

Assignment 1 - Network Environments

Adeeb Hadi Seeyad (B190737CS)

Introduction

Following tools I used to explore and summarize the network environment available in my system:

1. ping
2. traceroute
3. tracepath
4. ip / ifconfig
5. nslookup
6. whois
7. route
8. tcpdump
9. netstat / ss
10. dstat
11. ifstat
12. wget

1. ping

The ping (Packet Internet Groper) command sends packets of ICMP echo requests to a specific IP address on a network, and then lets you know how long it took to transmit that data and get a response.

Eg. *ping google.com*

```
adeeb in ~
→ ping -c 10 google.com
PING google.com(fjr01s01-in-x0e.1e100.net (2a00:1450:4019:805::200e)) 56 data bytes
64 bytes from fjr01s01-in-x0e.1e100.net (2a00:1450:4019:805::200e): icmp_seq=1 ttl=117 time=5.06 ms
64 bytes from fjr01s01-in-x0e.1e100.net (2a00:1450:4019:805::200e): icmp_seq=2 ttl=117 time=5.95 ms
64 bytes from fjr01s01-in-x0e.1e100.net (2a00:1450:4019:805::200e): icmp_seq=3 ttl=117 time=5.64 ms
64 bytes from fjr01s01-in-x0e.1e100.net (2a00:1450:4019:805::200e): icmp_seq=4 ttl=117 time=6.26 ms
64 bytes from fjr01s01-in-x0e.1e100.net (2a00:1450:4019:805::200e): icmp_seq=5 ttl=117 time=5.27 ms
64 bytes from fjr01s01-in-x0e.1e100.net (2a00:1450:4019:805::200e): icmp_seq=6 ttl=117 time=5.11 ms
64 bytes from fjr01s01-in-x0e.1e100.net (2a00:1450:4019:805::200e): icmp_seq=7 ttl=117 time=4.96 ms
64 bytes from fjr01s01-in-x0e.1e100.net (2a00:1450:4019:805::200e): icmp_seq=8 ttl=117 time=4.85 ms
64 bytes from fjr01s01-in-x0e.1e100.net (2a00:1450:4019:805::200e): icmp_seq=9 ttl=117 time=6.08 ms
64 bytes from fjr01s01-in-x0e.1e100.net (2a00:1450:4019:805::200e): icmp_seq=10 ttl=117 time=5.94 ms

--- google.com ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9013ms
rtt min/avg/max/mdev = 4.845/5.511/6.255/0.493 ms

adeeb in ~
→ █
```

- ❖ from : The destination and its IP address.
- ❖ icmp_seq : The sequence number of each ICMP packet. Increase by one for every echo request.
- ❖ ttl (Time to Live) : represents the number of network hops a packet can take before a router discards it.
- ❖ time : The time it took for a packet to reach its destination and come back to the source in milliseconds.

2. traceroute

It sends 3 packets to each intermediate hop to the network host and traces the packets. It displays the ip address of each hop and time taken for the packet to reach each hop. Some of its functionalities require super user privileges.

Eg. *traceroute google.com*

```
adeeb in ~
→ traceroute -T google.com
You do not have enough privileges to use this traceroute method.
socket: Operation not permitted

adeeb in ~
→ sudo !!
sudo traceroute -T google.com
[sudo] password for adeeb:
traceroute to google.com (216.58.208.238), 30 hops max, 60 byte packets
 1 _gateway (192.168.0.1)  1.337 ms  1.256 ms  1.934 ms
 2 * * *
 3 10.246.254.14 (10.246.254.14)  8.551 ms 10.246.254.13 (10.246.254.13)  10.828 ms 11.086 ms
 4 10.29.19.151 (10.29.19.151)  11.945 ms 10.29.19.149 (10.29.19.149)  13.042 ms 13.211 ms
 5 195.229.0.110 (195.229.0.110)  14.721 ms 195.229.0.112 (195.229.0.112)  15.337 ms 195.229.0.116 (195.229.0.116)  15.230 ms
 6 195.229.4.55 (195.229.4.55)  15.798 ms 195.229.6.7 (195.229.6.7)  15.707 ms 195.229.5.101 (195.229.5.101)  15.314 ms
 7 108.170.247.17 (108.170.247.17)  15.219 ms 172.253.51.55 (172.253.51.55)  11.241 ms 209.85.241.195 (209.85.241.195)  9.097 ms
 8 72.14.238.197 (72.14.238.197)  10.094 ms 74.125.253.227 (74.125.253.227)  11.111 ms 11.675 ms
 9 par10s22-in-f14.1e100.net (216.58.208.238)  9.807 ms  9.754 ms  9.458 ms

adeeb in ~
→ traceroute -I google.com
traceroute to google.com (216.58.208.238), 30 hops max, 60 byte packets
 1 _gateway (192.168.0.1)  0.859 ms  0.866 ms  1.761 ms
 2 31.215.228.2 (31.215.228.2)  3.064 ms  3.053 ms  3.558 ms
 3 10.246.254.14 (10.246.254.14)  5.998 ms  5.968 ms  5.959 ms
 4 195.229.0.116 (195.229.0.116)  8.380 ms  9.019 ms  9.013 ms
 5 195.229.4.51 (195.229.4.51)  9.004 ms  8.998 ms  9.026 ms
 6 172.253.51.205 (172.253.51.205)  8.988 ms  7.613 ms  9.056 ms
 7 74.125.253.227 (74.125.253.227)  8.793 ms  7.432 ms  7.322 ms
 8 par10s22-in-f238.1e100.net (216.58.208.238)  6.892 ms  6.810 ms  6.791 ms
```

- ❖ The T flag uses TCP SYN for probs and it requires superuser privileges.
- ❖ The I flag uses ICMP ECHO for probs and it does **not** require superuser privileges.

3. tracepath

Tracepath traces a path to a designated network address, reporting on the TTL lag and maximum transmission units (MTU) along the way. This command can be run by any user without superuser privileges unlike traceroute.

