Manage networking

Linux #redhat #network

In this note:

- Static vs. dynamic IP
- OS network components (Network interface, subnet mask, gateway, MAC address etc.)
- Getting started with NetworkManager.
- Network configuration methods
  - `nmtui`, `nmcli`, `nm-connection-editor`, and GNOME settings
    - GNOME is the desktop environment that can be accessed through the GUI.
- Network files and basic commands to manage networking and troubleshoot related issues.

Static IP vs. DHCP

- DHCP stands for "Dynamic Host Control Protocol"
  - If an IP is not static, it is referred to as DHCP
- IP stands for "Internet Protocol" which is assigned to your computer so it can access the network. There are two types of IPs that can be assigned.
  - Private IP is assigned for internal communication. (within your home network)
  - Public IP is for the internet/external communication.
- **Static** -> Does not change
  - You reboot your computer, you reconfigure it, you have it shut down for a month, it will have that IP manually configured to your computer.
- **Dynamic** -> Changes after system reboot
- Why do we have static vs. dynamic?
  - The main reason is that static IP is assigned to computers and it has its host name IP assignment as a DNS setup if there are computers outside in the world that accessing it. You don't want that on your computer IP to change because if it's changing every day, then it's gonna be so much confusion to find out what is your IP address.
  - Whereas dynamic IP, if you're running in the local lab environment, you don't really care what IP is given, as long as you could access it, that's what really matters.

> **Find IP**
>
> ifconfig

- Use the `ifconfig` command
  - It will list network adapters

Example using command:

```
[user@localhost ~]$ ifconfig
```

Output:

```
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.156  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::a00:27ff:feda:ce70  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:da:ce:70  txqueuelen 1000  (Ethernet)
        RX packets 172821  bytes 243056294 (231.7 MiB)
        RX errors 0  dropped 60  overruns 0  frame 0
        TX packets 16034  bytes 1322397 (1.2 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 18  bytes 2112 (2.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 18  bytes 2112 (2.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- This command will give you the network interfaces.
- In this case we can see 2 network adapters. The one listed with the name enp0s3 is that actual adapter responsible for your connection. And the IP address would be listed in here. In this case we found it is: 192.168.1.156
- The first adapter listed is 'enp0s3' and it is the interface given when you are running on Oracle virtual environment.
- The second adapter named "lo" stands for local.
  - It'll Always have the same IP.
  - Your processes inside of your computer, they have to communicate between each other, they use this interface to communicate.
- If the virtual machine is configured to bridge a network adapter, then you'll have a virtual bridge adapter listed under `ifconfig`, Usually ignored.
- `ip addr` has the same effect.

OS Network Components

- Network interface
  - The interfaces on you computer.
  - If you're running a physical server, you could see in behind your physical server, there are ports. Those ports are used to connect the internet, the Ethernet 5 cables. Those ports are connected onto a network card. Once that card is installed on your computer, they give you the network interfaces.
  - You need a network interface in order to communicate to other computers.
- MAC address
  - Every time that network interface gets installed, it has a hard-coded MAC address that never changes.
- Subnet Mask
  - Allows your IP range to go either to a certain number of IPs, or to an additional number of IPs in your environment.
- Gateway
  - Most likely an IP assigned to your router, which allows you to take your traffic from your computer to outside, to other computers.
- DNS (Domain Name System)
  - Allows you to have IP to host name, host name to IP, and host name to host name.
    - The bonding, the registration of your computer.
  - Back in the days, any computer that came on board, it has an IP address, so it was so hard to remember the IP addresses for your computer. That's when the DNS came along and we started assigning IP addresses to its host name so it's easier to remember.

Network interface

- How do we know how many network interfaces do we have?
  - run `ifconfig`
    - It will give us the network interfaces.
- In the previous example the first listed interface (enp0s3) is the one that you need to communicate to outside world. And this, by the way is assigned when you create your machine, your physical machine for the first time. Or if you're creating your machine on a virtual network, this is the network adapter.

MAC address

- if you type `ifconfig` again and you want to know the MAC address of your network interface, you will usually see the word "ether" and then the actual MAC address.

Output from `ifconfig`:

```
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.156  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::a00:27ff:feda:ce70  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:da:ce:70  txqueuelen 1000  (Ethernet)
        RX packets 172821  bytes 243056294 (231.7 MiB)
        RX errors 0  dropped 60  overruns 0  frame 0
        TX packets 16034  bytes 1322397 (1.2 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 18  bytes 2112 (2.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 18  bytes 2112 (2.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- In this example, for the enp0s3 network interface the MAC address is shown in the following line:
  - `ether 08:00:27:da:ce:70 txqueuelen 1000 (Ethernet)`
  - The actual MAC address is then: 08:00:27:da:ce:70
- This is the hard coded MAC address or an address of your interface that will never change.
- Your IP could change, if you assign the static one time, you could assign a different static the second time. It could change. But the MAC address never changes and it is assigned by the manufacturer when your computer first time built, or when your ethernet card is built.

Subnet mask

- When running `ifconfig` you can see the subnet mask next to the keyword "netmask".
- In the previous example it is found in the second line of the first listed network interface (enp0s3), the line would look like this:
  - `inet 192.168.1.156 netmask 255.255.255.0 broadcast 192.168.1.255`
  - The subnet mask is then: 255.255.255.0
- The subnet mask is the mask that actually allows or control your traffic.

- If your traffic has to go from one machine to another, it will find out whether it has to go through a router or it has to go directly within a machine to another machine.

Gateway

- In order to get the gateway information you'll have to type `netstat -rnv`

Example using `netstat` command:

```
[user@localhost ~]$ netstat -rnv
```

- This command will give you your gateway information.
- The `netstat` command prints network connections, routing tables, interface statistics, masquerade connections, and multicast memberships.
- The `-r` option stands for "route" and specifies to display the kernel routing tables.
- The `-n` option stands for "numeric"
- The `-v` option stands for "verbose"

Output:

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         192.168.1.1     0.0.0.0         UG        0 0          0 enp0s3
192.168.1.0     0.0.0.0         255.255.255.0   U         0 0          0 enp0s3
```

- Anywhere you want it to go outside of this computer you have to go through the gateway 192.168.1.1, by the way it is probable that this is the IP address of your router.

DNS (Domain Name System)

- DNS allows you to create records between host name to IP, IP to host name, and host name to host name.
- Lets say wou eant to go to google but you do not know its IP:

Example using `ping` command:

```
[user@localhost ~]$ ping www.google.com
```

- Trying to reach Google with the `ping` command.

Output:

```
PING www.google.com (216.239.38.120) 56(84) bytes of data.
64 bytes from any-in-2678.1e100.net (216.239.38.120): icmp_seq=1 ttl=110 time=47.2 ms
64 bytes from any-in-2678.1e100.net (216.239.38.120): icmp_seq=2 ttl=110 time=45.0 ms
64 bytes from any-in-2678.1e100.net (216.239.38.120): icmp_seq=3 ttl=110 time=44.3 ms
64 bytes from any-in-2678.1e100.net (216.239.38.120): icmp_seq=4 ttl=110 time=44.3 ms
```

- In this example, the IP address of Google is 216.239.38.120. And we are able to know this thanks to DNS.

How do you know where is your DNS?

- You could find about your DNS by running the command `nslookup` and then typing the hostname you want to reach.

Example using `nslookup` command:

```
[user@localhost ~]$ nslookup www.google.com
```

- The `nslookup` command queries internet name servers interactively.

Output:

```
Server:         192.168.1.1
Address:        192.168.1.1#53

Non-authoritative answer:
Name:   www.google.com
Address: 216.239.38.120
Name:   www.google.com
Address: 2001:4860:4802:32::78
```

- It tells you your DNS server is this, 192.168.1.1
  - Isn't that the same as the gateway?
    - Yes, in the local environment, in your home environment most of the time your server, your gateway, or your router performs two function. It performs DNS as well as the gateway.

Getting started with `NetworkManager`

- NetworkManager is a service and set of tools designed specifically to make it easier to manage the networking configuration on Linux systems and is the default network management service on RHEL 9
- It makes network management easier
- It provides easy setup of connection to the user
- NetworkManager offers management through different tools such as GUI, nmtui, and nmcli

Example using `systemctl` command:

```
[user@localhost ~]$ systemctl status NetworkManager
```

- Check if the service is running.

Output:

```
● NetworkManager.service - Network Manager
     Loaded: loaded (/usr/lib/systemd/system/NetworkManager.service; enabled; preset: enabled)
     Active: active (running) since Mon 2024-06-17 10:03:19 CST; 3h 28min ago
       Docs: man:NetworkManager(8)
   Main PID: 941 (NetworkManager)
      Tasks: 3 (limit: 65779)
     Memory: 9.9M
        CPU: 3.155s
```

```
     CGroup: /system.slice/NetworkManager.service
             └─941 /usr/sbin/NetworkManager --no-daemon

Jun 17 10:03:23 localhost.localdomain NetworkManager[941]: <info>  [1718640203.6492] device (enp0s3):
state change: >
Jun 17 10:03:23 localhost.localdomain NetworkManager[941]: <info>  [1718640203.7644] device (enp0s3):
state change: >
...
```

- NetworkManager is up and running.
- You can additionally run `ps -ef | grep NetworkManager` to verify that the process is running.
- NetworkManager controls your entire network, every time you make chanegs to your network you most likely would have to restart your NetworkManager service

Network Configuration methods

- `nmcli` – Short for "network manager command line interface". This tool is useful when access to a graphical environment is not available and can also be used within scripts to make network configuration changes.
- `nmtui` – Short for "network manager text user interface". This tool can be run within any terminal window and allows changes to be made by making menu selections and entering data.
- `nm-connection-editor` - A full graphical management tool providing access to most of the NetworkManager configuration options. It can only be accessed through the desktop or console.
- `GNOME Settings` - The network screen of the GNOME desktop settings application allows basic network management tasks to be performed.

The `nmcli` command

- You should run this command when you are not given a text graphical user interface.

Example using `nmcli` command:

```
[user@localhost ~]$ nmcli
```

- `nmcli` is a command-line tool for controlling NetworkManager

Output:

```
enp0s3: connected to enp0s3
        "Intel 82540EM"
        ethernet (e1000), 08:00:27:DA:CE:70, hw, mtu 1500
        ip4 default
        inet4 192.168.1.156/24
        route4 192.168.1.0/24 metric 100
        route4 default via 192.168.1.1 metric 100
        inet6 fe80::a00:27ff:feda:ce70/64
        route6 fe80::/64 metric 1024

lo: connected (externally) to lo
        "lo"
        loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536
        inet4 127.0.0.1/8
```

```
        inet6 ::1/128
        route6 ::1/128 metric 256

DNS configuration:
        servers: 192.168.1.1
        interface: enp0s3

Use "nmcli device show" to get complete information about known devices and
"nmcli connection show" to get an overview on active connection profiles.

Consult nmcli(1) and nmcli-examples(7) manual pages for complete usage details.
```

- When you run `nmcli` you'll get the information about your network adapters, and, other adapters.
- You can get help to use this command by running `nmcli --help`.

Example using `nmcli` command:

```
[user@localhost ~]$ nmcli connection
```

- The `connection` or `c` function lists networkManager's connections.

Output:

```
NAME    UUID                                  TYPE      DEVICE
enp0s3  7cc1f3c4-26e3-36ef-8412-32b94c016a6f  ethernet  enp0s3
lo      c04445be-eaa3-4ed6-8471-cf08b3b70b3c  loopback  lo
```

You can actually incorporate `nmcli` in your script.

- Let's say if you wanted to configure a static IP using `nmcli`, then you would have to use `nmcli connection` to know the name of the interface connected to NetworkManager, and then you have to use a mod (`modify` followed by the network interface).

Using `nmcli` to configure static IP

Case: let's say right now when you run `ifconfig` you find that you have an IP address of 192.168.1.156, but I wanted to assign a static IP.

- In case you are using a Virtual Machine it is recommended to use the console natively in your machine instead of an SSH connection or a PuTTY window because you will be modifying your network interface.
- Log in as root user.

Steps:

1. `nmcli device`
    - Gets the listing of network interfaces

Example using `nmcli` command:

```
[root@localhost ~]# nmcli device
```

Output:

```
DEVICE  TYPE      STATE                   CONNECTION
enp0s3  ethernet  connected               enp0s3
lo      loopback  connected (externally)  lo
```

- This will give us the listing of all the network devices that you get through `ifconfig` as well.
  - You can also use `ip addr` or `ip address` for the same purpose.

2. `nmcli connection modify enp0s3 ipv4.addresses 192.168.1.200/24`
   - Modify the IP address

Example using `nmcli` command:

```
[root@localhost ~]# nmcli connection modify enp0s3 ipv4.addresses 192.168.1.200/24
```

- The `connection` function is used to access the connections to NetworkManager.
- The `modify` function allows us to modify a NetworkManager connection.
- The network interface connected to NetworkManager in this example is enp0s3
- We want to modify the ipv4.addresses field from the enp0s3 interface (We are assigning an IP)
- The new IP that will be established to that interface will be 192.168.1.200
- The `/24` is because we want it in the 24 subnet

3. `nmcli connection modify enp0s3 ipv4.gateway 192.168.1.1`
   - Assign the gateway

Example using `nnmcli` command:

```
[root@localhost ~]# nmcli connection modify enp0s3 ipv4.gateway 192.168.1.1
```

4. `nmcli connection modify enp0s3 ipv4.method manual`
   - Set the method of assigning an IP to manual as we are assigning it manually.

Example using `nmcli` command:

```
[root@localhost ~]# nmcli connection modify enp0s3 ipv4.method manual
```

5. `nmcli connection modify enp0s3 ipv4.dns 8.8.8.8`
   - Modify the DNS. If you already have a DNS, then you do not necessarily have to. It is just done here for training purposes.

Example using `nmcli` command:

```
[root@localhost ~]# nmcli connection modify enp0s3 ipv4.dns 8.8.8.8
```

- The new DNS server will be 8.8.8.8
  - Or you could use the same gateway IP address as for DNS.

6. `nmcli connection down enp0s3 && nmcli connection up enp0s3`
    - You need to reset your interface

Example using `nmcli` command:

```
[root@localhost ~]# nmcli connection down enp0s3 && nmcli connection up enp0s3
```

- This command shuts the connection for the enp0s3 interface down and up in the same line
    - It serves as a "restart function"
    - Note the "`down`" and "`up`" keywords
- The ampersand sign (`&&`) is used to continue the command.
- You could use the entire command in one shot or you could use first down command and then up command.

Output:

```
Connetion 'enp0s3' successfully deactivated (...)
Connection successfully activated (...)
```

7. `ip address show enp0s3`
    - Verify that the new specified IP is set to the desired network interface.

Example using `ip address` command:

```
[root@localhost ~]# ip address show enp0s3
```

- The `show` function is used to show the information only of certain interfaces instead of all of them.
- `ifconfig enp0s3` would have the same result.

Output:

```
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:da:ce:70 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.200/24 brd 192.168.1.255 scope global noprefixroute enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:feda:ce70/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

- Note the new IP (192.168.1.200) previously specified by us is now set to the interface.
- It worked!

Adding secondary static IP using `nmcli`

- What if you wanted to assign a secondary static IP to the same interface using `nmcli` utility?
- Yes you could do that without disturbing the existing IP address.

Steps:

1. `nmcli device status`

- Check the status (connected)

Example using `nmcli` command:

```
[root@localhost ~]# nmcli device status
```

Output:

```
DEVICE  TYPE      STATE                   CONNECTION
enp0s3  ethernet  connected               enp0s3
lo      loopback  connected (externally)  lo
```

- Make sure the status for the 'enp0s3' is connected.
- Note it has the same results as the `nmcli device` command

2. `nmcli connection show --active`
    - Check and show active connections.

Example using `nmcli` command:

```
[root@localhost ~]# nmcli connection show --active
```

Output:

```
NAME    UUID                                  TYPE      DEVICE
enp0s3  7cc1f3c4-26e3-36ef-8412-32b94c016a6f  ethernet  enp0s3
lo      c04445be-eaa3-4ed6-8471-cf08b3b70b3c  loopback  lo
```

3. `ifconfig`
    - See the current IP assigned to enp0s3 ipv4 (This was assigned previously)
    - For this example the current IP of the interface is 192.168.1.200
4. `nmcli connection modify enp0s3 +ipv4.addresses 192.168.1.201/24`
    - Add the new secondary IP address.

Example using `nmcli` command:

```
[root@localhost ~]# nmcli connection modify enp0s3 +ipv4.addresses 192.168.1.201/24
```

- The difference with changing its own IP address as done previously is that here we use a `+` sign.
    - Remember now we have to ADD (`+`) an IP address.

5. `nmcli connection reload`
    - Will do the same thing as if you would do an `nmcli connection down enp0s3` or `up`

Example using `nmcli` command:

```
nmcli connection reload
```

6. `systemctl reboot`
    - Now you could reboot your system by doing the reboot command, or `systemctl` command, just to confirm that your settings stay the same even after the reboot.

Example using `systemctl` command:

```
[root@localhost ~]# systemctl reboot
```

- `systemctl` command to reboot your system.

7. `ip address show`
    - Confirm whether the second IP was added or not.

Example using `ip address` command:

```
[user@localhost ~]$ ip address show
```

Output:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:da:ce:70 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.200/24 brd 192.168.1.255 scope global noprefixroute enp0s3
       valid_lft forever preferred_lft forever
    inet 192.168.1.201/24 brd 192.168.1.255 scope global secondary noprefixroute enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:feda:ce70/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

- We can see that on the second interface 'enp0s3' the first IP that we assigned is 192.168.1.200
- The secondary IP is also listed under the same interface and is assigned 192.168.201

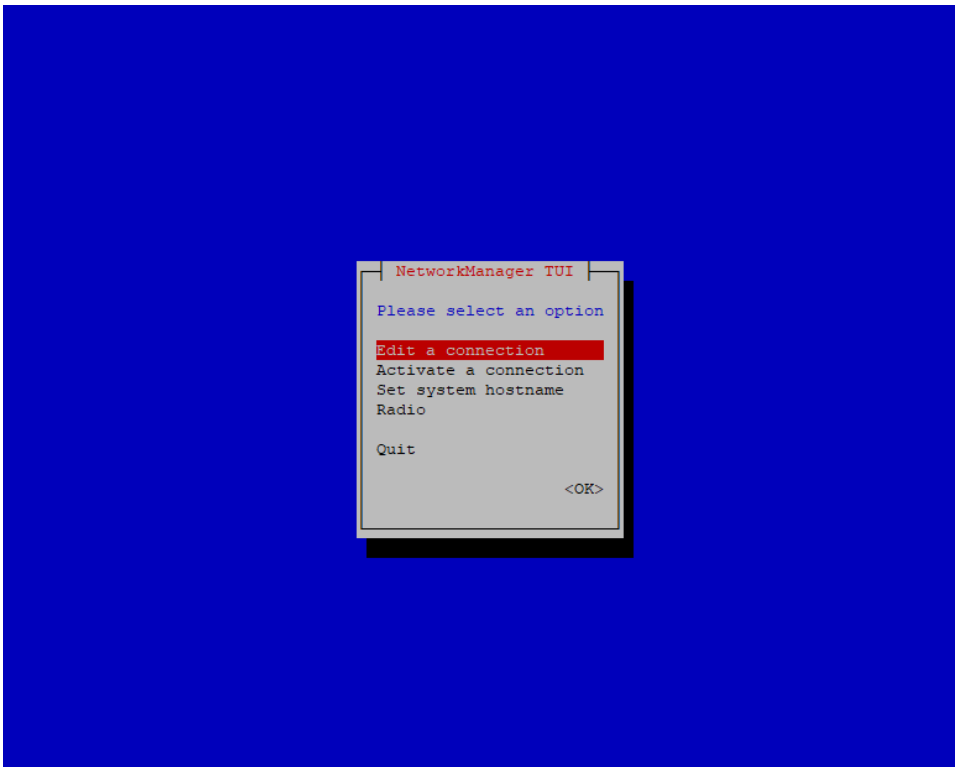
The `nmtui` command

- Can be accessed through console as well as on the PuTTY session.
- Run `nmtui` as root user in order to be able to edit interfaces.

