Lab 5: Subnetting and Linux Networking -

Parth Kalkar

Q1. You have a range 172.16.200.0/22

1. Identify subnet range - This is a class B address, therefore it has a reference block size of 16 bits, leaving us with 2^(22−16) = 2^6 = 64 subnets
2. How many usable IP addresses? - 2^10 − 2 = 1022
3. Identify starting IP and ending IP - Starting IP: 172.16.200.1 & Ending IP: 172.16.203.254
    - Address: 172.16.200.0 10101100.00010000.110010 00.00000000
    - Netmask: 255.255.252.0 = 22 11111111.11111111.111111 00.00000000
    - Wildcard: 0.0.3.255 00000000.00000000.000000 11.11111111
    - Network: 172.16.200.0/22 10101100.00010000.110010 00.00000000 (Class B)
    - Broadcast: 172.16.203.255 10101100.00010000.110010 11.11111111
    - HostMin: 172.16.200.1 10101100.00010000.110010 00.00000001
    - HostMax: 172.16.203.254 10101100.00010000.110010 11.11111110
    - Hosts/Net: 1022 (Private Internet)

Q2. You have a range 10.16.200.12/17

1. Identify subnet range - This is a class A address, therefore it has a reference block size of 8 bits, leaving us with 2^(17−8) = 2^9 = 512 subnets
2. How many usable ip addresses? - 2^15 − 2 = 32766
3. Identify starting IP and ending IP - First IP: 10.16.128.1 & Ending IP: 10.16.255.254
    - Address: 10.16.200.12 00001010.00010000.1 1001000.00001100
    - Netmask: 255.255.128.0 = 17 11111111.11111111.1 0000000.00000000
    - Wildcard: 0.0.127.255 00000000.00000000.0 1111111.11111111
    - Network: 10.16.128.0/17 00001010.00010000.1 0000000.00000000 (Class A)
    - Broadcast: 10.16.255.255 00001010.00010000.1 1111111.11111111
    - HostMin: 10.16.128.1 00001010.00010000.1 0000000.00000001
    - HostMax: 10.16.255.254 00001010.00010000.1 1111111.11111110
    - Hosts/Net: 32766 (Private Internet)

Q3. You have a range 192.168.0.0/24 and divide into small subnets

1. Subnet with 29 hosts - For 29 hosts, we need a minimum subnet of /27. So, we split out initial range into 8 subnets:
    - 192.168.0.0/27
    - 192.168.0.32/27
    - 192.168.0.64/27
    - 192.168.0.96/27
    - 192.168.0.128/27
    - 192.168.0.160/27
    - 192.168.0.192/27
    - 192.168.0.224/27
2. Subnet with 120 hosts - For 120 hosts, we need a minimum subnet of /25. So, we split out initial range into 2 subnets:

    - 192.168.0.0/25

    - 192.168.0.128/25

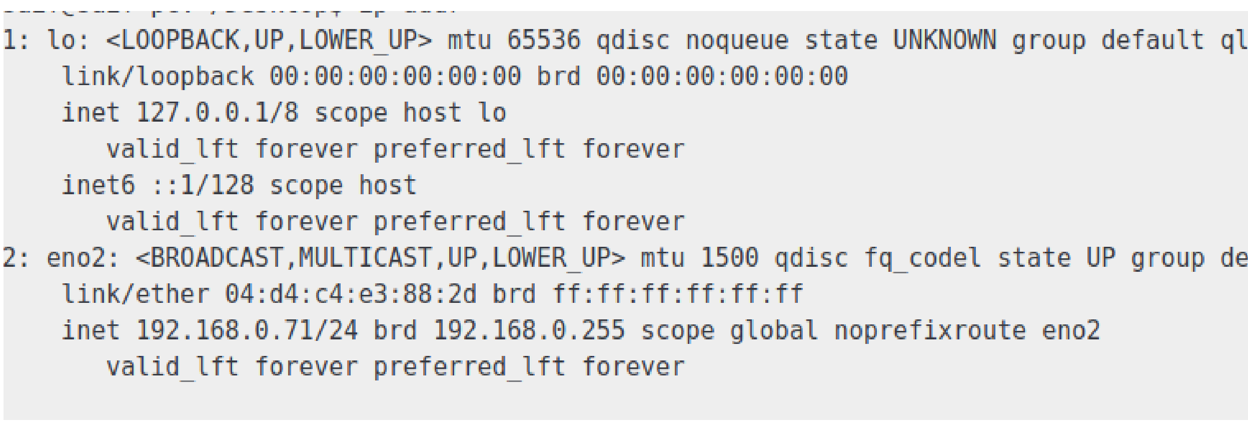
3. Subnet with 60 hosts - For 29 hosts, we need a minimum subnet of /26. So, we split out initial range into 4 subnets:
    - 192.168.0.0/26
    - 192.168.0.64/26
    - 192.168.0.128/26
    - 192.168.0.192/26