Eg. *tracepath google.com*

```
adeeb in ~ took 10s
→ man tracepath

adeeb in ~ took 45s
→ tracepath google.com
1?: [LOCALHOST]          0.013ms pmtu 1500
1: 2001:8f8:1521:abef:7a98:e8ff:fe72:55ac      3.109ms
1: 2001:8f8:1521:abef:7a98:e8ff:fe72:55ac      3.362ms
2: 2001:8f8:1521:abef:7a98:e8ff:fe72:55ac      3.320ms pmtu 1492
2: no reply
3: 2001:8f8:3:a106::2                         13.381ms
4: 2001:8f8:0:10:0:23:224:5                   9.625ms asymm 5
5: 2001:8f8:0:13::2e                          9.757ms asymm 9
6: no reply
7: no reply
8: no reply
9: no reply
10: no reply
11: no reply
12: no reply
^C

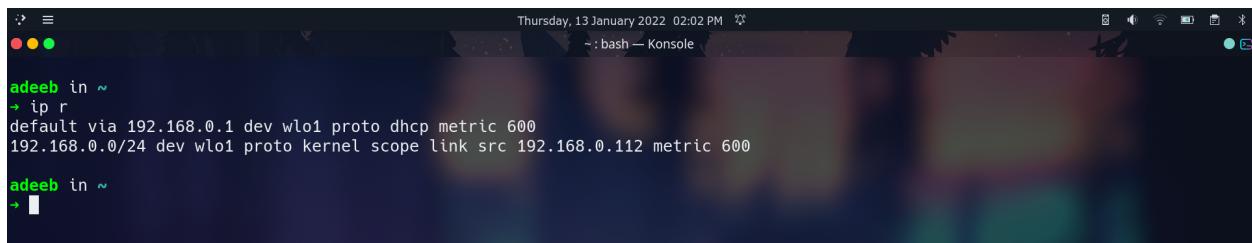
adeeb in ~ took 26s
→ tracepath -b google.com
1?: [LOCALHOST]          0.014ms pmtu 1492
1: 2001:8f8:1521:abef:7a98:e8ff:fe72:55ac (2001:8f8:1521:abef:7a98:e8ff:fe72:55ac)  3.387ms
1: 2001:8f8:1521:abef:7a98:e8ff:fe72:55ac (2001:8f8:1521:abef:7a98:e8ff:fe72:55ac)  3.482ms
2: no reply
3: 2001:8f8:3:a106::2 (2001:8f8:3:a106::2)    9.672ms
4: 2001:8f8:0:10:0:20:224:5 (2001:8f8:0:10:0:20:224:5)  9.908ms
5: 2001:8f8:0:13::2e (2001:8f8:0:13::2e)        13.633ms asymm 9
6: no reply
7: no reply
```

4. ip / ifconfig

IP

IP (Internet Protocol) Address is an address of your network hardware. It helps in connecting your computer to other devices on your network and all over the world.

The ip command shows or manipulates routing, network devices, interfaces and tunnels.



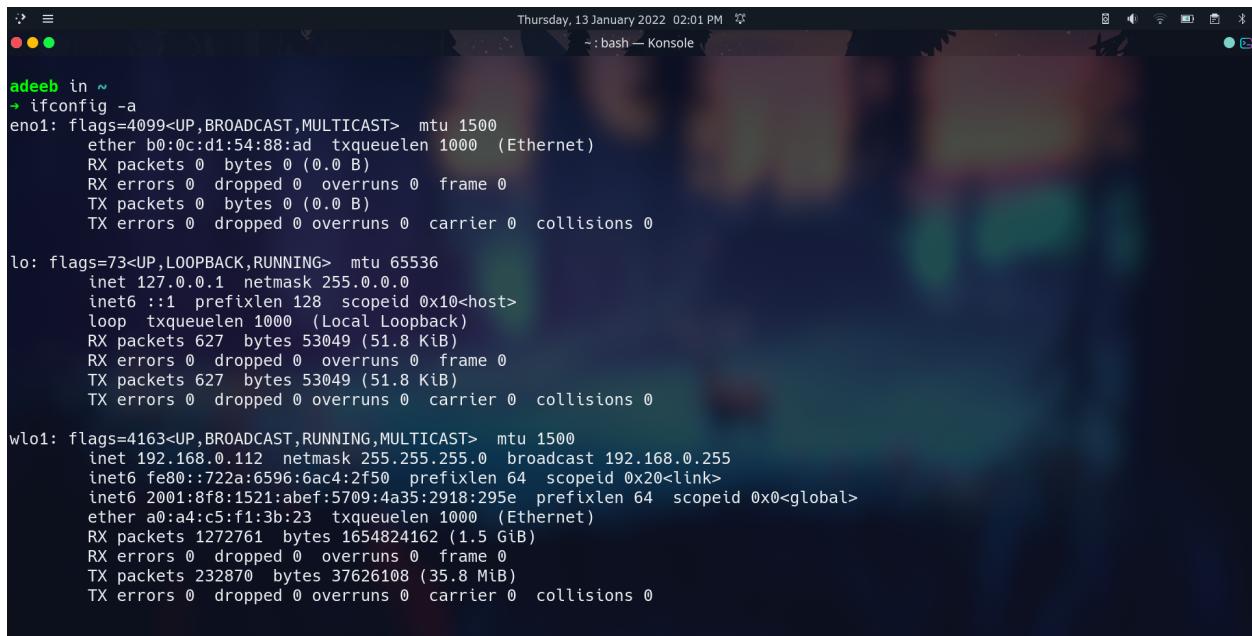
```
adeeb in ~
→ ip r
default via 192.168.0.1 dev wlo1 proto dhcp metric 600
192.168.0.0/24 dev wlo1 proto kernel scope link src 192.168.0.112 metric 600

adeeb in ~
→ █
```

ifconfig

Stands for Interface configuration. It is often used for troubleshooting network connectivities. Ifconfig is used at the boot time to set-up the interfaces as necessary.

