Lecture 5: Networking Commands

Introduction

Network commands are an essential toolkit for any network administrator or sysadmin. The commands help set up, troubleshoot, diagnose, and manage a Linux system's network connections.

Prerequisites

- Access to the command line/terminal.
- An administrator account with sudo privileges.

Linux Network Commands

Linux provides many helpful networking commands and tools. The commands typically perform complex networking tasks like monitoring, troubleshooting, and network configuration. Most networking utilities are part of the older (legacy) **net-tools** package or the more modern **iproute2**.

Note: Both **net-tools** and **iproute2** command are available on most Linux distributions. However, it is recommended to use **iproute2** tools due to their flexibility and speed.

Although **net-tools** is deemed as outdated, it is still widely used by legacy scripts and configurations.

Specific command syntax may differ depending on the command version. Double-check a command's syntax with:

1. man [command]

The man command displays the manual page for the specified command in the terminal.

Below is a brief overview of 20 Linux networking commands.

2. ip

The ip command is a unified networking tool for Linux systems. The **ip** command helps view and configure routing, interfaces, network devices, and tunnels.

The command is part of the **iproute2** package and replaces many older networking tools, such as the **route**, **ifconfig**, and **netstat** commands.

Syntax

The syntax for the **ip** command is:

ip [options] object [command]

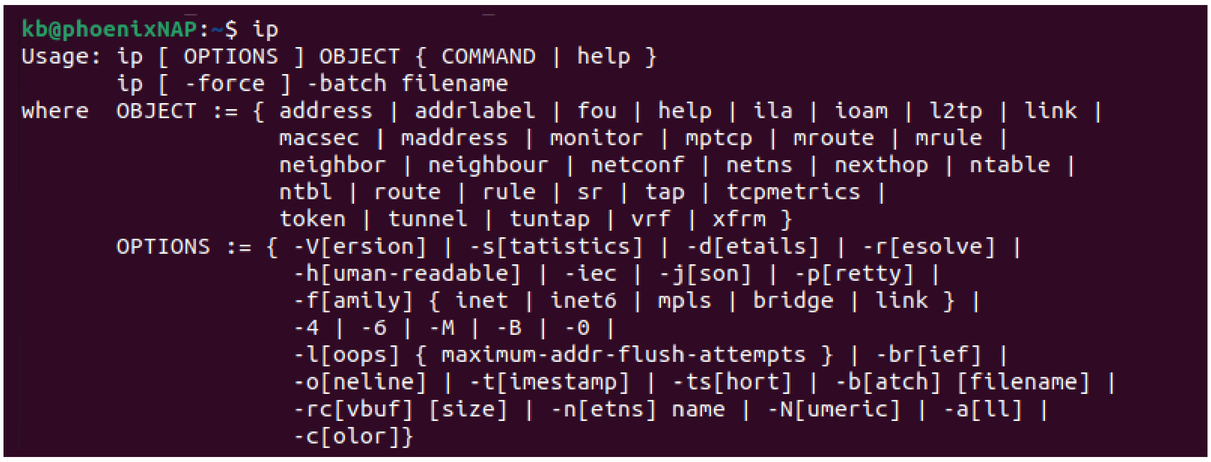
Each part of the command does the following:

- **[options]** are the command-line parameters that modify the command's behavior.
- **object** represents the available objects for configuration.
- **[command]** is a subcommand, an action performed on an object. The available commands differ depending on the object.

Example

The **ip** command shows the help menu when used without any options, objects, or commands:

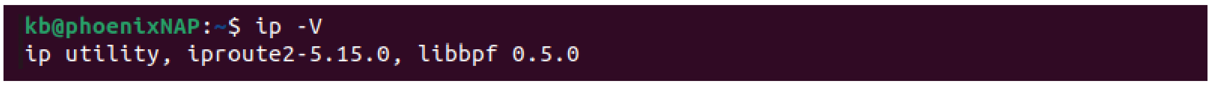
ip



```
kb@phoenixNAP:~$ ip
Usage: ip [ OPTIONS ] OBJECT { COMMAND | help }
       ip [ -force ] -batch filename
where  OBJECT := { address | addrlabel | fou | help | ila | ioam | l2tp | link |
                   macsec | maddress | monitor | mptcp | mroute | mrule |
                   neighbor | neighbour | netconf | netns | nexthop | ntable |
                   ntbl | route | rule | sr | tap | tcpmetrics |
                   token | tunnel | tuntap | vrf | xfrm }
       OPTIONS := { -V[ersion] | -s[tatistics] | -d[etails] | -r[esolve] |
                    -h[uman-readable] | -iec | -j[son] | -p[retty] |
                    -f[amily] { inet | inet6 | mpls | bridge | link } |
                    -4 | -6 | -M | -B | -0 |
                    -l[oops] { maximum-addr-flush-attempts } | -br[ief] |
                    -o[neline] | -t[imestamp] | -ts[hort] | -b[atch] [filename] |
                    -rc[vbuf] [size] | -n[etns] name | -N[umeric] | -a[ll] |
                    -c[olor]}
```

Add the **-V** option to see the current version:

ip -V



```
kb@phoenixNAP:~$ ip -V
ip utility, iproute2-5.15.0, libbpf 0.5.0
```

The output prints the package and library version for the **ip** utility.

ip addr

The **ip addr** command manages and shows network interface IP addresses. The command aliases are **ip address** or **ip a**.

Syntax

The syntax for the **ip addr** command is:

ip addr [subcommand]

The available subcommands on the object are:

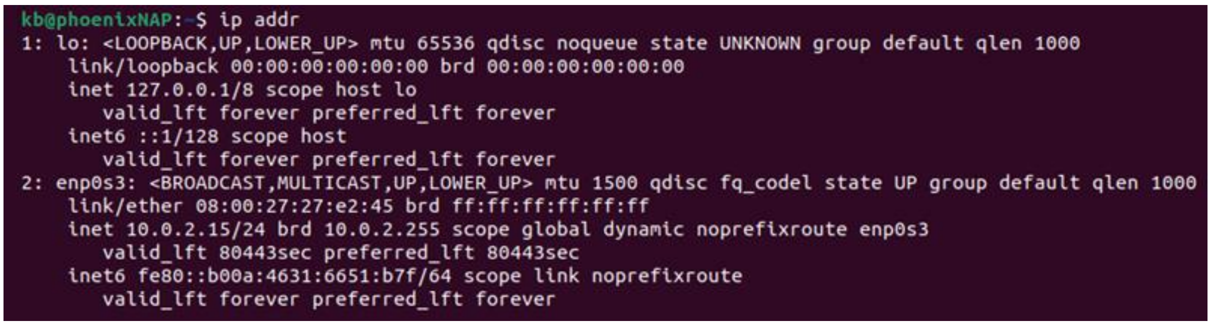
- **add** - Adds a new address.
- **show** - Shows protocol addresses.
- **del** - Removes an address.
- **flush** - Flushes addresses based on specified criteria.

Every subcommand has additional options and keywords to perform specific tasks for the network interface addresses.

