Computer Network

Name : Antuley Aman Siraj.

Roll No. : 23CO25.

Batch : 01

Experiment - 02

Aim : Use of basic Networking commands in Linux (ping , traceroute , nslookup , netstart , ARP . RARP , ip , ipconfig , dig , route).

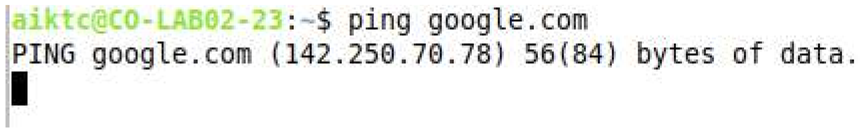
Theory : Every computer is connected to some other computer through a network whether internally or externally to exchange some information. This network can be small as some computers connected in your home or office , or can be large or complicated as in a large university or the entire internet.

1) **PING :** It is used to check the availability of a host or an IP network .

   A basic ping command looks like ping <destination>, where <destination> can be an IP address or a domain name.

   Syntax : ping <destination>

   Example :



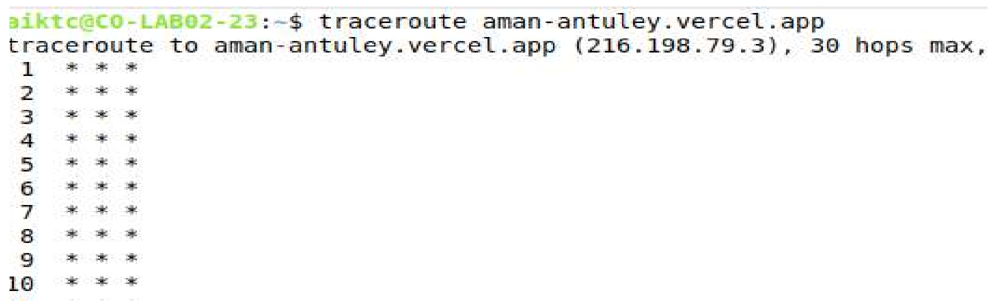
```
aiktc@CO-LAB02-23:~$ ping google.com
PING google.com (142.250.70.78) 56(84) bytes of data.
█
```

2) **TRACEROUTE** : It is a network diagnostic tool used to trace the path that a packet takes from your computer to a specific destination, such as a website or another computer.

   The core command is traceroute <destination> (Linux/macOS) or tracert <destination> (Windows), where <destination> is the target IP address or domain name.

   Syntax : **traceroute <destination>**

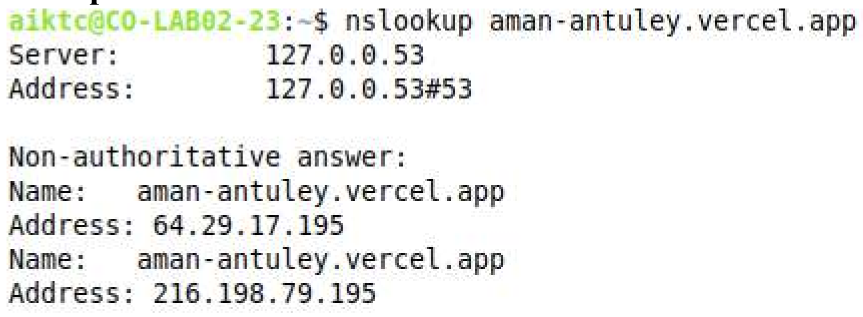
   Example :


```
aiktc@CO-LAB02-23:~$ traceroute aman-antuley.vercel.app
traceroute to aman-antuley.vercel.app (216.198.79.3), 30 hops max,
 1  * * *
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
```

3) **NSLOOKUP** : This command is used to query Domain Name System (DNS) servers to find information about domain names and IP addresses. It can be used in both interactive and non-interactive modes. Common uses include finding the IP address associated with a domain name, the name servers for a domain, or other DNS record types

   Syntax : **nslookup <destination>**

   Example :

