

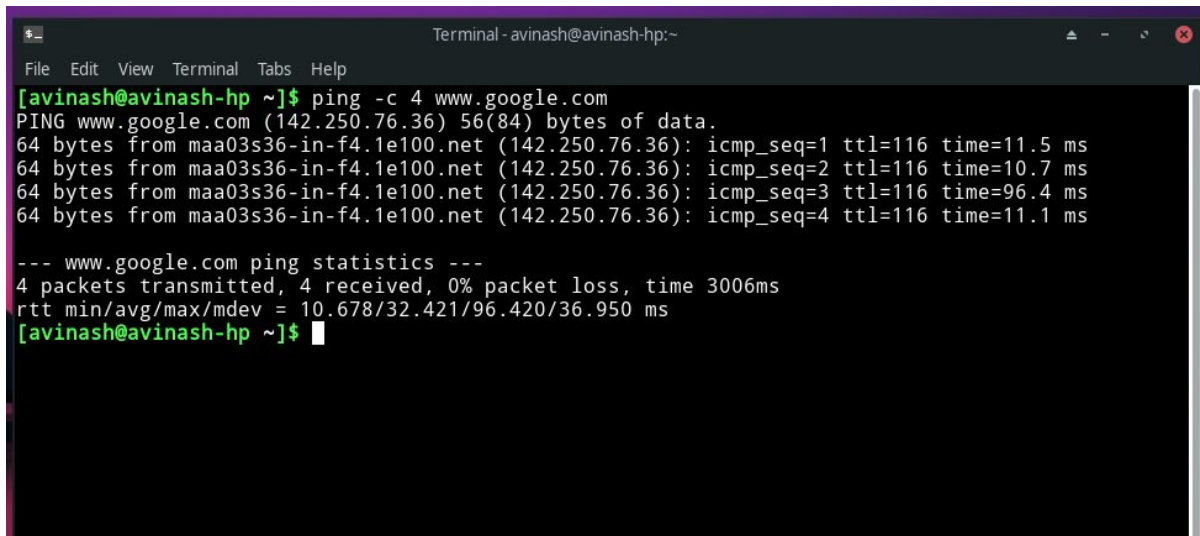
NETWORKS LAB

Experiment-1

Samudrala Aviansh
B180409CS

1. Ping :

Syntax : “ ***ping [OPTIONS] DESTINATION*** ”

A screenshot of a Linux terminal window titled "Terminal - avinash@avinash-hp:~". The terminal shows the execution of the command "ping -c 4 www.google.com". The output displays four successful ping responses from the IP address 142.250.76.36, each with a TTL of 116 and varying response times (11.5 ms, 10.7 ms, 96.4 ms, and 11.1 ms). Below the individual responses, it shows the ping statistics: 4 packets transmitted, 4 received, 0% packet loss, and a total time of 3006ms. The round-trip time (rtt) statistics are also listed: min/avg/max/mdev = 10.678/32.421/96.420/36.950 ms. The prompt "[avinash@avinash-hp ~]\$" is visible at the bottom of the terminal.

```
Terminal - avinash@avinash-hp:~
File Edit View Terminal Tabs Help
[avinash@avinash-hp ~]$ ping -c 4 www.google.com
PING www.google.com (142.250.76.36) 56(84) bytes of data:
64 bytes from maa03s36-in-f4.1e100.net (142.250.76.36): icmp_seq=1 ttl=116 time=11.5 ms
64 bytes from maa03s36-in-f4.1e100.net (142.250.76.36): icmp_seq=2 ttl=116 time=10.7 ms
64 bytes from maa03s36-in-f4.1e100.net (142.250.76.36): icmp_seq=3 ttl=116 time=96.4 ms
64 bytes from maa03s36-in-f4.1e100.net (142.250.76.36): icmp_seq=4 ttl=116 time=11.1 ms

--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 10.678/32.421/96.420/36.950 ms
[avinash@avinash-hp ~]$
```

The Linux ping command is a simple utility used to check whether a network is available and if a host is reachable. With this command, you can test if a server is up and running. It also helps with troubleshooting various connectivity issues.

The ping command resolves the domain name into an IP address and starts sending ICMP packages to the destination IP. If the destination IP is reachable it will respond back and the ping command prints a line that includes the following fields:

- The number of data bytes.
- The IP address of the destination.
- The ICMP sequence number for each packet.
- The Time to Live.
- The ping time, measured in milliseconds which is the round trip time for the packet to reach the host, and the response to return to the sender.

