

Lab 1: Test Connectivity

Tasks:

1. Ping your gateway and observe TTL value.

```
root@ip-172-31-6-44:/home/ubuntu# ip route
default via 172.31.0.1 dev ens5 proto dhcp src 172.31.6.44 metric 100
172.31.0.0/20 dev ens5 proto kernel scope link src 172.31.6.44 metric 100
172.31.0.1 dev ens5 proto dhcp scope link src 172.31.6.44 metric 100
172.31.0.2 dev ens5 proto dhcp scope link src 172.31.6.44 metric 100
root@ip-172-31-6-44:/home/ubuntu# █
```

```
root@ip-172-31-6-44:/home/ubuntu# ping 172.31.0.1
PING 172.31.0.1 (172.31.0.1) 56(84) bytes of data.
64 bytes from 172.31.0.1: icmp_seq=1 ttl=64 time=0.057 ms
64 bytes from 172.31.0.1: icmp_seq=2 ttl=64 time=0.171 ms
64 bytes from 172.31.0.1: icmp_seq=3 ttl=64 time=0.074 ms
64 bytes from 172.31.0.1: icmp_seq=4 ttl=64 time=0.066 ms
64 bytes from 172.31.0.1: icmp_seq=5 ttl=64 time=0.084 ms
64 bytes from 172.31.0.1: icmp_seq=6 ttl=64 time=0.088 ms
^C
--- 172.31.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5119ms
rtt min/avg/max/mdev = 0.057/0.090/0.171/0.037 ms
root@ip-172-31-6-44:/home/ubuntu# █
```

2. Ping a public server like 8.8.8.8 (Google DNS).

```
root@ip-172-31-6-44:/home/ubuntu# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=113 time=1.25 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=113 time=1.28 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=113 time=1.38 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=113 time=1.31 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=113 time=1.30 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=113 time=1.32 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=113 time=1.37 ms
^C
--- 8.8.8.8 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6009ms
rtt min/avg/max/mdev = 1.253/1.315/1.383/0.042 ms
root@ip-172-31-6-44:/home/ubuntu# █
```

3. Ping a domain name (e.g., google.com) — does DNS resolution work?

Commands Hint:

ping <ip>, ping <domain>

```
root@ip-172-31-6-44:/home/ubuntu# ping google.com
PING google.com (142.251.111.139) 56(84) bytes of data.
64 bytes from bk-in-f139.1e100.net (142.251.111.139): icmp_seq=1 ttl=101 time=2.09 ms
64 bytes from bk-in-f139.1e100.net (142.251.111.139): icmp_seq=2 ttl=101 time=2.12 ms
64 bytes from bk-in-f139.1e100.net (142.251.111.139): icmp_seq=3 ttl=101 time=2.09 ms
64 bytes from bk-in-f139.1e100.net (142.251.111.139): icmp_seq=4 ttl=101 time=2.08 ms
64 bytes from bk-in-f139.1e100.net (142.251.111.139): icmp_seq=5 ttl=101 time=2.15 ms
```

Lab 2: Essential Networking Commands

Objective: Use diagnostic tools.

Tasks:

- Display your network interfaces with ifconfig or ip a.

```
root@ip-172-31-6-44:/home/ubuntu# ip a
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: ens5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc mq state UP group default qlen 1000
    link/ether 02:41:e9:a:d5:6b brd ff:ff:ff:ff:ff:ff
        altname enp0s5
        inet 172.31.6.44/20 metric 100 brd 172.31.15.255 scope global dynamic ens5
            valid_lft 2754sec preferred_lft 2754sec
        inet6 fe80::41:e9ff:fe9a:d56b/64 scope link
            valid_lft forever preferred_lft forever
root@ip-172-31-6-44:/home/ubuntu#
```

- Trace the route to google.com using traceroute.

```
root@ip-172-31-6-44:/home/ubuntu# traceroute google.com
traceroute to google.com (172.253.115.101), 64 hops max
 1  240.3.180.79  1.564ms  1.534ms  1.560ms
 2  99.82.14.178  1.284ms  1.255ms  1.151ms
 3  99.82.14.179  1.455ms  1.351ms  1.376ms
 4  216.239.40.107  1.462ms  1.354ms  1.302ms
 5  192.178.243.2  1.589ms  1.591ms  1.552ms
 6  142.251.49.19  14.633ms  8.347ms  1.567ms
```

- Show ARP table and analyze entries.

```
root@ip-172-31-6-44:/home/ubuntu# arp -n
Address          HWtype  HWaddress          Flags Mask      Iface
172.31.0.1      ether    02:e2:0a:8a:8b:c9  C        ens5
172.31.0.2      ether    02:e2:0a:8a:8b:c9  C        ens5
root@ip-172-31-6-44:/home/ubuntu#
```

- Examine the routing table using route -n.

