Network administration:

Old Networking package used in RHEL 6 Network it was very limited called network.service.

In RHEL 7 and RHEL 8 we used Network Manager Tool It provides, more features like Creating profiles (Home, Work and Public) for each network Interface and the type of this profiles can be dedicated to bonding, teaming, wi-fi and Ethernet.

Note: each profile is active at a time, but we cannot use two profiles in the same time for the same NIC.

There is two ways to change network configuration one way is temporary means it is lost after the restart (ip tool) or permanent way be save our network configuration in files to be initiated by the NetworkManager service (nmtui, nmcli tools).

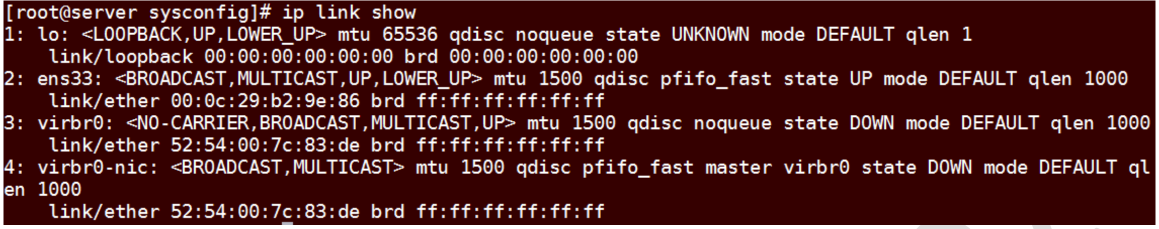
In RHEL8 we must be aware of the network naming so the name of the NIC card is called device. Older versions of Red Hat Enterprise Linux used names like eth0, eth1, and eth2 for each network interface. Newer versions of Red Hat Enterprise Linux use a different naming system. Instead of being based on detection order, the names of network interfaces are assigned based on information from the firmware, the PCI bus topology, and type of network device. In that device we can have multiple connection for example connection that connect to my home network and connection that connects to Work.

Network manager service save the network configuration in /etc/sysconfig/network-scripts/ifname.cfg (IP Config), DNS Config in /etc/resolv.conf, /etc/hosts, Machine name in /etc/hostname and /etc/services for Network Services.

NetworkManager is a daemon that monitors and manages network settings, In addition to the daemon, there is a GNOME notification are applet that provides network status info.

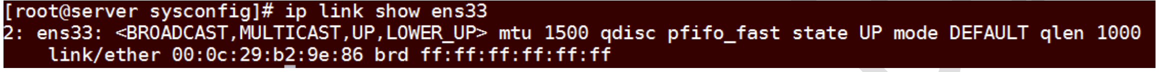
When working with IP address configuration it will be either in two ways. The first is to get the IP configuration Via DHCP server or the other way via static configuration.

1. Using IP Commands:
   To know the help of IP command we use IP help command, here we need to see the important sections link, addr and route.
   Link Means the physical layer of the device



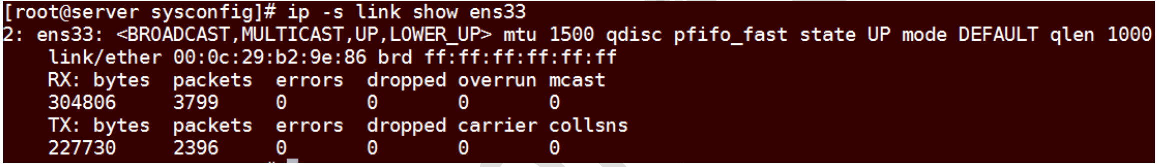
```
[root@server sysconfig]# ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT qlen 1000
    link/ether 00:0c:29:b2:9e:86 brd ff:ff:ff:ff:ff:ff
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DEFAULT qlen 1000
    link/ether 52:54:00:7c:83:de brd ff:ff:ff:ff:ff:ff
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast master virbr0 state DOWN mode DEFAULT ql
en 1000
    link/ether 52:54:00:7c:83:de brd ff:ff:ff:ff:ff:ff
```

To show specific device:



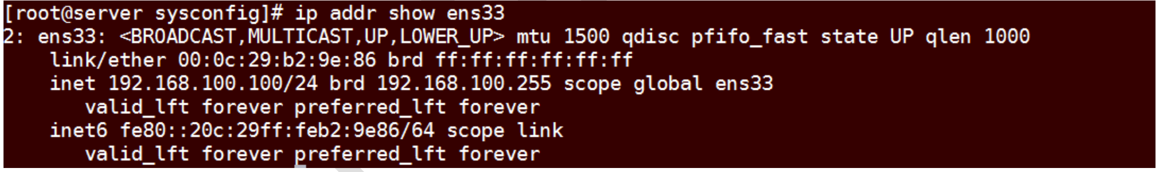
```
[root@server sysconfig]# ip link show ens33
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT qlen 1000
    link/ether 00:0c:29:b2:9e:86 brd ff:ff:ff:ff:ff:ff
```

To show statistics about specific device:
==>will show statistics TX, RX, Error, dropped.



```
[root@server sysconfig]# ip -s link show ens33
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT qlen 1000
    link/ether 00:0c:29:b2:9e:86 brd ff:ff:ff:ff:ff:ff
    RX: bytes  packets  errors  dropped overrun mcast
    304806     3799     0       0       0       0
    TX: bytes  packets  errors  dropped carrier collsns
    227730     2396     0       0       0       0
```

To show the current address configuration of specific NIC:

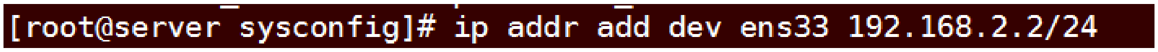


```
[root@server sysconfig]# ip addr show ens33
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:0c:29:b2:9e:86 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.100/24 brd 192.168.100.255 scope global ens33
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:feb2:9e86/64 scope link
       valid_lft forever preferred_lft forever
```

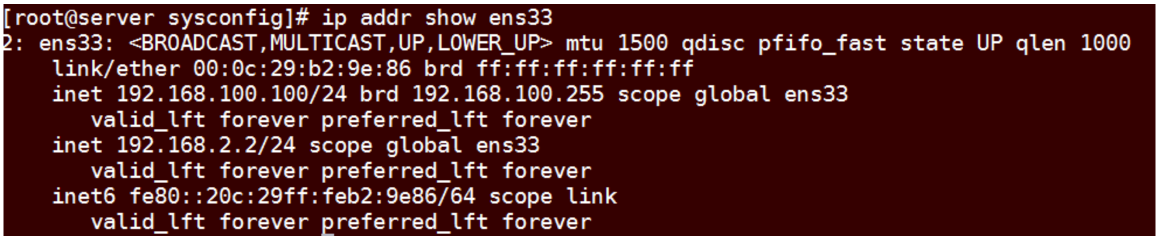
NOTE:

1- up →an active interface has the status of up
2- the link line specifies the hardware (MAC) address of the device
3- the inet line show the IPv4 address and prefix
4- the broadcast address, scope and device name are also on this line
5- the inet6 line show IPv6 info

If we want to add address to a NIC:



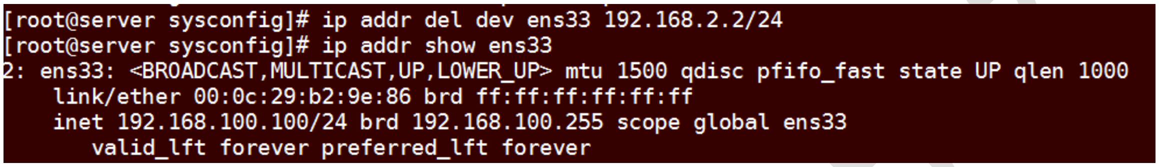
```
[root@server sysconfig]# ip addr add dev ens33 192.168.2.2/24
```



```
[root@server sysconfig]# ip addr show ens33
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:0c:29:b2:9e:86 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.100/24 brd 192.168.100.255 scope global ens33
       valid_lft forever preferred_lft forever
    inet 192.168.2.2/24 scope global ens33
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:feb2:9e86/64 scope link
       valid_lft forever preferred_lft forever
```

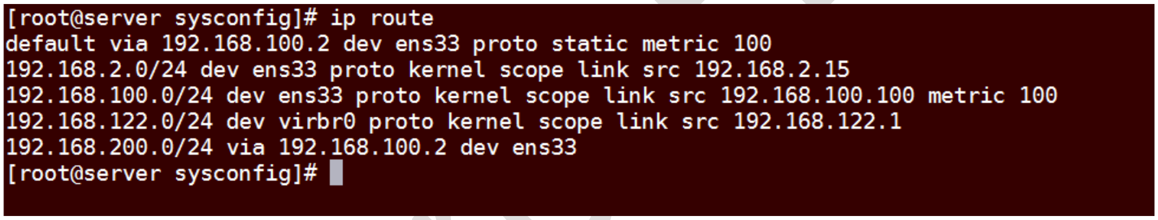
Note: we have to address with its subnet mask otherwise It will be /32.