Here i used “ping -c 4 www.google.com” to set the count of packets to 4.

Once the command stops, it displays a statistic, including the percentage of packet loss. The packet loss means the data was dropped somewhere in the network, indicating an issue within the network. If there is a packet loss, you can use the traceroute command to identify where the packet loss occurs.

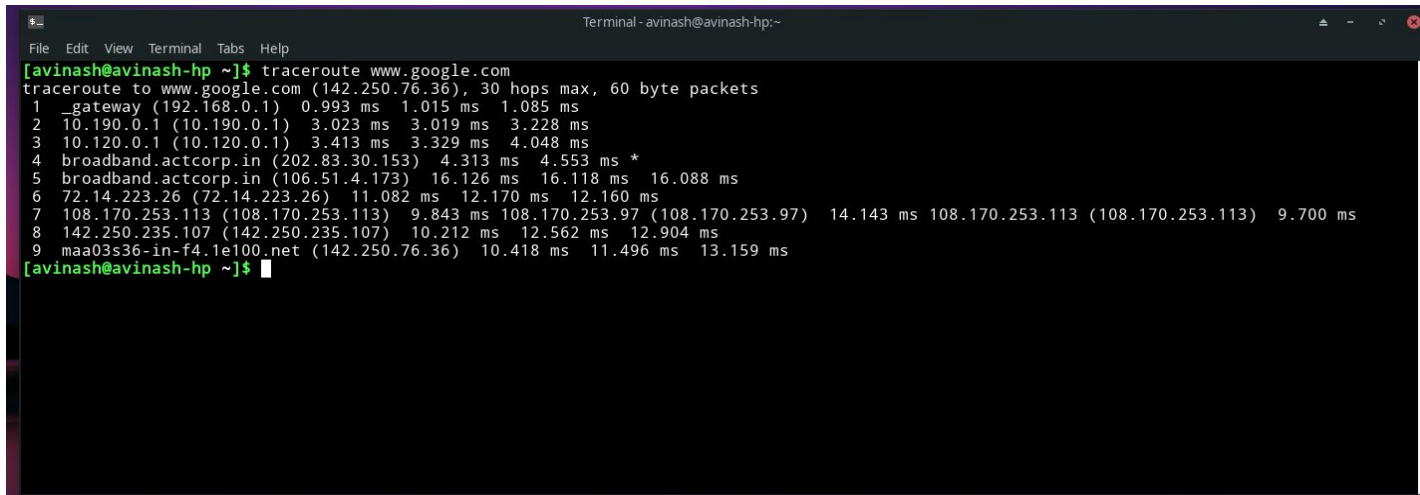
Options :

-c : to specify the count of packets to be sent to destination

-s : to specify the size of the packet that is to be sent.

-4 or -6 : to specify whether to use ipv4 or ipv6.

2. tracert / traceroute :



```
Terminal - avinash@avinash-hp:~  
[avinash@avinash-hp ~]$ traceroute www.google.com  
traceroute to www.google.com (142.250.76.36), 30 hops max, 60 byte packets  
1 _gateway (192.168.0.1) 0.993 ms 1.015 ms 1.085 ms  
2 10.190.0.1 (10.190.0.1) 3.023 ms 3.019 ms 3.228 ms  
3 10.120.0.1 (10.120.0.1) 3.413 ms 3.329 ms 4.048 ms  
4 broadband.actcorp.in (202.83.30.153) 4.313 ms 4.553 ms *  
5 broadband.actcorp.in (106.51.4.173) 16.126 ms 16.118 ms 16.088 ms  
6 72.14.223.26 (72.14.223.26) 11.082 ms 12.170 ms 12.160 ms  
7 108.170.253.113 (108.170.253.113) 9.843 ms 108.170.253.97 (108.170.253.97) 14.143 ms 108.170.253.113 (108.170.253.113) 9.700 ms  
8 142.250.235.107 (142.250.235.107) 10.212 ms 12.562 ms 12.904 ms  
9 maa03s36-in-f4.1e100.net (142.250.76.36) 10.418 ms 11.496 ms 13.159 ms  
[avinash@avinash-hp ~]$
```

Syntax : “ ***traceroute [options] host_Address*** “

traceroute command in Linux prints the route that a packet takes to reach the host. This command is useful when you want to know about the route and about all the hops that a packet takes. Above image depicts how traceroute command is used to reach the Google (142.250.76.36) host from the local machine and it also prints detail about all the hops that it visits in between.