Commands: ifconfig, traceroute, arp -n, route -n, ip route

```
root@ip-172-31-6-44:/home/ubuntu# route -n
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref  Use Iface
0.0.0.0         172.31.0.1   0.0.0.0       UG    100    0    0 ens5
172.31.0.0     0.0.0.0       255.255.240.0  U     100    0    0 ens5
172.31.0.1     0.0.0.0       255.255.255.255 UH    100    0    0 ens5
172.31.0.2     0.0.0.0       255.255.255.255 UH    100    0    0 ens5
root@ip-172-31-6-44:/home/ubuntu# ifconfig
ens5: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
      inet 172.31.6.44 netmask 255.255.240.0 broadcast 172.31.15.255
          inet6 fe80::41:e9ff:fe9a:d56b prefixlen 64 scopeid 0x20<link>
            ether 02:41:e9:9a:d5:6b txqueuelen 1000 (Ethernet)
              RX packets 32026 bytes 45136082 (45.1 MB)
              RX errors 0 dropped 0 overruns 0 frame 0
              TX packets 2796 bytes 354716 (354.7 KB)
              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)
```

Lab 3: DNS and Name Resolution

Objective: Investigate how DNS works.

Tasks:

- Use nslookup or dig to resolve www.github.com.

```
root@ip-172-31-6-44:/home/ubuntu# nslookup www.github.com
Server:           127.0.0.53
Address:          127.0.0.53#53

Non-authoritative answer:
www.github.com canonical name = github.com.
Name:  github.com
Address: 140.82.112.4

root@ip-172-31-6-44:/home/ubuntu#
```

Obtain information about Domain Names

- Identify the IP address of your DNS server.

```
root@ip-172-31-6-44:~# resolvectl status
Global
    Protocols: -LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported
resolv.conf mode: stub

Link 2 (ens5)
    Current Scopes: DNS
        Protocols: +DefaultRoute +LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported
Current DNS Server: 172.31.0.2
    DNS Servers: 172.31.0.2
        DNS Domain: ec2.internal
root@ip-172-31-6-44:~#
```

- Flush and recheck DNS cache.

```
root@ip-172-31-6-44:~# resolvectl statistics
DNSSEC supported by current servers: no

Transactions
Current Transactions: 0
    Total Transactions: 98

Cache
    Current Cache Size: 0
        Cache Hits: 21
        Cache Misses: 85

DNSSEC Verdicts
    Secure: 0
    Insecure: 0
        Bogus: 0
    Indeterminate: 0
root@ip-172-31-6-44:~#
```

- Explain difference between recursive and iterative DNS queries.
Commands: dig, nslookup, cat /etc/resolv.conf

Recursive	Iterative
Client expects final answer	Client follows referrals
Used by browsers	Used by DNS servers
Example: EC2 → Route53	Root → TLD → Authoritative

Lab 4: Use Traceroute

Tasks:

1. Run a traceroute to google.com.

```
root@ip-172-31-6-44:~# traceroute google.com
traceroute to google.com (142.251.111.138), 64 hops max
 1  100.100.32.28  1.285ms  0.838ms  1.565ms
 2  240.0.184.33  1.139ms  1.089ms  1.675ms
 3  100.100.2.32  6.163ms  40.847ms  13.401ms
```

2. How many hops are there?

3

3. Identify which hop belongs to your ISP.

Commands Hint:

traceroute google.com

```
root@ip-172-31-6-44:~# traceroute google.com
traceroute to google.com (142.251.111.138), 64 hops max
 1  100.100.32.28  1.285ms  0.838ms  1.565ms
 2  240.0.184.33  1.139ms  1.089ms  1.675ms
 3  100.100.2.32  6.163ms  40.847ms  13.401ms
```

Lab 5: Check Host Reachability with telnet/nc

Tasks:

1. Check if port 80 on a server is reachable.

```
root@ip-172-31-6-44:~# nc -zv google.com 80
Connection to google.com (64.233.180.139) 80 port [tcp/http] succeeded!
root@ip-172-31-6-44:~#
```

2. Check SSH port 22 connectivity.

```
root@ip-172-31-6-44:~# nc -zv 172.31.6.44 22
Connection to 172.31.6.44 22 port [tcp/ssh] succeeded!
root@ip-172-31-6-44:~#
```