If we want to delete the address from the interface



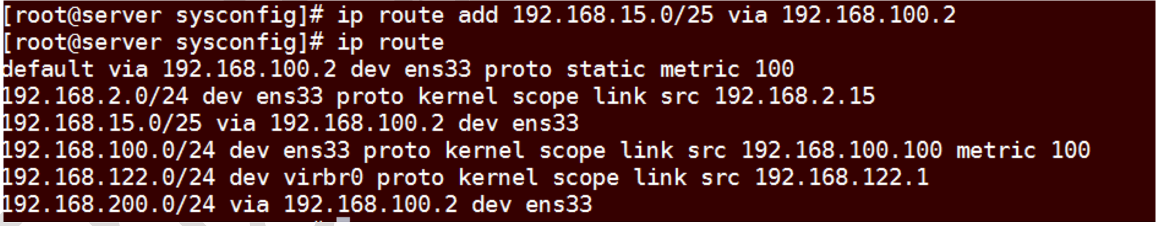
```
[root@server sysconfig]# ip addr del dev ens33 192.168.2.2/24
[root@server sysconfig]# ip addr show ens33
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:0c:29:b2:9e:86 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.100/24 brd 192.168.100.255 scope global ens33
       valid_lft forever preferred_lft forever
```

To show the device R.T.:
==>will display the Network , Interface ip , GW



```
[root@server sysconfig]# ip route
default via 192.168.100.2 dev ens33 proto static metric 100
192.168.2.0/24 dev ens33 proto kernel scope link src 192.168.2.15
192.168.100.0/24 dev ens33 proto kernel scope link src 192.168.100.100 metric 100
192.168.122.0/24 dev virbr0 proto kernel scope link src 192.168.122.1
192.168.200.0/24 via 192.168.100.2 dev ens33
[root@server sysconfig]# 
```

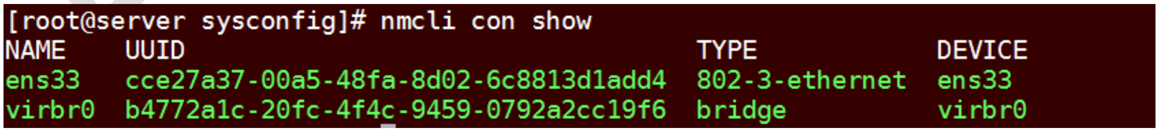
To add a route to the routing tabel:



```
[root@server sysconfig]# ip route add 192.168.15.0/25 via 192.168.100.2
[root@server sysconfig]# ip route
default via 192.168.100.2 dev ens33 proto static metric 100
192.168.2.0/24 dev ens33 proto kernel scope link src 192.168.2.15
192.168.15.0/25 via 192.168.100.2 dev ens33
192.168.100.0/24 dev ens33 proto kernel scope link src 192.168.100.100 metric 100
192.168.122.0/24 dev virbr0 proto kernel scope link src 192.168.122.1
192.168.200.0/24 via 192.168.100.2 dev ens33
```

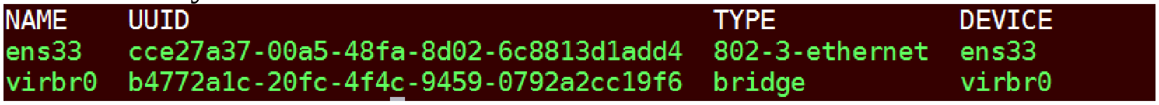
a. NMCLI:

To show the connections:



```
[root@server sysconfig]# nmcli con show
NAME    UUID                                  TYPE            DEVICE
ens33   cce27a37-00a5-48fa-8d02-6c8813d1add4  802-3-ethernet  ens33
virbr0  b4772a1c-20fc-4f4c-9459-0792a2cc19f6  bridge          virbr0
```

To show only the active connections:

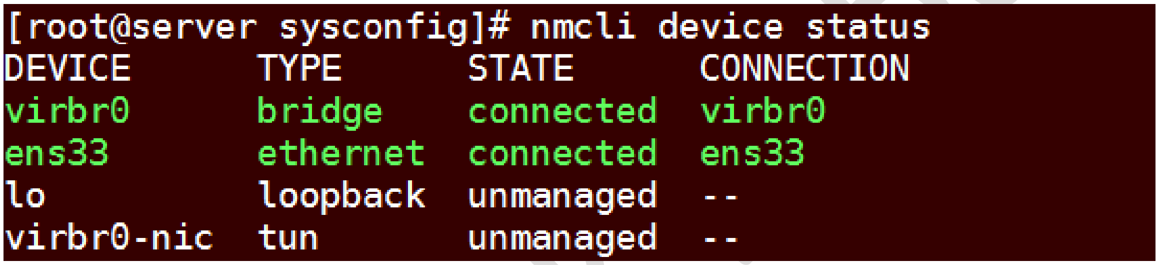


```
NAME    UUID                                  TYPE            DEVICE
ens33   cce27a37-00a5-48fa-8d02-6c8813d1add4  802-3-ethernet  ens33
virbr0  b4772a1c-20fc-4f4c-9459-0792a2cc19f6  bridge          virbr0
```

Here we have Two Profiles the first is ens33 for the physical Ethernet network 802-3-ethernet connect with Network interface or device ens33. the other profile is Bridge profile (virbr0) which used to connect multiple VMs to this virtual bridge and connected to the device or vNIC virbr0.

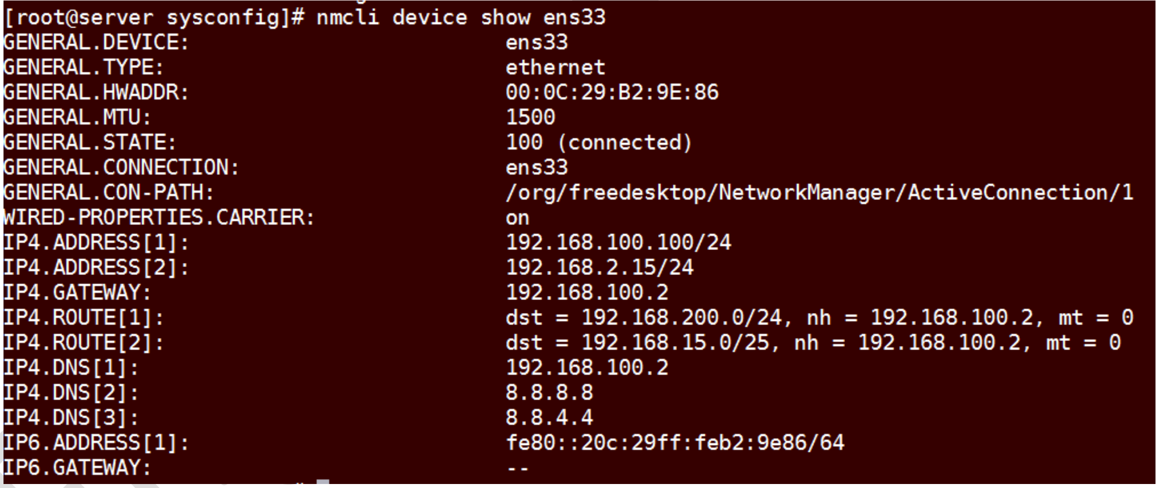
Note: to make each profile active from nmcli tool we use auto connect or from the default location of network Profiles /etc/sysclonfig/network-scripts/ifcfg-ens33 we use the section ONBOOt = yes.

To show more details about the connections:
[root@server sysconfig]# nmcli con show "ens33" | less
To see the devices status:



```
[root@server sysconfig]# nmcli device status
DEVICE      TYPE      STATE      CONNECTION
virbr0      bridge    connected  virbr0
ens33       ethernet  connected  ens33
lo          loopback  unmanaged  --
virbr0-nic  tun       unmanaged  --
```

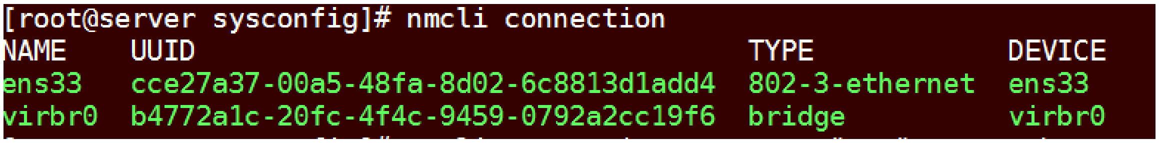
To see details about specific device:



```
[root@server sysconfig]# nmcli device show ens33
GENERAL.DEVICE:                         ens33
GENERAL.TYPE:                           ethernet
GENERAL.HWADDR:                         00:0C:29:B2:9E:86
GENERAL.MTU:                            1500
GENERAL.STATE:                          100 (connected)
GENERAL.CONNECTION:                     ens33
GENERAL.CON-PATH:                       /org/freedesktop/NetworkManager/ActiveConnection/1
WIRED-PROPERTIES.CARRIER:               on
IP4.ADDRESS[1]:                         192.168.100.100/24
IP4.ADDRESS[2]:                         192.168.2.15/24
IP4.GATEWAY:                            192.168.100.2
IP4.ROUTE[1]:                           dst = 192.168.200.0/24, nh = 192.168.100.2, mt = 0
IP4.ROUTE[2]:                           dst = 192.168.15.0/25, nh = 192.168.100.2, mt = 0
IP4.DNS[1]:                             192.168.100.2
IP4.DNS[2]:                             8.8.8.8
IP4.DNS[3]:                             8.8.4.4
IP6.ADDRESS[1]:                         fe80::20c:29ff:feb2:9e86/64
IP6.GATEWAY:                            --
```