The first column corresponds to the hop count. The second column represents the address of that hop and after that, you see three space-separated time in milliseconds. *traceroute* command sends three packets to the hop and each of the time refers to the time taken by the packet to reach the hop. In our case maximum no.of hops to reach google host is 9.

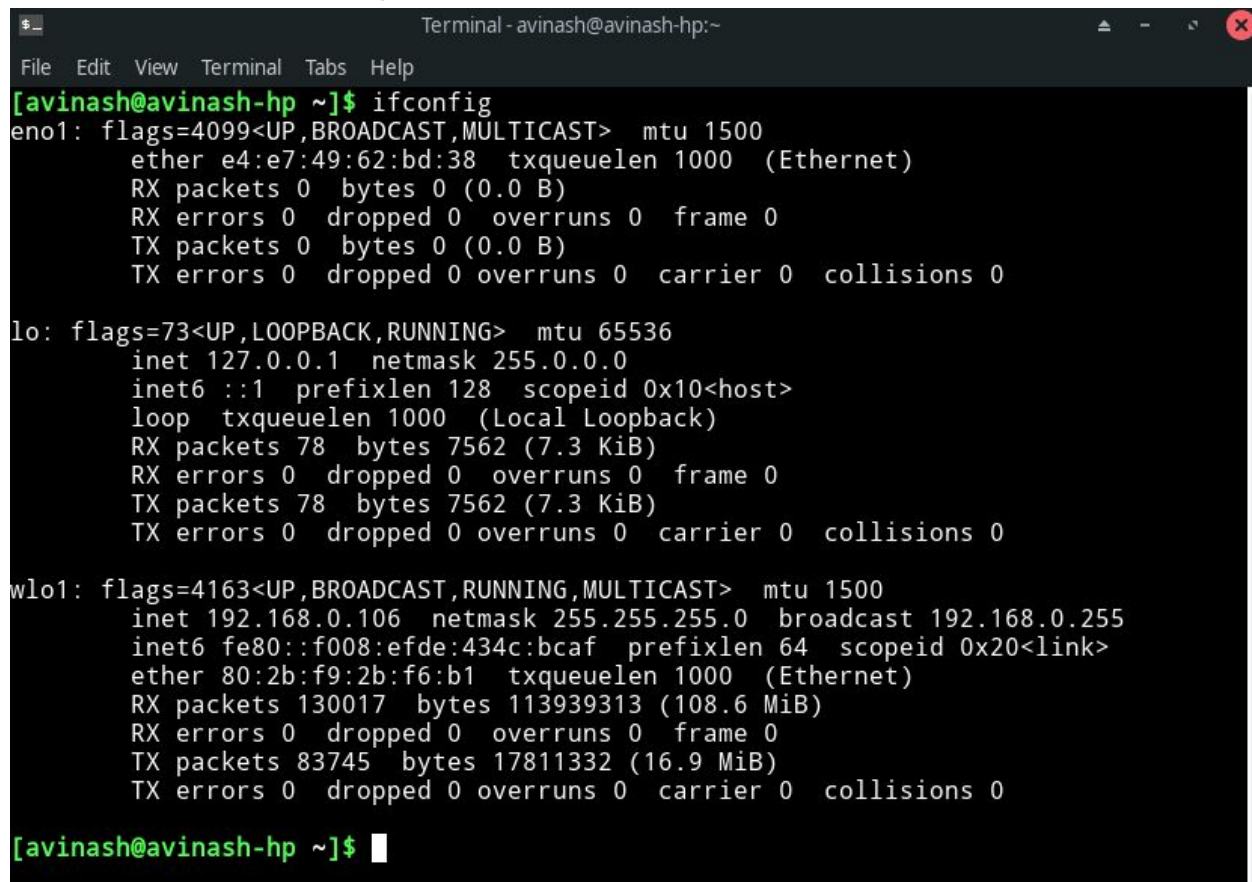
Options:

- p : used to specify the destination port
- F : do not fragment packet.
- g : Route the packet through gate.
- n : Do not resolve IP addresses to their domain names.