3. Test a closed port and observe behavior.

Commands Hint:

telnet <ip> <port> or nc -zv <ip> <port>

```
root@ip-172-31-6-44:~# nc -zv google.com 81
nc: connect to google.com (192.178.218.101) port 81 (tcp) failed: Connection timed out
root@ip-172-31-6-44:~#
```

Lab 6: Default Gateway & Routing Table

Tasks:

1. Display your routing table.

```
root@ip-172-31-6-44:~# ip route show
default via 172.31.0.1 dev ens5 proto dhcp src 172.31.6.44 metric 100
172.31.0.0/20 dev ens5 proto kernel scope link src 172.31.6.44 metric 100
172.31.0.1 dev ens5 proto dhcp scope link src 172.31.6.44 metric 100
172.31.0.2 dev ens5 proto dhcp scope link src 172.31.6.44 metric 100
root@ip-172-31-6-44:~#
```

2. Add a temporary static route to another network.

```
root@ip-172-31-6-44:~# sudo ip route add 10.10.10.0/24 via 172.31.0.1
root@ip-172-31-6-44:~#
```

- 3 . Delete that static route.

Commands Hint:

ip route show,

ip route add,
ip route del

```
root@ip-172-31-6-44:~# sudo ip route del 10.10.10.0/24  
root@ip-172-31-6-44:~# █
```

Lab 7: Network Security and Attack Simulation

Objective: Identify and defend against common attacks.

Tasks:

- Simulate a simple ping flood using ping -f (in a safe, local test).

```
root@ip-172-31-6-44:~# ping -f 127.0.0.1█
```

- Use netstat to detect unusual open connections.

```
root@ip-172-31-6-44:~# netstat -tulnp  
Active Internet connections (only servers)  
Proto Recv-Q Send-Q Local Address           Foreign Address         State      PID/Program name  
tcp        0      0 0.0.0.0:22              0.0.0.0:*          LISTEN     614/sshd: /usr/sbin  
tcp        0      0 127.0.0.53:53            0.0.0.0:*          LISTEN     345/systemd-resolve  
tcp6       0      0 :::22                  :::*               LISTEN     614/sshd: /usr/sbin  
udp        0      0 127.0.0.1:323             0.0.0.0:*          LISTEN     447/chronyrd  
udp        0      0 127.0.0.53:53            0.0.0.0:*          LISTEN     345/systemd-resolve  
udp        0      0 172.31.6.44:68            0.0.0.0:*          LISTEN     343/systemd-network  
udp6       0      0 :::323                 :::*               LISTEN     447/chronyrd  
root@ip-172-31-6-44:~# █
```

- Discuss mitigation techniques for DDoS and Man-in-the-Middle attacks.

DDoS:

AWS Shield

WAF

Rate limiting

Auto Scaling

MITM:

HTTPS / TLS

Certificate pinning

VPN

ARP inspection

- Identify open ports and services using nmap.
Commands: ping, netstat -tulnp, nmap -sS localhost

Scan Open Ports

```
root@ip-172-31-6-44:~# sudo nmap -sS localhost
Starting Nmap 7.80 ( https://nmap.org ) at 2026-01-28 10:14 UTC
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000030s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh

Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds
root@ip-172-31-6-44:~#
```

Lab 8: Hosts File Resolution Priority

Tasks:

1. Add a fake entry for facebook.com in /etc/hosts.

```
127.0.0.1 localhost

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
1.1.1.1 facebook.com
```

2. Curl to see if it resolves to your fake IP.

```
root@ip-172-31-6-44:~# curl facebook.com
error code: 1001root@ip-172-31-6-44:~#
```

3. Remove the entry and test again.

```
root@ip-172-31-6-44:~# curl facebook.com
root@ip-172-31-6-44:~#
```

Lab 9: Understand Ports, Services & Firewall Rules

Tasks:

1. List which services are listening on your machine.

```
root@ip-172-31-6-44:~# ss -tulnp
Netid      State     Recv-Q      Send-Q      Local Address:Port          Peer Address:Port      Process
udp        UNCONN    0            0           127.0.0.1:323          0.0.0.0:*              users:(("chronyd",pid=447,fd=5))
udp        UNCONN    0            0           127.0.0.53:lo:53        0.0.0.0:*              users:(("systemd-resolve",pid=345,fd=13))
udp        UNCONN    0            0           172.31.6.44:ens5:68      0.0.0.0:*              users:(("systemd-network",pid=343,fd=15))
udp        UNCONN    0            0           [::]:323                [::]:*                  users:(("chronyd",pid=447,fd=6))
tcp        LISTEN    0            128          0.0.0.0:22              0.0.0.0:*              users:(("sshd",pid=614,fd=3))
tcp        LISTEN    0            4096         127.0.0.53:lo:53        0.0.0.0:*              users:(("systemd-resolve",pid=345,fd=14))
tcp        LISTEN    0            128          [::]:22                [::]:*                  users:(("sshd",pid=614,fd=4))
```

