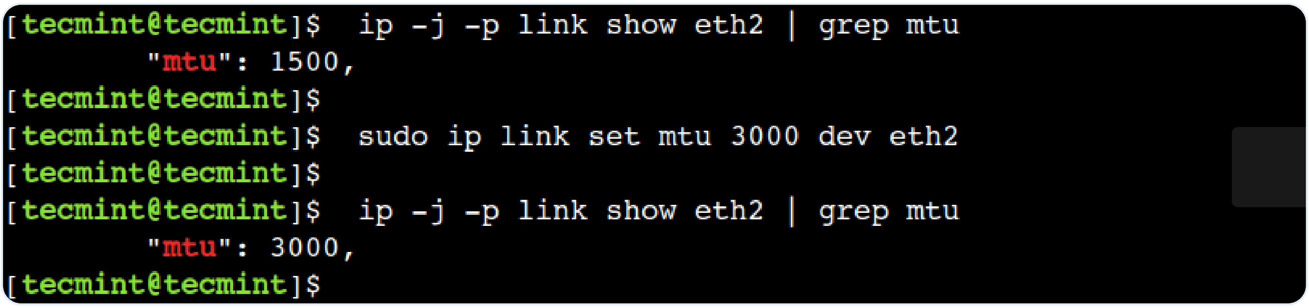
```
$ ip -j -p link show eth2 | grep mtu
```

Next, update the **MTU** size of the **eth2** network interface to **3000**:

```
$ sudo ip link set mtu 3000 dev eth2
```

Finally, verify that the MTU has been updated successfully:

```
$ ip -j -p link show eth2 | grep mtu
```



```
[tecmint@tecmint]$  ip -j -p link show eth2 | grep mtu
        "mtu": 1500,
[tecmint@tecmint]$
[tecmint@tecmint]$  sudo ip link set mtu 3000 dev eth2
[tecmint@tecmint]$
[tecmint@tecmint]$  ip -j -p link show eth2 | grep mtu
        "mtu": 3000,
[tecmint@tecmint]$
```

Set Network MTU in Linux

24. How to Change the Network Mac Address

The **ip command** allows us to change the MAC address of the network interface. To achieve this, we can use the set sub-command with the link object:

First, list the current **MAC** address of the **eth2** network interface:

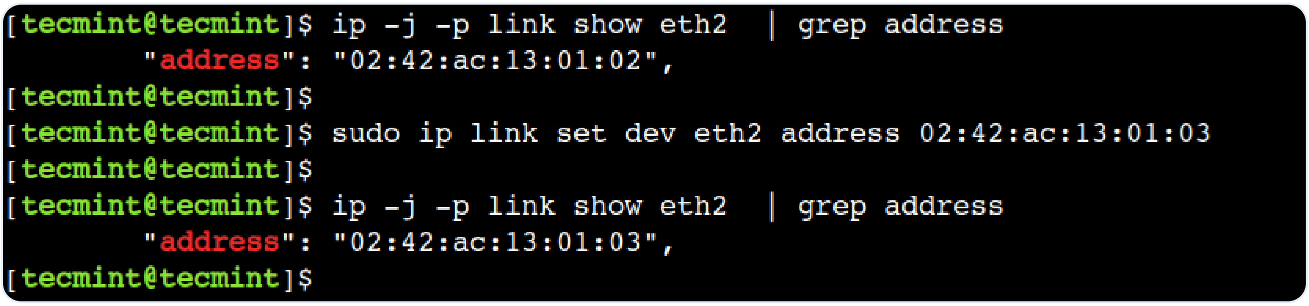
```
$ ip -j -p link show eth2 | grep address
```

Next, change the MAC address of the network interface using the below command:

```
$ sudo ip link set dev eth2 address 02:42:ac:13:01:03
```