Example using `nmtui` command:

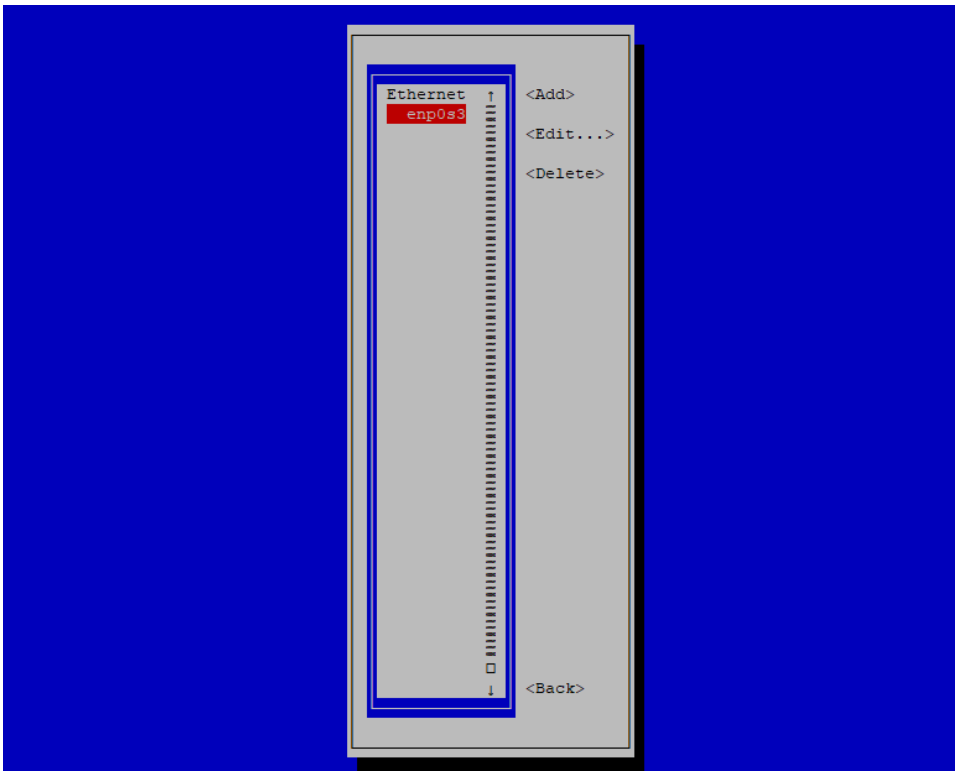
```
[user@localhost ~]$ nmtui
```

nmtui



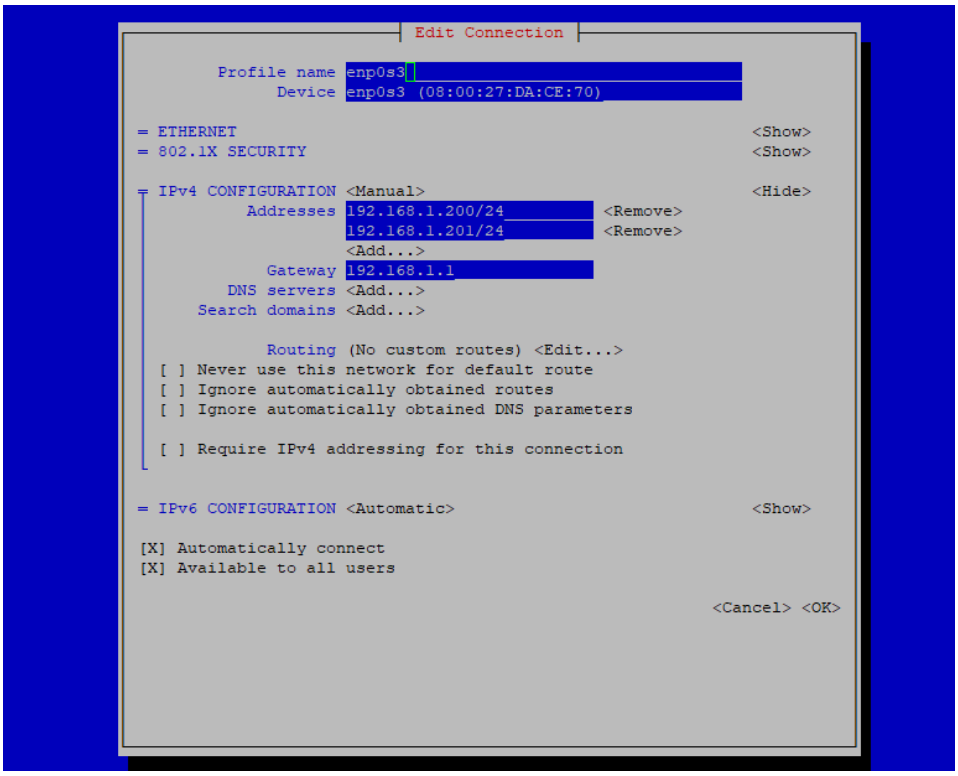
- This is the text interface, it looks like a graphical, but it's not graphical, it's text-based.
- Hit enter to select an option. Use the keys to change between options.

