System Networking

Networking configuration and commands

- Networking Commands
- Network Managers
- DNS and Hosts Resolution
- ARP

Networking Commands

==========

Show networking information

show current networking configuration and status

```
ip a
```

Show IPv4 or IPv6 specifically

```
ip -4 a


ip -6 a
```

Show IPs assigned to interfaces in simple format

```
ip -br addr
```

show network interfaces

```
ip l
```

show configuration for a specific interface

```
ip a s interfacename
```

show route table

```
ip r
```

show arp cache

```
ip n
```

Changing networking information

Add an IP address to an interface:

```
ip a a IPADDRESS dev interfacename
```

Delete an IP from an interface

==========

Network Troubleshooting

---

traceroute/6

The traceroute command is used to send a request to a domain/IP. The output then includes each server that the request has passed through to get to the target:

```
root@test:~# traceroute google.com
traceroute to google.com (142.250.185.78), 30 hops max, 60 byte packets
 1  pfsense.b4sed.xyz (192.168.1.1)  0.162 ms  0.120 ms  0.097 ms
 2  100.88.196.1 (100.88.196.1)  0.374 ms  0.392 ms  0.372 ms
 3  core23.fsn1.hetzner.com (213.239.203.141)  0.432 ms core22.fsn1.hetzner.com
(213.239.254.113)  5.252 ms core23.fsn1.hetzner.com (213.239.203.141)  0.383 ms
 4  core21.fsn1.hetzner.com (213.239.224.14)  5.029 ms core5.fra.hetzner.com (213.239.224.78)
 5.003 ms  4.982 ms
 5  72.14.218.176 (72.14.218.176)  5.067 ms 142.250.160.234 (142.250.160.234)  5.087 ms
72.14.218.94 (72.14.218.94)  6.109 ms
 6  * * *
 7  142.250.214.190 (142.250.214.190)  5.225 ms fra16s48-in-f14.1e100.net (142.250.185.78)
 5.223 ms 142.250.210.208 (142.250.210.208)  6.215 ms
```

You can also use traceroute for IPv6 connections:

```
traceroute6 ip/hostname
```

---

mtr (My TraceRoute)

The mtr command is very similar to traceroute, except the data is formatted more nicely, and the output continuously updates.

```
mtr google.com

 Host                                    Loss%   Snt   Last   Avg  Best  Wrst StDev
 1. pfsense.b4sed.xyz                    0.0%    33    0.3   0.2   0.2   0.4   0.1
 2. 100.88.196.1                         0.0%    33    9.8   1.1   0.4   9.8   1.7
 3. core21.fsn1.hetzner.com              0.0%    33    3.2   1.2   0.5   6.1   1.4
 4. hos-tr4.ex3k5.dc4.fsn1.hetzner.com   0.0%    32    5.3   5.3   5.1   5.6   0.1
 5. 142.250.160.234                      0.0%    32    5.2   5.4   5.1   6.5   0.3
 6. 72.14.239.217                        0.0%    32    5.4   5.6   5.2   7.1   0.4
 7. 142.250.62.151                       0.0%    32    5.4   5.3   5.2   5.5   0.1
 8. fra16s48-in-f14.1e100.net            0.0%    32    5.3   5.3   5.1   5.5   0.1
```

---

tracepath/6

Tracepath is similar to both of the above commands, however, it has more of a focus on the connection to each host. Tracepath spends 30 seconds analysis the connection between the local machine, and each machine identified as a hop in the trace, making it ideal to identify whether slow responses from a device on the trace are causing slow loading issues.

You can also use tracepath for IPv6 connections:

```
traceroute6 ip/hostname
```

==========

Network Managers

==========

Ubuntu - Netplan

Before making changes to the network configuration on an Ubuntu machine, ensure that Cloud-init isn't enabled. Cloud-init is essentially Canonicals attempt at having Ubuntu servers fully configure themselves, in terms of networking, Cloud-init will force an Ubuntu machine to attempt to automatically configure it's networking

Netplan uses .yaml files for network configuration, these are stored in /etc/netplan

Check DHCP IP leases info

```
netplan ip leases interfacename
```

Changes to interfaces are made in the .yaml files, once changes have been made they need to be applied.