Finally, verify that the MAC address has been changed:

```
$ ip -j -p link show eth2 | grep address
```



```
[tecmint@tecmint]$ ip -j -p link show eth2  | grep address
        "address": "02:42:ac:13:01:02",
[tecmint@tecmint]$
[tecmint@tecmint]$ sudo ip link set dev eth2 address 02:42:ac:13:01:03
[tecmint@tecmint]$
[tecmint@tecmint]$ ip -j -p link show eth2  | grep address
        "address": "02:42:ac:13:01:03",
[tecmint@tecmint]$
```

Change Network MAC Address in Linux

Please refer manual page doing **man ip** from the terminal/command prompt to know more about IP Command.

```
$ man ip
```

Conclusion

In this article, we discussed some common examples of the **ip command**. One can use these examples in day-to-day life to perform network administration.

Do you know of any other best example of the ip command in Linux? Let us know your views in the comments below.

Hey TecMint readers,

Exciting news! Every month, our top blog commenters will have the chance to win fantastic rewards, like free Linux eBooks such as **RHCE**, **RHCSA**, **LFCS**, **Learn Linux**, and **Awk**, each worth **$20**!

Learn more about the contest and stand a chance to win by sharing your thoughts below!

GIVEAWAY!
Win eBooks
LEARN LINUX QUICKLY
Level Up Your Career with a 7-Day Linux Mastery Course
By TecMint (@tecmint)
RHCE (EX294) Red Hat Certified Engineer Red Hat
By TecMint (@tecmint)
RHCSA (EX200) Red Hat Certified System Administrator Red Hat
By TecMint (@tecmint)
www.tecmint.com

PREVIOUS ARTICLE:

A Career in Linux is What You Should Be Pursuing In 2024

NEXT ARTICLE:

[How to Connect Remote Linux via SSH ProxyJump and ProxyCommand]

Narendra K

I'm an experienced and passionate software engineer with in-depth experience in Linux, Distributed systems, DevOps, and Cloud. My expertise lies in back-end web development, and the main languages in my tech stack are Java, Spring, Python, and Go. I'm a lifelong learner and an open-source enthusiast.

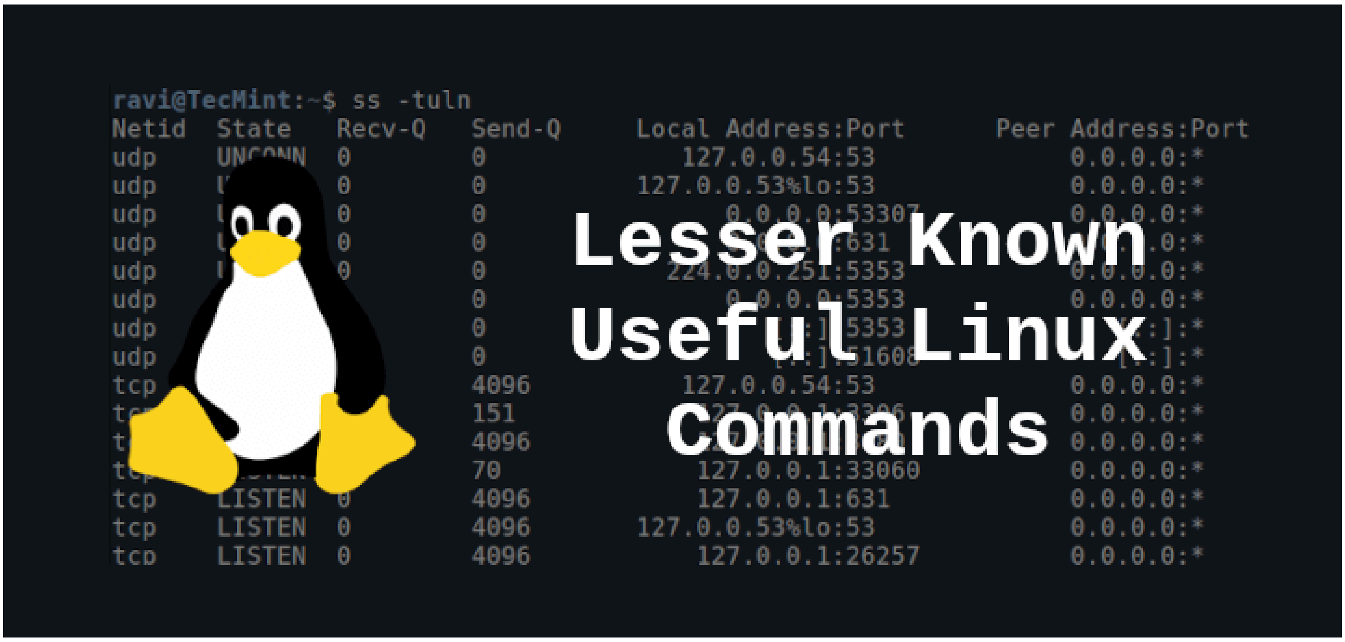
Each tutorial at ***TecMint*** *is created by a team of experienced Linux system administrators* so that it meets our high-quality *standards*.