- Address: 192.168.0.0 11000000.10101000.00000000 .00000000
- Netmask: 255.255.255.0 = 24 11111111.11111111.11111111 .00000000
- Wildcard: 0.0.0.255 00000000.00000000.00000000 .11111111
- Network: 192.168.0.0/24 11000000.10101000.00000000 .00000000 (Class C)
- Broadcast: 192.168.0.255 11000000.10101000.00000000 .11111111
- HostMin: 192.168.0.1 11000000.10101000.00000000 .00000001
- HostMax: 192.168.0.254 11000000.10101000.00000000 .11111110
- Hosts/Net: 254 (Private Internet)

192.168.0.0/25 , 192.168.0.128/26, 192.168.0.192/27 and 192.168.0.224/27 are available as unused subnets that can have up to 30 hosts.

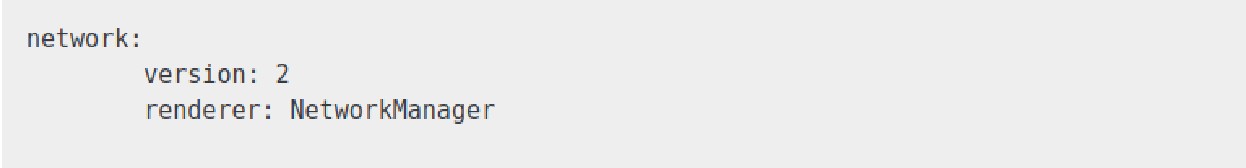
Q4. Add several IP addresses to the interface using netplan and ping them.

- To solve this task, do the following:
   a. **Get the physical interface name** - Type $ *ip addr* command to get the physical interface names to edit the yaml file. I have typed the same and my interface name is **eno2**.



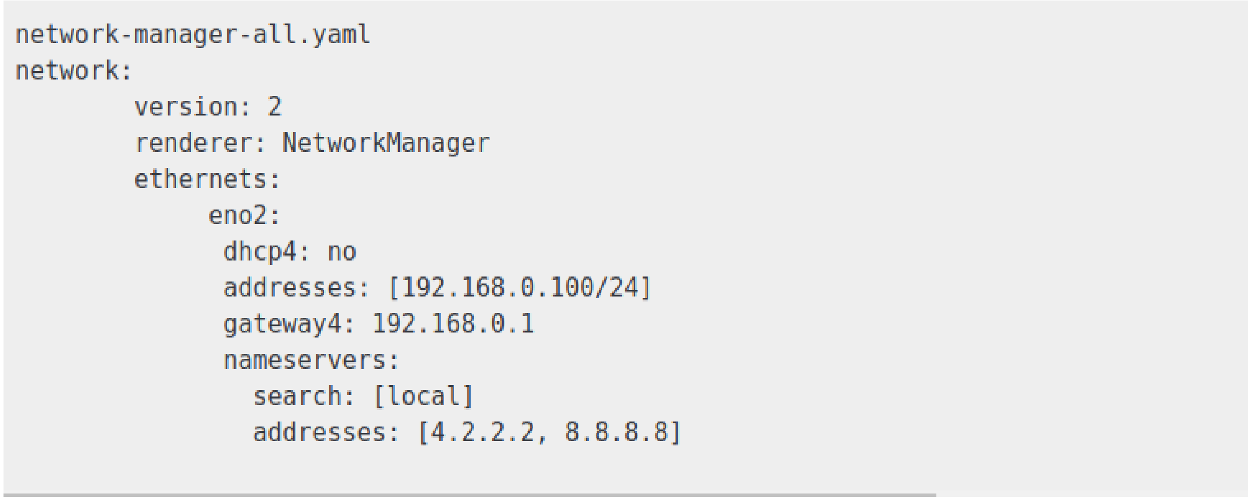
```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default ql
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eno2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group de
    link/ether 04:d4:c4:e3:88:2d brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.71/24 brd 192.168.0.255 scope global noprefixroute eno2
       valid_lft forever preferred_lft forever
```