Example

The **ip addr** command without any subcommands shows the network interface information, including the associated IP addresses:

ip addr

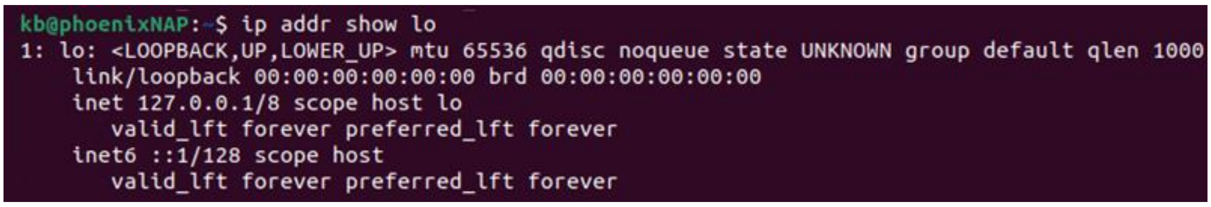


```
kb@phoenixNAP:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:27:e2:45 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 80443sec preferred_lft 80443sec
    inet6 fe80::b00a:4631:6651:b7f/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

The output for **ip addr show** is identical.

To show a specific network interface, use the **ip addr show** subcommand and add the interface name. For example:

ip addr show [interface]



```
kb@phoenixNAP:~$ ip addr show lo
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
```

The command filters the **ip addr** output and shows only information relevant to the specified interface.

ip link

The **ip link** command manages and shows network interface information. It allows viewing, changing, enabling, and disabling network interfaces.

Syntax

The syntax for the command is:

ip link [subcommand] [options] [interfaces]

The subcommands enable the following actions:

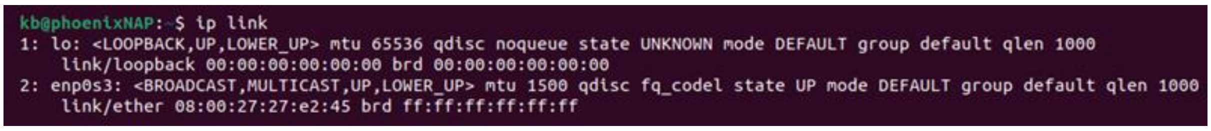
- **show** - Prints network interface information.
- **set** - Changes or adds information to a network interface.
- **add** - Adds a new network interface.
- **del** - Deletes a network interface.

Subcommands have additional options and allow targeting specific interfaces.

Example

The **ip link** command without any additional subcommands and options shows all network interface link information:

ip link

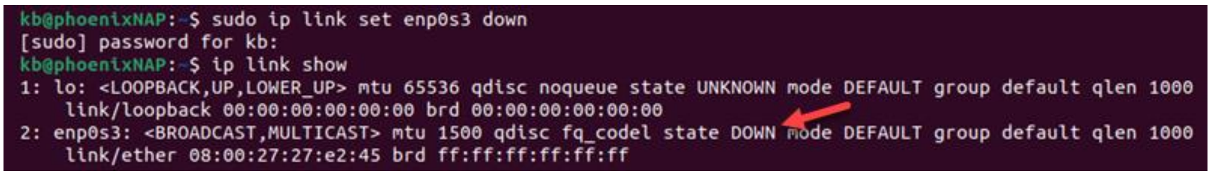


```
kb@phoenixNAP:~$ ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:27:e2:45 brd ff:ff:ff:ff:ff:ff
```

The **ip link show** command provides the same output.

To turn off an interface, use the following syntax as a superuser:

sudo ip link set [interface] down

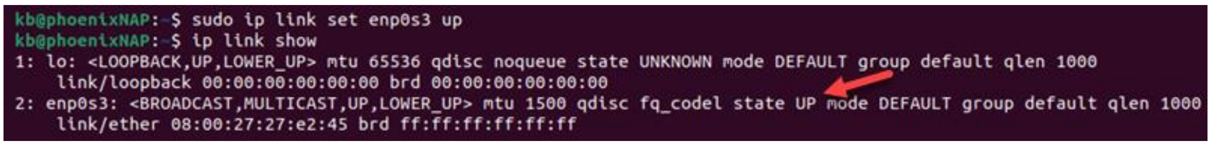


```
kb@phoenixNAP:~$ sudo ip link set enp0s3 down
[sudo] password for kb:
kb@phoenixNAP:~$ ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel state DOWN mode DEFAULT group default qlen 1000
    link/ether 08:00:27:27:e2:45 brd ff:ff:ff:ff:ff:ff
```

The interface shows the state as **DOWN** after executing the command.

Similarly, to disable an interface, use the **up** keyword:

sudo ip link set [interface] up



```
kb@phoenixNAP:~$ sudo ip link set enp0s3 up
kb@phoenixNAP:~$ ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:27:e2:45 brd ff:ff:ff:ff:ff:ff
```

The interface state changes to **UP**.

ip route

The **ip route** command shows and configures the IP routing table. The command allows users to adjust the routing table and perform other crucial networking tasks with the routing table.

Syntax

The command follows a specific syntax, as shown below:

ip route [subcommand] [options] [destination]

The following actions are available as subcommands:

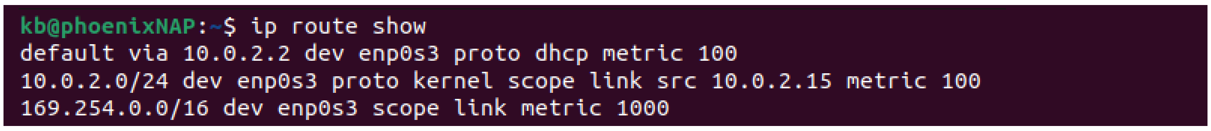
- **show** - Shows the routing table.
- **add** - Adds a new route to the table.
- **del** - Deletes a route from the table.
- **change** - Modifies an existing route.

The **[destination]** parameter determines where the network traffic is directed. Additional options help control the traffic flow further.

Example

To view the routing table, run the following command:

ip route show


```
kb@phoenixNAP:~$ ip route show
default via 10.0.2.2 dev enp0s3 proto dhcp metric 100
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15 metric 100
169.254.0.0/16 dev enp0s3 scope link metric 1000
```

Each line in the output represents individual routes in the table.

Note: For additional options on managing network interfaces, read about ifdown command.

3. **ifconfig**

The ifconfig (**i**nter**f**ace **config**uration) command manages and shows network interface information on a system. The command is part of the **net-tools** package.

Although the command has limited functions compared to the **ip** command, the **ifconfig** command is still commonly used for configuring network interfaces.

Syntax

The syntax for the command is:

ifconfig [interface] [options]

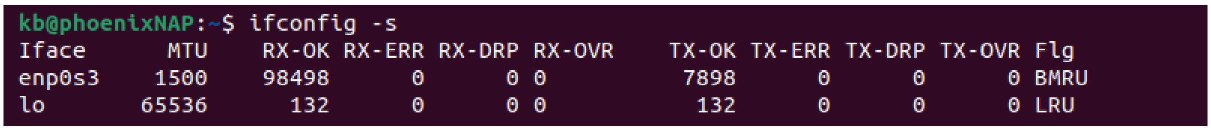
The syntax breaks down into the following:

- **[interface]** - The network interface to configure or show information for. The parameter is optional, and not specifying an interface shows the status of all active interfaces.
- **[options]** - Command-line options to perform specific actions or configure certain parameters. The parameter is also optional.

Example

To display the summary of all active network interfaces, run:

ifconfig -s


