Linux Operating System and Application

Basic Network Management

Introduction to Network Management

- What is network management?
- Importance in Linux system administration
- Tools & utilities overview (CLI-focused)

/etc/hosts File

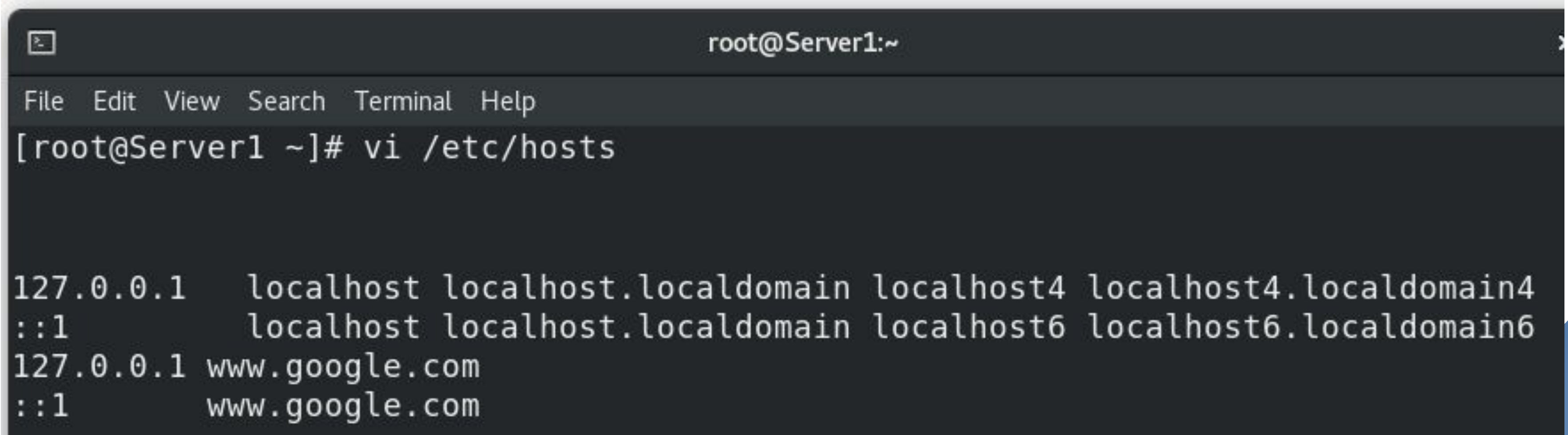
- Maps IP addresses to hostnames on the network.
- Similar to the hosts file in Windows.
- File syntax:

```
IP_address<Tab>Fully.Qualified.Domain.Name<space>[host_alias]*
```

- Example:

```
192.168.1.10     khtn.edu.vn     khtn
```

- Applications will consult this file first when resolving a hostname.

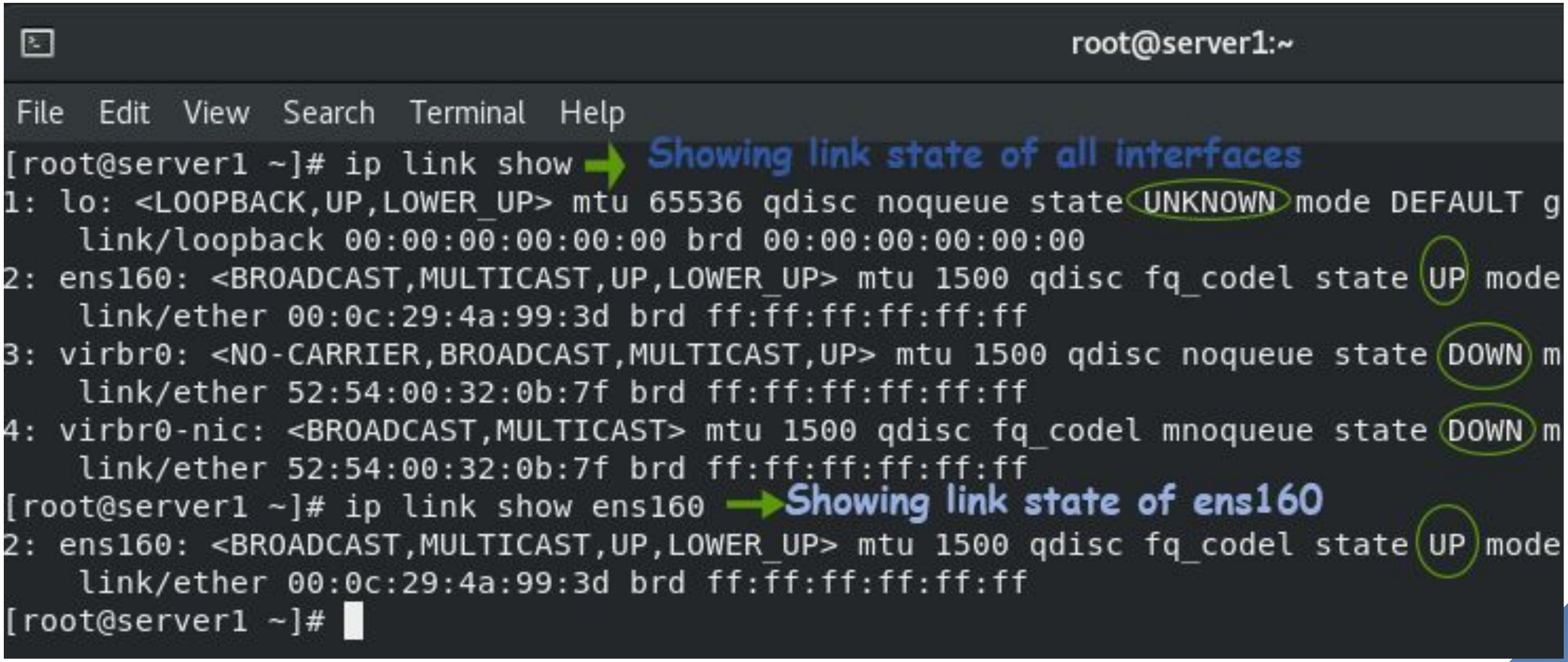


```
[root@Server1 ~]# vi /etc/hosts


127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
127.0.0.1 www.google.com
::1       www.google.com
```

Understanding Network Interfaces

- ❑ Types of network interfaces (e.g., `eth0`, `ens33`, `lo`)
- ❑ Check interfaces with:

```
ip link show

ip link show [network_interface]
```



```
[root@server1 ~]# ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT g
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode
    link/ether 00:0c:29:4a:99:3d brd ff:ff:ff:ff:ff:ff
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN m
    link/ether 52:54:00:32:0b:7f brd ff:ff:ff:ff:ff:ff
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel mnoqueue state DOWN m
    link/ether 52:54:00:32:0b:7f brd ff:ff:ff:ff:ff:ff
[root@server1 ~]# ip link show ens160
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode
    link/ether 00:0c:29:4a:99:3d brd ff:ff:ff:ff:ff:ff
[root@server1 ~]#
```