Eg. *Ifconfig -a* (display all interfaces which are currently available in the system)



```
adeeb in ~
→ ifconfig -a
en0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether b0:0c:d1:54:88:ad 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 627 bytes 53049 (51.8 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 627 bytes 53049 (51.8 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlo1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.112 netmask 255.255.255.0 broadcast 192.168.0.255
        inet6 fe80::722a:6596:6ac4:2f50 prefixlen 64 scopeid 0x20<link>
            inet6 2001:8f8:1521:abef:5709:4a35:2918:295e prefixlen 64 scopeid 0x0<global>
                ether a0:a4:c5:f1:3b:23 txqueuelen 1000 (Ethernet)
                RX packets 1272761 bytes 1654824162 (1.5 GiB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 232870 bytes 37626108 (35.8 MiB)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

5. nslookup

About

The command queries Internet name servers interactively and non-interactively from DNS (Domain Name Server).

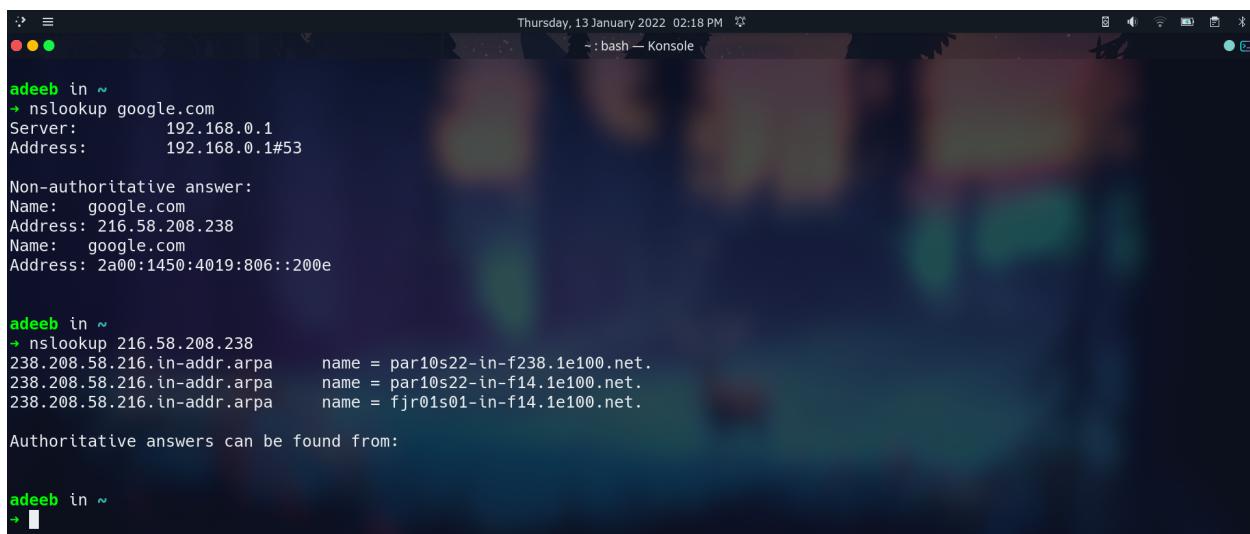
It can also perform reverse lookups, that is, given the ip address it can find the corresponding domain name.

Host is used by default to determine what domain a particular IP address resolves to. It can be changed interactively with the command `server 8.8.8.8` (uses google's server to determine the IP address)

DNS (Domain Name System)

The DNS turns domain names into IP addresses, which browsers use to load internet pages.

Command Execution

A screenshot of a Linux desktop environment showing a terminal window titled "Konsole". The terminal window has a dark background and displays the output of several nslookup commands. The first command, "nslookup google.com", shows the server as 192.168.0.1 and the address as 192.168.0.1#53. It then provides a non-authoritative answer for the domain google.com, listing both IPv4 (216.58.208.238) and IPv6 (2a00:1450:4019:806::200e) addresses. The second command, "nslookup 216.58.208.238", performs a reverse lookup for the IP address 216.58.208.238, returning three domain names: par10s22-in-f238.1e100.net, par10s22-in-f14.1e100.net, and fjr01s01-in-f14.1e100.net. The terminal prompt "adeeb in ~" is visible at the bottom.

```
adeeb in ~
+ nslookup google.com
Server:      192.168.0.1
Address:     192.168.0.1#53

Non-authoritative answer:
Name:   google.com
Address: 216.58.208.238
Name:   google.com
Address: 2a00:1450:4019:806::200e

adeeb in ~
+ nslookup 216.58.208.238
238.208.58.216.in-addr.arpa    name = par10s22-in-f238.1e100.net.
238.208.58.216.in-addr.arpa    name = par10s22-in-f14.1e100.net.
238.208.58.216.in-addr.arpa    name = fjr01s01-in-f14.1e100.net.

Authoritative answers can be found from:

adeeb in ~
+ 
```

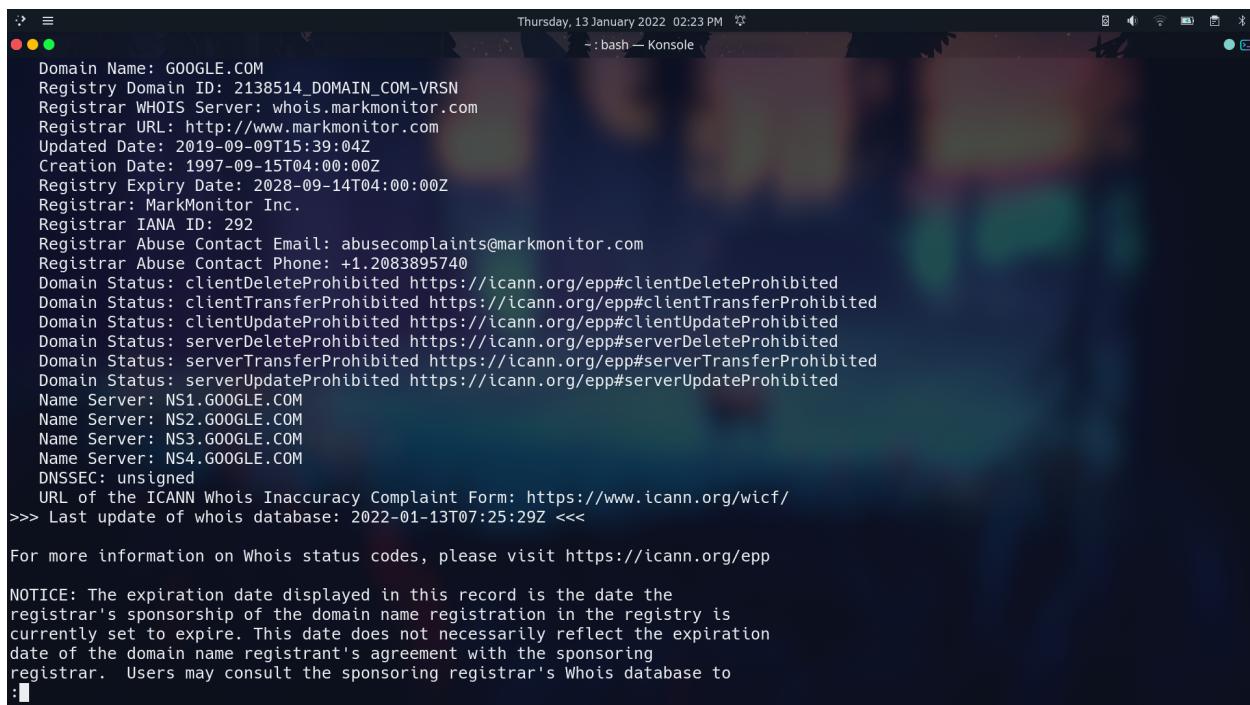
As seen in the first command *nslookup google.com*, it displays both IPv4 and IPv6 address of the domain name, first address being the IPv4 address and second one being the IPv6 address.

The second command performs reverse lookup.

6. whois

This command searches for an object in a RFC 3912 database for listing of records that contain details about the ownership of domains and the owners. The whois client tries to guess the right server to ask for the specified object. If no guess can be made it will connect to whois.networksolutions.com for NIC handles or whois.arin.net for IPv4 addresses and network names.

Eg. *whois google.com*



```
Thursday, 13 January 2022 02:23 PM ~ : bash — Konsole

Domain Name: GOOGLE.COM
Registry Domain ID: 2138514_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2019-09-09T15:39:04Z
Creation Date: 1997-09-15T04:00:00Z
Registry Expiry Date: 2028-09-14T04:00:00Z
Registrar: MarkMonitor Inc.
Registrar IANA ID: 292
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
Registrar Abuse Contact Phone: +1.2083895740
Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited
Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited
Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited
Name Server: NS1.GOOGLE.COM
Name Server: NS2.GOOGLE.COM
Name Server: NS3.GOOGLE.COM
Name Server: NS4.GOOGLE.COM
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2022-01-13T07:25:29Z <<<

For more information on Whois status codes, please visit https://icann.org/epp

NOTICE: The expiration date displayed in this record is the date the
registrar's sponsorship of the domain name registration in the registry is
currently set to expire. This date does not necessarily reflect the expiration
date of the domain name registrant's agreement with the sponsoring
registrar. Users may consult the sponsoring registrar's Whois database to
:■
```

```
Thursday, 13 January 2022 02:23 PM ~: bash — Konsole

The Registry database contains ONLY .COM, .NET, .EDU domains and
Registrars.
Domain Name: google.com
Registry Domain ID: 2138514_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.markmonitor.com
Registrar URL: http://www.markmonitor.com
Updated Date: 2019-09-09T15:39:04+0000
Creation Date: 1997-09-15T07:00:00+0000
Registrar Registration Expiration Date: 2028-09-13T07:00:00+0000
Registrar: MarkMonitor, Inc.
Registrar IANA ID: 292
Registrar Abuse Contact Email: abusecomplaints@markmonitor.com
Registrar Abuse Contact Phone: +1.2083895770
Domain Status: clientUpdateProhibited (https://www.icann.org/epp#clientUpdateProhibited)
Domain Status: clientTransferProhibited (https://www.icann.org/epp#clientTransferProhibited)
Domain Status: clientDeleteProhibited (https://www.icann.org/epp#clientDeleteProhibited)
Domain Status: serverUpdateProhibited (https://www.icann.org/epp#serverUpdateProhibited)
Domain Status: serverTransferProhibited (https://www.icann.org/epp#serverTransferProhibited)
Domain Status: serverDeleteProhibited (https://www.icann.org/epp#serverDeleteProhibited)
Registrant Organization: Google LLC
Registrant State/Province: CA
Registrant Country: US
Registrant Email: Select Request Email Form at https://domains.markmonitor.com/whois/google.com
Admin Organization: Google LLC
Admin State/Province: CA
Admin Country: US
Admin Email: Select Request Email Form at https://domains.markmonitor.com/whois/google.com
Tech Organization: Google LLC
Tech State/Province: CA
Tech Country: US
Tech Email: Select Request Email Form at https://domains.markmonitor.com/whois/google.com
:■
```

```
Thursday, 13 January 2022 02:24 PM ~: bash — Konsole

https://domains.markmonitor.com/whois

If you have a legitimate interest in viewing the non-public WHOIS details, send
your request and the reasons for your request to whoisrequest@markmonitor.com
and specify the domain name in the subject line. We will review that request and
may ask for supporting documentation and explanation.

The data in MarkMonitor's WHOIS database is provided for information purposes,
and to assist persons in obtaining information about or related to a domain
name's registration record. While MarkMonitor believes the data to be accurate,
the data is provided "as is" with no guarantee or warranties regarding its
accuracy.