```
kb@phoenixNAP:~$ ifconfig -s
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
enp0s3    1500    98498      0      0 0          7898      0      0      0 BMRU
lo       65536      132      0      0 0           132      0      0      0 LRU
```

The command prints a shortlist with crucial information about active interfaces.

4. **dig**

The dig command queries Domain Name Systems (DNS) and finds information for DNS records. The command collects domain name information and associated records.

Use **dig** to troubleshoot DNS issues and to verify DNS configuration on a Linux system. It is suitable for creating scripts and automating tasks related to network troubleshooting. The robust command is so prevalent in network troubleshooting that a Windows version of dig is available.

Syntax

The **dig** command syntax is as follows:

dig [options] [domain] [record type] [DNS server]

The components of the command are:

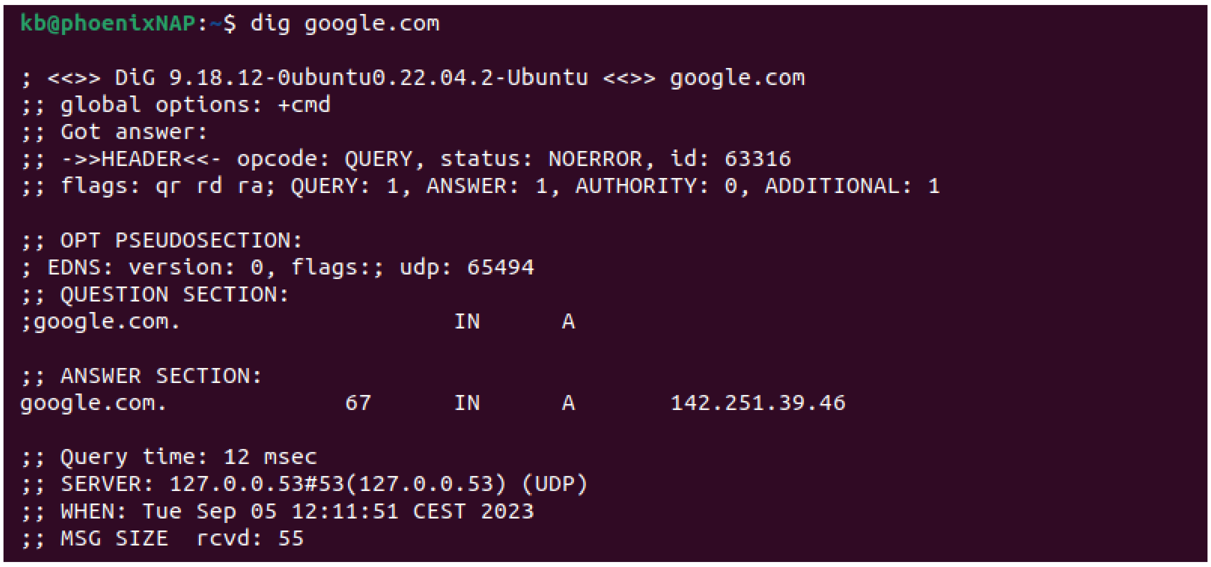
- **[options]** - Parameters that modify the behavior of the command.
- **[domain]** - The domain name to query.
- **[record type]** - The DNS record type to query. Defaults to A records.
- **[DNS server]** - A specified DNS server for the query.

All parameters are optional. The command shows the default DNS resolver information and query statistics without additional options.

Example

To perform a simple DNS lookup, run the command with a domain name:

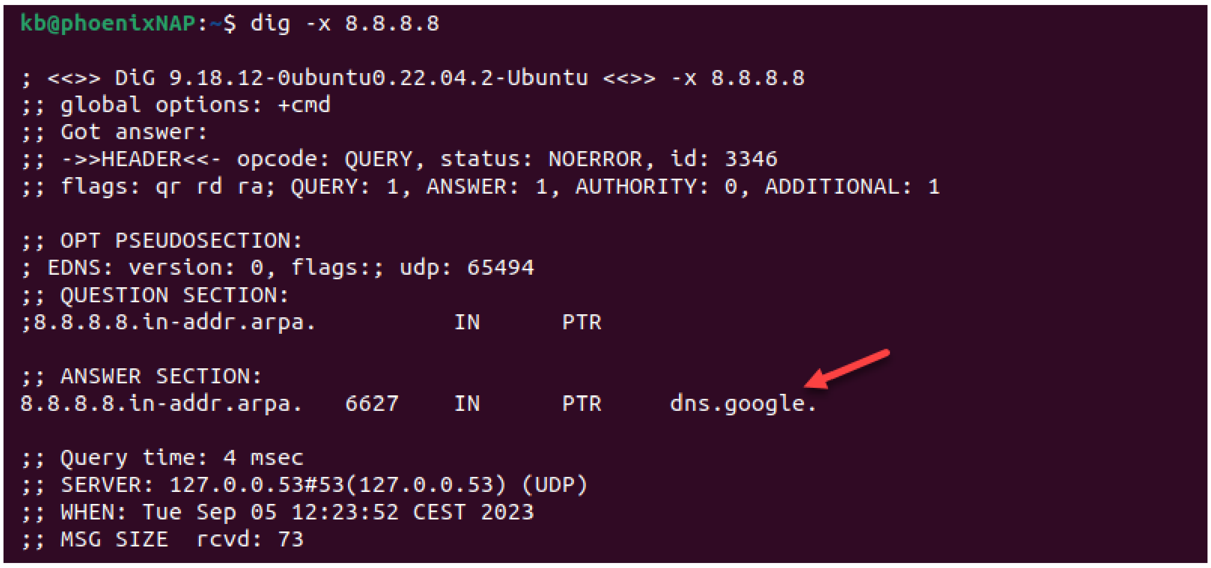
dig google.com



```
kb@phoenixNAP:~$ dig google.com

; <<>> DiG 9.18.12-0ubuntu0.22.04.2-Ubuntu <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63316
;; 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.             67      IN      A       142.251.39.46

;; Query time: 12 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Tue Sep 05 12:11:51 CEST 2023
;; MSG SIZE  rcvd: 55
```

Alternatively, provide the IP address and the **-x** option to perform a reverse DNS lookup. For example:

dig -x 8.8.8.8



```
kb@phoenixNAP:~$ dig -x 8.8.8.8

; <<>> DiG 9.18.12-0ubuntu0.22.04.2-Ubuntu <<>> -x 8.8.8.8
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3346
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;8.8.8.8.in-addr.arpa.          IN      PTR

;; ANSWER SECTION:
8.8.8.8.in-addr.arpa.   6627    IN      PTR     dns.google.

;; Query time: 4 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Tue Sep 05 12:23:52 CEST 2023
;; MSG SIZE  rcvd: 73
```

The **ANSWER SECTION** in the output shows the requested domain name.

Note: Excessive DNS lookups impact website performance. Reducing DNS lookups lowers server load and network latency.

5. **nslookup**

The nslookup command is similar to the **dig** command. The main difference between the two commands is that **nslookup** features an interactive mode. It enables diagnosing and querying DNS servers, which is helpful for network troubleshooting and DNS tasks.

The command is available for most Unix-like and Windows operating systems.

Syntax

The general syntax for the **nslookup** command is:

nslookup [domain] [DNS server]

The command components are:

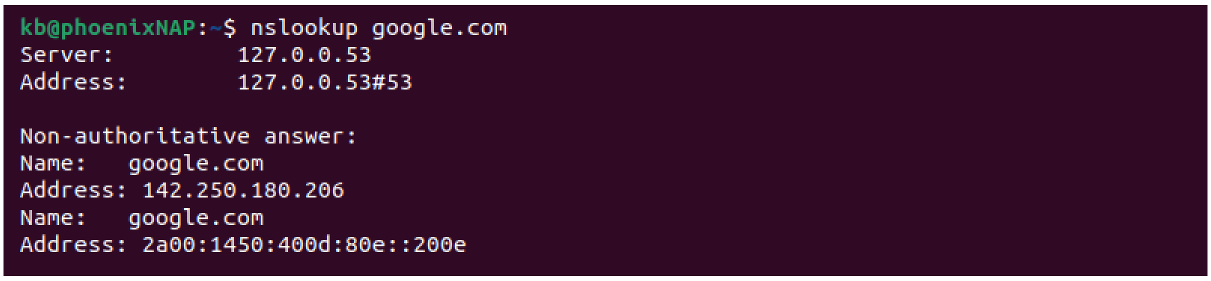
- **[domain]** - The domain name to look up. Not specifying a name enables querying multiple domains in interactive mode.
- **[DNS server]** - The DNS server to use for the lookup. Defaults to the system DNS server when left out.

The query performs A record domain lookups by default.

Example

The following example shows how to perform a DNS lookup for a domain:

nslookup google.com