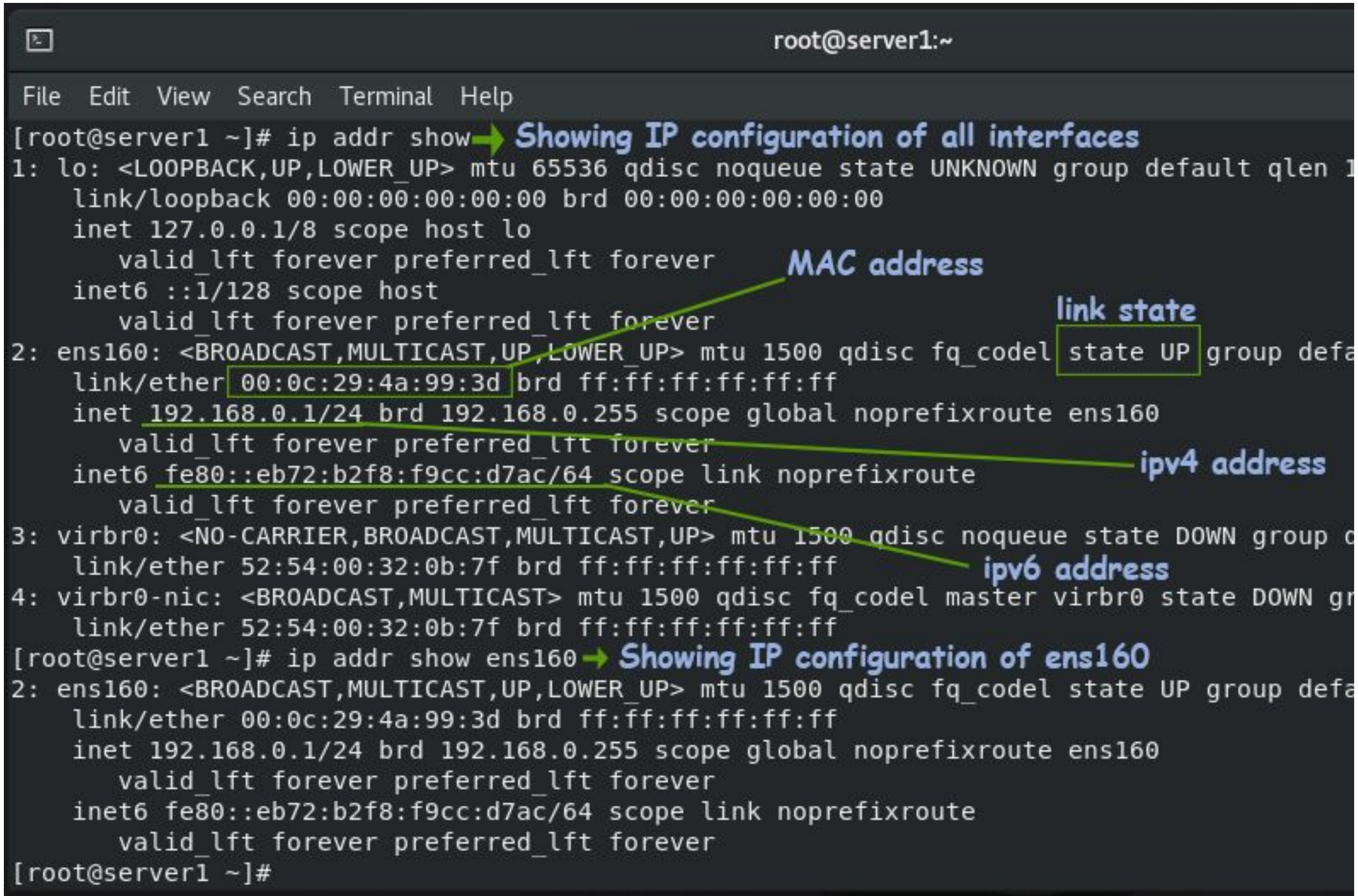
Showing link state of all interfaces

Showing link state of ens160

Checking IP Configuration

- ❑ View IP addresses:

```
ip addr show

ip addr show [network_interface]
```



```
[root@server1 ~]# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    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: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group defa
    link/ether 00:0c:29:4a:99:3d brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.1/24 brd 192.168.0.255 scope global noprefixroute ens160
       valid_lft forever preferred_lft forever
    inet6 fe80::eb72:b2f8:f9cc:d7ac/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group d
    link/ether 52:54:00:32:0b:7f brd ff:ff:ff:ff:ff:ff
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel master virbr0 state DOWN gr
    link/ether 52:54:00:32:0b:7f brd ff:ff:ff:ff:ff:ff
[root@server1 ~]# ip addr show ens160
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group defa
    link/ether 00:0c:29:4a:99:3d brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.1/24 brd 192.168.0.255 scope global noprefixroute ens160
       valid_lft forever preferred_lft forever
    inet6 fe80::eb72:b2f8:f9cc:d7ac/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
[root@server1 ~]#
```

Showing IP configuration of all interfaces

MAC address

link state

ipv4 address

ipv6 address

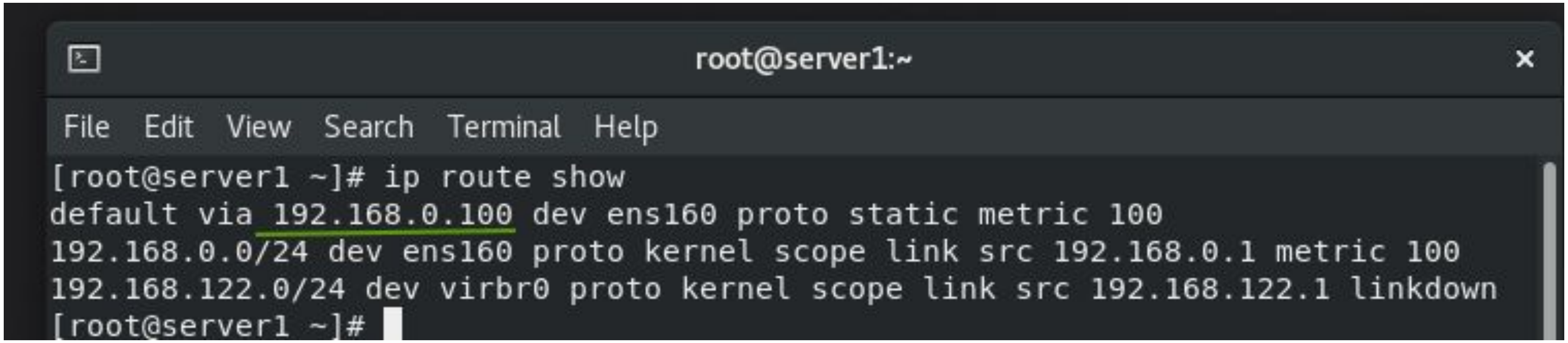
Showing IP configuration of ens160

Checking IP Configuration

- ❑ Show routing table:

```
ip route show
```

This command displays the default gateway IP address. The default gateway connects the host to the remote network.