2. Identify which ports are open for TCP and UDP.

```
root@ip-172-31-6-44:~# sudo nmap -sS -sU localhost
Starting Nmap 7.80 ( https://nmap.org ) at 2026-01-28 10:25 UTC
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000040s latency).
Not shown: 1999 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh

Nmap done: 1 IP address (1 host up) scanned in 0.10 seconds
root@ip-172-31-6-44:~#
```

3. Enable or disable a port using ufw (Ubuntu firewall).

```
root@ip-172-31-6-44:~# sudo ufw allow 80
Rule added
Rule added (v6)
root@ip-172-31-6-44:~# sudo ufw deny 22
Rule added
Rule added (v6)
root@ip-172-31-6-44:~#
```

4. Verify whether a port is reachable using nc or telnet.

```
root@ip-172-31-6-44:~# nc -zv localhost 80
Connection to localhost (127.0.0.1) 80 port [tcp/http] succeeded!
root@ip-172-31-6-44:~#
```

Error	Meaning
-------	---------

Connection refused	No service
--------------------	------------

Connection timed out	Firewall drop
----------------------	---------------

Tasks:

1. Use curl to fetch the homepage of a website (e.g., example.com).

```
root@ip-172-31-6-44:~# curl example.com
<!DOCTYPE html><html lang="en"><head><title>Example Domain</title><meta name="viewport" content="width=device-width, initial-scale=1"><style>body{background-color:#eee; width:60vw; margin:15vh auto; font-family:system-ui,sans-serif}h1{font-size:1.5em}div{opacity:0.8}a:link,a:visited{color:#348}</style><body><div><h1>Example Domain</h1><p>This domain is for use in documentation examples without needing permission. Avoid use in operations.<p><a href="https://iana.org/domains/example">Learn more</a></div></body>
//iana.org/domains/example
root@ip-172-31-6-44:~#
```

2. View only HTTP response headers.

```
root@ip-172-31-6-44:~# curl -I example.com
HTTP/1.1 200 OK
Date: Wed, 28 Jan 2026 10:42:09 GMT
Content-Type: text/html
Connection: keep-alive
CF-RAY: 9c4fe1ab096cae7d-IAD
Last-Modified: Tue, 27 Jan 2026 13:21:58 GMT
Allow: GET, HEAD
Accept-Ranges: bytes
Age: 4252
cf-cache-status: HIT
Server: cloudflare

root@ip-172-31-6-44:~#
```

3. Send a HEAD request to check server details.

```
root@ip-172-31-6-44:~# curl --head https://example.co
HTTP/2 200
date: Wed, 28 Jan 2026 10:44:36 GMT
content-type: text/html
cf-ray: 9c4fe53dbabb2fd6-IAD
last-modified: Tue, 27 Jan 2026 13:22:54 GMT
allow: GET, HEAD
accept-ranges: bytes
age: 10480
cf-cache-status: HIT
server: cloudflare

root@ip-172-31-6-44:~#
```

4. Test an HTTPS endpoint and observe certificate details

```
root@ip-172-31-6-44:~# curl -v https://google.com
*   Trying 142.251.167.139:443...
* Connected to google.com (142.251.167.139) port 443 (#0)
* ALPN, offering h2
* ALPN, offering http/1.1
* CAfile: /etc/ssl/certs/ca-certificates.crt
* CApath: /etc/ssl/certs
* TLSv1.0 (OUT), TLS header, Certificate Status (22):
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS header, Certificate Status (22):
* TLSv1.3 (IN), TLS handshake, Server hello (2):
* TLSv1.2 (IN), TLS header, Finished (20):
* TLSv1.2 (IN), TLS header, Supplemental data (23):
* TLSv1.3 (IN), TLS handshake, Encrypted Extensions (8):
* TLSv1.3 (IN), TLS handshake, Certificate (11):
* TLSv1.3 (IN), TLS handshake, CERT verify (15):
* TLSv1.3 (IN), TLS handshake, Finished (20):
* TLSv1.2 (OUT), TLS header, Finished (20):
```