```
kb@phoenixNAP:~$ nslookup google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.180.206
Name:   google.com
Address: 2a00:1450:400d:80e::200e
```

The output shows the DNS resolution information for the provided domain.

6. **netstat**

The netstat command (**net**work **stat**istics) is a networking utility that shows various networking statistics. The command provides statistics for network ports and shows port availability.

The command is part of the **net-tools** package and is considered obsolete. The recommended replacement is the **ss** command, which is part of **iproute2**. Other functionalities of the **netstat** command are available with the **ip** command.

Syntax

The syntax for the **netstat** command is simple:

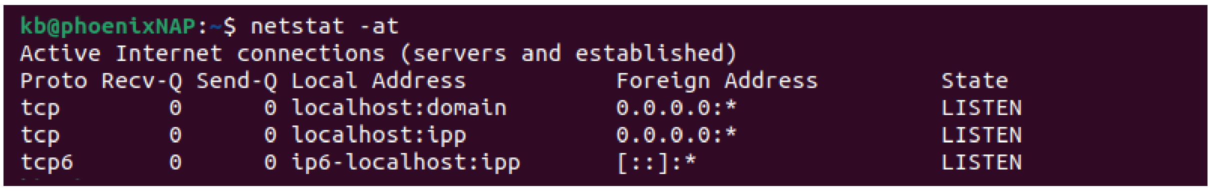
netstat [options]

The command allows combining various options to customize the output and to show specific network information types. The command lists open sockets for all configured address families without any options.

Example

For example, to list all TCP ports with the **netstat** command, use the **-at** options:

netstat -at



```
kb@phoenixNAP:~$ netstat -at
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN
tcp6       0      0 ip6-localhost:ipp       [::]:*                  LISTEN
```

The output shows all active TCP connections on the system.

7. traceroute

The traceroute command is a networking diagnostics tool available for Linux, macOS, and Windows. The command tracks the route that packets take to reach a destination on a TCP/IP network.

Use the command to discover routing issues and bottlenecks by showing a packet's intermediate hops while traveling from source to destination.

The default trace is 30 hops with a packet size of 60 bytes for IPv4 (80 bytes for IPv6).

Syntax

The syntax for the **traceroute** command is:

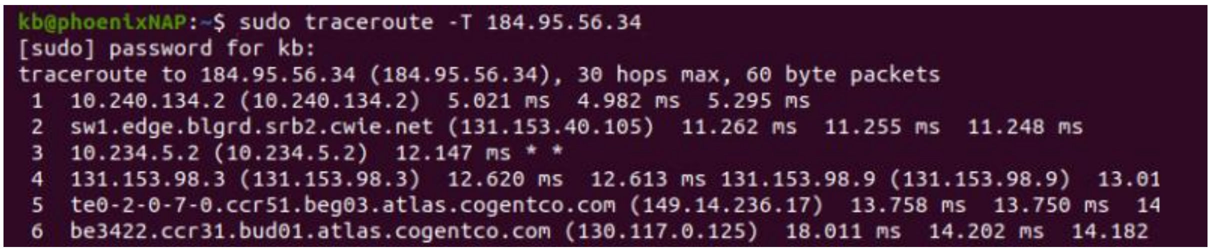
traceroute [options] [hostname/IP]

The **[hostname/IP]** parameter is required, while additional options control whether to perform DNS lookups, the TTL parameter, and the packet type.

Example

To trace a packet route using the TCP protocol, run the **traceroute** command as an administrator with the **-T** option. For example:

sudo traceroute -T 184.95.56.34



```
kb@phoenixNAP:~$ sudo traceroute -T 184.95.56.34
[sudo] password for kb:
traceroute to 184.95.56.34 (184.95.56.34), 30 hops max, 60 byte packets
 1  10.240.134.2 (10.240.134.2)  5.021 ms  4.982 ms  5.295 ms
 2  sw1.edge.blgrd.srb2.cwie.net (131.153.40.105)  11.262 ms  11.255 ms  11.248 ms
 3  10.234.5.2 (10.234.5.2)  12.147 ms * *
 4  131.153.98.3 (131.153.98.3)  12.620 ms  12.613 ms 131.153.98.9 (131.153.98.9)  13.01
 5  te0-2-0-7-0.ccr51.beg03.atlas.cogentco.com (149.14.236.17)  13.758 ms  13.750 ms  14
 6  be3422.ccr31.bud01.atlas.cogentco.com (130.117.0.125)  18.011 ms  14.202 ms  14.182
```

The output shows the sequential route from source to destination.

8. tracepath

The **tracepath** command is similar to the **traceroute** command. The command identifies paths and latencies from source to destination, mapping the router and network hops.

Although **traceroute** is a well-known command with comprehensive options, the **tracepath** command is a simple network mapping tool available on most Linux systems. For more details, see the comparison between tracepath and traceroute.

Syntax

The syntax for the **tracepath** command is:

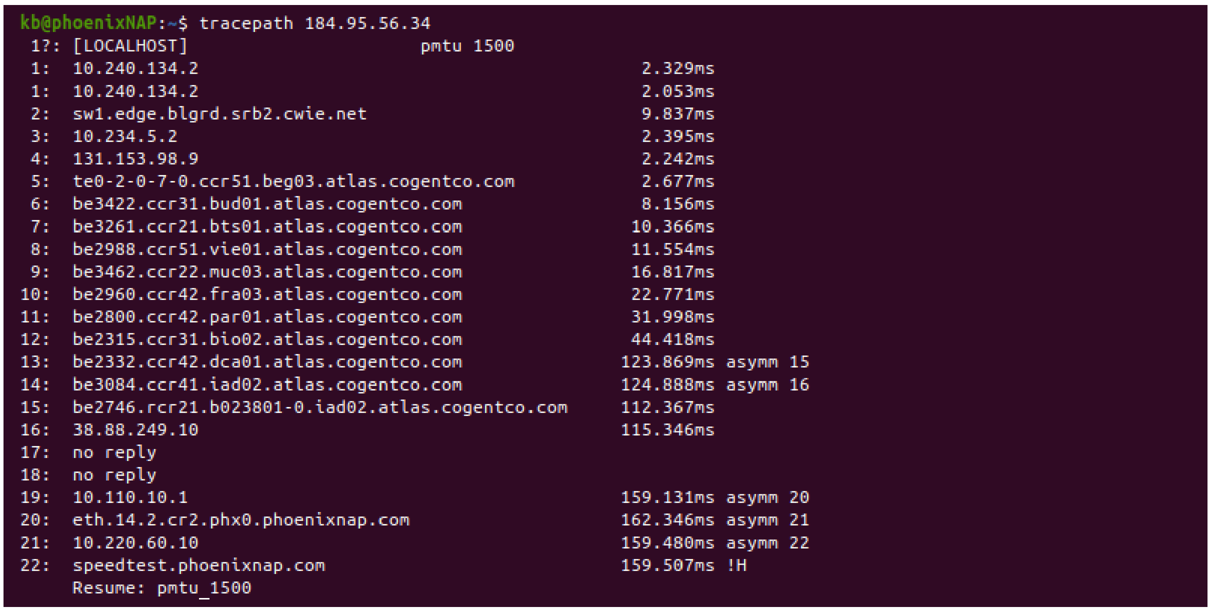
tracepath [options] [hostname/IP]

The additional **[options]** control the query behavior, such as the number of hops and whether to perform a reverse DNS lookup for the addresses. The **[hostname/IP]** field is required and represents the destination.

Example

Run the **tracepath** command without any options to perform a simple trace from destination to host:

tracepath [hostname/IP]