Related Posts

```
tecmint@tecmint ~/testing $ find . -type f \( -name "*.txt" -o -
name "*.sh" -o -name "*.c" \)
./emails.txt
./script-1.sh
./header.c
./examples.txt
./script.sh
./expenses.txt
```

How to Search Files by Name or Extension Using find Command

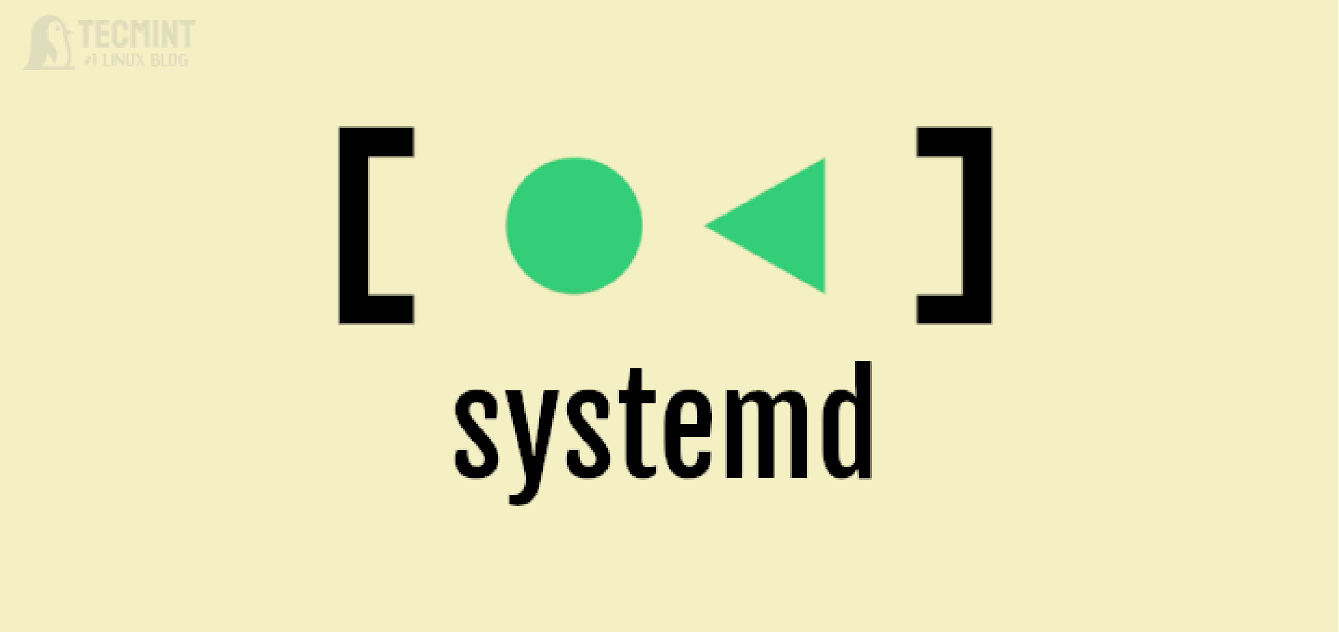
10 Lesser Known Linux Commands – Part 2



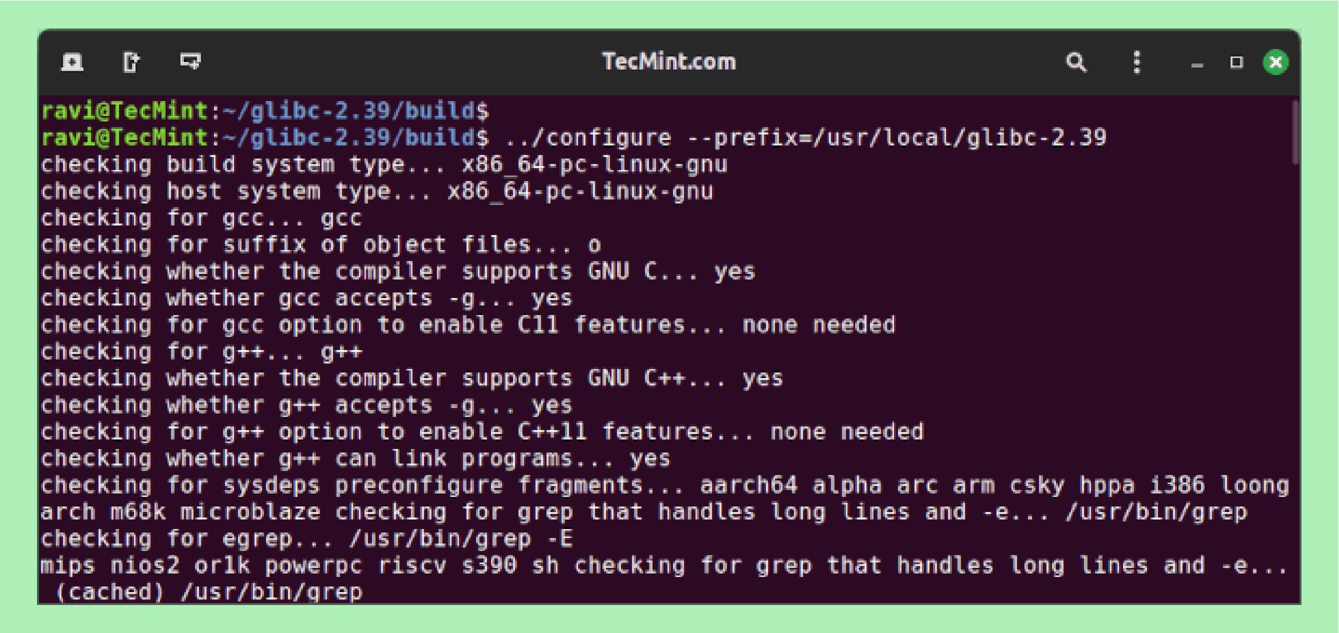
```
ravi@TecMint:~$ ss -tuln
Netid  State    Recv-Q   Send-Q     Local Address:Port        Peer Address:Port
udp    UNCONN   0        0             127.0.0.54:53               0.0.0.0:*
udp             0        0          127.0.0.53%lo:53               0.0.0.0:*
udp             0        0                                         
udp             0        0                                         
udp             0        0            224.0.0.251:5353             0.0.0.0:*
udp                      0                0.0.0.0:5353             0.0.0.0:*
udp                      0                                         
udp                      0                                         
tcp                      4096          127.0.0.54:53               0.0.0.0:*
                         151                                       0.0.0.0:*
                         4096                                      0.0.0.0:*
                         70             127.0.0.1:33060            0.0.0.0:*
tcp    LISTEN   0        4096           127.0.0.1:631              0.0.0.0:*
tcp    LISTEN   0        4096       127.0.0.53%lo:53               0.0.0.0:*
tcp    LISTEN   0        4096           127.0.0.1:26257            0.0.0.0:*
```