3. Ip / ifconfig / ipconfig :

Syntax : “ **ifconfig** [...**OPTIONS**] [**INTERFACE**] “

This commands is used to configure the network interfaces. Also, this command is used to assign the IP address and netmask to an interface or to enable or disable a given interface.



```
Terminal - avinash@avinash-hp:~
File Edit View Terminal Tabs Help
[avinash@avinash-hp ~]$ ifconfig
eno1: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether e4:e7:49:62:bd:38 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

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 78 bytes 7562 (7.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 78 bytes 7562 (7.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlo1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.106 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::f008:efde:434c:bcaf prefixlen 64 scopeid 0x20<link>
    ether 80:2b:f9:2b:f6:b1 txqueuelen 1000 (Ethernet)
    RX packets 130017 bytes 113939313 (108.6 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 83745 bytes 17811332 (16.9 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[avinash@avinash-hp ~]$
```

Here, **eno1**, **lo** and **wlo1** are the names of the active network interfaces on the system.

- **eno1** is the first Ethernet interface.
- **lo** is the loopback interface. This is a special network interface that the system uses to communicate with itself.
- **wlo1** is the name of the first wireless network interface on the system.

Options :

up : to activate driver for interface , **down** : to deactivate driver from interface

Here **inet** indicates the machine's ip address(ipv4) and **inet6** indicates the machine's ipv6 address.

RX and TX are the number of received and transmitted packets.And the state <RUNNING> specifies that the interface is ready to accept the data.

In my case, my machine's ipv4 address is '192.168.0.106' and ipv6 address is 'fe80::f008:efde:434c:bcaf'

4. dig / nslookup / host :

Syntax : ***"host [OPTIONS] hostname [server]."***

"dig [SERVER] [NAME] [TYPE]"

"nslookup [OPTIONS] hostname [SERVER]"

```
terminal - avinash@avinash-hp:~
File Edit View Terminal Tabs Help
[avinash@avinash-hp ~]$ dig www.google.com

; <<>> DiG 9.16.10 <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 17485
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.google.com.                IN      A

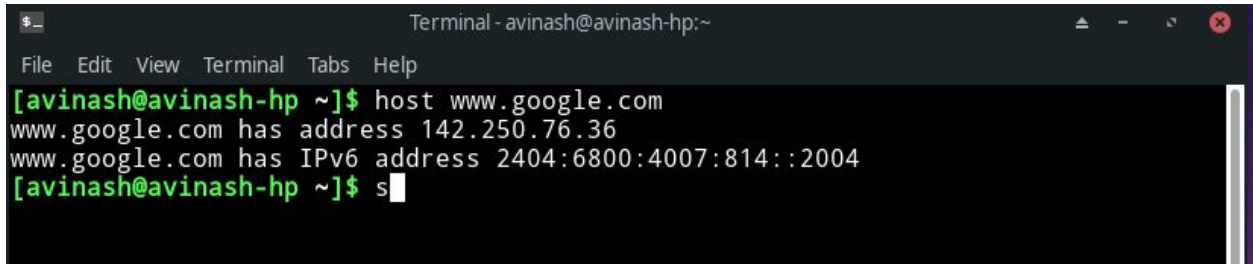
;; ANSWER SECTION:
www.google.com.                111     IN      A      142.250.76.36

;; Query time: 3 msec
;; SERVER: 202.83.30.162#53(202.83.30.162)
;; WHEN: Sun Jan 24 16:13:55 IST 2021
;; MSG SIZE rcvd: 59

[avinash@avinash-hp ~]$ nslookup www.google.com
Server:                202.83.30.162
Address:                202.83.30.162#53

Non-authoritative answer:
Name:   www.google.com
Address: 142.250.76.36
Name:   www.google.com
Address: 2404:6800:4007:814::2004

[avinash@avinash-hp ~]$ s
```

A terminal window titled "Terminal - avinash@avinash-hp:~" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The prompt is [avinash@avinash-hp ~]\$. The command "host www.google.com" is entered. The output shows "www.google.com has address 142.250.76.36" and "www.google.com has IPv6 address 2404:6800:4007:814::2004". The prompt is then [avinash@avinash-hp ~]\$ s.

```
Terminal - avinash@avinash-hp:~
File Edit View Terminal Tabs Help
[avinash@avinash-hp ~]$ host www.google.com
www.google.com has address 142.250.76.36
www.google.com has IPv6 address 2404:6800:4007:814::2004
[avinash@avinash-hp ~]$ s
```

These commands are basically used for DNS (Domain Name System) lookup operations. In simple words, these commands are used to find the IP address of a particular domain name or if you want to find out the domain name of a particular IP address these commands become handy. You can also find more specific details of a domain by specifying the corresponding option along with the domain name.