By submitting a WHOIS query, you agree that you will use this data only for
lawful purposes and that, under no circumstances will you use this data to:
(1) allow, enable, or otherwise support the transmission by email, telephone,
or facsimile of mass, unsolicited, commercial advertising, or spam; or
(2) enable high volume, automated, or electronic processes that send queries,
data, or email to MarkMonitor (or its systems) or the domain name contacts (or
its systems).

MarkMonitor reserves the right to modify these terms at any time.

By submitting this query, you agree to abide by this policy.

MarkMonitor Domain Management(TM)
Protecting companies and consumers in a digital world.

Visit MarkMonitor at https://www.markmonitor.com
Contact us at +1.8007459229
In Europe, at +44.02032062220
--  
(END)
```

7. route

About

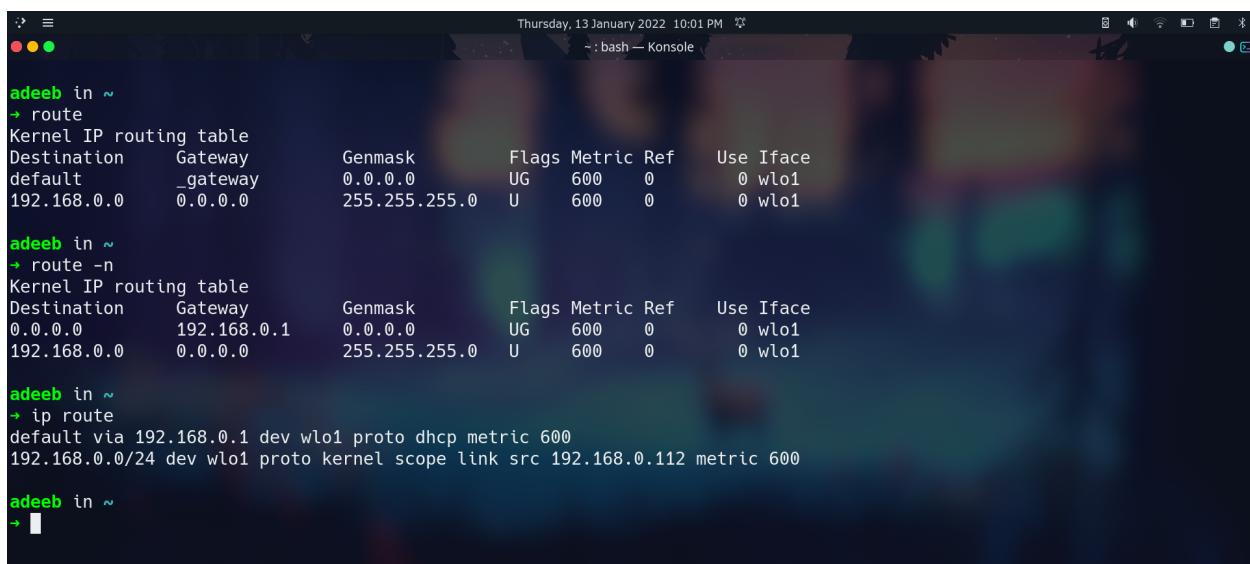
The command shows or manipulates the IP routing table. Its primary use is to set up static routes to specific hosts or networks via an interface after it has been configured with [ifconfig](#).

Routing Table

It is a data table stored in a router or a network host that lists the routes to particular network destinations, and in some cases, metrics associated with those routes.

Command Execution

1. *route* -- displays routing table entries.
2. *route -n* -- displays routing table entries in numerical form
3. *ip route* -- displays details of IP routing table



```
Thursday, 13 January 2022 10:01 PM ~ : bash — Konsole

adeeb in ~
→ 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

adeeb in ~
→ route -n
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref    Use Iface
0.0.0.0         192.168.0.1   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

adeeb in ~
→ ip route
default via 192.168.0.1 dev wlo1 proto dhcp metric 600
192.168.0.0/24 dev wlo1 proto kernel scope link src 192.168.0.112 metric 600

adeeb in ~
→ █
```

8. tcpdump

It allows the user to display TCP/IP and other packets being transmitted or received over a network to which the computer is attached.

Eg 1. `sudo tcpdump` -- captures packets from the interface of the network through which the system is connected to the internet

Eg 2. `sudo tcpdump -D` -- print list of networks that this tool can capture packets from

Eg 3. `sudo tcpdump -i any` -- captures packets from the interface of any network

```
adeeb in ~
→ sudo tcpdump -c 10
[sudo] password for adeeb:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on wlo1, link-type EN10MB (Ethernet), snapshot length 262144 bytes
23:05:52.687602 IP Adeeb.36120 > ec2-23-21-18-146.compute-1.amazonaws.com.https: Flags [P.], seq 1958684030:1958684422,
ack 2197820678, win 501, options [nop,nop,TS val 3437119260 ecr 274225406], length 392
23:05:52.703682 IP Adeeb.41277 > _gateway.domain: 48934+ PTR? 146.18.21.23.in-addr.arpa. (43)
23:05:52.922503 IP _gateway.domain > Adeeb.41277: 48934 1/0/0 PTR ec2-23-21-18-146.compute-1.amazonaws.com. (97)
23:05:52.922601 IP ec2-23-21-18-146.compute-1.amazonaws.com.https > Adeeb.36120: Flags [.], ack 392, win 236, options [
nop,nop,TS val 274237186 ecr 3437119260], length 0
23:05:53.699772 IP Adeeb.56592 > 151.101.1.140.https: Flags [P.], seq 2021261296:2021261413, ack 4293392530, win 501, o
ptions [nop,nop,TS val 864071029 ecr 4124615520], length 117
23:05:53.699959 IP Adeeb.56592 > 151.101.1.140.https: Flags [.], seq 117:1497, ack 1, win 501, options [nop,nop,TS val
864071029 ecr 4124615520], length 1380
23:05:53.699981 IP Adeeb.56592 > 151.101.1.140.https: Flags [P.], seq 1497:2655, ack 1, win 501, options [nop,nop,TS va
l 864071029 ecr 4124615520], length 1158
23:05:53.706377 IP 151.101.1.140.https > Adeeb.56592: Flags [.], ack 117, win 297, options [nop,nop,TS val 4124643682 e
cr 864071029], length 0
23:05:53.708673 IP 151.101.1.140.https > Adeeb.56592: Flags [.], ack 1497, win 302, options [nop,nop,TS val 4124643684
ecr 864071029], length 0
23:05:53.708692 IP 151.101.1.140.https > Adeeb.56592: Flags [P.], seq 1:40, ack 1497, win 302, options [nop,nop,TS val
4124643684 ecr 864071029], length 39
10 packets captured
16 packets received by filter
0 packets dropped by kernel

