List of Linux networking tools

netstat (ss)

Displays contents of /proc/net files. It works with the Linux Network Subsystem, it will tell you what the status of ports are ie. open, closed, waiting, masquerade connections. It will also display various other things. It has many different options. Netstat (Network Statistic) command display connection info, routing table information etc. To displays routing table information use option as -r.

Sample output:

```
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp4       0      0  127.0.0.1.62132        127.0.0.1.http         ESTABLISHED
tcp4       0      0  127.0.0.1.http         *                      LISTEN
```

tcpdump

This is a sniffer, a program that captures packets off a network interface and interprets them for you. It understands all basic internet protocols, and can be used to save entire packets for later inspection.

ping

The ping command (named after the sound of an active sonar system) sends echo requests to the host you specify on the command line, and lists the responses received their round trip time. PING (Packet INternet Groper) command is the best way to test connectivity between two nodes. Whether it is Local Area Network (LAN) or Wide Area Network (WAN). Ping use ICMP (Internet Control Message Protocol) to communicate to other devices. You can ping host name of ip address using below command.

```
$ ping google.com
PING google.com (216.58.198.78): 56 data bytes
64 bytes from 216.58.198.78: icmp_seq=0 ttl=46 time=6.108 ms
64 bytes from 216.58.198.78: icmp_seq=1 ttl=46 time=6.222 ms
^C
--- google.com ping statistics ---
2 packets transmitted, 2 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 6.108/6.165/6.222/0.057 ms
```

traceroute

traceroute will show the route of a packet. It attempts to list the series of hosts through which your packets travel on their way to a given destination. Also have a look at xtraceroute (one of several graphical equivalents of this program). traceroute

is a network troubleshooting utility which shows number of hops taken to reach destination also determine packets traveling path. Below we are tracing route to global DNS server IP Address and able to reach destination also shows path of that packet is traveling.

```
$ traceroute -I google.com
traceroute to google.com (216.58.198.78), 128 hops max, 72 byte packets
1  52.93.7.1 (52.93.7.1)  6.361 ms  6.229 ms  6.106 ms
2  72.14.215.85 (72.14.215.85)  5.939 ms  5.460 ms  5.914 ms
3  209.85.252.198 (209.85.252.198)  6.012 ms  5.694 ms  5.761 ms
4  64.233.174.27 (64.233.174.27)  5.079 ms  4.776 ms  4.662 ms
5  dub08s02-in-f78.1e100.net (216.58.198.78)  6.650 ms  5.509 ms  5.596 ms
```

tracepath

tracepath performs a very simlar function to traceroute the main difference is that tracepath doesn't take complicated options.

nmap

“ network exploration tool and security scanner”. nmap is a very advanced network tool used to query machines (local or remote) as to whether they are up and what ports are open on these machines.

dig

Dig (domain information groper) query DNS related information like A Record, CNAME, MX Record etc. This command mainly use to troubleshoot DNS related query.

```
$ dig amazon.com

; <<>> DiG 9.8.3-P1 <<>> amazon.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30832
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;amazon.com.                    IN      A

;; ANSWER SECTION:
amazon.com.             17      IN      A       54.239.25.200
amazon.com.             17      IN      A       54.239.25.208
amazon.com.             17      IN      A       54.239.26.128
amazon.com.             17      IN      A       54.239.17.6
amazon.com.             17      IN      A       54.239.17.7
amazon.com.             17      IN      A       54.239.25.192

;; Query time: 3 msec
;; SERVER: 10.78.97.142#53(10.78.97.142)
```

```
;; WHEN: Thu Dec 29 13:56:22 2016
;; MSG SIZE  rcvd: 124
```

nslookup

nslookup command also use to find out DNS related query. The following examples shows A Record (IP Address) of tecmint.com.

```
nslookup amazon.com
Server:         10.78.97.142
Address: 10.78.97.142#53

Non-authoritative answer:
Name:   amazon.com
Address: 54.239.25.192
Name:   amazon.com
Address: 54.239.25.200
Name:   amazon.com
Address: 54.239.25.208
Name:   amazon.com
Address: 54.239.26.128
Name:   amazon.com
Address: 54.239.17.6
Name:   amazon.com
Address: 54.239.17.7
```