Options for host :

- a or -v:** It is used to enable the verbose output.
- t :** It is used to specify the type of query(ns, SOA, txt).
- R :** In order to specify the number of retries you can do in case one try fails. If any one succeeds then the command stops.

5. Whois :

Syntax : ***whois [-h HOST] [-p PORT] [-aCFHILMmrRSVx] [-g SOURCE:FIRST-LAST] [-i ATTR] [-S SOURCE] [-T TYPE] object***

whois searches for an object in a WHOIS database. WHOIS is a query and response protocol that is widely used for querying databases that store the registered users of an Internet resource, such as a domain name or an IP address block, but is also used for a wider range of other information.

As you can see the ipv4 returned by **host** command for "www.google.com" is "142.250.76.36". So in the image below I checked whether the ipv4 "142.250.73.36" belongs to Google or not using **whois** command.

```
Terminal - avinash@avinash-hp:~
File Edit View Terminal Tabs Help
[avinash@avinash-hp ~]$ whois 142.250.76.36
#
# ARIN WHOIS data and services are subject to the Terms of Use
# available at: https://www.arin.net/resources/registry/whois/tou/
#
# If you see inaccuracies in the results, please report at
# https://www.arin.net/resources/registry/whois/inaccuracy_reporting/
#
# Copyright 1997-2021, American Registry for Internet Numbers, Ltd.
#

NetRange:      142.250.0.0 - 142.251.255.255
CIDR:          142.250.0.0/15
NetName:       GOOGLE
NetHandle:     NET-142-250-0-0-1
Parent:        NET142 (NET-142-0-0-0-0)
NetType:       Direct Allocation
OriginAS:      AS15169
Organization:  Google LLC (GOGL)
RegDate:       2012-05-24
Updated:       2012-05-24
Ref:           https://rdap.arin.net/registry/ip/142.250.0.0

OrgName:       Google LLC
OrgId:         GOGL
Address:        1600 Amphitheatre Parkway
City:          Mountain View
StateProv:     CA
PostalCode:    94043
Country:       US
RegDate:       2000-03-30
Updated:       2019-10-31
Comment:       Please note that the recommended way to file abuse complaints are located in the following links.
```

6. route :

route command in Linux is used when you want to work with the IP/kernel routing table. It is mainly used to set up static routes to specific hosts or networks via an interface. It is used for showing or update the IP/kernel routing table. Syntax : “**route [OPTIONS]**”

```
Terminal - avinash@avinash-hp:~
File Edit View Terminal Tabs Help
[avinash@avinash-hp ~]$ route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
default _gateway 0.0.0.0 UG 600 0 0 wlo1
192.168.0.0 0.0.0.0 255.255.255.0 U 600 0 0 wlo1
[avinash@avinash-hp ~]$
```

We can get details of kernel/IP routing table using the command “**ip route**”

7. tcpdump :

tcpdump is a packet sniffing and packet analyzing tool to troubleshoot connectivity issues in Linux. It is used to capture, filter, and analyze network traffic such as TCP/IP packets going through your system. It is many times used as a security tool as well.

Syntax : “**tcpdump [OPTIONS] [EXPRESSION]**”

Output Format :

[Timestamp] [Protocol] [Src IP].[Src Port] > [Dst IP].[Dst Port]: [Flags], [Seq], [Ack], [Win Size], [Options], [Data Length]

Example:

15:47:24.248737 IP 192.168.1.185.22 > 192.168.1.150.37445: Flags [P.], seq 201747193:201747301, ack 1226568763, win 402, options [nop,nop,TS val 1051794587 ecr 2679218230], length 108

- 15:47:24.248737 - The timestamp of the captured packet (format: hours:minutes:seconds.frac)
- IP - The packet protocol. In this case, IP means the Internet protocol version 4 (IPv4).
- 192.168.1.185.22 - The source IP address and port, separated by a dot (.).
- 192.168.1.150.37445 - The destination IP address and port, separated by a dot (.).
- Flags [P.] - TCP Flags field. In this example, [P.] means Push Acknowledgment packet, which is used to acknowledge the previous packet and send data.
- seq 201747193:201747301 - The sequence number is in the first:last notation. It shows the number of data contained in the packet.
- ack 1226568763 The acknowledgment number is the sequence number of the next data expected by the other end of this connection.