11 Lesser Known Useful Linux Commands

26 Security Hardening Tips for Modern Linux Servers



How to Remove Systemd Services on Linux



```
ravi@TecMint:~/glibc-2.39/build$
ravi@TecMint:~/glibc-2.39/build$ ../configure --prefix=/usr/local/glibc-2.39
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking for gcc... gcc
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking for g++... g++
checking whether the compiler supports GNU C++... yes
checking whether g++ accepts -g... yes
checking for g++ option to enable C++11 features... none needed
checking whether g++ can link programs... yes
checking for sysdeps preconfigure fragments... aarch64 alpha arc arm csky hppa i386 loong
arch m68k microblaze checking for grep that handles long lines and -e... /usr/bin/grep
checking for egrep... /usr/bin/grep -E
mips nios2 or1k powerpc riscv s390 sh checking for grep that handles long lines and -e...
 (cached) /usr/bin/grep
```

How to Install and Run Multiple glibc Libraries in Linux

36 Comments

Leave a Reply

Tom

May 23, 2021 at 7:00 pm

I have Debian buster on my laptop. It defaults to nm. But I used network/interfaces years ago.

Reply

BionicBeaver
September 9, 2018 at 4:37 pm

Calm down. Netplan is not in charge yet here, in Ubuntu 18 bionic beaver desktop. We may also run into trouble with resolvconf.

grep '^ *renderer: *NetworkManager' /etc/netplan/01-network-manager-all.yaml
&& {
[do stuff]
} || echo netplan has superseded NetworkManager here\, \"man netplan\"

Reply

Eugene
July 21, 2018 at 8:22 pm

Your info is deprecated. For example:

```
$ cat /etc/network/interfaces
# ifupdown has been replaced by netplan(5) on this system.  Do not
edit.
```

There is netplan in Ubuntu now.

To add permanent Static route, edit file **/etc/sysconfig** file. There is no **/etc/sysconfig** directory at all.

And so on.

Reply

Colin

July 24, 2018 at 4:41 pm

Do you know why Ubuntu has implemented netplan? Is it across all Debian distro's?

We are quite rapidly heading towards 2 distinct branches of Linux; RH based for enterprise, and Ubuntu/Debian for developers.

Reply

Eugene

July 26, 2018 at 4:06 am

I think you should ask Canonical about this but not me. I'm usually remove this and configuring network by networkd.

Reply

BionicBeaver

September 9, 2018 at 4:40 pm

September 9, 2018, Ubuntu bionic beaver desktop, netplan is not in charge here:

grep '^ *renderer: *NetworkManager' /etc/netplan/01-network-manager-all.yaml
&& {
sudo /etc/init.d/networking stop
sudo ip addr add 192.168.1.1/24 dev $eth1

} || echo netplan may be in charge\, \"man netplan\"

Reply

Saniss ElMehdi
February 21, 2018 at 12:08 am

A big thank's

Reply

Cyberat
February 1, 2018 at 3:01 am

Problem with setting static IP.

Done static route through GUI and on **reboot** I get no WAN. Also my "**eth0**" shows up as "**enp2s0**" not sure how it got that way. (wired only, wireless off, set IPv6 as Link Local Only) Does Linux Mint 18.3 require IPv6 to connect to WAN ? The only way to restore my network is set to Auto DHCP and restart.

Reply

Colin

January 17, 2018 at 10:46 pm

Thank you .

Your process was clean and tidy and it worked.

May I ask: I have taken over a slightly mis-configured system in my new position, and found some **ifcfg-***** files that have quotation marks around some of the parameters.

I am wondering if these quotes have an affect on the system, or if they are simply one persons preference?

Please see below:

```
TYPE="Ethernet"
BOOTPROTO="none"
DEVICE="eno50"
ONBOOT="yes"
MASTER=bond0
SLAVE=yes
```

I don’t find any problem with the quotes but wonder why they are there.

Thanks

Reply

max blank

December 2, 2017 at 3:03 pm

Thanks it’s really a useful tutorial. May I ask, how can i config wireless network in red hat 6?