Edit a connection



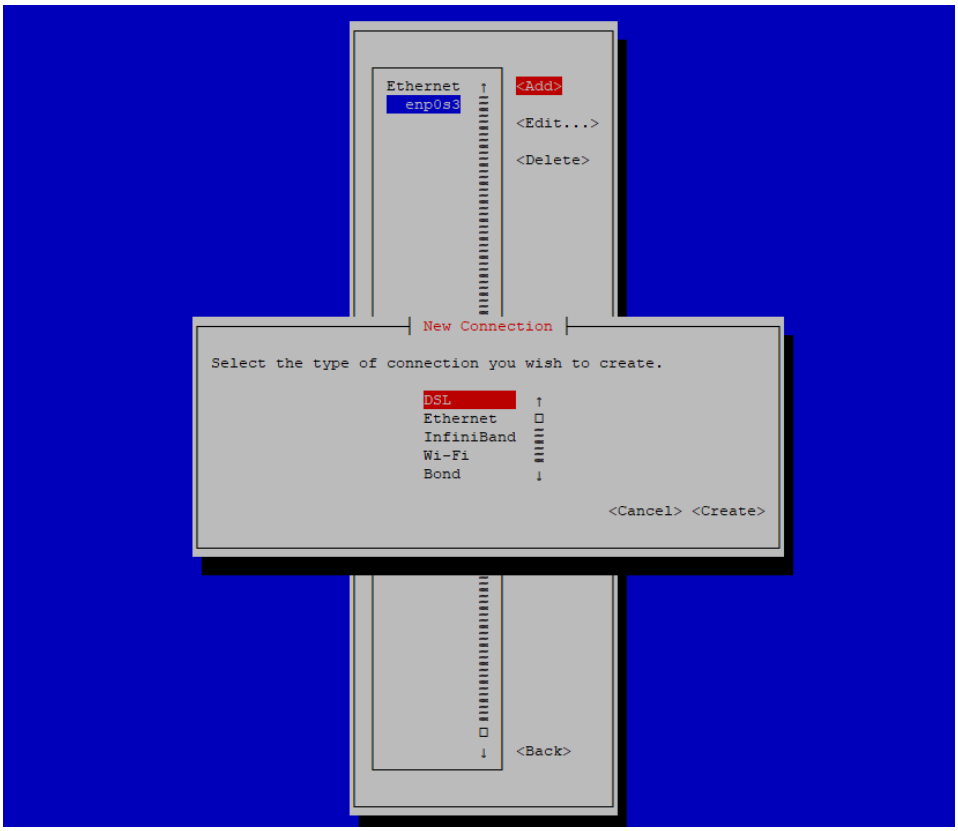
- Network interfaces are listed here.
- If you want to work on the first adapter, 'enp0s3', you move your cursor to the right (using the arrow keys) and click Edit (hit Enter).

<Edit...>



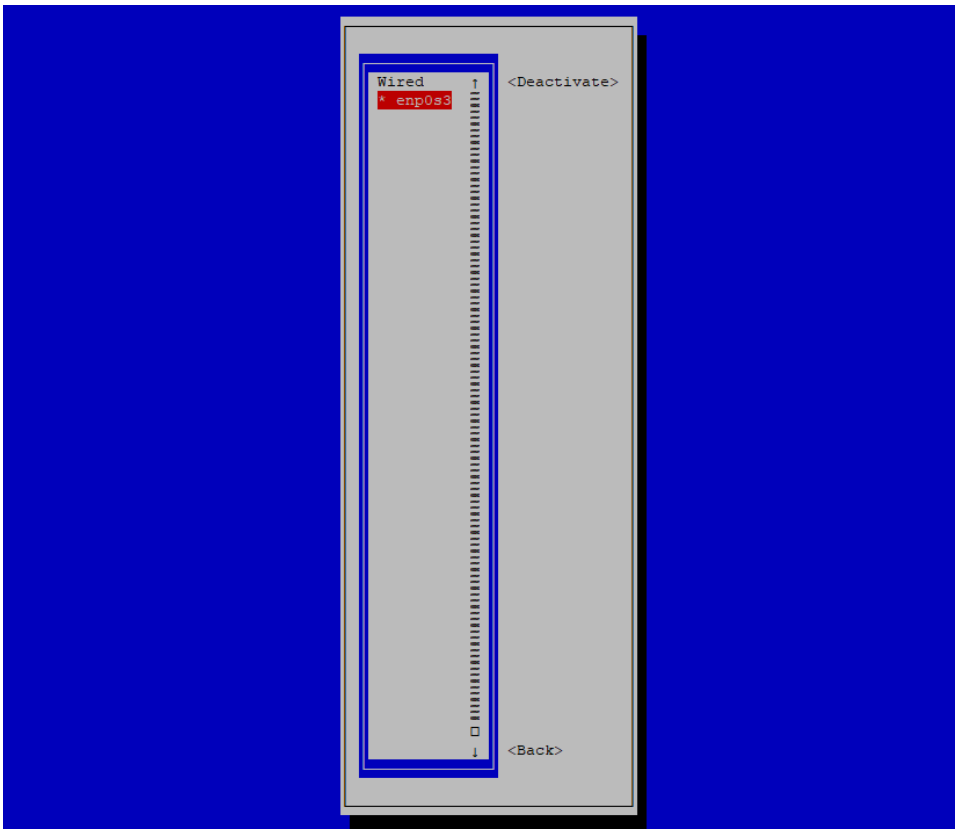
- The profile name, and the device MMAC address is listed.
- In this example the IPv4 configuration is set to Manual, but in other instances and if default, it is probable that this is set t Automatic.
- If you wanted to assign a manual static IP, you have to select it, hit Enter, and then select Manual.
  - Then click Show, and you would have to add the IP address to whatever the static IP address that you what to assign to.
- You can go back to previous screens by hitting the Esc key. Or you can go to the Back option sometimes provided.
- Note that if you are editing the interface without being root user , it will not give you the option to click OK

**<Add.>



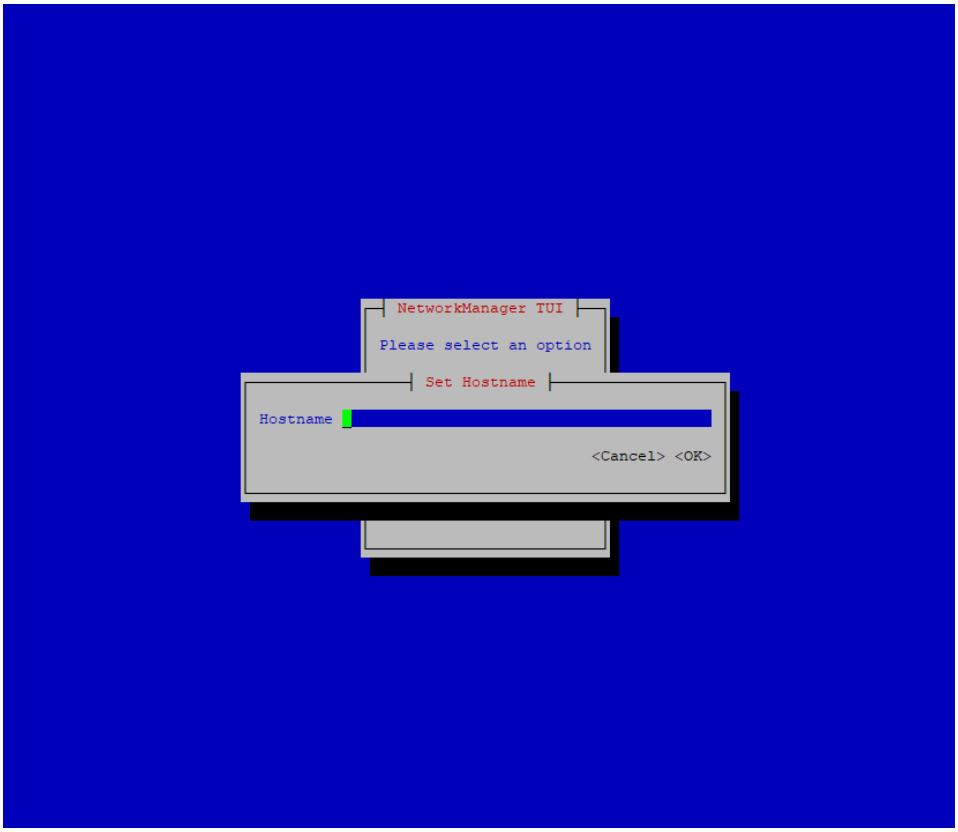
- If you wanted to add a connection, once that connection is added to your Virtual Box, then you could actually do NIC bonding, or teaming through this same tool. And it's a lot easier to do that
- You have the options to:
  - Bond
  - Team
  - ...
- This is where you could pick the option to bond two network interfaces together