- win 402 - The window number is the number of available bytes in the receiving buffer.
- options [nop,nop,TS val 1051794587 ecr 2679218230] - TCP options.
- length 108 - The length of payload data.

```

Terminal - avinash@avinash-hp:~
File Edit View Terminal Tabs Help
[avinash@avinash-hp ~]$ sudo tcpdump
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on wlo1, link-type EN10MB (Ethernet), snapshot length 262144 bytes
17:28:35.800064 IP 74.118.186.210.https > avinash-hp.local.58010: Flags [.], ack 628690612, win
  14970, options [nop,nop,TS val 876711061 ecr 3920052449], length 0
17:28:35.811998 IP avinash-hp.local.35258 > broadband.actcorp.in.domain: 10927+ PTR? 106.0.168.
  192.in-addr.arpa. (44)
17:28:35.814241 IP broadband.actcorp.in.domain > avinash-hp.local.35258: 10927 NXDomain 0/1/0 (
  121)
17:28:35.815410 IP avinash-hp.local.48196 > broadband.actcorp.in.domain: 22425+ PTR? 210.186.11
  8.74.in-addr.arpa. (45)
17:28:35.823189 IP avinash-hp.local.35222 > a23-9-20-102.deploy.static.akamaitechnologies.com.h
  ttps: Flags [.], ack 1750942760, win 501, options [nop,nop,TS val 524452153 ecr 3855038186], le
  ngth 0
17:28:35.823211 IP avinash-hp.local.51390 > a104-120-154-233.deploy.static.akamaitechnologies.c
  om.https: Flags [.], ack 1773604924, win 501, options [nop,nop,TS val 1436820195 ecr 2461705517
  ], length 0
17:28:35.833865 IP a23-9-20-102.deploy.static.akamaitechnologies.com.https > avinash-hp.local.3
  5222: Flags [.], ack 1, win 501, options [nop,nop,TS val 3855048326 ecr 524390999], length 0
17:28:35.837601 IP a104-120-154-233.deploy.static.akamaitechnologies.com.https > avinash-hp.loc
  al.51390: Flags [.], ack 1, win 501, options [nop,nop,TS val 2461715656 ecr 1436756900], length
  0
17:28:36.249853 IP avinash-hp.local.43252 > 69.173.159.33.https: Flags [.], ack 2501187464, win
  501, options [nop,nop,TS val 2173938592 ecr 944810222], length 0
17:28:36.979696 IP avinash-hp.local.58752 > 596.bm-nginx-loadbalancer.mgmt.sin3.adnexus.net.htt
  ps: Flags [.], ack 2592515217, win 501, options [nop,nop,TS val 3589356551 ecr 2400261744], len
  gth 0
17:28:37.069291 IP broadband.actcorp.in.domain > avinash-hp.local.48196: 22425 ServFail 0/0/0 (
  45)
17:28:37.069540 IP avinash-hp.local.60221 > broadband.actcorp.in.domain: 22425+ PTR? 210.186.11
  8.74.in-addr.arpa. (45)
17:28:37.075756 IP 596.bm-nginx-loadbalancer.mgmt.sin3.adnexus.net.https > avinash-hp.local.587
  52: Flags [P.], seq 1:32, ack 1, win 70, options [nop,nop,TS val 2400264244 ecr 3589346549], le
  ngth 31
17:28:37.075804 IP avinash-hp.local.58752 > 596.bm-nginx-loadbalancer.mgmt.sin3.adnexus.net.htt
  ps: Flags [.], ack 32, win 501, options [nop,nop,TS val 3589356647 ecr 2400264244], length 0
17:28:37.076350 IP avinash-hp.local.58752 > 596.bm-nginx-loadbalancer.mgmt.sin3.adnexus.net.htt
  ps: Flags [P.], seq 1:32, ack 32, win 501, options [nop,nop,TS val 3589356648 ecr 2400264244],
  length 31
17:28:37.076387 IP avinash-hp.local.58752 > 596.bm-nginx-loadbalancer.mgmt.sin3.adnexus.net.htt
  ps: Flags [F.], seq 32, ack 32, win 501, options [nop,nop,TS val 3589356648 ecr 2400264244], le
  ngth 0
^C17:28:37.229835 IP avinash-hp.local.40708 > 216.52.2.39.https: Flags [.], ack 673924514, win
  501, options [nop,nop,TS val 1863159481 ecr 3255678717], length 0

17 packets captured
162 packets received by filter
129 packets dropped by kernel
[avinash@avinash-hp ~]$

