Network Devices.

* Vmnet0, vmnet1, vmnet2 etc, for virtual devices for communicating with virtual clients
* Wlan0, wlnan1, etc for wireless devices
* Eth0, eth1, etc for Ethernet devices
* Br0, br1, br2, etc ….. for bridge interfaces.

**Predictable Network Interface Device Names (PNIDN)** is strongly correlated with the use of the udev and integration with sysmtend. There are now 5 types of names that devices cab be given.

* **Incorporating Firmware or BIOS provided index numbers for on-board devices:**

eno1

* **Incorporating Firmaware or BIOS provided PCI Express hotplug slot index numbers:**

ens1

* **Incorporating physical and/or geographical lication oof the hardware connection:**

**enp2s0**

* **Incorpirating the mac address:**

**Example:** enx7837d1ea46da

* Olda classic method:

eth0

**IP. Is a multiplex utility ;**  the  **OBJECT** argument describes what kind of action is going to be performed. The possible **COMMANDS**  depends on which **OBJECT** is detected.

**Is more versatile and more efficient because it uses netlink** sockets rather that **ioctl systems** calls. It can be uses to **configure, control and query devices and interfaces parameters,**  as well as manipulating routing, policy bases routing, and tunneling.

Basic syntax:

**Ip [options] OBJECT { COMMAND | help }**

**Ip [ -force ] –batch filename**  ……this can read commands from a designated file.

MAIN IP OBJECTS

* + **ADDRESS**
  + **LINK** NETWORK DEVICES
  + **Maddress** Multicast Address
  + **Minitor** watch for netlink messages
  + **Rule** rule the routing policy database
  + **Tunnel** tunnel over IP

IP USAGE:

* + Show information for the **eth0** net interface.

$ Ip –s link show eth0

* + Show information for all network interfaces::

$ ip link

* + Set the IP address for eth0 :

$ sudo ip addr add 192.168.1.7 dev eth0

* + Bring eth0 down:

$sudo ip link set eth0 down

* + Set **MTU** to 1480 bytes for eth0:

$ sudo ip link set eth0 mtu 1480

* + Set the networking route:

$ sudo ip route add 172.16.1.0 /24 via 192.168.1.5

**IFCONFIG:**

* **Display information about all interfaces**

$ Ifconfig

* Set ip address to 192.168.1.50 on ingerface eth0:

$ sudo ifconfig eth0 192.168.1.50

* Set the netmask to 24-bit

$ sudo ifconfig eth0 netmask 255.255..255.0

* Bring interface eth0 up:

$ sudo ifconfig eth0 up

* Bring eth0 down:

$ sudo ifconfig eth0 down

* Set the  **MTU ( M**aximum Transfer Unit ) to 1480 bytes for eth0

$ sudo ifconfig eth mtu 1480

**NIC Configuration files:**

**RED HAT:**

/etc/sysconfig/network

/etc/sysconfig/network-scripts/ifcfg-ethX

/etc/sysconfig/network-scripts/ifcfg-ethX:Y

/etc/sysconfig/network-scripts/route-ethX

**DEBAN:**

**/etc/**network/interfaces

**SUCE:**

**/etc/**sysconfig/network

**NETWORK MANAGER:**

Nmtui: GUI

Nmcli:

If we need to run scripts that change the network configuration and use command line.

<https://fedoraproject.org/wiki/Networking/CLI>

**$ man nmcli-examples**

* + Display all connections

$ nmcli connection show

* + List available access points(AP) to connect to

$ nmcli device wifi list

* + Create a new connection to an open AP

$ nmcli device wifi connect <SSID|BSSID>

* + edit an already existing connection using an interactive editor

$ nmcli connection edit <connection name>

**Routing:**

**Is the process of selecting paths on a network along which to send network traffic. The routing table**  is a list of routes to other networks managed by the system. It defines paths to all networks and hosts, sending remote traffic to routers.

Show the current routing table:

$ route –n

**Default route:**

**: To configured manualy (static) with nmcli**

$sudo nmcli con mod virbr0 ipv4.routes 192.168.10.0/24 \+ipv4.gateway 192.168.122.0

Or

$sudo nmcli con up virbr0

Or  **In Red Hat systems** modify the **/etc/sysconfig/network**

**To set the deault getway at runtime:**

**$** sudo route add default gw 192.168.1.10 enp2s0

$route

**Note :** this might wipe out network connection !! can be restored by resetting the network, or setting old route ( changes are note persistent**)**

**Static Routes:**

Used to control packet flow when ether is more than one router or route. They are defined for each interface and can be either persistent or non-persistent.

Either **Route** or **ip** can be used to set **a non-persistent** route as in:

$sudo ip route add 10.5.0.0/16 via 192.168.1.100

To set a persisten route:

Edit  **/etc/sysconfig/network-scripts/route-ethX**

$ cat /etc/sysconfig/network-scripts/route-eth0

**NAME RESOLUTION:**

is the act of translating hostnames to the IP addresses of their hosts.

There are two facilities for doing this translation:

* + - static name resolution ( using **/etc/hosts** )

/etc/hosts **holds a local database of hostnames and ip addresses**.

Its priority can be controlled by **/etc/nsswitch.conf.**

* + - dynamic name resolution ( using **DNS servers )**

**From the command line:**

$[dig |host |nslookup] pagename.org

dig: generates the most information and has many optinos

host: more compact

nslookup: older

**DNS:**

**Is configured in /etc/resolv/conf.**

**Network Diagnostics Utilities.**

* **Ping**
* **Traceroute**  used to trace a network path to a destination. Shows the routers **packets flow** though to get to a host, as well as the time it takes for each hop.
* **Mtr** conbines ping and traceroute and creates a continuoly update
* **Dig** is usefull for testing **DNS.** Also can use **host** and **nslooup** ( older programs)