

CentOS Networking

1. Document network setup. Your setup should include how to find all the networking information, as well as the basics on how your server does networking by default. This should include locations of any files you've created or need to reference. You should include any programs you've installed or updated as well, with dates of when you've done those things. Include a short document including how you did any of the installs or updates.

1a.

Tools and Updates

- The server was last updated on 10/11/24 using `sudo yum update`
- Installed network tool `openssh` on 9/7/24

`Openssh` allows for ssh connections on the ubuntu server, in turn allowing me to connect with my preferred ssh tool `putty`.

- Installed network tool `fail2ban` on 10/10/24

`Fail2ban` is a critical tool that enhances the security of the server by banning malicious connections that attempt to brute force their way into the server over the network.

Commands and their function

Now, to quickly find a comprehensive list of networking information on the server setup you can use the following command:

- `nmcli`

When inputted, you will see that your DNS information, interface connections and their mac addresses, ip addresses of interfaces, and the route information. This is a great way to generally see some of the most important network information about a server. Centos handles network management differently from Ubuntu. The biggest difference between `netplan` (ubuntu) and `nmcli` is that `nmcli` configuration is set via the command line where Ubuntu's `netplan` is handled through YAML files.

```
[flaureano@localhost ~]$ nmcli
ens33: connected to ens33
    "Intel 82545EM"
    ethernet (e1000), 00:0C:29:C7:7B:AB, hw, mtu 1500
    ip4 default
    inet4 192.168.186.131/24
    route4 192.168.186.0/24 metric 100
    route4 default via 192.168.186.2 metric 100
    inet6 fe80::20c:29ff:fec7:7bab/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: 192.168.186.2
    domains: localdomain
    interface: ens33
```

Additionally, you can dive deeper into the network configuration by running a few other commands:

- `ip a`

`Ip` will show more in depth information about the network interfaces that are connected to the server.

```
[flaureano@localhost ~]$ ip a
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: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:c7:7b:ab brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.186.131/24 brd 192.168.186.255 scope global dynamic noprefixroute ens33
        valid_lft 966sec preferred_lft 966sec
    inet6 fe80::20c:29ff:fec7:7bab/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[flaureano@localhost ~]$
```

- `route -n`

`route -n` will show a routing table of your network without resolving the addresses. If you want to see a routing table with resolved addresses, you can simply type `route`

```
[flaureano@localhost ~]$ route -n
Kernel IP routing table
Destination    Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0        192.168.186.2  0.0.0.0         UG    100    0      0 ens33
192.168.186.0  0.0.0.0        255.255.255.0   U     100    0      0 ens33
[flaureano@localhost ~]$
```

- `sudo ss`

`sudo ss` will show socket information and the port they're listing on which can be useful for troubleshooting network issues. However this list can be extremely long so if you want to see a summary you can add the `-s` switch (`ss -s`)

```
[flaureano@localhost ~]$ sudo ss -s
Total: 141
TCP:    3 (estab 1, closed 0, orphaned 0, timewait 0)

Transport Total      IP        IPv6
RAW       1           0         1
UDP       3           2         1
TCP       3           2         1
INET      7           4         3
FRAG      0           0         0
```

- `nmcli device show`

This command will display the dns information of your server and if you would like to change it from the default, you can edit the dns information with 'nmcli connection modify' to add in your new dns server(s) but this is not recommended unless you are entire 100% sure of what youre doing.

NetworkManager and how it works with CentOS

As my server is based on CentOS, all the networking is handled through NetworkManager which houses the network configuration. NetworkManager works through the connection profiles that are applied to network interfaces. These profiles are stored in `/etc/NetworkManager/system-connections/` and handle the network information like DNS, ip, and routing table for each interface. As the server is hosted on a VM, the network traffic gets routed from the VM onto my computer and from there it routes out to the internet. This setup is standard as I have not made any changes with dns or the routing table.