```

8. netstat / ss :

Syntax : “**netstat** [OPTIONS]”, “**ss** [OPTIONS]”

netstat command displays various network related information such as network connections, routing tables, interface statistics, masquerade connections, multicast memberships etc.,

The **ss** command is a tool used to dump socket statistics and displays information in similar fashion (although simpler and faster) to netstat. The ss command can also display even more TCP and state information than most other tools.

```
Terminal - avinash@avinash-hp:~
File Edit View Terminal Tabs Help
[avinash@avinash-hp ~]$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 avinash-hp.local:42122  104.17.131.171:https    ESTABLISHED
tcp      0      0 avinash-hp.local:42984  maa03s36-in-f4.1e:https ESTABLISHED
tcp      0      0 avinash-hp.local:48182  117.18.237.29:www-http  TIME_WAIT
tcp      0      0 avinash-hp.local:42136  69.173.159.50:https     TIME_WAIT
tcp      0      0 avinash-hp.local:58156  whatsapp-cdn-shv-:https ESTABLISHED
tcp      0      0 avinash-hp.local:42144  69.173.159.50:https     TIME_WAIT
tcp      0      0 avinash-hp.local:44932  maa05s13-in-f10.1:https ESTABLISHED
tcp      0      0 avinash-hp.local:44590  ec2-52-41-198-156:https ESTABLISHED
tcp      0      0 avinash-hp.local:60126  maa03s28-in-f2.1e:https ESTABLISHED
tcp      0      0 avinash-hp.local:44480  maa03s28-in-f8.1e:https ESTABLISHED
tcp      0      0 avinash-hp.local:38488  104.17.70.176:https     ESTABLISHED
tcp      0      0 avinash-hp.local:57416  104.19.154.83:https     ESTABLISHED
tcp      0      0 avinash-hp.local:59564  maa03s28-in-f2.1e:https ESTABLISHED
tcp      0      0 avinash-hp.local:44180  maa03s28-in-f3.1e:https ESTABLISHED
tcp      0      0 avinash-hp.local:34834  108-174-11-69.fwd:https ESTABLISHED
tcp      0      0 avinash-hp.local:52452  218.64.98.34.bc.g:https ESTABLISHED
tcp      0      0 avinash-hp.local:54832  maa05s14-in-f14.1:https ESTABLISHED
tcp      0      0 avinash-hp.local:60074  maa03s28-in-f2.1e:https ESTABLISHED
```

```
Terminal - avinash@avinash-hp:~
File Edit View Terminal Tabs Help
[avinash@avinash-hp ~]$ ss -t
State Recv-Q Send-Q Local Address:Port      Peer Address:Port      Process
ESTAB  0      0      192.168.0.106:51930     172.217.163.206:https
ESTAB  0      0      192.168.0.106:33984     34.217.188.66:https
ESTAB  0      0      192.168.0.106:59360     34.98.64.218:https
ESTAB  0      0      192.168.0.106:58014     172.217.26.202:https
ESTAB  0      0      192.168.0.106:37300     142.250.76.34:https
ESTAB  0      0      192.168.0.106:37302     142.250.76.34:https
ESTAB  0      0      192.168.0.106:38250     172.217.31.194:https
ESTAB  0      0      192.168.0.106:36810     142.250.76.46:https
[avinash@avinash-hp ~]$
```

netstat -at : To list all tcp ports.

netstat -au : To list all udp ports.

netstat -l : To list only the listening ports.

Options for ss :

-t: display all TCP connections.

-u: display all UDP connections.

-x: display all UNIX connections.

9. **dstat** :

dstat is a tool that is used to retrieve information or statistics form components of the system such as network connections, IO devices, or CPU, etc.