   b. **Edit the Netplan configuration** - Type $ *sudo nano etc/netplan/01-network-manager-all.yaml*



```
network:
        version: 2
        renderer: NetworkManager
```

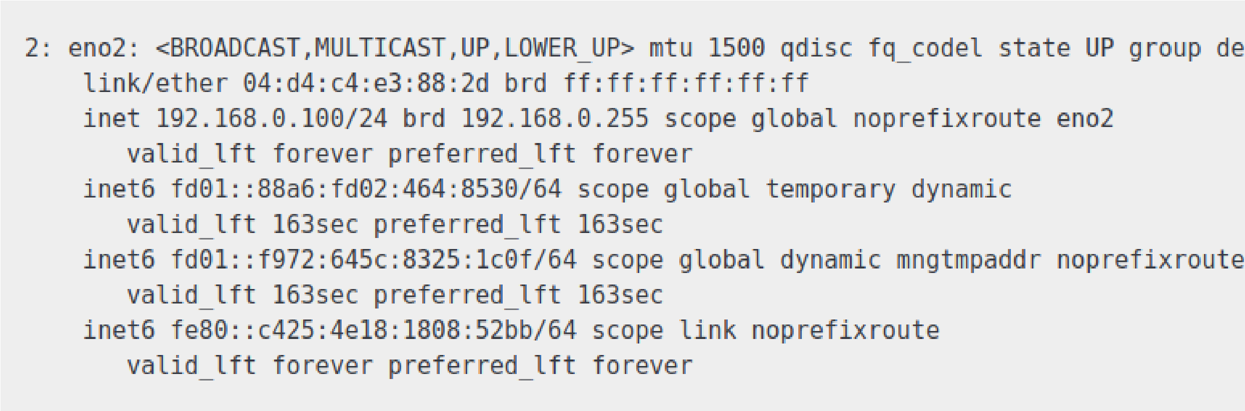
As we can see in the diagram we should assign an **IP address 192.168.0.100/24 to the host.** Hence add the addresses and gateways according to the image below



```
network-manager-all.yaml
network:
        version: 2
        renderer: NetworkManager
        ethernets:
             eno2:
              dhcp4: no
              addresses: [192.168.0.100/24]
              gateway4: 192.168.0.1
              nameservers:
                search: [local]
                addresses: [4.2.2.2, 8.8.8.8]
```

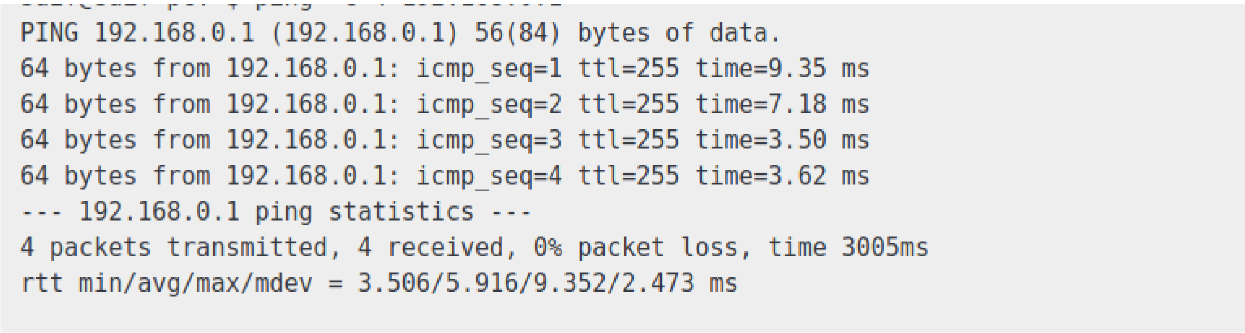
c. **Apply the config -** Type $ *sudo netplan apply*