Reply

Mohamed Elhosary
May 7, 2017 at 9:07 pm

Thanks for this great post, but may I have two questions, not related directly to your article, but I think you can help me.

1. How to change IP address to another IP “Say I am from Egypt and I need ip from USA”

2. Is there any way to know the current speed of my internet connection.

Looking for your reply,
Mohamed Elhosary

Reply

Admin

Ravi Saive
May 8, 2017 at 1:24 pm

@Mohamed,

Most welcome to ask any queries, we love to help you out, regarding your first question.

1. I didn’t understand what you mean here? I think you want to change IP address of the server? if yes read this article:

How to Set Static IP Address in Linux

2. There are tons of online speed test tools, one the most popular one is: http://www.speedtest.net/

Reply

Rob

April 15, 2016 at 12:48 am

Thanks for this, any chance you could write something to show how to set up home network, for noobs, when you are trying to access shares from a win 7 machine from a Linux machine?

Reply

Vahid feizi

February 23, 2016 at 8:16 pm

Hi
That was so practical and useful, thank you.

Reply

Helper

January 10, 2016 at 6:32 pm

in my centos 6.5, I am using two interfaces. eth0 for public ip and eth1 private ip. I have made default gateway to public ip. But for other purpose, I need private ip for connetion to other system.

I have added static routes but they are lost when system reboots. So I add persistent route

```
# vi /etc/sysconfig/network-scripts/route-eth1
192.168.20.0/24 via 192.168.50.100 dev eth1
```

when I save this , then the system indicates that eth1 is disconnected. Any suggestion?

Reply

Admin

Ravi Saive

January 11, 2016 at 1:19 pm

@Dipendra,

To keep this route at system boot you should add this entry to /etc/rc.local file, this file executes when system boots so whatever you add this will auto start at system reboot..

Reply

Venkata Subbarao

November 6, 2015 at 12:00 pm

Hi Ravi,

Very good article. Could you please suggest me on the following ?

I have a requirement where I have a Ubuntu system and many devices wants to talk to this and this system wants to talk to them. Some are having different subnet though they are connected to same layer-2 switch. I am in dilemma, whether to use ip aliasing or vlan. Any suggestion please.

Thanks,
Subbarao

Reply

Admin

Ravi Saive

November 6, 2015 at 12:10 pm

@Venkata,
Better I suggest you to assign same network range to all machines or either do a routing..

Reply

Veasna Thong

October 19, 2015 at 9:53 am

Hi Sir!
i have follow by your command line successful , i’m so thank you so much !! :D

Reply

Daniel Moreno

October 16, 2015 at 10:25 pm

Very usefull post
;-)

Reply

pio777

March 30, 2015 at 2:04 pm

Compact size tutorial – good work! I used it once already. Thanks.

Reply

Ji Ji

March 11, 2015 at 3:21 pm

great tutorial

Reply

Anuradha

December 17, 2014 at 4:47 pm

Very good article with simple steps

Reply

xshadow yass

December 13, 2014 at 5:42 pm

hello,
thank you for tutorial that you & your team made they are so good
please i have a problem i have already a public @ IP assigned to my dedicated server
but i bought another failover @ip please how can i add it to my dedicated server so i can assign to it a domain name

Best regards,

PS: my interface file :

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto em1
iface em1 inet dhcp
```

Ubuntu 14.04.1

Reply

DaveB

November 27, 2014 at 4:36 am

In your instructions above, the first time you mention to restart the network services you have:

```
# /etc/init.d/networking restart
```

```
$ sudo /etc/init.d/networking restart
```

This should be

```
# /etc/init.d/network restart
```

```
$ sudo /etc/init.d/network restart
```

like you have near the end of the article.

Reply

Pierre

August 14, 2014 at 2:24 am

I noticed that these two commands behave differently:

```
ip link set eth0 down
&
ip link set dev eth0 down
```

When dev is not used, the link state is brought down and stays down. Otherwise it will come up again soon after executing the command. The difference between the commands does not appear to be documented anywhere. Would you happen to know exactly what happens when running the commands and where the difference in behaviour stems from?

Reply

Gene

July 18, 2014 at 6:14 am

Hey Ravi!

great article as always

but quick question thou what’s the difference between ifup and ip link set ???

Reply

Admin

Ravi Saive

July 18, 2014 at 6:45 pm

Bro, ifup command is used to up network interface and ip command is used to assign IP address, Netmask, Gateway etc..

Reply

prafulla nayak

March 10, 2014 at 8:45 pm

hello,

can we tweak keyboard keys at user level. we use Redhat 5.9 64 bit workstations.

For example i want Num Lock as Tab & so on.....

Thanks in advance

Reply

Deepanjan

September 16, 2013 at 11:28 am

Nice one...

Okk if some one change my ip address ,subnet mask or try to play with it can i view the trail logs or anything simillar ????

Reply

Admin

Ravi Saive

September 16, 2013 at 5:07 pm

Use history command to view last executed commands.

Reply

venkatesh

November 23, 2013 at 6:00 pm

hello,

My static ipv4 address was not showing on eth0 but it is showing on ifconfig... I want to change my ip address how it is possible without eth0.

Reply

nik

August 14, 2013 at 9:48 pm

Superb article……

Reply

Mandar

July 22, 2013 at 10:10 pm

Great article first of all !

This command is very much similar to “IFCONFIG”.
“sudo ifconfig eth1 up” replaced by “sudo ip link set eth1 up”.

That’s really cool !

Reply

manoj

May 7, 2013 at 11:31 am

provide tips for gcov and lcov in linux

Reply

Admin

Ravi Saive

May 7, 2013 at 4:32 pm

Never heard about these tools, but we do try our best to provide you an how-to soon. Please stay updated..

Reply

Got Something to Say? Join the Discussion...

Thank you for taking the time to share your thoughts with us. We appreciate your decision to leave a comment and value your contribution to the discussion. It's important to note that we moderate all comments in accordance with our comment policy to ensure a respectful and constructive conversation.

Rest assured that your email address will remain private and will not be published or shared with anyone. We prioritize the privacy and security of our users.

Name *

Email *

□ Save my name, email, and website in this browser for the next time I comment.

Post Comment

Search...

Linux Commands and Tools

fdupes – A Command Line Tool to Find and Delete Duplicate Files in Linux

How to Enable Syntax Highlighting in Vi/Vim Editor

How to Transfer Files Between Two Computers using nc and pv Commands

Newsroom – A Modern CLI to Get Your Favorite News in Linux

How to Install and Use ‘yum-utils’ to Maintain Yum and Boost its Performance

How to Use ‘Yum History’ to Find Out Installed or Removed Packages Info

Linux Server Monitoring Tools

Conky – A System Monitor Tool for Linux Desktop

How to Set Priority of a Running Process in Linux

4 Ways to Watch or Monitor Log Files in Real Time

How to Install Nagios XI on Ubuntu 22.04

How to Install 'atop' to Monitor Logging Activity of Linux System Processes

How to Monitor System Usage, Outages and Troubleshoot Linux Servers – Part 9

Learn Linux Tricks & Tips

How to Count Number of Files and Subdirectories inside a Given Directory

How to Create a New Ext4 File System (Partition) in Linux

How to Monitor Progress of (Copy/Backup/Compress) Data using 'pv' Command

How to Find Out List of All Open Ports in Linux

Find Top Running Processes by Highest Memory and CPU Usage in Linux

How to Run MySQL/MariaDB Queries Directly from the Linux Command Line

Best Linux Tools

15 Best Kali Linux Web Penetration Testing Tools

Top 5 Open-Source Productivity Tools for Linux

8 Useful GUI Email Clients for Linux Desktop

4 Best Tools for Creating Fillable PDF Forms on Linux

19 Best Open Source WYSIWYG HTML Editors in 2023

5 Most Frequently Used Open Source Shells for Linux