host

host command to find name to IP or IP to name in IPv4 or IPv6 and also query DNS records.

```
$ host amazon.com
amazon.com has address 54.239.17.6
amazon.com has address 54.239.17.7
amazon.com has address 54.239.25.192
amazon.com has address 54.239.25.200
amazon.com has address 54.239.25.208
amazon.com has address 54.239.26.128
amazon.com mail is handled by 10 amazon-smtp.amazon.com.
```

hostname

hostname is to identify in a network. Execute hostname command to see the hostname of your box. You can set hostname permanently in /etc/sysconfig/network. Need to reboot box once set a proper hostname.

```
$ hostname -f
miglen.development.box
```

arp

ARP (Address Resolution Protocol) is useful to view / add the contents of the kernel's ARP tables. To see default table use the command as.

TLDR

```
Show and manipulate your system's ARP cache.

- Show current arp table:
    arp -a

- Clear the entire cache:
    sudo arp -a -d

- Delete a specific entry:
    arp -d address

- Create an entry:
    arp -s address mac_address
```

Configuration

ifconfig

This command is used to configure network interfaces, or to display their current configuration. In addition to activating and deactivating interfaces with the "up" and "down" settings, this command is necessary for setting an interface's address information if you don't have the ifcfg script.

TLDR

```
- View network settings of an ethernet adapter:
    ifconfig eth0

- Display details of all interfaces, including disabled interfaces:
    ifconfig -a

- Disable eth0 interface:
    ifconfig eth0 down

- Enable eth0 interface:
    ifconfig eth0 up

- Assign IP address to eth0 interface:
    ifconfig eth0 ip_address
```

- ifup - Use ifup device-name to bring an interface up by following a script (which will contain your default networking settings). Simply type ifup and you will get help on using the script.

- ifdown - Use ifdown device-name to bring an interface down using a script (which will contain your default network settings). Simply type ifdown and you will get help on using the script.
- ifcfg - Use ifcfg to configure a particular interface. Simply type ifcfg to get help on using this script.

route

The route command is the tool used to display or modify the routing table. To add a gateway as the default you would type:

```
$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         172.31.16.1     0.0.0.0         UG    0      0        0 eth0
172.17.0.0      0.0.0.0         255.255.0.0     U     0      0        0 docker0
172.31.16.0     0.0.0.0         255.255.240.0   U     0      0        0 eth0
```

TLDR

```
Manually manipulate the routing tables.
Necessitates to be root.

- Add a route to a destination through a gateway:
    sudo route add dest_ip_address gateway_address

- Add a route to a /24 subnet through a gateway:
    sudo route add subnet_ip_address/24 gateway_address

- Run in test mode (does not do anything, just print):
    sudo route -t add dest_ip_address/24 gateway_address

- Remove all routes:
    sudo route flush

- Delete a specific route:
    sudo route delete dest_ip_address/24

- Lookup and display the route for a destination (hostname or IP address):
    sudo route get destination
```

iwconfig

iwconfig command in Linux is use to configure a wireless network interface. You can see and set the basic Wi-Fi details like SSID channel and encryption. You can refer man page of iwconfig to know more.

ip

Show / manipulate routing, devices, policy routing and tunnels.

- List interfaces with detailed info:

```
ip address
```

- List interfaces with brief network layer info:

```
ip -brief address
```

- List interfaces with brief link layer info:

```
ip -brief link
```

- Display the routing table:

```
ip route
```

- Show neighbors (ARP table):

```
ip neighbour
```

- Make an interface up/down:

```
ip link set {{interface}} up/down
```

- Add/Delete an ip address to an interface:

```
ip addr add/del {{ip}}/{{mask}} dev {{interface}}
```

- Add a default route:

```
ip route add default via {{ip}} dev {{interface}}
```