adeeb in ~
→ █
```

```
adeeb in ~
→ sudo tcpdump -D
1.wlo1 [Up, Running, Wireless, Associated]
2.any (Pseudo-device that captures on all interfaces) [Up, Running]
3.lo [Up, Running, Loopback]
4.eno1 [Up, Disconnected]
5.bluetooth0 (Bluetooth adapter number 0) [Wireless, Association status unknown]
6.bluetooth-monitor (Bluetooth Linux Monitor) [Wireless]
7.nflog (Linux netfilter log (NFLOG) interface) [none]
8.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
9 dbus-system (D-Bus system bus) [none]
10 dbus-session (D-Bus session bus) [none]

adeeb in ~
→ █
```

9. netstat / ss

netstat

It displays network connections for TCP/UDP and stats for Interfaces, Network protocols, routing tables, etc

ss

It's another utility to investigate sockets, replacing netstat. ss command tool which dumps socket stats and displays information similarly but it is faster than netstat. Usually used in troubleshooting network problems.

Command Execution

1. `ss -s` *-- displays summary stats*
2. `ss -t` *-- displays all TCP connections*
3. `ss -u` *-- displays all UDP connections*
4. `ss -x` *-- displays all unix connections*
5. `netstat -ie` *-- displays all the network interfaces (similar to [ifconfig](#) with a flag)*
6. `netstat -pnl | grep <port number>` *-- Check if specific port is listening*

```

adeeb in ~
→ ss -t
State      Recv-Q      Send-Q      Local Address:Port      Peer Address:Port      Process
ESTAB      0            0           192.168.0.112:43794  44.226.241.1:https
ESTAB      0            0           192.168.0.112:43428  89.187.162.107:dynamid

adeeb in ~
→ ss -u
Recv-Q      Send-Q      Local Address:Port      Peer Address:Port      Process
0            0           192.168.0.112%wlo1:bootpc  192.168.0.1:bootps
0            0           [::1]:48425

adeeb in ~
→ ss -x
Netid  State  Recv-Q  Send-Q      Local Address:Port      Peer Address:Port      Process
u_dgr  ESTAB  0        0           /run/systemd/notify 13752  * 0
u_dgr  ESTAB  0        0           /run/systemd/journal/dev-log 13776  * 0
u_dgr  ESTAB  0        0           /run/systemd/journal/socket 13778  * 0
u_seq   ESTAB  0        0           @00004 23421       * 23420
u_seq   ESTAB  0        0           @00005 24981       * 24982
u_seq   ESTAB  0        0           @00006 24983       * 24984
u_seq   ESTAB  0        0           @00001 23420       * 23421
u_seq   ESTAB  0        0           @00002 24343       * 24344
u_seq   ESTAB  0        0           @00003 24344       * 24343
u_str   ESTAB  0        0           * 24625          * 24626
u_str   ESTAB  0        0           @/tmp/.ICE-unix/799 17339  * 17338
u_str   ESTAB  0        0           /run/systemd/journal/stdout 16015  * 17279
u_str   ESTAB  0        0           * 70306          * 70307
u_str   ESTAB  0        0           * 21492          * 23037
u_str   ESTAB  0        0           * 30653          * 38921
u_str   ESTAB  0        0           * 23114          * 19354
u_str   ESTAB  0        0           * 11716          * 17218

```

```

adeeb in ~
→ ss -s
Total: 846
TCP:  11 (estab 2, closed 4, orphaned 0, timewait 0)

Transport Total      IP          IPv6
RAW     1            0           1
UDP     4            1           3
TCP     7            4           3
INET    12           5           7
FRAG    0            0           0

adeeb in ~
→ █

```

```

adeeb in ~
→ netstat -pl | grep ssh
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
tcp      0      0 0.0.0.0:ssh          0.0.0.0:*      LISTEN      -
tcp6     0      0 [::]:ssh            [::]:*        LISTEN      -
unix    2      [ ACC ]      STREAM      LISTENING    21428      699/systemd  /run/user/1000/gcr/ssh
unix    2      [ ACC ]      STREAM      LISTENING    21434      699/systemd  /run/user/1000/gnupg/S.gpg-agent.ssh

adeeb in ~
→ netstat -pnl | grep :22
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
tcp      0      0 0.0.0.0:22          0.0.0.0:*      LISTEN      -
tcp6     0      0 :::22              :::*        LISTEN      -