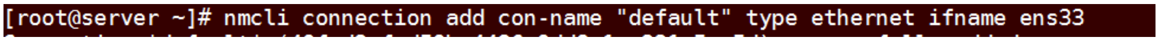
To create network connection with nmcli:
When creating a new connection with nmcli, the order of the argument is important, the common arguments appear first and must include the type and interface, next, specify any type-specific argument and finally specify the ipaddress, prefix and gateway info. Multiple addresses may be specified for a single device, additional setting such as DNS is set as modification once the connection exists.

Show connections:



```
[root@server sysconfig]# nmcli connection
NAME    UUID                                  TYPE            DEVICE
ens33   cce27a37-00a5-48fa-8d02-6c8813d1add4  802-3-ethernet  ens33
virbr0  b4772a1c-20fc-4f4c-9459-0792a2cc19f6  bridge          virbr0
```

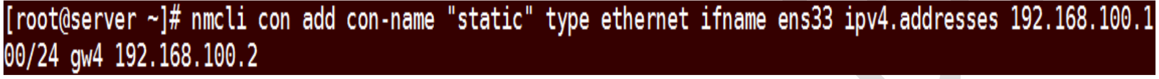
To add new connection:



```
[root@server ~]# nmcli connection add con-name "default" type ethernet ifname ens33
```

Note: If auto-connect is enabled with the previous profile and there is no any address specified so it will take the Network Configuration from the DHCP server if it is available in this network.

To add connection with IP and Gateway:



```
[root@server ~]# nmcli con add con-name "static" type ethernet ifname ens33 ipv4.addresses 192.168.100.1
00/24 gw4 192.168.100.2
```

Note: the device must match the connection ifname, if it does not match the connection cannot be up

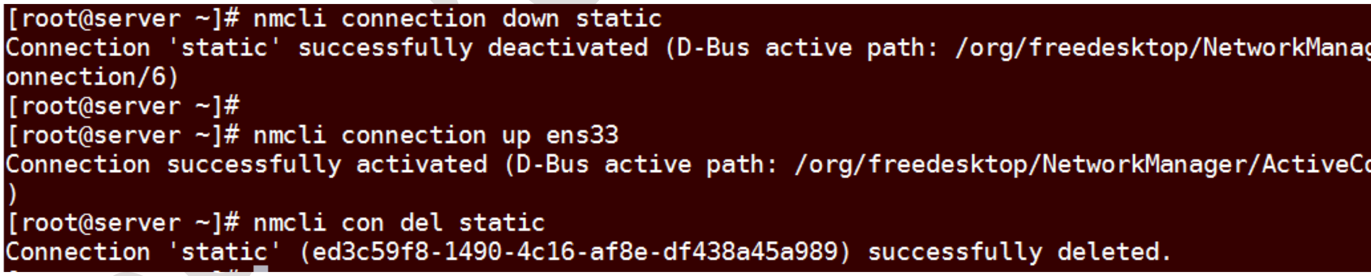
We first need to down one of the interfaces:

[root@server ~]# nmcli connection down "ens33"

We need to make the connection up:

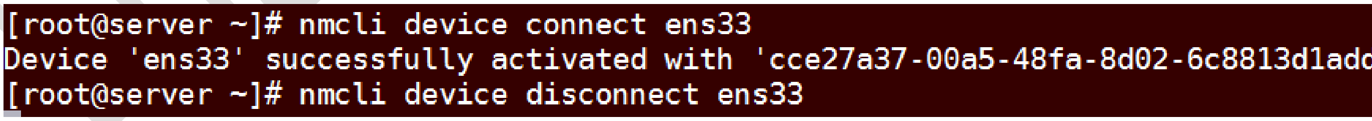
[root@server ~]# nmcli connection up "static"