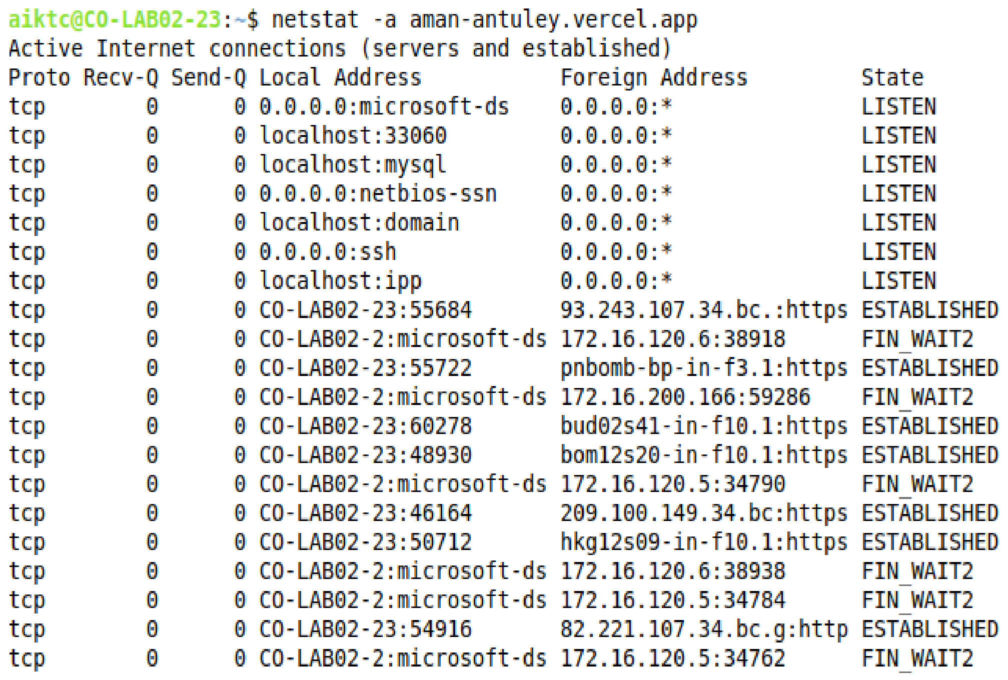

```
aiktc@CO-LAB02-23:~$ nslookup aman-antuley.vercel.app
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   aman-antuley.vercel.app
Address: 64.29.17.195
Name:   aman-antuley.vercel.app
Address: 216.198.79.195
```

4) **NETSTAT :** This command displays network connections, routing tables, interface statistics, and more. It's a versatile tool for diagnosing network issues and understanding network activity

   Syntax : **netstat -a <destination>**

   Example :

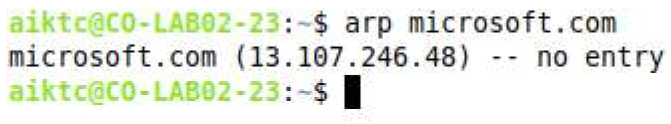


```
aiktc@CO-LAB02-23:~$ netstat -a aman-antuley.vercel.app
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:microsoft-ds    0.0.0.0:*               LISTEN
tcp        0      0 localhost:33060         0.0.0.0:*               LISTEN
tcp        0      0 localhost:mysql         0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:netbios-ssn     0.0.0.0:*               LISTEN
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN
tcp        0      0 CO-LAB02-23:55684       93.243.107.34.bc.:https ESTABLISHED
tcp        0      0 CO-LAB02-2:microsoft-ds 172.16.120.6:38918      FIN_WAIT2
tcp        0      0 CO-LAB02-23:55722       pnbomb-bp-in-f3.1:https ESTABLISHED
tcp        0      0 CO-LAB02-2:microsoft-ds 172.16.200.166:59286    FIN_WAIT2
tcp        0      0 CO-LAB02-23:60278       bud02s41-in-f10.1:https ESTABLISHED
tcp        0      0 CO-LAB02-23:48930       bom12s20-in-f10.1:https ESTABLISHED
tcp        0      0 CO-LAB02-2:microsoft-ds 172.16.120.5:34790      FIN_WAIT2
tcp        0      0 CO-LAB02-23:46164       209.100.149.34.bc:https ESTABLISHED
tcp        0      0 CO-LAB02-23:50712       hkg12s09-in-f10.1:https ESTABLISHED
tcp        0      0 CO-LAB02-2:microsoft-ds 172.16.120.6:38938      FIN_WAIT2
tcp        0      0 CO-LAB02-2:microsoft-ds 172.16.120.5:34784      FIN_WAIT2
tcp        0      0 CO-LAB02-23:54916       82.221.107.34.bc.g:http ESTABLISHED
tcp        0      0 CO-LAB02-2:microsoft-ds 172.16.120.5:34762      FIN_WAIT2
```