d. **Validate the change -** Type $ *ip addr*



```
2: eno2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group de
    link/ether 04:d4:c4:e3:88:2d brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.100/24 brd 192.168.0.255 scope global noprefixroute eno2
       valid_lft forever preferred_lft forever
    inet6 fd01::88a6:fd02:464:8530/64 scope global temporary dynamic
       valid_lft 163sec preferred_lft 163sec
    inet6 fd01::f972:645c:8325:1c0f/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 163sec preferred_lft 163sec
    inet6 fe80::c425:4e18:1808:52bb/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

e. **Verify the connectivity -** ping the IP address **192.168.0.1**



```
PING 192.168.0.1 (192.168.0.1) 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=255 time=9.35 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=255 time=7.18 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=255 time=3.50 ms
64 bytes from 192.168.0.1: icmp_seq=4 ttl=255 time=3.62 ms
--- 192.168.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 3.506/5.916/9.352/2.473 ms
```

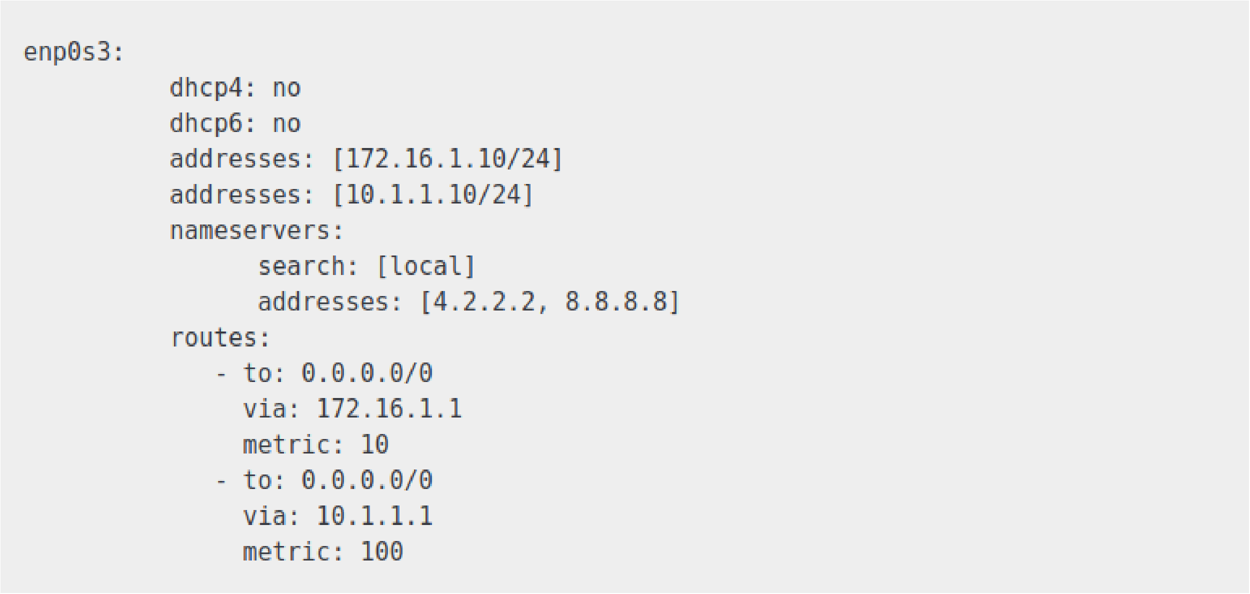
Adding a second IP address, the steps are quite similar to what is shown earlier.

I have an Ubuntu machine connected to the network using a single interface and that interface has the default gateway 172.16.1.1 and 10.1.1.1 as a secondary IP. This is what we are going to do.

- Add the primary address and secondary address 172.16.1.10 and 10.1.1.10.
- As we cannot add a default gateway for two IP addresses, we need to use something called routing.
- With the help of netplan routing command, you can add two default routes one with a lower metric which will be preferred and another with the higher metric. so the machine can go out to the internet.

a. **Sample configuration** - The configuration looks the same as previous but a new field added as addresses and we have configured a secondary address there on the second field.

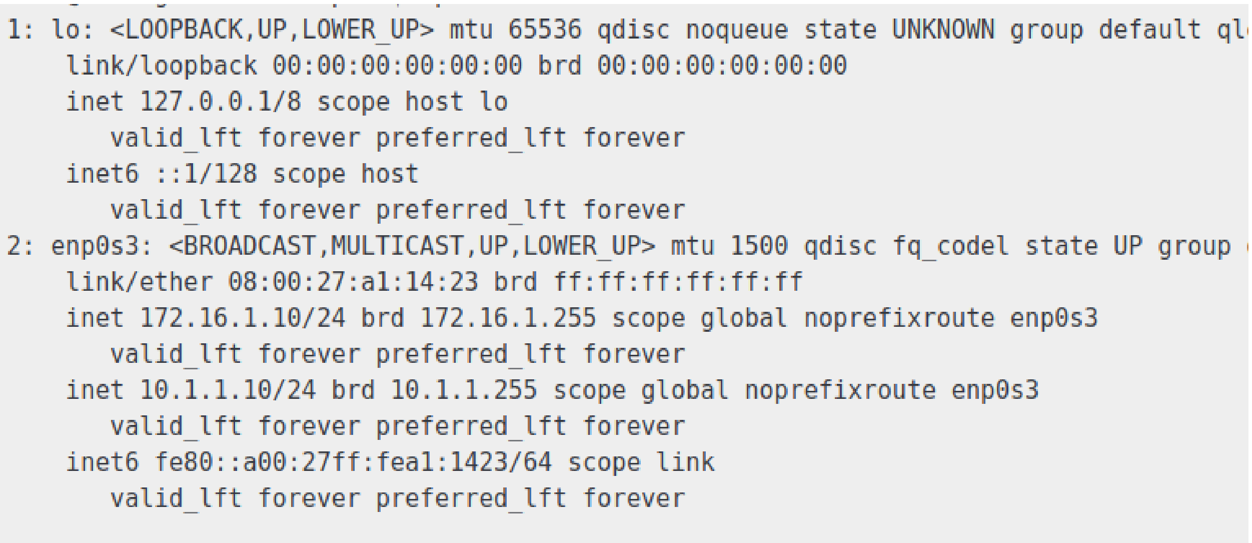
1. First, we added two IP addresses.
2. Second, we added the DNS servers.
3. Finally, the routes command to route the packet to two different networks. One with the metric of 10 and other with 100.



