

Exercise : 2

SOME IMPORTANT LINUX NETWORKING

Date : 14.7.25

COMMANDS

1. ip

`ip <OPTIONS> <OBJECT> <COMMAND>`

Here are some common use cases for the ip command.

a) `ip address show` : to show the IP addresses assigned to an interface on your server.

OUTPUT :

1. `lo : <LOOPBACK, UP, LOWER_UP> mtu 65536 qdisc`

`noqueue state unknown group default qlen 1000.`

2. `enp0s31f6 : <NO-CARRIER, BROADCAST, MULTICAST, UP>`
`mtu 1500 qdisc fq_codel state DOWN group default`
`qlen 1000`

3. `wlpas0 : <BROADCAST, MULTICAST, UP, LOWER_UP> mtu 1500`
`qdisc noqueue state UP group default qlen 1000`

b) `ip address add 192.168.1.254/24 dev enp0s31f6`
to assign an IP address to an interface

OUTPUT :

password for student :

Assigned an IP to an interface.

c) to delete an IP on an interface

`ip address del 192.168.1.254/24 dev enp0s31f6`

deleted an IP on an interface

d) `ip link set eth0 down`: Alter the status of the interface by bringing the interface eth0 offline.

e) `ip link set eth0 up`: Alter the status of the interface by bringing the interface online:

f) `ip link set eth0 promisc on`: Alter the status of the interface by enabling promiscuous mode for eth0:

g) `ip link set eth0`

Add a default route (for all addresses) via the local gateway 192.168.1.254 that can be reached on device eth0:

h) `ip route add 192.168.1.0/24 via 192.168.1.254`

Add a route to 192.168.1.0/24 via the gateway at 192.168.1.254:

i) `ip route add 192.168.1.0/24 via 192.168.1.254`

Add a route to 192.168.1.0/24 that can be reached on device eth0.

j) `ip route delete 192.168.1.0/24 via 192.168.1.254`

Delete the route for 192.168.1.0/24 via the gateway at 192.168.1.254.

k) ip route get 10.10.1.4

Display the route taken for IP 10.10.1.4

OUTPUT:

```
10.10.1.4 dev enp0s31f6 src 192.168.1.254  
uid 1000 cache
```

2) ifconfig

the ifconfig Command was /is a staple in many sysadmin's tool belt for configuration and troubleshooting networks. It has since been replaced by the ip command discussed above.

OUTPUT:

```
enp0s31f6: flags=4355<UP, BROADCAST, PROMISC,  
MULTICAST> mtu 1500.
```

```
lo: flags=73<UP, LOOPBACK, RUNNING>
```

```
mtu 65536
```

```
wlp2s0: flags=4163<UP, BROADCAST, RUNNING,  
MULTICAST> mtu 1500
```

3. mtr

MTR is a program with a command-line interface that serves as a network diagnostic and troubleshooting tool.

The syntax of the command is as follows

mtr <options> hostname/IP

Let's look at some common use cases.

a) mtr google.com

The basic mtr command shows you the statistics including each hop (hostnames) with time and loss %.

b) ~~mtr -b google.com~~

Output:

Host	Packets		Pings				
	Loss %	Sent	last	Avg	Best	Worst	Stdev
1. JN 115.245.95.245	0.0%	83	6.2	14.2	5.8	20.2	24.1
2. VS 72.14.217.252	0.0%	321	6.2	15.5	5.9	36.8	25.3

b) mtr -b google.com

Show the numeric IP address hostnames too.

Output

Output	Packets		Pings			
	Loss %	snt	last	avg	Best	worst
172.16.12.122	0.0%	8.3	6.2	14.2	5.8	24
172.16.12.122	0.0%	8.4	6.4	82.6	5.5	54.8

A) tcpdump:

This command is designed for capturing and displaying packets.

a) tcpdump -i wlp250:

This command captures the traffic on wlp250.

output:

dropped privs to tcpdump

tcpdump: Verbose output suppressed, use -v[v]... for full protocol decode

listening on wlp2s0, link-type EN10MB (Ethernet),
snapshot length 262144 bytes

23:15:48.819979 ARP, Request who-has climate-ay-1.1

b) `tcpdump -i wlp2s0 -c 10 host 3.8.8.8:`

To capture traffic to and coming from one specific host.

output:

dropped privs to tcpdump

tcpdump: verbose output suppressed, use -v[v]... for full protocol decode

listening on wlp2s0, link-type EN10MB (Ethernet)
snapshot length 262144 bytes

0 packets captured

0 packets received by filter

0 packets dropped by kernel

c) `tcpdump -i wlp2s0 net 10.1.0.0 mask 255.255.255.0:`

to capture traffic to and from a specific network.

output:

dropped privs to tcpdump

tcpdump verbose output suppressed, use -v[v]... for full protocol decode

listening on wlp2s0, link

0 packets captured

0 packets received by filter

d) `tcpdump -i wlp2s0 port 80:`

To capture traffic to and from port 80

output:

dropped privs to

tcpdump: Verbose

for full

protocol decode

0 packets captured

0 packets received

e) Ping:

It is used to troubleshoot network connectivity and name resolution.

ping google.com

ping google.com (12

bytes of data.

from fedora (192

destination - Can not reach

from fedora (192

destination most unreachable

Result:

thus the command was executed successfully

listening on wlan0, link-type EN10MB(Ethernet)

0 packets captured

0 packets received by filter

d) tcpdump -i wlan0 port 58:

To capture traffic to and from port numbers
output:

dropped prior to tcpdump

tcpdump: Verbose output suppressed, use -v[v].
for full:

protocol decode

0 packets captured

0 packets received by filter

5) Ping:

It is used to troubleshoot connectivity reachability
and name resolution.

ping google.com

ping google.com (142.253.221.266) 56(34)

bytes of data.

from fedora (192.168.1.294) icmp-seq=1

destination - can't reach unreachable.

from fedora (192.168.1.294) icmp-seq=2

destination most unreachable.

Result:

thus the commands has been
executed successfully using linux and windows
4/8/20