Activate a connection



- If you have a network adapter already added to your computer but it's not active, then you could go to Activate a connection, select the Activate option and hit Enter.
- You can also Deactivate adapters if desired.

Set hostname



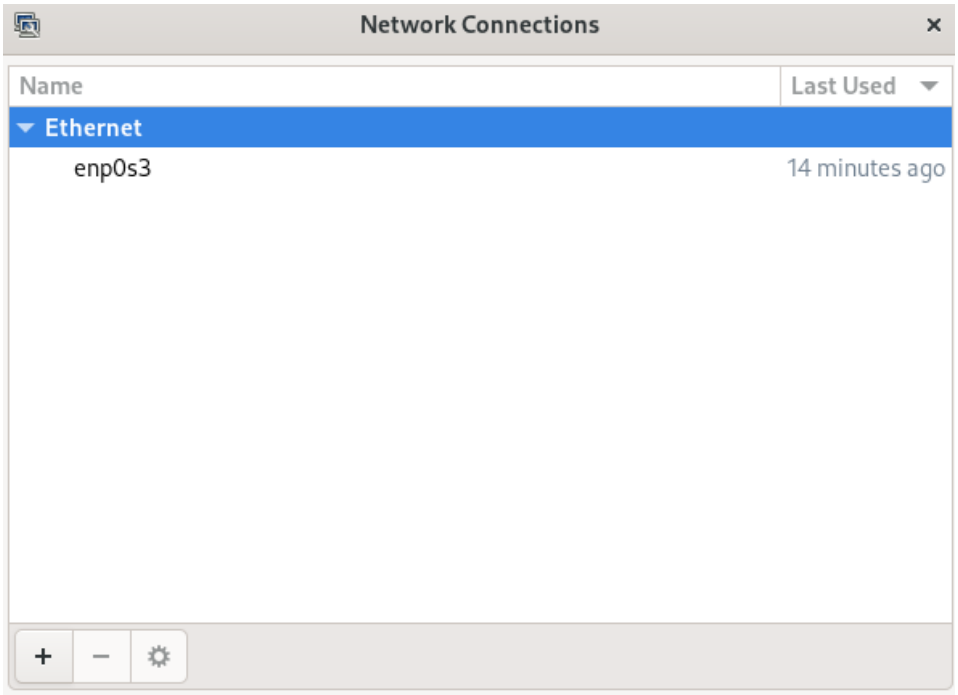
- If you wanted to change your Linux machine host name, you could run `nmtui`, come to the option Set Hostname and change that host name.

The `nm-connection-editor` GUI editor

Example using `nm-connection-editor` command:

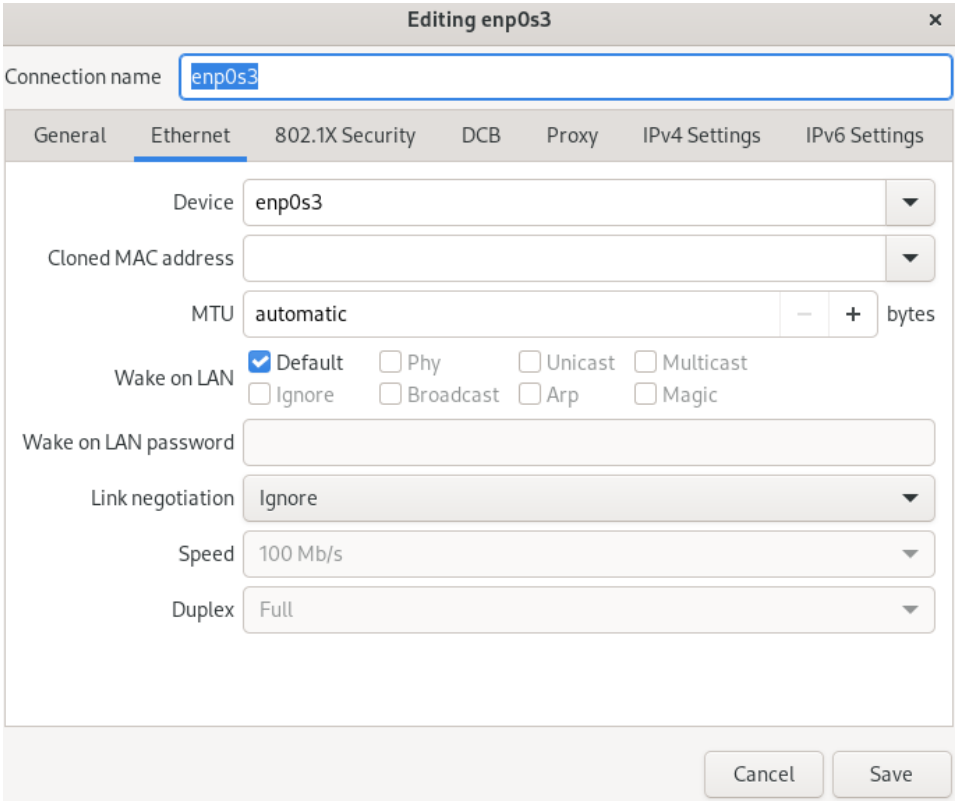
```
[user@localhost ~]$ nm-connection-editor
```

nm-connection-editor



- This is a GUI app, or graphical window.
- Select the ethernet adapter you wanted to pick