To test changes, use the try command. This essentially implements the change for a set amount of time, after which the change is reverted.

```
netplan try
```

You can also set a custom timeout time using the below command:

```
netplan try --timeout=15
```

To permanently apply a change, use the below command:

```
netplan apply
```

==========

RHEL - nmcli

Network configuration files are stored in /etc/syconfig/network-scripts. Files stored within this directory shouldn't really be edited - these are used for functionality of the ifconfig command, rather than being the interface configuration itself.

Viewing network configuration

View interface connections

```
nmcli connection show
```

View interface device status

```
nmcli device status
```

View network device configuration

```
nmcli device show devicename
```

Editing network settings & configuration

Edit connection settings

```
nmcli connection edit connectionname
```

From here, you can edit any of the settings shown in the 'nmcli device show devicename' command

As an example, the below could be used to change the default gateway IP

```
set ipv4.gateway 10.0.0.2
```

You would then want to save changes, you're given the option to trial the change or save it permanently

```
save persistent
```

or

```
save temporary
```

The above uses the nmcli cli to make changes, you can also format commands as below to make changes without entering the dedicated cli:

Add an additional IP to a connection

```
nmcli connection modify connectionname ipv4.addresses oldIP, newIP
```

Delete a connection

```
nmcli connection delete connectionname
```

Once changes have been made via nmcli, the interface will need restarting

```
nmcli connection down


nmcli connection up
```

or

```
nmcli connection reload
```

==========

OpenSUSE - Wicked

Network configuration files are stored in /etc/sysconfig/networks

systemctl status network

Show all interfaces

```
wicked show all
```

Show info for specific interface

```
wicked show eth1
```

Take interface down or up

```
wicked ifdown eth1
```

```
wicked ifup eth1
```

DNS and Hosts Resolution

==========

DNS and Hosts Testing

```
dig domainname
```

dig from a specific DNS server

```
dig domainname @DNS_ServerIP
```

==========

DNS and Host Resolution

/etc/resolv.conf

The /etc/resolv.conf file is used to configure DNS server that your server will use for DNS lookups.

Important Note; The below documentation is related to the /etc/resolve.conf file. This is not the primary file that Linux machines will use for the resolution configuration. Instead, this file is symlinked to /run/systemd/resolve/stub-resolve.conf which is referenced by systemd-resolvd. The primary configuration file used by systemd-resolvd is /run/systemd/resolve/resolve.conf, but stub-resolv.conf is also referenced. TLDR; /etc/resolve.conf is still used, but it's not the primary place referenced by systemd-resolvd.

```
root@test:~# ls -l /etc/resolv.conf
lrwxrwxrwx 1 root root 39 Aug 10  2023 /etc/resolv.conf -> ../run/systemd/resolve/stub-
resolv.conf
```

The systemd-resolved service listens on port 53 locally: this port needs to be open in order for DNS resolution to function.

```
root@test:~# lsof -i:53
COMMAND      PID            USER   FD   TYPE  DEVICE SIZE/OFF NODE NAME
systemd-r 127318 systemd-resolve   13u  IPv4 1028696      0t0  UDP localhost:domain
systemd-r 127318 systemd-resolve   14u  IPv4 1028697      0t0  TCP localhost:domain (LISTEN)
```

/etc/Resolve.conf config:

Below is a typical default configuration you might see on a Linux system:

```
# This file configures your system's DNS resolution.


nameserver 127.0.0.53  # Local DNS server (systemd-resolved)
options edns0 trust-ad  # Enables EDNS for performance
search b4sed.xyz        # Search domain to append to incomplete names
```

`nameserver` - specifies where the system looks for DNS resolution

`search` - This is the default search domain. For example, if a lookup is made to google, this option would append .b4sed.xyz to the end: google.b4sed.xyz

`options:`

`edns0` - enables a potentially performance-enhancing feature.

`trust-ad` - instructs your resolver to accept and potentially use the information in the Additional Records section without further verification.

View current DNS configuration:

```
resolvectl status
```

/etc/hosts

The /etc/hosts file can be thought of as essentially a local DNS configuration. This means that DNS entries can be mapped here, overwriting any DNS entries provided by an external service.

Entries into the /etc/hosts file can be formatted as follows:

```
IP domainname
```


ARP

show arp cache

arp -a