Syntax: "**dstat [OPTIONS]**"

```
Terminal - avinash@avinash-hp:~
File Edit View Terminal Tabs Help
[avinash@avinash-hp ~]$ dstat
You did not select any stats, using -cdngy by default.
--total-cpu-usage-- -dsk/total- -net/total- ---paging-- ---system--
usr sys idl wai stl read writ recv send in out int csw
 4  1  95  1  0   74k 129k    0    0    0    0  990 2783
 0  0 100  0  0    0    0 2216B 1030B  0    0  184  793
 0  0 100  0  0    0   612k 206B 170B  0    0  131  429
 0  0 100  0  0    0    0    0    0  0    0  167  524
 1  0  97  2  0    0   64k 2429B 18k    0    0  217  964
 0  0 100  0  0    0    0  911B 227B  0    0  187  615
 0  0 100  0  0    0  624k    0    0  0    0  138  415
 1  0  99  0  0    0    0    0    0  0    0  617  698
 0  0 100  0  0    0    0  86B 106B  0    0  375  430
 0  0 100  0  0    0    0  42B  70B  0    0  150  501
 0  0  98  1  0    0  20k 119B  94B  0    0  196  489
 0  0  99  0  0    0    0    0    0  0    0  682  653
 0  0  99  0  0    0   12k  78B    0  0    0  299  476
 0  0 100  0  0    0    0  78B 114B  0    0  217  580
 0  0 100  0  0    0    0    0    0  0    0  129  503
 1  0  99  0  0    0    0    0    0  0    0  589  703
 0  0 100  0  0    0    0    0    0  0    0  573  424
 0  0 100  0  0    0    0    0    0  0    0  132  413
 1  0  99  0  0    0    0 4404B 2664B  0    0  181  731
 1  0  99  0  0    0    0    0    0  0    0  428  706
```

10. ifstat :

The ifstat command prints network interface statistics. The interface keeps records of the previous data displayed in history files.

Syntax : “***ifstat [OPTIONS][INTERFACE]***”

```
Terminal - avinash@avinash-hp:~
File Edit View Terminal Tabs Help
[avinash@avinash-hp ~]$ ifstat
#kernel
Interface      RX Pkts/Rate    TX Pkts/Rate    RX Data/Rate    TX Data/Rate
                RX Errs/Drop   TX Errs/Drop   RX Over/Rate    TX Coll/Rate
lo              6 0            6 0             532 0           532 0
                0 0            0 0             0 0             0 0
eno1            0 0            0 0             0 0             0 0
                0 0            0 0             0 0             0 0
wlo1           35548 0        35360 0         12519K 0        10487K 0
                0 0            0 0             0 0             0 0
[avinash@avinash-hp ~]$ s
```

There are five columns of data per interface, with each interface having two rows. The first column is the interface name. The remaining columns contain two data fields each.

You see packets *received* and rate in the RX column, as well as any errors or drops on those packets. In the TX column, we have packets *transmitted* and rate, as well as errors and drops. After the packet columns, we change the unit of measure to data sent and received. This value is measured in kilobytes. We have data received and rate, as well as data transferred and rate.

11. wget :

Syntax : “ **wget** [*OPTIONS*] *url* ”

Wget is the non-interactive network downloader which is used to download files from the server even when the user has not logged on to the system and it can work in the background without hindering the current process.

- wget is non-interactive, meaning that it can work in the background, while the user is not logged on. This allows you to start a retrieval and disconnect from the system, letting wget finish the work. By contrast, most of the Web browsers require constant user's presence, which can be a great hindrance when transferring a lot of data.
- wget can follow links in HTML and XHTML pages and create local versions of remote web sites, fully recreating the directory structure of the original site. This is sometimes referred to as recursive downloading. While doing that, wget respects the Robot Exclusion Standard (/robots.txt). wget can be instructed to convert the links in downloaded HTML files to the local files for offline viewing.
- wget has been designed for robustness over slow or unstable network connections; if a download fails due to a network problem, it will keep retrying until the whole file has been retrieved. If the server supports resuming, it will instruct the server to continue the download from where it left off.

Options :

-q : turn off output in terminal while downloading.

-b : to download file in background.

-c : To resume a partially downloaded file.

```
Terminal - avinash@avinash-hp:~
File Edit View Terminal Tabs Help
[avinash@avinash-hp ~]$ wget www.google.com
--2021-01-24 20:04:47-- http://www.google.com/
Resolving www.google.com (www.google.com)... 142.250.76.36, 2404:6800:4007:814::
2004
Connecting to www.google.com (www.google.com)|142.250.76.36|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'index.html'

index.html          [ <=>          ] 14.84K  ---KB/s   in 0.01s

2021-01-24 20:04:47 (1.44 MB/s) - 'index.html' saved [15192]

[avinash@avinash-hp ~]$
```