If we need to delete a connection it must be down first:



```
[root@server ~]# nmcli connection down static
Connection 'static' successfully deactivated (D-Bus active path: /org/freedesktop/NetworkManag
onnection/6)
[root@server ~]#
[root@server ~]# nmcli connection up ens33
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveC
)
[root@server ~]# nmcli con del static
Connection 'static' (ed3c59f8-1490-4c16-af8e-df438a45a989) successfully deleted.
```

To connect or disconnect device:

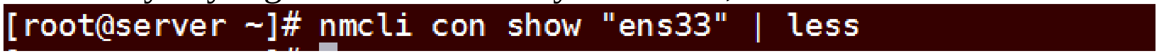


```
[root@server ~]# nmcli device connect ens33
Device 'ens33' successfully activated with 'cce27a37-00a5-48fa-8d02-6c8813d1add
[root@server ~]# nmcli device disconnect ens33
```

Note: the type of the device can be either ethernet, wlan or other like VLAns

Modify network interfaces with nmcli:

To modify any argument we need key and value, to show the values



```
[root@server ~]# nmcli con show "ens33" | less
```

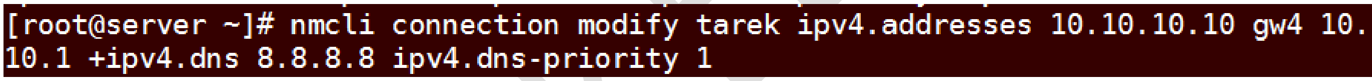
1- turn off auto connect
nmcli con mod “static” connection.autoconnect no

2- specify DNS server
nmcli con mod “static” ipv4.dns 172.25.0.254

3- some configuration arguments may have added or removed. Add a +/- symbol in front of the argument. Add a secondary DNS server
nmcli con mod “static” +ipv4.dns 8.8.8.8

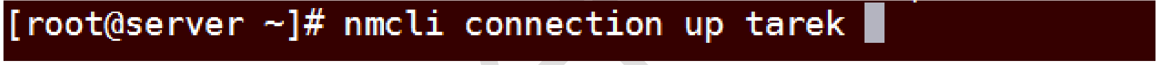
4- replace the static IP address and gateway
nmcli con mod “static” ipv4.addresses “172.25.0.10/24 172.25.0.254”

5- add secondary IP address without a gateway
nmcli con mod “static” +ipv4.addresses 10.10.10.10/16



```
[root@server ~]# nmcli connection modify tarek ipv4.addresses 10.10.10.10 gw4 10.
10.1 +ipv4.dns 8.8.8.8 ipv4.dns-priority 1
```

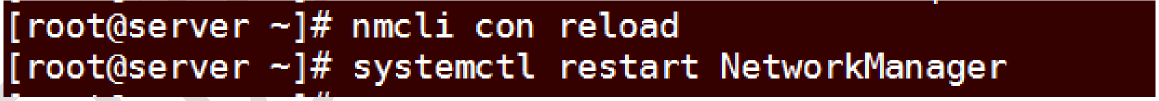
To make this changes activated:



```
[root@server ~]# nmcli connection up tarek
```

To change the connection from dhcp to static configuration we have to change the ipv4.method
[root@server ~]# nmcli connection modify tarek ipv4.method manual