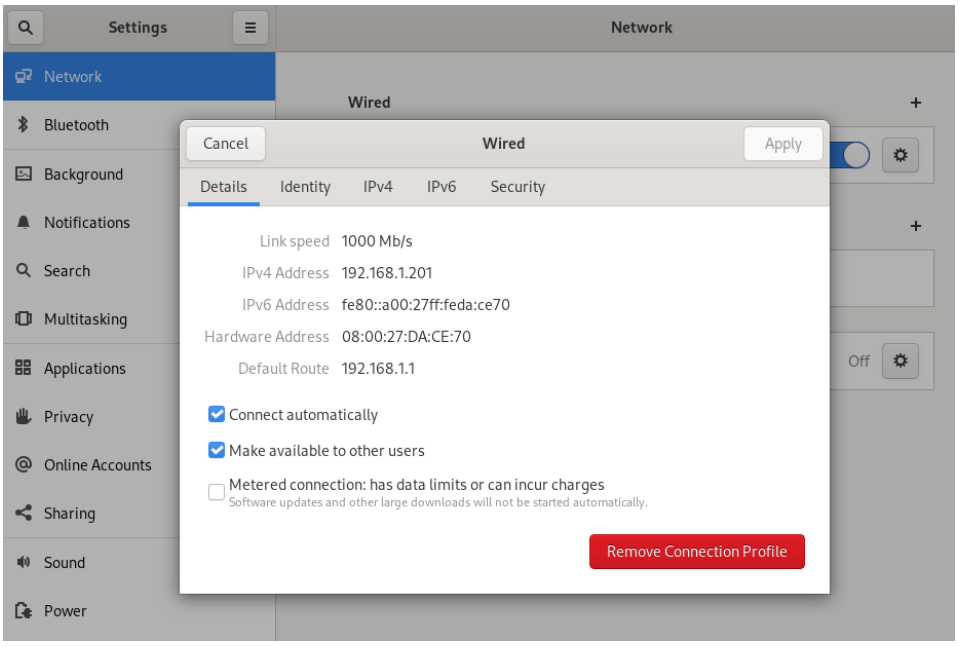
Editing enp0s3



- Make any changes you want.
- If you have access to the console, it is a very useful tool to manage networks.

GNOME settings

- Simply look for the Settings app on your Linux machine and search for the Network tab.
- In the Network tab of the Settings app you will find a "Gear" symbol next to your connection. You can click on this symbol and there you could make certain changes to whatever that you wanted to make changes to.



Network Files and Basic Commands

Files: (Files that you should know to manage your network)

- `/etc/sysconfig/network-scripts`
- `/etc/hosts`
- `/etc/hostname`
- `/etc/resolv.conf`
- `/etc/nsswitch.conf`

Commands:

- `ping`
- `ifconfig` or `ip`
- `ifup` or `ifdown`
- `netstat`
- `traceroute`
- `tcpdump`
- `nslookup` or `dig`

The `/etc/sysconfig/network-scripts` directory

- If you are modifying your network manually by going into the files, which was the case in the older versions of Linux, now you have these fancy tools, you don't have to modify the files, you could do it through the tools.
- But if you have to come to the files and modify it, you could come to this directory, do `ls -ltr` and you will see your network interfaces.

Example using `ls` command:

```
[root@localhost network-scripts]# ls -ltr
```

- Note the directory location.
- Listing the contents of the `/etc/sysconfig/network-scripts` file.

Output:

```
total 4
-rw-r--r--. 1 root root 1244 Mar 26 09:51 readme-ifcfg-rh.txt
```

- This is the network interface you have to modify in case you want to assign a Static IP or make any changes.
  - In this example the file is was not existent and there was only a readme file. The file that we are looking for should have the name 'ifcfg-enp0s3'.
  - Previously you could see some fields listed, one of them was 'BOOTPROTO=', there is where you could see what is the IP assignment protocol for your Linux machine (dhcp or static).

Output from `more readme-ifcfg-rh.txt`:

```
NetworkManager stores new network profiles in keyfile format in the
/etc/NetworkManager/system-connections/ directory.
```

```
Previously, NetworkManager stored network profiles in ifcfg format
in this directory (/etc/sysconfig/network-scripts/). However, the ifcfg
format is deprecated. By default, NetworkManager no longer creates
new profiles in this format.

Connection profiles in keyfile format have many benefits. For example,
this format is INI file-based and can easily be parsed and generated.

Each section in NetworkManager keyfiles corresponds to a NetworkManager
setting name as described in the nm-settings(5) and nm-settings-keyfile(5)
man pages. Each key-value-pair in a section is one of the properties
listed in the settings specification of the man page.

If you still use network profiles in ifcfg format, consider migrating
them to keyfile format. To migrate all profiles at once, enter:

# nmcli connection migrate

This command migrates all profiles from ifcfg format to keyfile
format and stores them in /etc/NetworkManager/system-connections/.

Alternatively, to migrate only a specific profile, enter:

# nmcli connection migrate <profile_name|UUID|D-Bus_path>

For further details, see:
* nm-settings-keyfile(5)
* nmcli(1)
```

- Now we know that the network-scripts directory is now Deprecated in RHEL 9.
- Now NetworkManager stores new network profiles in keyfile format in the `/etc/NetworkManager/system-connections/` directory.

The `/etc/hosts` file

- This is the local file that it's a local file to your machine to keep track of IP to host name.
- It's like a local DNS in your machine.

Example using `cat` command:

```
[root@localhost ~]# cat /etc/hosts
```

Output:

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
```

- If you wanted to ping your own hostname you could run `ping localhost`, because it has the IP address to 127.0.0.1
- If you wanted to ping your hostname you could run `ping linuxtest` ('linuxtest' is just an example hostname) and it will ping because this is the actual host name assigned to your network interface. So you will also see your IP address while pinging.

The `/etc/hostname` file

- If you wanted to change the host name you could go to this file and modify your host name of your computer right there and then restart your network manager.

Example using `cat` command:

```
[root@linuxtest ~]# cat /etc/hostname
```

- Note the name of the machine is 'linuxtest'

Output:

```
linuxtest
```

The `/etc/resolv.conf` file

More on: In /etc/resolv.conf, what exactly does the "search" configuration option do?

- This is the file where you could actually keep the information of your DNS server.
- Every time you have to go to another machine through the host name, your operating system is gonna come to this file, consult this file, and ask this file, "Hey, who is my DNS?"
- This file is gonna tell your OS, your DNS, or in other words, it's also referred to as 'name server'

Example using `cat` command:

```
[root@linuxtest ~]#cat /etc/resolv.conf
```

Output:

```
# Generated by NetworkManager
nameserver 192.168.1.1
```

The `/etc/nsswitch.conf` file

- This file is used if you want your system to go to first files, or DNS first.
- For example, I want every time, my OS needs IP to host name lookups, I want it to go to files first, which is `/etc/host`.
  - If, and then if the entry is not there, then I want it to go to DNS, so that's why the same way, if I want my shadow or passwords to go to files first, then I'll put files there.
  - If I wanted to go to single sign-ons, then a second prop method would be the SSS.

Example using `cat` command:

```
[root@linuxtest ~]# cat /etc/nsswitch.conf
```

Output:

```
...
# In order of likelihood of use to accelerate lookup.
shadow:     files
hosts:      files dns myhostname

aliases:    files
ethers:     files
gshadow:    files
# Allow initgroups to default to the setting for group.
# initgroups: files
networks:   files dns
protocols:  files
publickey:  files
rpc:        files
```

The `ping` command

- One of the basic and most popular commands that you should know when you are troubleshooting network issues.
- So if you have a computer, a remote computer, and you are trying to access it and it's not accessible, the very first thing you gotta do is ping it.
- Let's say I cannot get to my gateway, I wanted to see my gateway is reachable, I'll do...

Example using `ping` command:

```
[root@linuxtest ~]# ping 192.168.1.1
```

Output:

```
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=1.97 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=2.22 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=2.09 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=2.56 ms
...
^C
--- 192.168.1.1 ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10045ms
rtt min/avg/max/mdev = 1.651/1.963/2.561/0.240 ms
```

- When you get this response (64 bytes from 192.168.1.1), it means the computer is there and it's responding to my request. It is like a "Yes, I'm here"

The `ifup` and `ifdown` command

- If you have a deactivated network interface, you could activate it from `nmtui`, or you could run `ifup`.

The `netstat` command

Example using `netstat` command:

```
[root@linuxtest ~]# netstat -a
```

- The `-a` option stanDs for "all" and lists all the connections.
- If you run this command, it will tell you what is connected to your machine and what is going outside of your machine. The ports up and running.

Output:

```
...
unix  3      [ ]         STREAM     CONNECTED     26819
unix  3      [ ]         STREAM     CONNECTED     24980
unix  3      [ ]         STREAM     CONNECTED     28190
unix  3      [ ]         STREAM     CONNECTED     29879    /run/user/1000/bus
unix  3      [ ]         STREAM     CONNECTED     28887
Active Bluetooth connections (servers and established)
Proto  Destination       Source            State         PSM DCID   SCID      IMTU    OMTU Security
Proto  Destination       Source            State     Channel
```

The `traceroute` command

- `traceroute` tells you how your computer is going from one machine to another.
- The `traceroute` package is not included by default in a fresh RHEL 9 install so you will probably want to install it.

Example using `traceroute` command:

```
[root@linuxtest ~]# traceroute
```

Output:

```
bash: traceroute: command not found...
Install package 'traceroute' to provide command 'traceroute'? [N/y]
```

The `tcpdump` command

- Very powerful command
- Traces everything that's coming into your computer and going outside of your computer.
- Before you run `tcpdump`, you have to know your interface where you're tunning that `tcpdump`

Example using `tcpdump` command:

```
[root@linuxtest ~]# tcpdump -i enp0s3
```

- The `-i` option stands for "interface" and lets you specify an interface.

Output:

```
...
12:08:08.721284 IP 192.168.1.58.51819 > linuxtest.ssh: Flags [.], ack 21696, win 8191, length 0
12:08:08.783848 IP linuxtest.ssh > 192.168.1.58.51819: Flags [P.], seq 21696:21968, ack 161, win 341,
length 272
12:08:08.824654 IP 192.168.1.58.51819 > linuxtest.ssh: Flags [.], ack 21968, win 8190, length 0
12:08:08.887664 IP linuxtest.ssh > 192.168.1.58.51819: Flags [P.], seq 21968:22240, ack 161, win 341,
length 272
12:08:08.928792 IP 192.168.1.58.51819 > linuxtest.ssh: Flags [.], ack 22240, win 8189, length 0
^C
171 packets captured
298 packets received by filter
123 packets dropped by kernel
```

- Right now all the traffic that are going out of your computer, coming into your computer, you'll see it coming right in front of your eyes.
- you could `grep` or filter it based on your packet filtering requirement.

The `nslookup` and `dig` command

- If you wanted to know what is the IP address of a host name, or of a website, or server.

Example using `nslookup` command:

```
[root@linuxtest ~]# nslookup www.hotmail.com
```

- Looking for the IP address of hotmail.

Output:

```
Server:         192.168.1.1
Address:        192.168.1.1#53

Non-authoritative answer:
www.hotmail.com canonical name = outlook-fd-0010.live.com.
outlook-fd-0010.live.com        canonical name = a-0010.a-msedge.net.
Name:   a-0010.a-msedge.net
Address: 204.79.197.212
Name:   a-0010.a-msedge.net
Address: 2620:1ec:c11::212
```

- The IP address for www.hotmail.com is found to be 204.79.197.212

The `dig` command will fulfill the same purpose
Example using `dig` command:

```
[root@linuxtest ~]# dig www.hotmail.com
```

- `dig` is a DNS lookup utility.

Output:

```
; <<>> DiG 9.16.23-RH <<>> www.hotmail.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 34623
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.hotmail.com.               IN      A

;; ANSWER SECTION:
www.hotmail.com.        477     IN      CNAME   outlook-fd-0010.live.com.
outlook-fd-0010.live.com. 2330  IN      CNAME   a-0010.a-msedge.net.
a-0010.a-msedge.net.    240     IN      A       204.79.197.212

;; Query time: 38 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Tue Jun 18 12:16:59 CST 2024
;; MSG SIZE  rcvd: 128
```

- It will give you the same information but its output will look a little different.

The `ethttol` command

- If you wanted to know your ethernet adapter, it's speed, whether it's up or not, it's link speed, status, then you could use `ethtool`.

Example using `ethtool` command:

```
[root@linuxtest ~]# ethtool enp0s3
```

- `ethtool` - query or control network driver and hardware settings
- You have to provide the name of a Network Interface

Output:

```
Settings for enp0s3:
        Supported ports: [ TP ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Supported pause frame use: No
        Supports auto-negotiation: Yes
        Supported FEC modes: Not reported
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Advertised pause frame use: No
        Advertised auto-negotiation: Yes
        Advertised FEC modes: Not reported
        Speed: 1000Mb/s
        Duplex: Full
        Auto-negotiation: on
```

```
        Port: Twisted Pair
        PHYAD: 0
        Transceiver: internal
        MDI-X: off (auto)
        Supports Wake-on: umbg
        Wake-on: d
        Current message level: 0x00000007 (7)
                               drv probe link
        Link detected: yes
```

- It'll give you a lot of information about your interface.