adeeb in ~
→ █

```

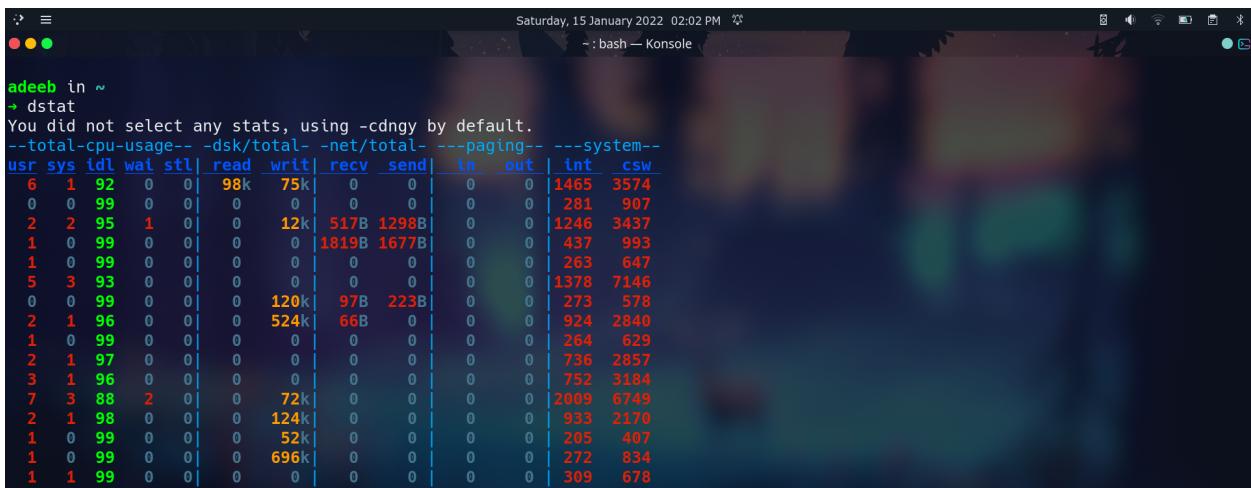
10. dstat

About

Dstat allows you to view all of your system resources instantly, you can eg. compare disk usage in combination with interrupts from your IDE controller, or compare the network bandwidth numbers directly with the disk throughput (in the same interval).

Command Execution

1. dstat



```
Saturday, 15 January 2022 02:02 PM ✘
adeeb in ~
→ dstat
You did not select any stats, using -cdng by default.
--total-cpu-usage-- -dsk/total- -net/total- ---paging--- ---system--
usr sys idl wai stl| read wrt| recv send| in out| int csw
 6  1 92  0  0| 98k  75k|   0   0|   0   0| 1465 3574
 0  0 99  0  0| 0    0|   0   0|   0   0| 281  907
 2  2 95  1  0| 0    0| 12k  517B 1298B|   0   0| 1246 3437
 1  0 99  0  0| 0    0| 0  1819B 1677B|   0   0| 437  993
 1  0 99  0  0| 0    0| 0    0|   0   0| 263  647
 5  3 93  0  0| 0    0| 0    0|   0   0| 1378 7146
 0  0 99  0  0| 0    0| 120k  97B 223B|   0   0| 273  578
 2  1 96  0  0| 0    0| 524k  66B|   0   0| 924 2840
 1  0 99  0  0| 0    0| 0    0|   0   0| 264  629
 2  1 97  0  0| 0    0| 0    0|   0   0| 736 2857
 3  1 96  0  0| 0    0| 0    0|   0   0| 752 3184
 7  3 88  2  0| 0    0| 72k|   0   0|   0   0| 2009 6749
 2  1 98  0  0| 0    0| 124k|   0   0|   0   0| 933 2170
 1  0 99  0  0| 0    0| 52k|   0   0|   0   0| 205  407
 1  0 99  0  0| 0    0| 696k|   0   0|   0   0| 272  834
 1  1 99  0  0| 0    0| 0    0|   0   0|   0   0| 309  678
```

Column	Meaning
CPU Stats	CPU usage by user, system processes and number of idle processes, and number of waiting processes, hardware and software interrupts.
Disk Stats	Total number of read and write operations on the disk.
Network Stats	Total amount of Bytes received and sent on network interfaces.
Paging Stats	Number of times information is copied into and moved out of memory.
System Stats	Number of interrupts and context switches.

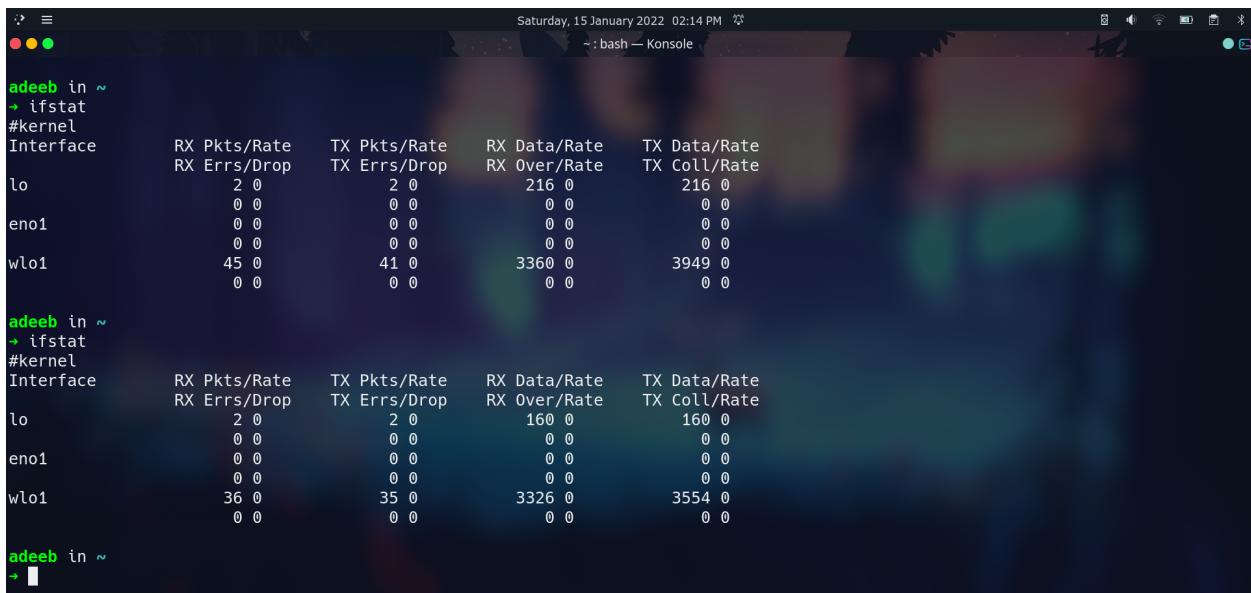
11. ifstat

About

Ifstat prints out network interface statistics. The utility keeps records of the previous data displayed in history files and by default only shows the difference between the last and the current call.

Command Execution

1. Ifstat -- Prints network interface statistics considering history
2. ifstat -a -- Ignores history
3. ifstat -e -- Shows errors
4. ifstat -r -- Resets history



```
Saturday, 15 January 2022 02:14 PM ~ :bash — Konsole

adeeb in ~
→ 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              2 0           2 0           216 0          216 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            45 0          41 0          3360 0         3949 0
                0 0           0 0           0 0            0 0

adeeb in ~
→ 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              2 0           2 0           160 0          160 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            36 0          35 0          3326 0         3554 0
                0 0           0 0           0 0            0 0

adeeb in ~
→ █
```

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.