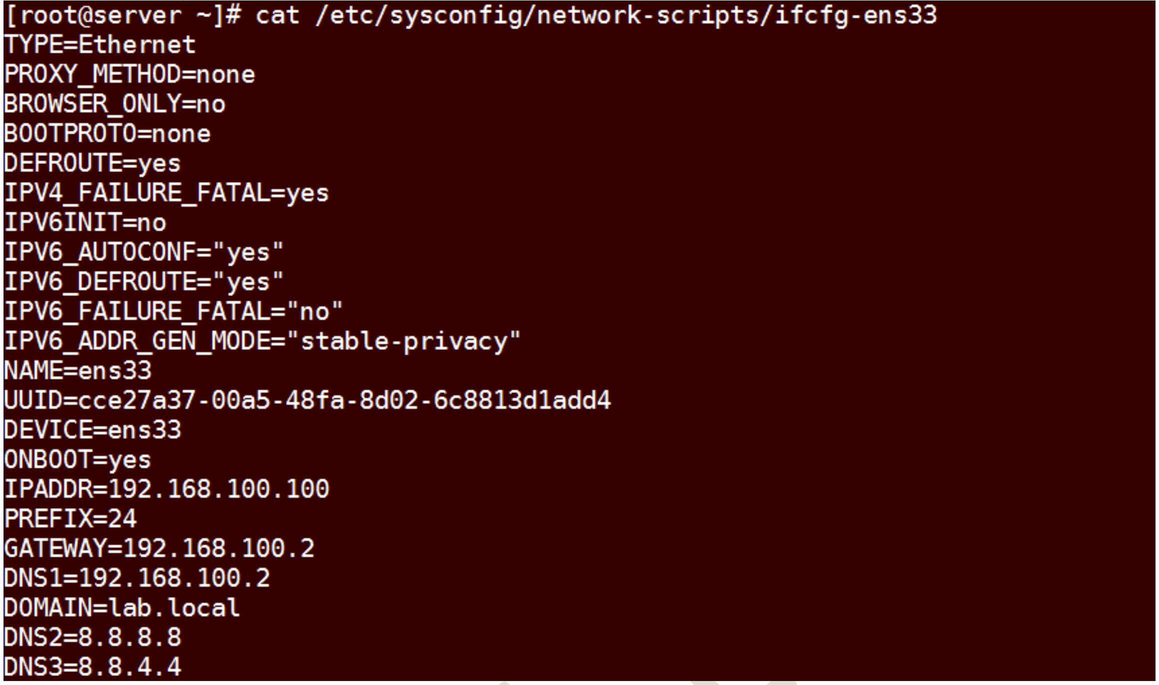
Note: Editing the network configuration using nmtui (graphical tool), needs from you to reload the configuration settings of the network using the following:



```
[root@server ~]# nmcli con reload
[root@server ~]# systemctl restart NetworkManager
```

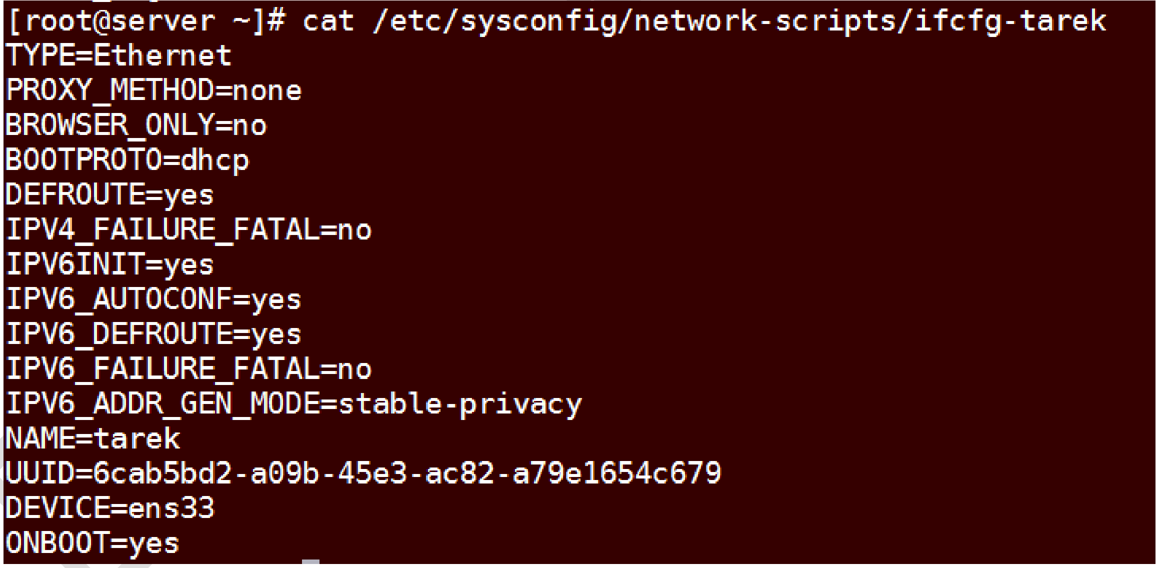
Editing network configuration files:
It is possible to configure the network by editing interface configuration files. These files are usually named /etc/sysconfig/network-scripts/ifcfg-name, where name refer to the name of the device or connection that the configuration file control. The following are standard variables found in the file used for static or dynamic configuration.