5) **ARP :** The arp -a command is used to display the Address Resolution Protocol (ARP) cache on a computer. It allows you to see the **MAC** addresses associated with IP addresses, add new entries, or delete existing ones.
arp -a (or just arp): This command displays all current ARP entries in the cache.
The output typically shows the IP address, the corresponding MAC address, and the interface through which the entry was learned.

Syntax : arp -a <destination>

Example :

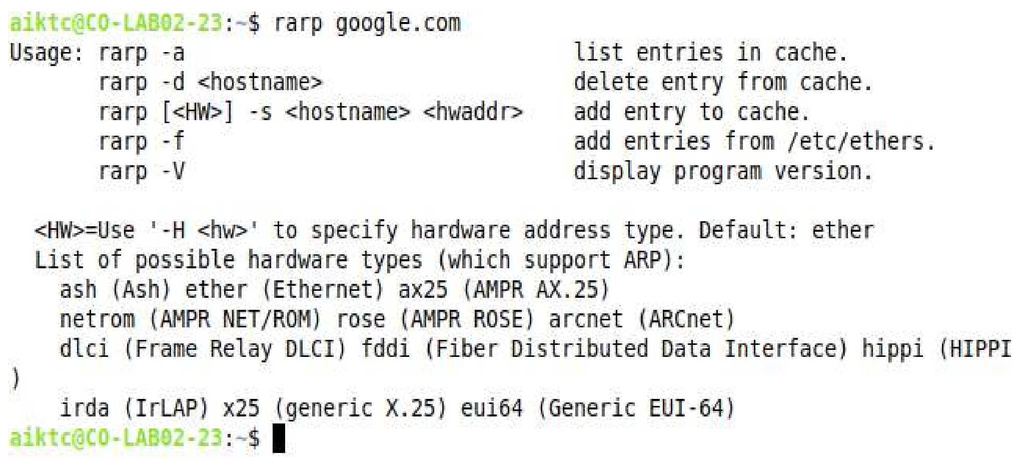


```
aiktc@CO-LAB02-23:~$ arp microsoft.com
microsoft.com (13.107.246.48) -- no entry
aiktc@CO-LAB02-23:~$ 
```

6) **RARP :** RARP (Reverse Address Resolution Protocol) is a legacy protocol used to discover an IP address from a known MAC address, typically for diskless workstations. It's mostly replaced by BOOTP and DHCP. The core functionality of RARP involves sending a RARP request (containing the MAC address) and receiving a RARP reply (containing the corresponding IP address)

Syntax : rarp <destination>

Example :



```
aiktc@CO-LAB02-23:~$ rarp google.com
Usage: rarp -a                               list entries in cache.
       rarp -d <hostname>                    delete entry from cache.
       rarp [<HW>] -s <hostname> <hwaddr>    add entry to cache.
       rarp -f                               add entries from /etc/ethers.
       rarp -V                               display program version.

  <HW>=Use '-H <hw>' to specify hardware address type. Default: ether
  List of possible hardware types (which support ARP):
    ash (Ash) ether (Ethernet) ax25 (AMPR AX.25)
    netrom (AMPR NET/ROM) rose (AMPR ROSE) arcnet (ARCnet)
    dlci (Frame Relay DLCI) fddi (Fiber Distributed Data Interface) hippi (HIPPI
)
    irda (IrLAP) x25 (generic X.25) eui64 (Generic EUI-64)
aiktc@CO-LAB02-23:~$
```

It shows an error because it works with a MAC address to give the IP address.