```
kb@phoenixNAP:~$ tracepath 184.95.56.34
 1?: [LOCALHOST]                      pmtu 1500
 1:  10.240.134.2                                          2.329ms
 1:  10.240.134.2                                          2.053ms
 2:  sw1.edge.blgrd.srb2.cwie.net                          9.837ms
 3:  10.234.5.2                                            2.395ms
 4:  131.153.98.9                                          2.242ms
 5:  te0-2-0-7-0.ccr51.beg03.atlas.cogentco.com            2.677ms
 6:  be3422.ccr31.bud01.atlas.cogentco.com                 8.156ms
 7:  be3261.ccr21.bts01.atlas.cogentco.com                10.366ms
 8:  be2988.ccr51.vie01.atlas.cogentco.com                11.554ms
 9:  be3462.ccr22.muc03.atlas.cogentco.com                16.817ms
10:  be2960.ccr42.fra03.atlas.cogentco.com                22.771ms
11:  be2800.ccr42.par01.atlas.cogentco.com                31.998ms
12:  be2315.ccr31.bio02.atlas.cogentco.com                44.418ms
13:  be2332.ccr42.dca01.atlas.cogentco.com               123.869ms asymm 15
14:  be3084.ccr41.iad02.atlas.cogentco.com               124.888ms asymm 16
15:  be2746.rcr21.b023801-0.iad02.atlas.cogentco.com     112.367ms
16:  38.88.249.10                                        115.346ms
17:  no reply
18:  no reply
19:  10.110.10.1                                         159.131ms asymm 20
20:  eth.14.2.cr2.phx0.phoenixnap.com                    162.346ms asymm 21
21:  10.220.60.10                                        159.480ms asymm 22
22:  speedtest.phoenixnap.com                            159.507ms !H
     Resume: pmtu_1500
```

The output shows the hop number, IP address or resolved hostname, and the round-trip time (RTT) for each hop.

9. host

The host command is a simple tool for performing DNS lookups. The command resolves IP addresses into domain names and vice versa.

Use the command to perform a query for DNS records and basic DNS troubleshooting.

Syntax

The syntax for the **host** command is:

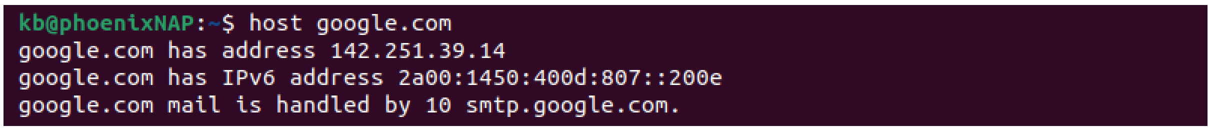
host [options] [hostname/IP]

The various **[options]** control the command's behavior, such as the query type or the start of authority (SOA) for the provided domain.

Example

To perform a simple DNS lookup, use the **host** command and provide a hostname or IP address. For example:

host google.com



```
kb@phoenixNAP:~$ host google.com
google.com has address 142.251.39.14
google.com has IPv6 address 2a00:1450:400d:807::200e
google.com mail is handled by 10 smtp.google.com.
```

The output shows the resolved IPv4 and IPv6 addresses for the provided hostname.

10. hostname

The hostname command helps display and change a system's hostname and domain and identifies devices within a network environment.

Use the command to display, change, or search for hostnames.

Syntax

The syntax for the **hostname** command is:

hostname [options] [name]

The **[options]** parameter control what the command displays, while the **[name]** parameter temporarily sets the hostname to the provided name.

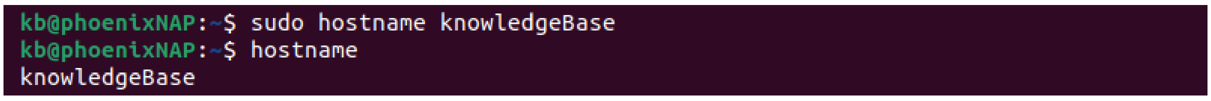
Example

To temporarily change the system hostname, run the command without any options and provide a name:

sudo hostname [name]

The command does not produce an output. Check the current hostname by running:

hostname



```
kb@phoenixNAP:~$ sudo hostname knowledgeBase
kb@phoenixNAP:~$ hostname
knowledgeBase
```

The current hostname prints to the screen.

11. ping

The ping command is a network utility for testing whether a host is reachable. The command sends ICMP requests to a host (a computer or server) and measures the round-trip time (RTT).

Pinging helps determine the network latency between two nodes and whether a network is reachable.

Syntax

The syntax for the **ping** command is:

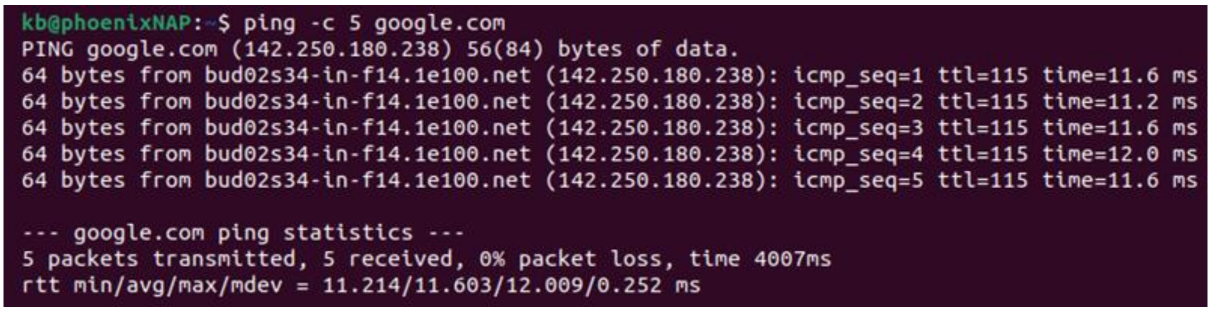
ping [options] [hostname/IP]

State the **[hostname/IP]** of the host to ping. Add options to control the command's behavior, such as the ping request number, intervals, or packet size.

Example

An example **ping** command request looks like the following:

ping -c 5 google.com



```
kb@phoenixNAP:~$ ping -c 5 google.com
PING google.com (142.250.180.238) 56(84) bytes of data.
64 bytes from bud02s34-in-f14.1e100.net (142.250.180.238): icmp_seq=1 ttl=115 time=11.6 ms
64 bytes from bud02s34-in-f14.1e100.net (142.250.180.238): icmp_seq=2 ttl=115 time=11.2 ms
64 bytes from bud02s34-in-f14.1e100.net (142.250.180.238): icmp_seq=3 ttl=115 time=11.6 ms
64 bytes from bud02s34-in-f14.1e100.net (142.250.180.238): icmp_seq=4 ttl=115 time=12.0 ms
64 bytes from bud02s34-in-f14.1e100.net (142.250.180.238): icmp_seq=5 ttl=115 time=11.6 ms