```
enp0s3:
          dhcp4: no
          dhcp6: no
          addresses: [172.16.1.10/24]
          addresses: [10.1.1.10/24]
          nameservers:
                search: [local]
                addresses: [4.2.2.2, 8.8.8.8]
          routes:
             - to: 0.0.0.0/0
               via: 172.16.1.1
               metric: 10
             - to: 0.0.0.0/0
               via: 10.1.1.1
               metric: 100
```

b. **Apply the configuration** - Type $ *sudo netplan apply*

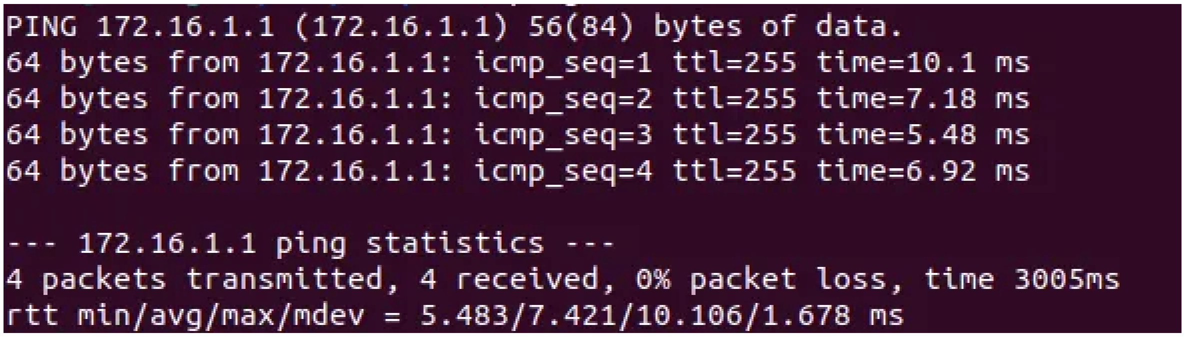
c. **Verification** - The IP configuration would look like below.



```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default ql
    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
    link/ether 08:00:27:a1:14:23 brd ff:ff:ff:ff:ff:ff
    inet 172.16.1.10/24 brd 172.16.1.255 scope global noprefixroute enp0s3
       valid_lft forever preferred_lft forever
    inet 10.1.1.10/24 brd 10.1.1.255 scope global noprefixroute enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fea1:1423/64 scope link
       valid_lft forever preferred_lft forever
```

d. **Ping** - Ping the newly added IP's default gateway

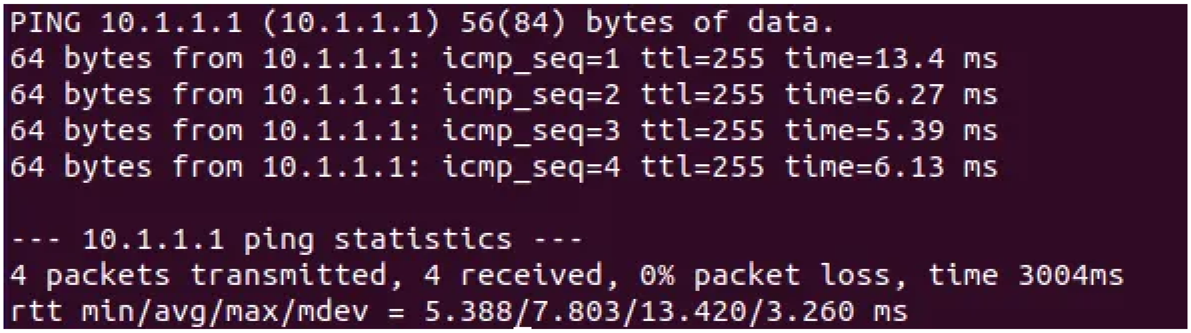
1. $ *ping 172.16.1.1 - 4*



```
PING 172.16.1.1 (172.16.1.1) 56(84) bytes of data.
64 bytes from 172.16.1.1: icmp_seq=1 ttl=255 time=10.1 ms
64 bytes from 172.16.1.1: icmp_seq=2 ttl=255 time=7.18 ms
64 bytes from 172.16.1.1: icmp_seq=3 ttl=255 time=5.48 ms
64 bytes from 172.16.1.1: icmp_seq=4 ttl=255 time=6.92 ms

--- 172.16.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 5.483/7.421/10.106/1.678 ms
```

2. $ *ping 10.1.1.1 -c 4*



```
PING 10.1.1.1 (10.1.1.1) 56(84) bytes of data.
64 bytes from 10.1.1.1: icmp_seq=1 ttl=255 time=13.4 ms
64 bytes from 10.1.1.1: icmp_seq=2 ttl=255 time=6.27 ms
64 bytes from 10.1.1.1: icmp_seq=3 ttl=255 time=5.39 ms
64 bytes from 10.1.1.1: icmp_seq=4 ttl=255 time=6.13 ms

--- 10.1.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 5.388/7.803/13.420/3.260 ms
```