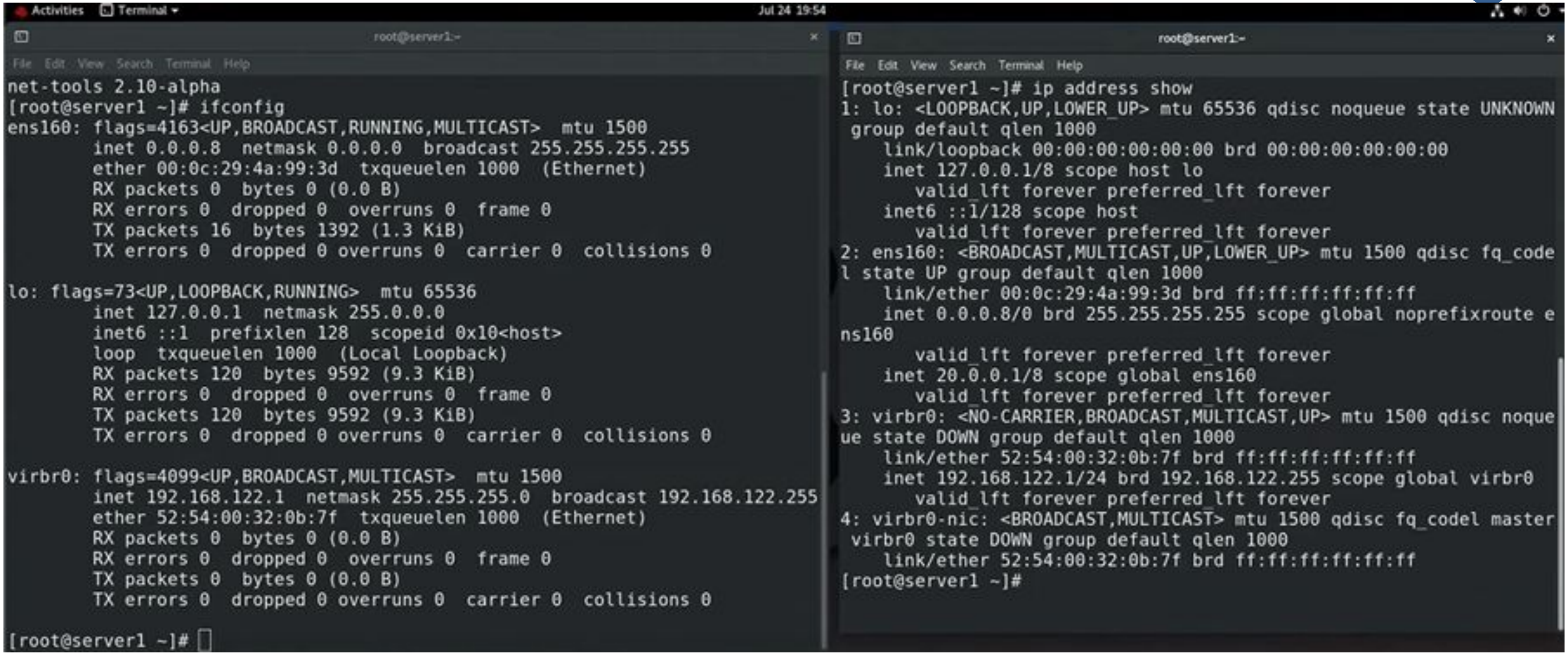
```
[root@server1 ~]# ip route show
default via 192.168.0.100 dev ens160 proto static metric 100
192.168.0.0/24 dev ens160 proto kernel scope link src 192.168.0.1 metric 100
192.168.122.0/24 dev virbr0 proto kernel scope link src 192.168.122.1 linkdown
[root@server1 ~]#
```

ifconfig Command (deprecated)

- ❑ The **ifconfig** command is deprecated. The **ip** command replaces the **ifconfig** command.
- ❑ In earlier versions of Linux, the **ifconfig** command was the default utility for checking and verifying IP configuration.

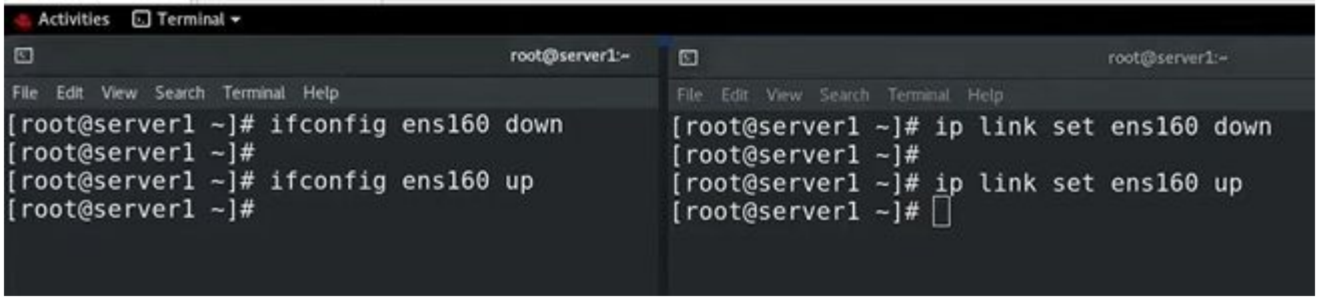
ifconfig command	**Task**	**ip command**
ifconfig	Show ip configuration of all interfaces	ip address show
ifconfig [*interface*]	Show ip configuration of the specified interface	ip address [*interface*]
ifconfig [*interface*] down	Bring the specified interface down	ip link set [*interface*] down
ifconfig [*interface*]	up Bring the specified interface up	ip link set [*interface*] up
ifconfig [*interface*] [*ip address*] ifconfig [*interface*] [*subnet mask*]	Assign a temporary IP address to the specified interface	ip address add [*ip address/subnet mask*] dev [*interface*]