--- google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 11.214/11.603/12.009/0.252 ms
```

The command sends five ICMP packets to the provided host and prints the statistics.

12. ss

The ss command is a CLI tool for displaying network statistics. The tool is part of the **iproute2** package and is a faster alternative to the **netstat** command.

Use the **ss** command to examine network sockets and view various network-related data.

Syntax

The basic syntax for the command is:

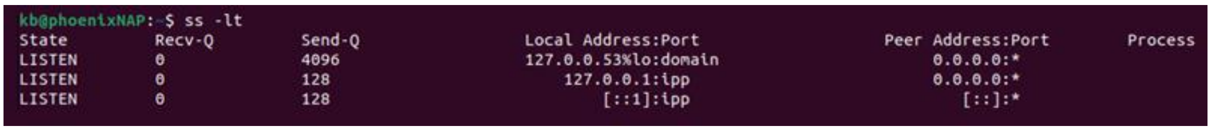
ss [options] [filter]

The **[options]** parameter allows filtering sockets by protocol, while the **[filter]** parameter helps queue sockets by state to narrow down the result view.

Example

For example, to show all listening TCP sockets using the **ss** command, add the **-lt** options:

ss -lt



```
kb@phoenixNAP:~$ ss -lt
State      Recv-Q      Send-Q           Local Address:Port          Peer Address:Port      Process
LISTEN     0           4096           127.0.0.53%lo:domain               0.0.0.0:*
LISTEN     0           128                127.0.0.1:ipp                  0.0.0.0:*
LISTEN     0           128                    [::1]:ipp                     [::]:*
```

The output shows all TCP sockets in the **LISTEN** state waiting for incoming connections.

13. route

The **route** command in Linux is a specialized command for displaying and configuring the routing table. The command modifies the kernel's IP routing tables and helps set up static routes to specific hosts or networks.

Use the command after configuring a network interface with a tool such as the **ifconfig** command.

Note: The preferable alternative to the **route** command is the **ip route** command.

Syntax

The syntax for the **route** command is:

route [options] [subcommand] [arguments]

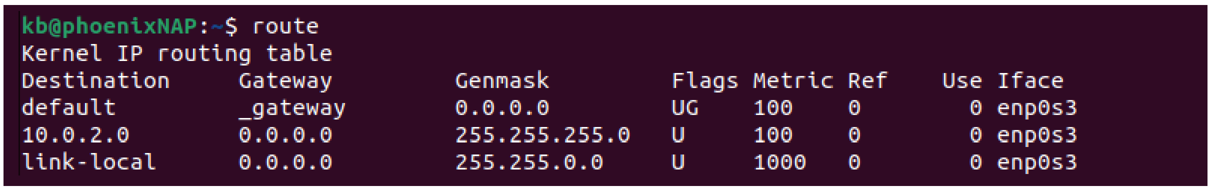
It contains the following components:

- **[options]** - Optional command-line parameters that control the output view, address family, and IP protocol.
- **[subcommand]** - An action to perform, such as **add** or **delete**.
- **[arguments]** - Additional arguments that differ depending on the subcommand.

Example

To view the current routing table, use the **route** command without any options:

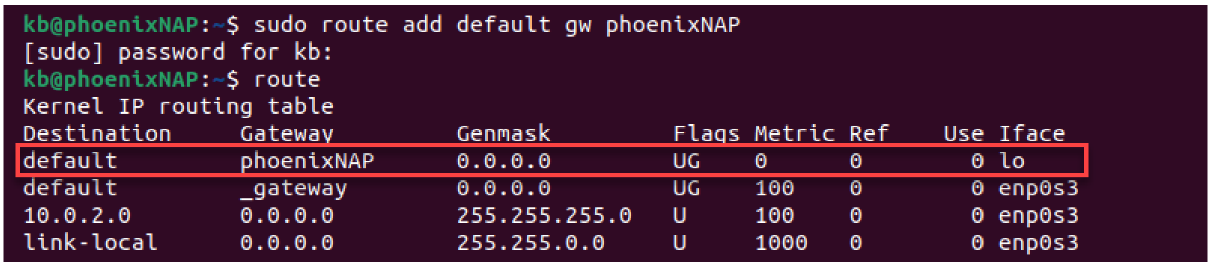
route



```
kb@phoenixNAP:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    100    0        0 enp0s3
10.0.2.0        0.0.0.0         255.255.255.0   U     100    0        0 enp0s3
link-local      0.0.0.0         255.255.0.0     U     1000   0        0 enp0s3
```

Use the following format to add a default gateway:

sudo route add default gw [gateway]



```
kb@phoenixNAP:~$ sudo route add default gw phoenixNAP
[sudo] password for kb:
kb@phoenixNAP:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         phoenixNAP      0.0.0.0         UG    0      0        0 lo
default         _gateway        0.0.0.0         UG    100    0        0 enp0s3
10.0.2.0        0.0.0.0         255.255.255.0   U     100    0        0 enp0s3
link-local      0.0.0.0         255.255.0.0     U     1000   0        0 enp0s3
```

The command adds a default route, which is used when no other routes match. The provided gateway must be a directly reachable route.

14. arp

The **arp** command shows and configures the Address Resolution Protocol (ARP) cache. The ARP protocol maps IP addresses to physical Media Access Control (MAC) addresses in a local network. The cache stores these mappings for all devices on the local network.

Syntax

The syntax for the **arp** command is in the following format:

arp [options] [hostname/IP]

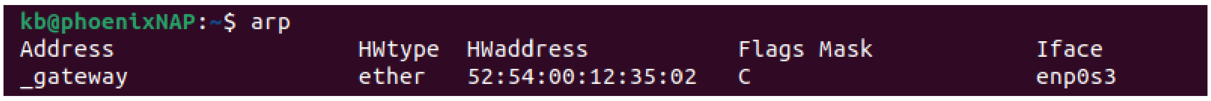
- The **[options]** parameter modifies the command's behavior, such as setting up and deleting actions, or controlling the output.

- The **[hostname/IP]** parameter is an optional identifier for a remote system for which to resolve a MAC address. If unprovided, the command checks the local ARP cache.

Example

To display the ARP cache, run the **arp** command without any additional parameters:

arp



```
kb@phoenixNAP:~$ arp
Address                  HWtype  HWaddress           Flags Mask            Iface
_gateway                 ether   52:54:00:12:35:02   C                     enp0s3
```

The output shows the ARP cache (IP and MAC addresses) in a table.

15. iwconfig

The **iwconfig** command shows and configures wireless network interface information. The command comes in handy for troubleshooting wireless network issues.

Use the command to view or change a wireless network's name, power management settings, and other wireless configurations.

Syntax

The syntax for the **iwconfig** command is:

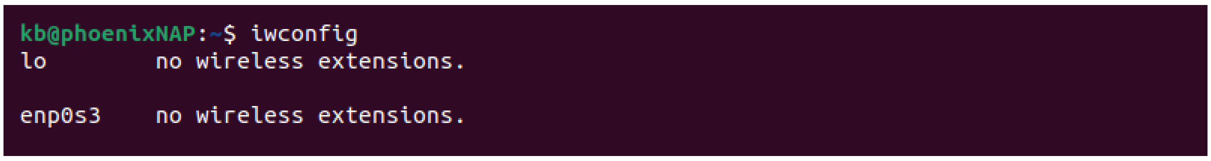
iwconfig [interface] [options]

The **[interface]** parameter filters the wireless network interface by name, whereas the **[options]** parameter controls various settings, such as the operation mode, rate limits, and the wireless encryption key.

Example

To view the available wireless interfaces on the system and the current setup, run the command without any parameters:

iwconfig



```
kb@phoenixNAP:~$ iwconfig
lo        no wireless extensions.