7) **IP :** This command is a powerful utility in Linux used for managing network interfaces, IP addresses, routing tables, and more. It's part of the iproute2 package and is designed to replace older commands like ifconfig and route. Essentially, it allows administrators to configure and view various aspects of a system's network setup.

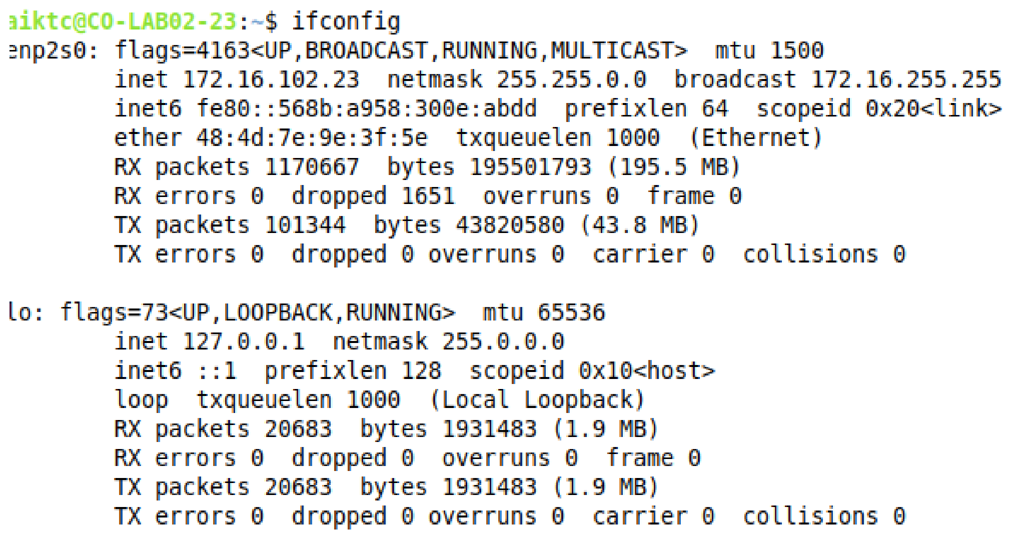
   Syntax : **ip**

   Example : Same as IFCONFIG

8) **IFCONFIG :** It is a command-line utility in Unix-like operating systems (like Linux and macOS) used to configure and display network interface settings. It displays Network Information.

   Syntax : **ifconfig**

   Example :