Static configuration:



```
[root@server ~]# cat /etc/sysconfig/network-scripts/ifcfg-ens33
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=none
DEFROUTE=yes
IPV4_FAILURE_FATAL=yes
IPV6INIT=no
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_FAILURE_FATAL="no"
IPV6_ADDR_GEN_MODE="stable-privacy"
NAME=ens33
UUID=cce27a37-00a5-48fa-8d02-6c8813d1add4
DEVICE=ens33
ONBOOT=yes
IPADDR=192.168.100.100
PREFIX=24
GATEWAY=192.168.100.2
DNS1=192.168.100.2
DOMAIN=lab.local
DNS2=8.8.8.8
DNS3=8.8.4.4
```

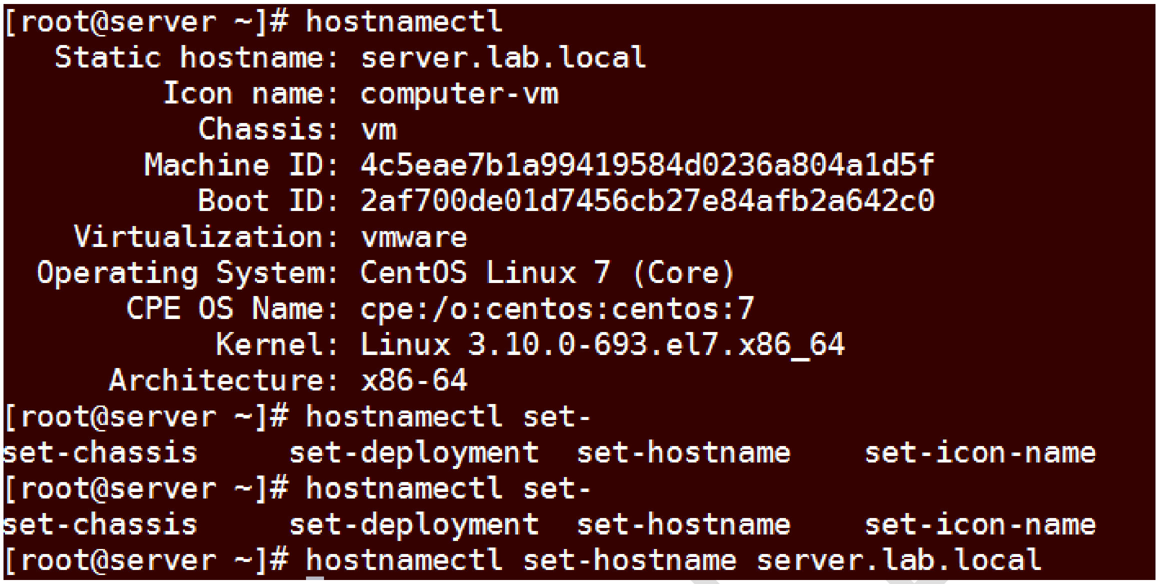
Dynamic configuration:



```
[root@server ~]# cat /etc/sysconfig/network-scripts/ifcfg-tarek
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=dhcp
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=tarek
UUID=6cab5bd2-a09b-45e3-ac82-a79e1654c679
DEVICE=ens33
ONBOOT=yes
```

Changing the Hostname:

To change the hostname which is on /etc/hostname we use tool called hostnamectl:



```
[root@server ~]# hostnamectl
   Static hostname: server.lab.local
         Icon name: computer-vm
           Chassis: vm
        Machine ID: 4c5eae7b1a99419584d0236a804a1d5f
           Boot ID: 2af700de01d7456cb27e84afb2a642c0
    Virtualization: vmware
  Operating System: CentOS Linux 7 (Core)
       CPE OS Name: cpe:/o:centos:centos:7
            Kernel: Linux 3.10.0-693.el7.x86_64
      Architecture: x86-64
[root@server ~]# hostnamectl set-
set-chassis     set-deployment  set-hostname    set-icon-name
[root@server ~]# hostnamectl set-
set-chassis     set-deployment  set-hostname    set-icon-name
[root@server ~]# hostnamectl set-hostname server.lab.local
```

To change the Hosts file which is in /etc/hosts we edit it directly using vim tool, but if we need to change the configuration of dns in /etc/resolv.conf we must use either nmtui or nmcli tool with the restriction of reload the network configuration or Network Manager settings.

NOTE: the static hostname is stored in /etc/hostname, previous versions of RHEL stored the hostname as a variable in the /etc/sysconfig/network file

Note: we are using host command and dig command to resolve the DNS Queries

[root@server ~]# netstat -ntulp ==> to show the ports and application that run this port. The same command ss -ta.
[root@server ~]# tracepath -b google.com ==> to show the route to the other destination.
[root@server ~]# ping -c5 192.168.10.139 ==> to limit the number of replies