ifconfig Command vs ip Command


Activities
Terminal
Jul 24 19:54
root@server1:~
File Edit View Search Terminal Help
root@server1:~
File Edit View Search Terminal Help

```
net-tools 2.10-alpha
[root@server1 ~]# ifconfig
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 0.0.0.8  netmask 0.0.0.0  broadcast 255.255.255.255
        ether 00:0c:29:4a:99:3d  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 16  bytes 1392 (1.3 KiB)
        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 120  bytes 9592 (9.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 120  bytes 9592 (9.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

virbr0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.122.1  netmask 255.255.255.0  broadcast 192.168.122.255
        ether 52:54:00:32:0b:7f  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

[root@server1 ~]# 
```

```
[root@server1 ~]# ip address show
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: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_code
l state UP group default qlen 1000
    link/ether 00:0c:29:4a:99:3d brd ff:ff:ff:ff:ff:ff
    inet 0.0.0.8/0 brd 255.255.255.255 scope global noprefixroute e
ns160
       valid_lft forever preferred_lft forever
    inet 20.0.0.1/8 scope global ens160
       valid_lft forever preferred_lft forever
3: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noque
ue state DOWN group default qlen 1000
    link/ether 52:54:00:32:0b:7f brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
       valid_lft forever preferred_lft forever
4: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel master
 virbr0 state DOWN group default qlen 1000
    link/ether 52:54:00:32:0b:7f brd ff:ff:ff:ff:ff:ff
[root@server1 ~]#
```


Activities
Terminal
root@server1:~
File Edit View Search Terminal Help
root@server1:~
File Edit View Search Terminal Help

```
[root@server1 ~]# ifconfig ens160 down
[root@server1 ~]#
[root@server1 ~]# ifconfig ens160 up
[root@server1 ~]#
```

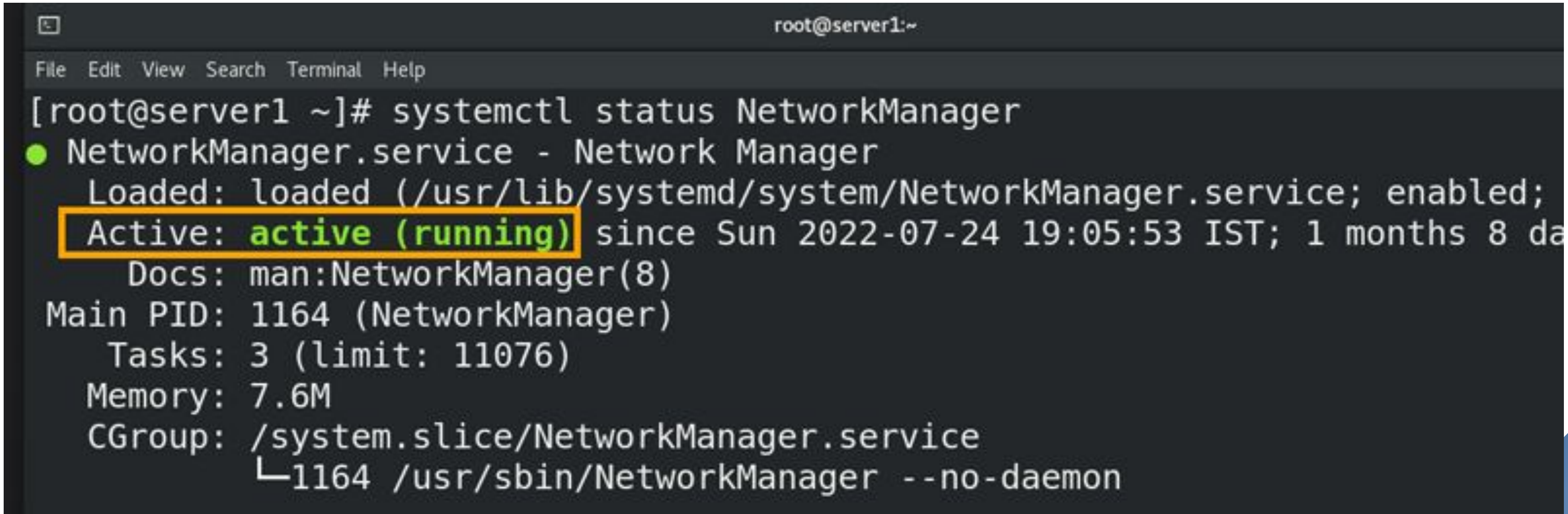
```
[root@server1 ~]# ip link set ens160 down
[root@server1 ~]#
[root@server1 ~]# ip link set ens160 up
[root@server1 ~]# 
```

Managing Network Services (systemd & NetworkManager)

`NetworkManager:` A tool for managing network connections on Linux.

- It can manage wired (Ethernet), wireless (Wi-Fi), VPN connections...
- Main tools: nmcli and nmtui
- Start/enable/disable/restart networking service:

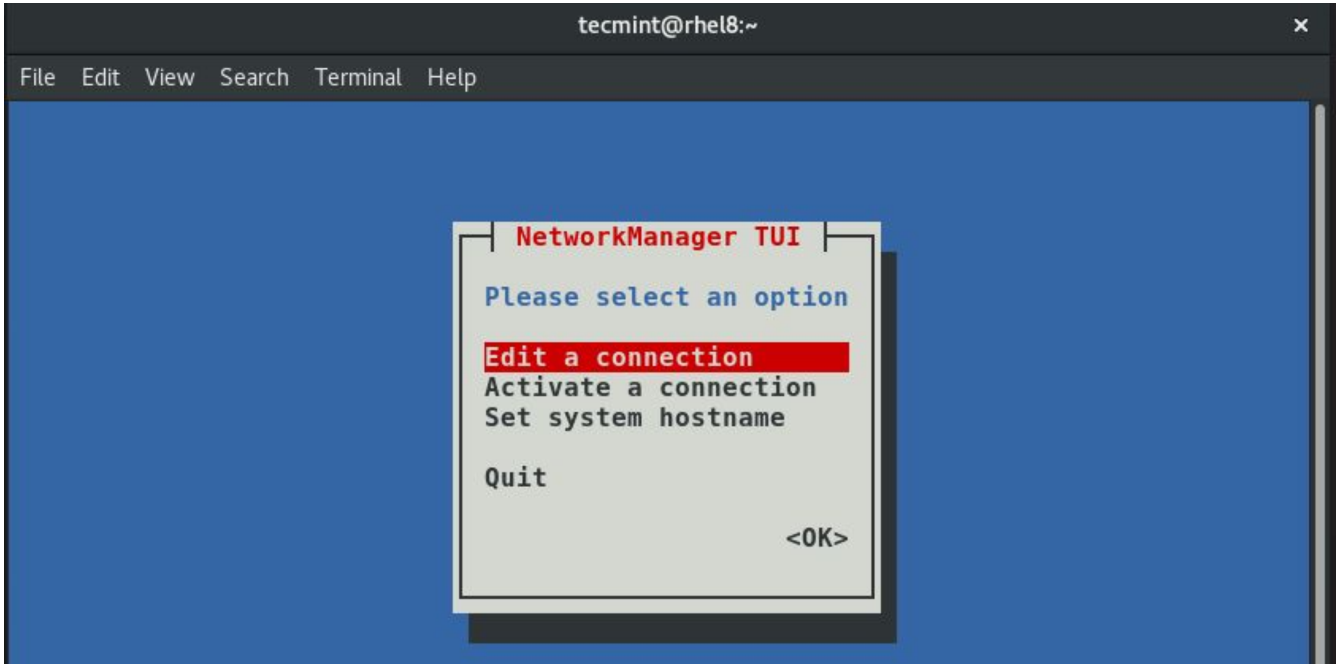
```
systemctl restart NetworkManager

systemctl enable NetworkManager
```



```
[root@server1 ~]# systemctl status NetworkManager
● NetworkManager.service - Network Manager
   Loaded: loaded (/usr/lib/systemd/system/NetworkManager.service; enabled;
   Active: active (running) since Sun 2022-07-24 19:05:53 IST; 1 months 8 da
     Docs: man:NetworkManager(8)
 Main PID: 1164 (NetworkManager)
    Tasks: 3 (limit: 11076)
   Memory: 7.6M
   CGroup: /system.slice/NetworkManager.service
           └─1164 /usr/sbin/NetworkManager --no-daemon
```