enp0s3    no wireless extensions.
```

The command shows all information on wireless interfaces on the system.

16. curl or wget

The **wget** and **curl** commands are command-line tools for downloading files from the internet. The two tools are similar, but there are slight differences in how they work and the options they offer:

- The wget command downloads files from the web using HTTP, HTTPS, or FTP protocols. The tool is simple to use for file downloads.
- The curl command is versatile and supports various network protocols, such as SCP, IMAP POP3, SMTP, etc. The tool also sends HTTP requests and interacts with web services.

Use **curl** or **wget** to test network download speeds.

Syntax

The syntaxes for the **wget** and **curl** commands are similar:

wget [options] [URL]

curl [options] [URL]

The **[options]** parameter controls the various download and output options, while the **[URL]** parameter is a file's download URL. The **curl** command features many advanced options and usage patterns compared to the **wget** command.

Example

To download a file using the **wget** command, use the following format:

wget -O [file name] [URL]

Alternatively, to use **curl** to achieve the same task, run:

curl -o [file name] [URL]

The file downloads from the specified URL and saves the contents to the provided file name.

17. mtr

The **mtr** command (**m**y **tr**ace**r**oute) is a diagnostics tool that combines elements from the **ping** and **traceroute** commands. The command sends real-time insights into network quality, making it an excellent tool for troubleshooting high latency and packet loss.

Syntax

The syntax for the **mtr** command is:

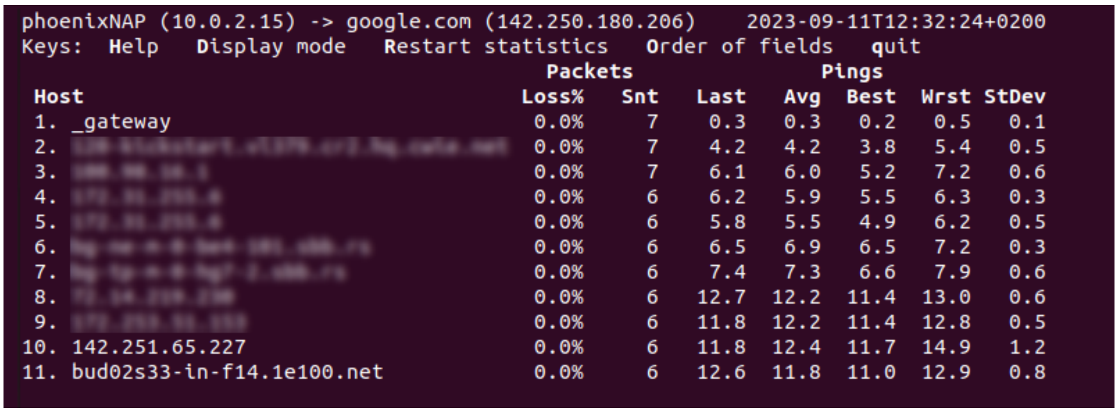
mtr [options] [hostname/IP]

The **[options]** parameter controls the packet number and size, while the **[hostname/IP]** parameter contains the destination.

Example

The **mtr** command, without any parameters, starts a trace session to the provided host. For example:

mtr google.com



```
phoenixNAP (10.0.2.15) -> google.com (142.250.180.206)    2023-09-11T12:32:24+0200
Keys:  Help   Display mode   Restart statistics   Order of fields   quit
                                       Packets               Pings
 Host                                Loss%   Snt   Last   Avg  Best  Wrst StDev
 1. _gateway                          0.0%     7    0.3   0.3   0.2   0.5   0.1
 2. [redacted]                        0.0%     7    4.2   4.2   3.8   5.4   0.5
 3. [redacted]                        0.0%     7    6.1   6.0   5.2   7.2   0.6
 4. [redacted]                        0.0%     6    6.2   5.9   5.5   6.3   0.3
 5. [redacted]                        0.0%     6    5.8   5.5   4.9   6.2   0.5
 6. [redacted]                        0.0%     6    6.5   6.9   6.5   7.2   0.3
 7. [redacted]                        0.0%     6    7.4   7.3   6.6   7.9   0.6
 8. [redacted]                        0.0%     6   12.7  12.2  11.4  13.0   0.6
 9. [redacted]                        0.0%     6   11.8  12.2  11.4  12.8   0.5
10. 142.251.65.227                    0.0%     6   11.8  12.4  11.7  14.9   1.2
11. bud02s33-in-f14.1e100.net         0.0%     6   12.6  11.8  11.0  12.9   0.8
```

To exit the window, press **q**.

18. whois

The **whois** command queries information about domain names, IP addresses, and other network-related information. Use the command to fetch domain ownership details, such as the domain's ownership details, registration date, and expiration date.

Syntax

The syntax for the **whois** command is:

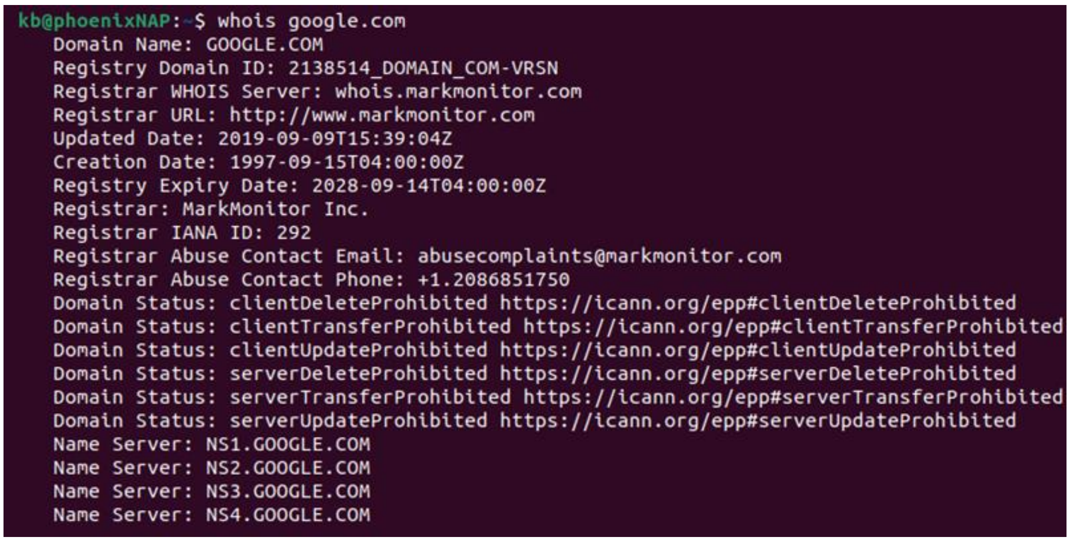
whois [options] [query]

- The **[options]** parameter allows setting a specific WHOIS server to query, changing the protocol, and adding additional query parameters.
- The **[query]** parameter is the domain name, IP address, or Autonomous System Number (ASN) to look up.

Example

Run the command without any options to perform a simple query for a given domain name. For example:

whois google.com



