**Networking**

**1.[**check the details of ethernet interface]

vi /etc/sysconfig/network-scripts/<interface name>

vi /etc/sysconfig/network-scripts/ifcfg-eno16780032

**contents**

TYPE=Ethernet

BOOTPROTO=static

DEFROUTE=yes

PEERDNS=yes

PEERROUTES=yes

IPV4\_FAILURE\_FATAL=no

IPADDR=172.16.103.62

GATEWAY=172.16.103.1

DNS1=192.168.1.25

DNS2=192.168.1.165

NETMASK=255.255.255.0

IPV6INIT=yes

IPV6\_AUTOCONF=yes

IPV6\_DEFROUTE=yes

IPV6\_PEERDNS=yes

IPV6\_PEERROUTES=yes

IPV6\_FAILURE\_FATAL=no

NAME=eno16780032

UUID=cfed4212-ddc0-4c48-80f6-da204c631a46

DEVICE=eno16780032

ONBOOT=yes

ZONE=public

"/etc/sysconfig/network-scripts/ifcfg-eno16780032" 22L, 402C

Edit the parameters and save

Device name uid type

eno16780032 cfed4212-ddc0-4c48-80f6-da204c631a46 802-3-ethernet

device

eno16780032

2.[removes the details to interface along with the routes]

Ip address flush <interface name>

[to restart network using systemctl utility]

# Systemctl restart network

[Adding a static ip address to a interface]

# Ip address add <ip address> connection <enp0s3>

[to show the ip address]

Ip address show

[Release the current ip address lease, and once the lease has been released, the client exits]

Dhclient –r

[to view the interface status]

nmcli connection show

[to make the interface up or down]

nmcli connection up(or) down <interface name>

[To remove all addresses]

Ip address flush <interface name>

[to restart network using systemctl utility]

# Systemctl restart network

**Network troubleshooting**

[ip route command without options to display the IP routing table]

# Ip route

[to list out all the tcp, udp socket connections and the unix socket connections.]

# netstat -a

[List only TCP connections]

# netstat -at

[list only udp connections]

# netstat -au

[List out only listening connections]

# netstat -tnl

[Get process name/pid and user id]

# sudo netstat -nlpt

[Get process name/pid and user id along with username]

# sudo netstat -ltpe

[Print statistics]

# netstat -s

[Display kernel routing information]

# netstat -rn

[Print network interfaces]

# netstat -i

[Print network interfaces in human friendly version]

# netstat -ie

[Get netstat output continuously]

# netstat -ct

[Display multicast group information]

# netstat –g

[to list active internet connections]

#netstat –tulpen

For more info, <http://www.binarytides.com/linux-netstat-command-examples/>

[ to display the “A Record” ( IP Address ) of the domain ]

# nslookup google.com

[to display the “A record” of the domain with question, answer, dns section, query section]

# dig google.com

For more info, <http://www.thegeekstuff.com/2012/02/dig-command-examples/>

[show you the path a packet of information takes from your computer to one you specify]

# traceroute google.com

[to find out what services a computer is running]

# nmap

For more info, <https://www.linux.com/learn/beginners-guide-nmap>

**Configuring hostname**

[ to check the hostname ]

# hostname

# hostname -f

# uname –n

[ to check all the details of hostname and kernel ]

# hostnamectl

[ to reset hostname ]

#  hostname <new hostname>

[ to update the hostname without opening a new terminal ]

# bash

[ to set fully qualified domain name ]

# hostnamectl  set-hostname <new hostname.example.com>

[ to check the hostname ]

# cat /etc/hostname

# hostnamectl set-hostname “cento’s.example.com”

# cat /etc/machine-info  ( shows pretty name )

[ to coinfugure pretty name ]

# hostnamectl set-hostname "hellolinux" –pretty

[to configure transcient name ]]

# hostnamectl set-hostname "hellolinux" –transcient

[ to configure standard hostname ]]

# hostnamectl set-hostname "hellolinux" –standard

Note : Hostnamectl tool distinguishes three different hostnames: the high-level "pretty" hostname which might include all kinds of special characters (e.g. "Lennart's Laptop"), the static hostname which is used to initialize the kernel hostname at boot (e.g. "lennarts-laptop"), and the transient hostname which is a fallback value received from network configuration. If a static hostname is set, and is valid (something other than localhost), then the transient hostname is not used.

**Hostname : localhosts file**

[to add alias name ]

# vi /etc/hosts

<IP Address> <HOSTNAME.example.org> <aliasname>

**DNS name resolution**

[to know the dns server details ]

# cat /etc/resolv.conf

# yum install -y bind-utils

# dig [www.google.com](http://www.google.com)

# dig +short [www.google.com](http://www.google.com) ( give only ip addresses)

**Network Manager**

[to add a new Ethernet interface]

Syntax:

# nmcli connection add type ethernet con-name connection-name ifname interface-name

Example:

# nmcli connection add type ethernet con-name my-office ifname ens3

[ to view interfaces]

# nmcli connection show

[to make interface down]

# nmcli conn down enp0s3

[to view ip address]

# ip add show

[to make interface up and running]

# nmcli conn up <interface name>

# nmcli conn up enp0s3

[to make interface up and running]

# ip add show

[to make interface down]

# nmcli conn down home

[ to check the status of the network]

# systemctl status network

Note : Difference between Nat, Bridge and Host only network in Virtual machine.

**Host-Only**: The VM will be assigned one IP, but it's only accessible by the host. No other computers can access it.

**NAT**: Just like your home network with a wireless router, the VM will be assigned in a separate subnet, like 192.168.6.1 is your host computer, and VM is 192.168.6.3, then your VM can access outside network like your host, but no outside access to your VM directly, it's protected.

**Bridged**: Your VM will be in the same network as your host, if your host IP is 172.16.120.45 then your VM will be like 172.16.120.50. It can be accessed by all computers in your host network.

**RouteTables**

[ to check the networks directly connected to interfaces]

# ip route show

[ to check the routing table ]

# route

[To add a static route to a host address, in other words to a single IP address ]

# ip route add default via <server ip address> <interface name>

[ to check routing is enables or not]

# cat /proc/sys/net/ipv4/ip\_forward ( ‘0’ shows routing is turned off )

[ to enable routing ]

# nano /etc/sysctl.conf

Go complete down and add net.ipv4.ip\_forward=1 and save the file (:wq)

[stopping firewall ]

# systemctl stop firewalld