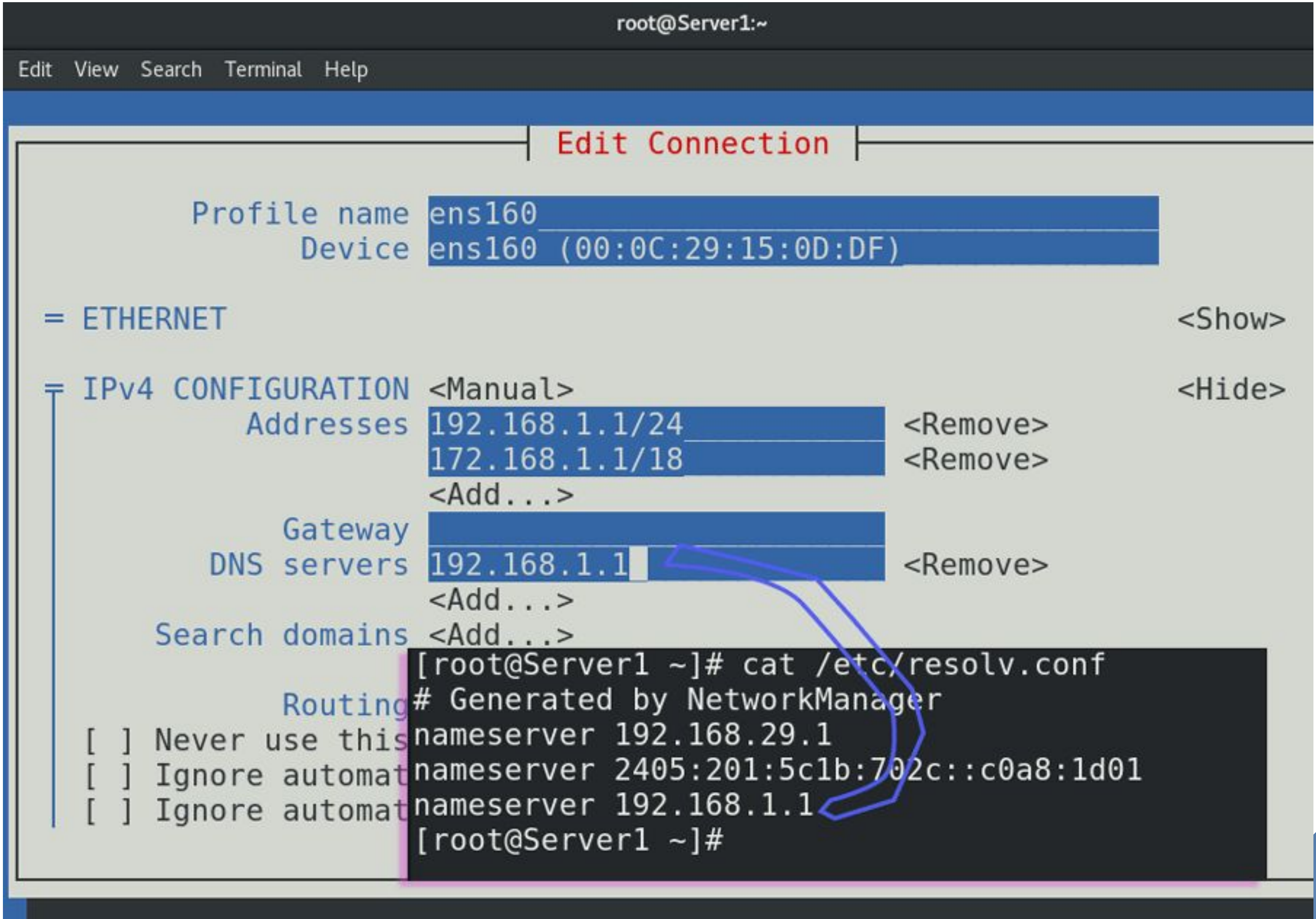
nmtui (NetworkManager Text User Interface)

- nmtui
  - Activate a connection: Enable or disable network connections.
  - Edit a connection: Create, modify, or delete network connections (Ethernet, Wi-Fi).
  - Set system hostname: Configure the system's hostname.



```
NetworkManager TUI

Please select an option

Edit a connection
Activate a connection
Set system hostname

Quit

                 <OK>
```

nmtui (NetworkManager Text User Interface)



```
                     ┤ Edit Connection ├

       Profile name ens160
             Device ens160 (00:0C:29:15:0D:DF)

 = ETHERNET                                                    <Show>

 ╤ IPv4 CONFIGURATION <Manual>                                 <Hide>
 │         Addresses 192.168.1.1/24         <Remove>
 │                   172.168.1.1/18         <Remove>
 │                   <Add...>
 │           Gateway
 │       DNS servers 192.168.1.1            <Remove>
 │                   <Add...>
 │    Search domains <Add...>
 │
 │           Routing
 │ [ ] Never use this
 │ [ ] Ignore automat
 │ [ ] Ignore automat
```

```
[root@Server1 ~]# cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 192.168.29.1
nameserver 2405:201:5c1b:702c::c0a8:1d01
nameserver 192.168.1.1
[root@Server1 ~]#
```

Assigning Static IP Address (CLI)

- ❑ Files to Edit: /etc/sysconfig/network-scripts/ifcfg-<interface>
- ❑ Example:

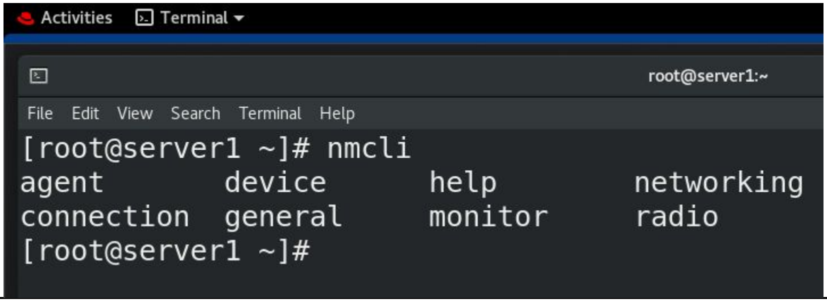
```
BOOTPROTO=static

ONBOOT=yes

IPADDR=192.168.1.100

NETMASK=255.255.255.0

GATEWAY=192.168.1.1

DNS1=8.8.8.8
```

- ❑ Reload settings:

```
# nmcli connection reload

# systemctl restart NetworkManager
```

nmcli Command

- ❑ Syntax: #nmcli [options] [section] [action]
- ❑ Section:



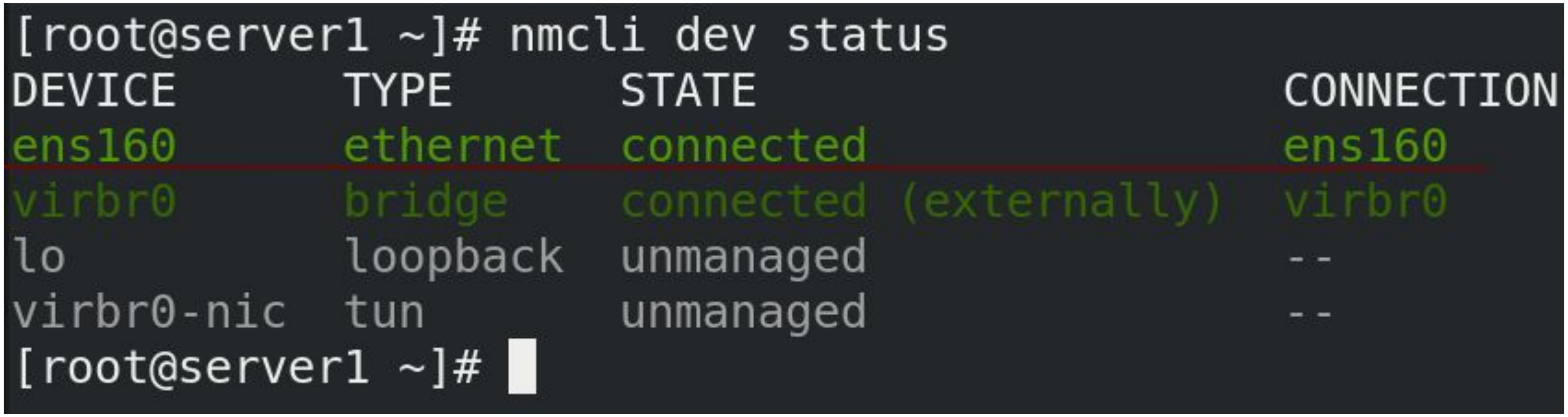
```
[root@server1 ~]# nmcli
agent       device      help        networking
connection  general     monitor     radio
[root@server1 ~]#
```

Section	Description
help	Used to get help related to the nmcli options and parameters
general	Used to get the status and global configuration of the NetworkManager
networking	Used to start, restart, and manage NetworkManager
radio	Used to manage wireless devices and protocols
connection	Used to manage connections
device	Used to manage network devices
agent	Used to configure and manage various security settings
monitor	Used to monitor network changes

nmcli Command Examples

- ❑ List all available network devices and their current status

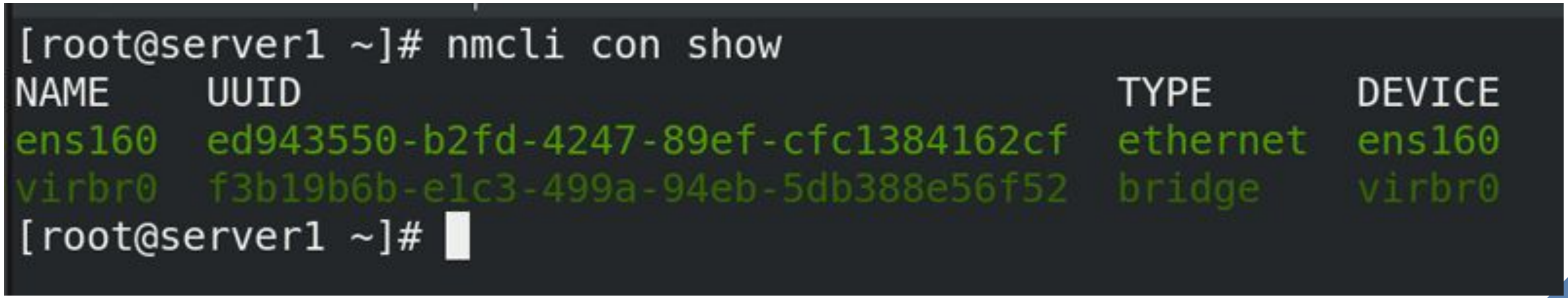
  # nmcli dev status



```
[root@server1 ~]# nmcli dev status
DEVICE      TYPE      STATE                   CONNECTION
ens160      ethernet  connected               ens160
virbr0      bridge    connected (externally)  virbr0
lo          loopback  unmanaged               --
virbr0-nic  tun       unmanaged               --
[root@server1 ~]# 
```

- ❑ A device can have multiple connections, view all of them.