```
adeeb in ~
→ ifstat -a
#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            390 0          390 0          34809 0         34809 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        248557 0        68075 0        330836K 0       9062K 0
              0 0           0 0           0 0            0 0
```

```
adeeb in ~
→ ifstat -e
#kernel
Interface      RX Pkts/Rate    TX Pkts/Rate    RX Data/Rate    TX Data/Rate
                  RX Errs/Rate   RX Drop/Rate   RX Over/Rate   RX Leng/Rate
                  RX Crc/Rate    RX Frm/Rate    RX Fifo/Rate   RX Miss/Rate
                  TX Errs/Rate   TX Drop/Rate   TX Coll/Rate  TX Carr/Rate
                  TX Abrt/Rate   TX Fifo/Rate   TX Hear/Rate  TX Wind/Rate
lo            0 0           0 0           0 0            0 0
              0 0           0 0           0 0            0 0
              0 0           0 0           0 0            0 0
              0 0           0 0           0 0            0 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
              0 0           0 0           0 0            0 0
              0 0           0 0           0 0            0 0
              0 0           0 0           0 0            0 0
wlo1        16 0          20 0          2444 0         3251 0
              0 0           0 0           0 0            0 0
              0 0           0 0           0 0            0 0
              0 0           0 0           0 0            0 0
              0 0           0 0           0 0            0 0
```

```
adeeb in ~
→ ifstat -r
#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            391 0          391 0          34889 0         34889 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        248576 0        68098 0        330839K 0       9065K 0
              0 0           0 0           0 0            0 0

adeeb in ~
→ 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            0 0           0 0           0 0            0 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          0 0           0 0           0 0            0 0
```

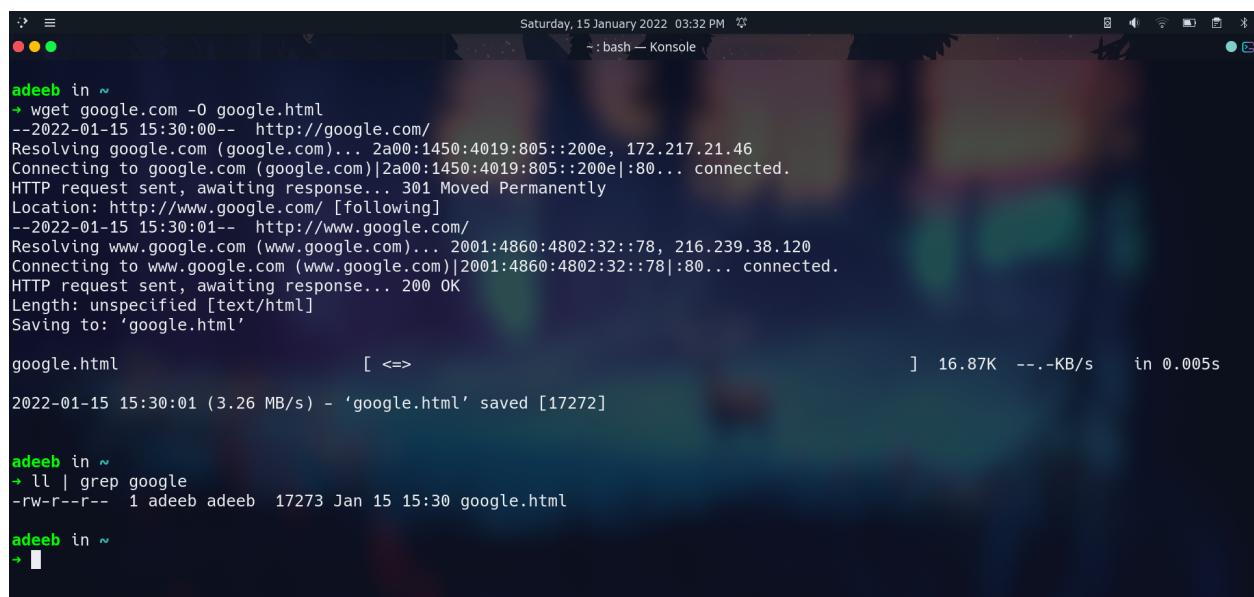
12. wget

About

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. With Wget, you can download files using HTTP, HTTPS, and FTP protocols.

Command Execution

1. `wget google.com`
2. `wget google.com -c` - - resume a partially downloaded file



The screenshot shows a terminal window titled "Konsole" running on a Linux desktop. The terminal output is as follows:

```
Saturday, 15 January 2022 03:32 PM
adeeb in ~
→ wget google.com -O google.html
--2022-01-15 15:30:00-- http://google.com/
Resolving google.com (google.com)... 2a00:1450:4019:805::200e, 172.217.21.46
Connecting to google.com (google.com)|2a00:1450:4019:805::200e|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://www.google.com/ [following]
--2022-01-15 15:30:01-- http://www.google.com/
Resolving www.google.com (www.google.com)... 2001:4860:4802:32::78, 216.239.38.120
Connecting to www.google.com (www.google.com)|2001:4860:4802:32::78|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'google.html'

google.html [ <=> ] 16.87K --.-KB/s in 0.005s

2022-01-15 15:30:01 (3.26 MB/s) - 'google.html' saved [17272]

adeeb in ~
→ ll | grep google
-rw-r--r-- 1 adeeb adeeb 17273 Jan 15 15:30 google.html

adeeb in ~
→ █
```