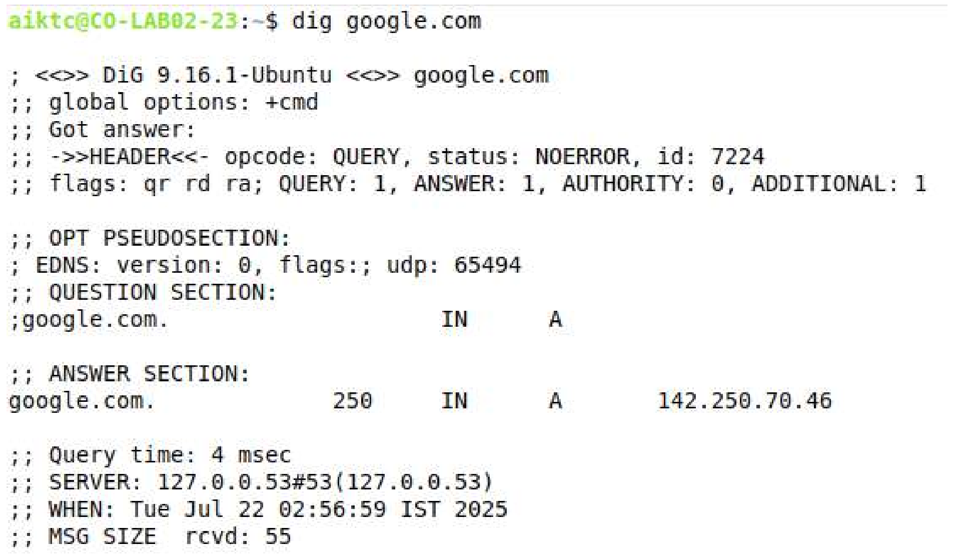
```
aiktc@CO-LAB02-23:~$ ifconfig
enp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.102.23  netmask 255.255.0.0  broadcast 172.16.255.255
        inet6 fe80::568b:a958:300e:abdd  prefixlen 64  scopeid 0x20<link>
        ether 48:4d:7e:9e:3f:5e  txqueuelen 1000  (Ethernet)
        RX packets 1170667  bytes 195501793 (195.5 MB)
        RX errors 0  dropped 1651  overruns 0  frame 0
        TX packets 101344  bytes 43820580 (43.8 MB)
        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 20683  bytes 1931483 (1.9 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 20683  bytes 1931483 (1.9 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

9) **DIG :** DIG (Domain Information Groper) is a flexible command-line tool for interrogating DNS (Domain Name System) name servers. It performs DNS lookups and displays the answers that are returned from the name servers. It is commonly used by system administrators and network engineers for troubleshooting DNS problems and for verifying DNS configurations.

Syntax : dig [server] [name] [type]

Example :



```
aiktc@CO-LAB02-23:~$ dig google.com

; <<>> DiG 9.16.1-Ubuntu <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7224
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;google.com.                    IN      A

;; ANSWER SECTION:
google.com.             250     IN      A       142.250.70.46

;; Query time: 4 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: Tue Jul 22 02:56:59 IST 2025
;; MSG SIZE  rcvd: 55
```

10) **ROUTE :** ROUTE is a command-line utility used to view and manipulate the IP routing table. It allows users to add, delete, and modify network routes, which determine the path that network packets take to reach their destinations. This is essential for configuring how a computer or server communicates with other networks.

Syntax :

route

add: Adds a new route.

del or delete: Deletes an existing route.

print: Displays the routing table.

netmask: Specifies the subnet mask for the destination network.

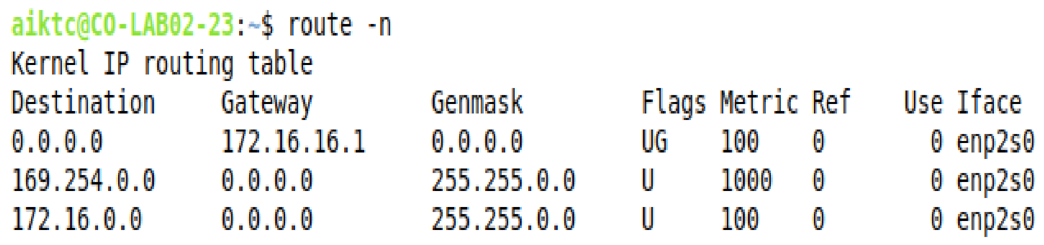
gw: Specifies the gateway (router) through which the traffic should be routed.

metric: Sets the metric (cost) for the route, which can influence route selection in some routing protocols.

dev: Specifies the network interface to use for the route.

if: Specifies the interface index for the route

Example :



```
aiktc@CO-LAB02-23:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         172.16.16.1     0.0.0.0         UG    100    0        0 enp2s0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 enp2s0
172.16.0.0      0.0.0.0         255.255.0.0     U     100    0        0 enp2s0
```

Conclusion:

Basic networking commands in Linux such as `ping`, `traceroute`, `nslookup`, `netstat`, `arp`, `ipconfig`, `dig`, and `route` are essential tools diagnosing and managing network-related issues. These commands help in testing connectivity, resolving DNS, checking routing tables, viewing IP configurations, and understanding how data travels across network