  # nmcli con show

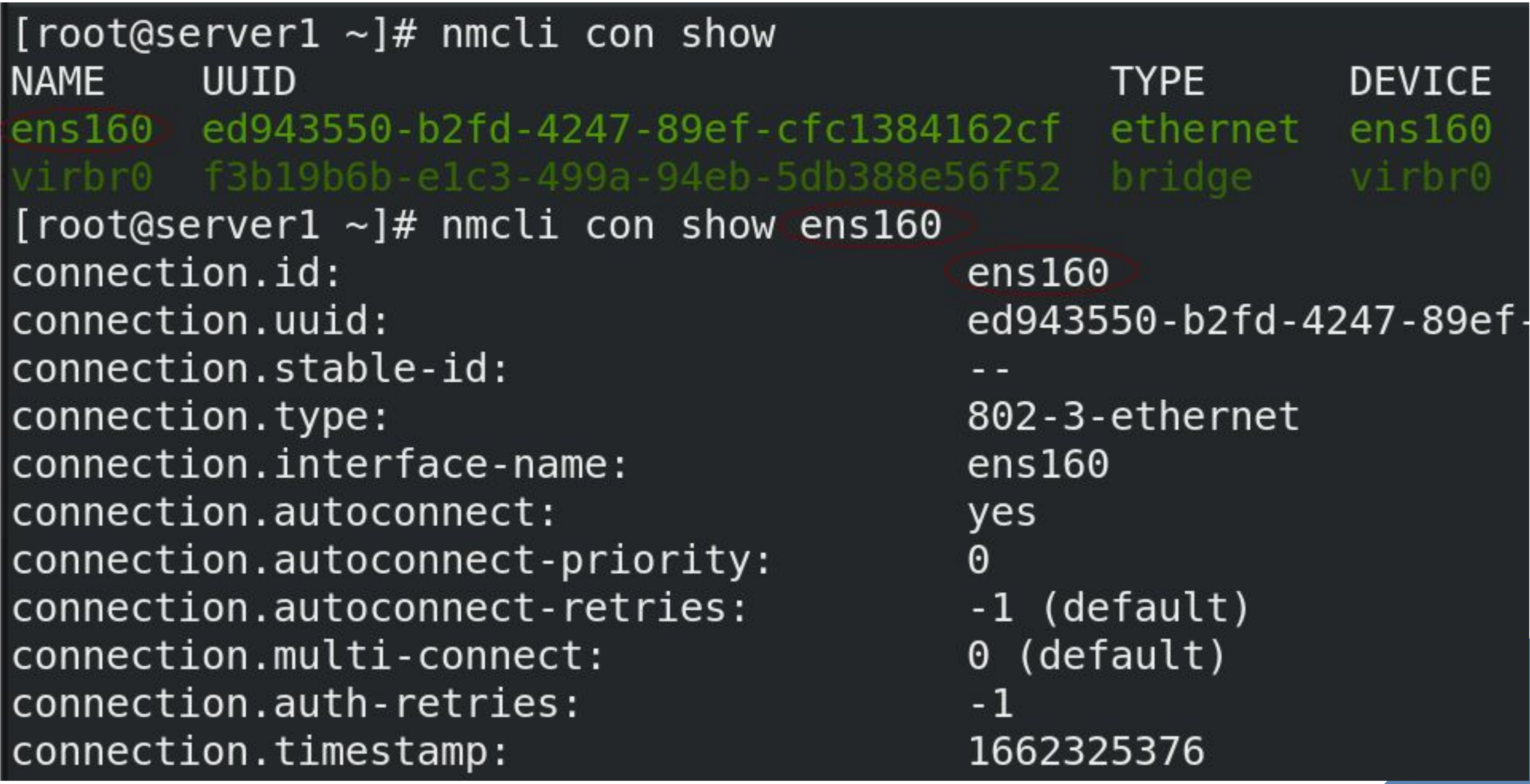


```
[root@server1 ~]# nmcli con show
NAME    UUID                                  TYPE      DEVICE
ens160  ed943550-b2fd-4247-89ef-cfc1384162cf  ethernet  ens160
virbr0  f3b19b6b-e1c3-499a-94eb-5db388e56f52  bridge    virbr0
[root@server1 ~]# 
```

nmcli Command Examples

- ❑ Lists all configured values of the **ens160** connection

  # nmcli con show ens160



```
[root@server1 ~]# nmcli con show
NAME     UUID                                  TYPE      DEVICE
ens160   ed943550-b2fd-4247-89ef-cfc1384162cf  ethernet  ens160
virbr0   f3b19b6b-e1c3-499a-94eb-5db388e56f52  bridge    virbr0
[root@server1 ~]# nmcli con show ens160
connection.id:                          ens160
connection.uuid:                        ed943550-b2fd-4247-89ef-
connection.stable-id:                   --
connection.type:                        802-3-ethernet
connection.interface-name:              ens160
connection.autoconnect:                 yes
connection.autoconnect-priority:        0
connection.autoconnect-retries:         -1 (default)
connection.multi-connect:               0 (default)
connection.auth-retries:                -1
connection.timestamp:                   1662325376
```

nmcli Command

- ❑ Remove an active connection

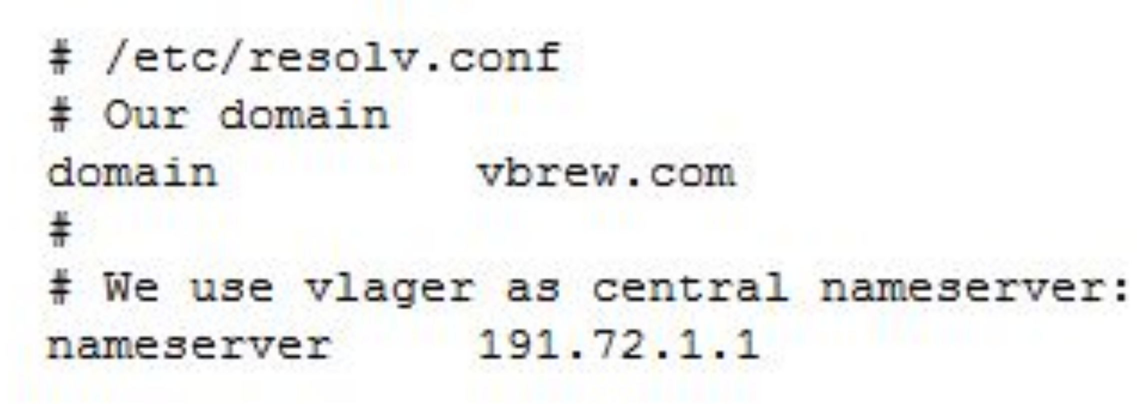
  #nmcli con down [*connection name*]

- ❑ Attach a new or updated connection

  #nmcli con up [*connection name*]

DNS Configuration

- The /etc/resolv.conf file is used to define the name servers that the system will use for DNS (domain name) resolution.
- Common directives:
  - domain: Specifies the DNS domain of the system.
  - nameserver: Specifies the IP address or hostname of a name server the system will use.



```
# /etc/resolv.conf
# Our domain
domain         vbrew.com
#
# We use vlager as central nameserver:
nameserver     191.72.1.1
```

Hostname Configuration

❑ View current hostname:

```
hostnamectl status
```

❑ Change hostname:

```
hostnamectl set-hostname myserver.local
```

/etc/services File

- Contains a list of **network ports** and the **services** that use them.
- Used by various networking programs to match service names with port numbers.
- When defining a **new service**, the system administrator must add a **service name** and **port number** pair to `/etc/services`.
- **Port Ranges**
  - **Ports 0–1024**: Reserved ports (well-known, system-defined).
  - **Ports >1024**: Can be assigned as needed by applications.

```
# /etc/services: database of service names -> port/protocol
# Format: service-name  port/protocol  [aliases]  [# comment]

http        80/tcp     www      # World Wide Web HTTP
https       443/tcp             # HTTP over TLS/SSL
ssh         22/tcp              # Secure Shell
smtp        25/tcp     mail     # Simple Mail Transfer
domain      53/udp              # Domain Name Server
mysql       3306/tcp            # MySQL Database Service
```