```
kb@phoenixNAP:~$ whois google.com
   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: +1.2086851750
   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
```

The output shows the results of the basic WHOIS lookup for the provided domain name.

19. iftop

The **iftop** command is a network monitoring utility. Use the command to view network connections and bandwidth usage in real time.

Syntax

The syntax for the **iftop** command is:

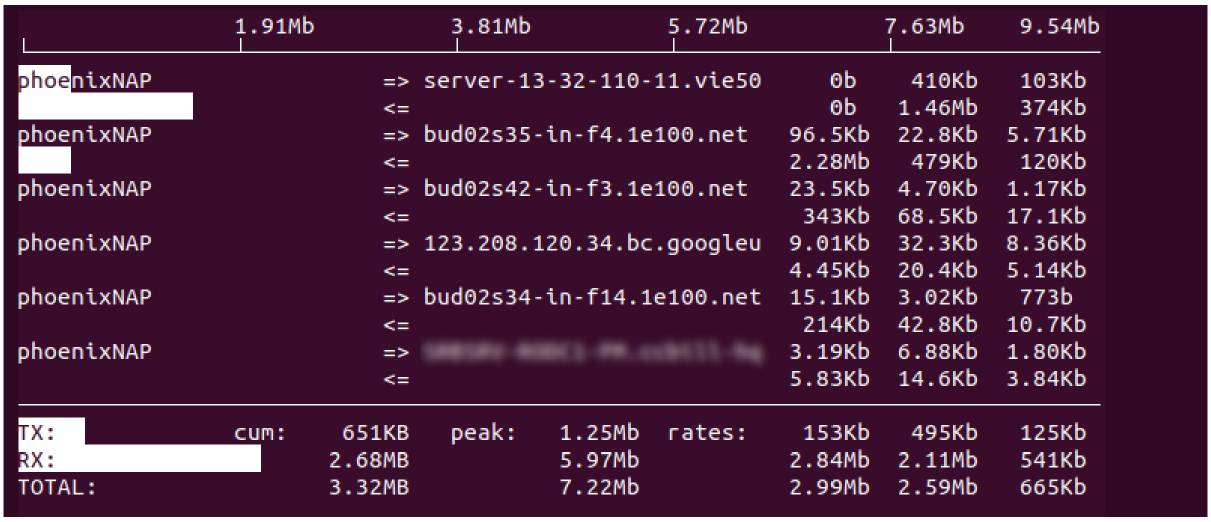
iftop [options]

The **[options]** parameter controls the display information. The command also requires sufficient privileges to monitor all traffic on the network interface.

Example

The primary usage of **iftop** is without any additional options:

sudo iftop



```
                1.91Mb          3.81Mb          5.72Mb          7.63Mb    9.54Mb
└───────────────┴───────────────┴───────────────┴───────────────┴─────────
phoenixNAP                  => server-13-32-110-11.vie50     0b    410Kb   103Kb
[redacted]                  <=                               0b   1.46Mb   374Kb
phoenixNAP                  => bud02s35-in-f4.1e100.net   96.5Kb  22.8Kb  5.71Kb
[redacted]                  <=                            2.28Mb   479Kb   120Kb
phoenixNAP                  => bud02s42-in-f3.1e100.net   23.5Kb  4.70Kb  1.17Kb
                            <=                             343Kb  68.5Kb  17.1Kb
phoenixNAP                  => 123.208.120.34.bc.googleu  9.01Kb  32.3Kb  8.36Kb
                            <=                            4.45Kb  20.4Kb  5.14Kb
phoenixNAP                  => bud02s34-in-f14.1e100.net  15.1Kb  3.02Kb   773b
                            <=                             214Kb  42.8Kb  10.7Kb
phoenixNAP                  => [redacted]                 3.19Kb  6.88Kb  1.80Kb
                            <=                            5.83Kb  14.6Kb  3.84Kb
──────────────────────────────────────────────────────────────────────────────
TX: [redacted]      cum:   651KB   peak:   1.25Mb  rates:  153Kb   495Kb   125Kb
RX: [redacted]            2.68MB           5.97Mb         2.84Mb  2.11Mb   541Kb
TOTAL:                    3.32MB           7.22Mb         2.99Mb  2.59Mb   665Kb
```

The command opens a new monitoring screen, which changes as data transfers via the network interface.

The interface allows controlling the display from the monitoring screen, such as toggling the source (**s**) or destination (**d**) views. To exit the screen, press **q**.

20. tcpdump

The **tcpdump** command is a packet sniffer and network security tool that captures real-time network packet information. Use the command to analyze traffic, troubleshoot issues, and monitor network security.

Syntax

The syntax for the **tcpdump** command is:

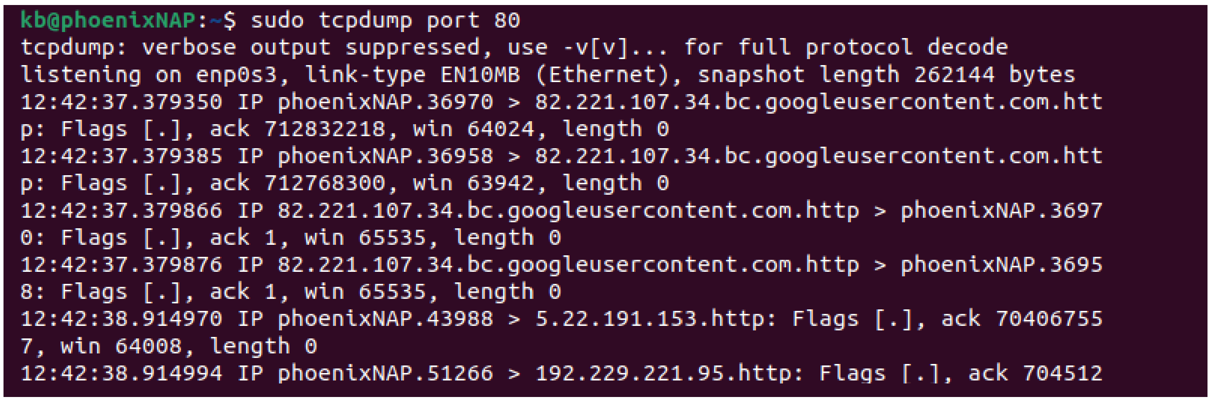
tcpdump [options] [filter]

The **[options]** parameter handles various display options, controls the packet number, and enables working with files. Use the [filter] parameter to enter the criteria for packet capturing.

Example

To capture packets on a specific port, use the following format:

sudo tcpdump port 80



```
kb@phoenixNAP:~$ sudo tcpdump port 80
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
12:42:37.379350 IP phoenixNAP.36970 > 82.221.107.34.bc.googleusercontent.com.htt
p: Flags [.], ack 712832218, win 64024, length 0
12:42:37.379385 IP phoenixNAP.36958 > 82.221.107.34.bc.googleusercontent.com.htt
p: Flags [.], ack 712768300, win 63942, length 0
12:42:37.379866 IP 82.221.107.34.bc.googleusercontent.com.http > phoenixNAP.3697
0: Flags [.], ack 1, win 65535, length 0
12:42:37.379876 IP 82.221.107.34.bc.googleusercontent.com.http > phoenixNAP.3695
8: Flags [.], ack 1, win 65535, length 0
12:42:38.914970 IP phoenixNAP.43988 > 5.22.191.153.http: Flags [.], ack 70406755
7, win 64008, length 0
12:42:38.914994 IP phoenixNAP.51266 > 192.229.221.95.http: Flags [.], ack 704512
```

The filter **port 80** captures packets on the specified port to monitor HTTP traffic.

21. ifplugstatus

The **ifplugstatus** command is a simple utility to check the network interface status. The command helps determine whether an ethernet cable is connected to an interface.

Use **ifplugstatus** to check a network's physical link, especially after changes to the network interface.

Syntax

The syntax for the **ifplugstatus** command is:

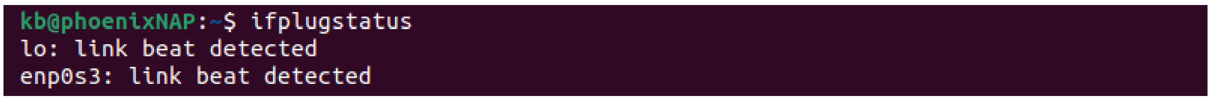
ifplugstatus [options] [interface]

The **[options]** parameter allows setting a specific configuration file or running in batch mode for scripting. State the **[interface]** parameter to check the status of the specified interface.

Example

To list the status for all network interfaces, run the command without any parameters:

ifplugstatus



```
kb@phoenixNAP:~$ ifplugstatus
lo: link beat detected
enp0s3: link beat detected
```

If the output states **link beat detected**, the interface has an active physical link.