Testing Network Connectivity

Tool	Purpose	Example
`ping`	Check if a host is reachable	`ping google.com`
`traceroute`	Show the path to a host	`traceroute google.com`
`curl`	Test HTTP/HTTPS request (get data)	`curl https://example.com`
`wget`	Download files from URLs	`wget https://example.com`
`dig`	Detailed DNS lookup	`dig google.com`
`nslookup`	Simple DNS query	`nslookup google.com`

Tip: Install `traceroute`, `dig`, and `nslookup` with `sudo yum install traceroute bind-utils`

Useful Network Troubleshooting Tools

Tool	Purpose	Example
`ss -tuln`	Show open TCP/UDP ports (replace `netstat`)	`ss -tuln`
`tcpdump`	Capture and analyze network traffic	`tcpdump -i eth0`
`ip -s link`	Show interface statistics (RX/TX)	`ip -s link`

Note:

- `netstat` is deprecated — use `ss` instead.
- `tcpdump` may need to be installed: `sudo yum install tcpdump`

route Command (deprecated)

Overview:

- Used to **view**, **edit**, and **manage** the system’s routing table.
- Allows admins to define **custom default routes**.
- Enables manual configuration of the **default gateway**.

Use Cases:

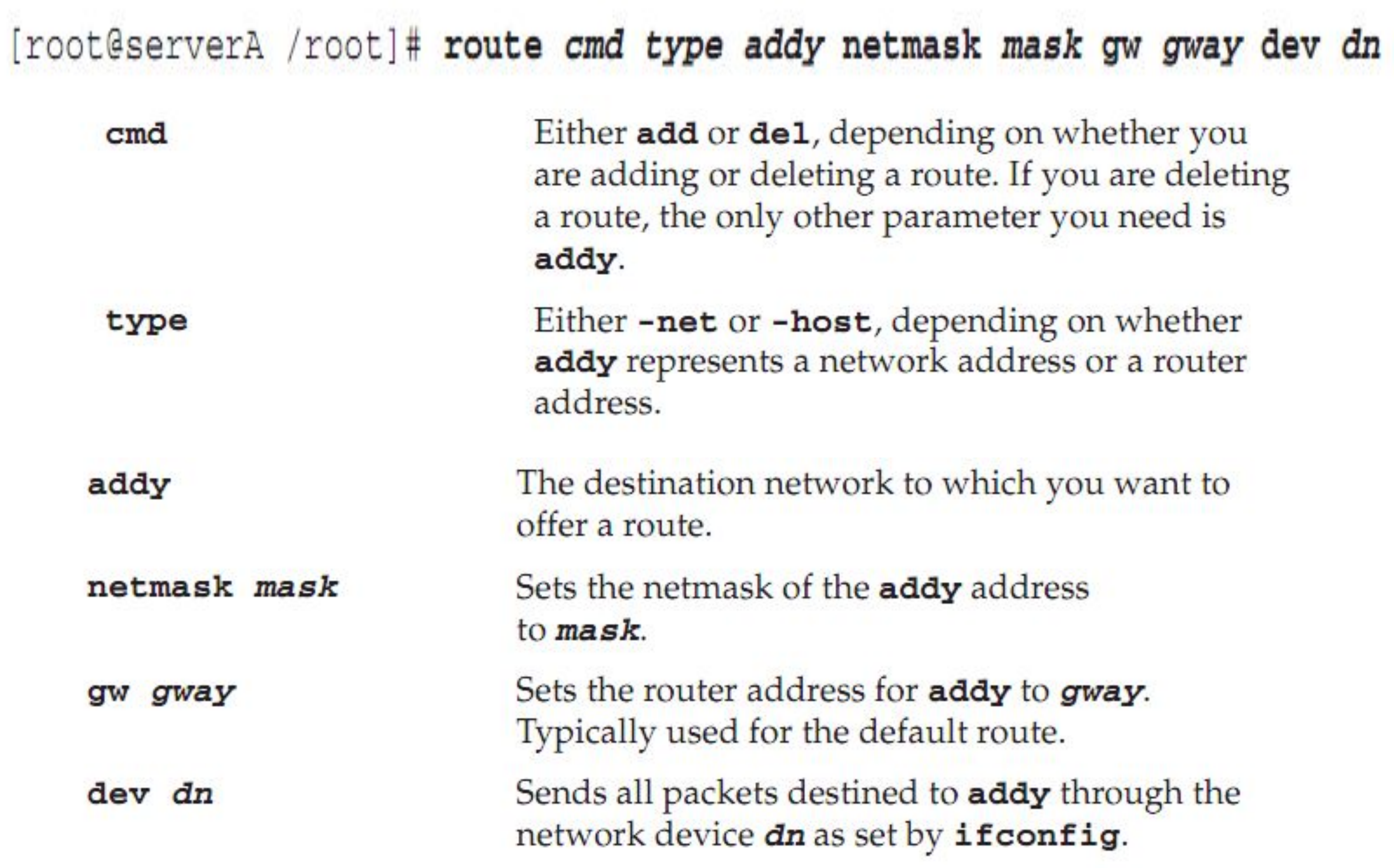
- Configure **static routes** to specific networks.
- Set or change the system’s **default gateway**.

Note:

- `route` is **deprecated** in modern systems.
- Use `ip route` instead for new configurations.

route Command

- Syntax:



```
[root@serverA /root]# route cmd type addy netmask mask gw gway dev dn
```

`cmd` Either **add** or **del**, depending on whether you are adding or deleting a route. If you are deleting a route, the only other parameter you need is **addy**.

`type` Either **-net** or **-host**, depending on whether **addy** represents a network address or a router address.

`addy` The destination network to which you want to offer a route.

`netmask mask` Sets the netmask of the **addy** address to ***mask***.

`gw gway` Sets the router address for **addy** to ***gway***. Typically used for the default route.

`dev dn` Sends all packets destined to **addy** through the network device ***dn*** as set by **ifconfig**.

route Command vs ip Command

Task	Legacy Command	Modern `ip` Command
View routing table	`route -n`	`ip route show`
Add route to network	`route add -net 192.56.76.0 netmask 255.255.255.0 eth0`	`ip route add 192.56.76.0/24 dev eth0`
Add default gateway	`route add default gw 192.168.1.254`	`ip route add default via 192.168.1.254`

Q&A