

## Day 15 – Networking Concepts: DNS, IP, Subnets & Ports

### Task 1: DNS – How Names Become IPs

#### 1-What happens when you type google.com in a browser?

When you type google.com, your system asks a DNS resolver for its IP address.

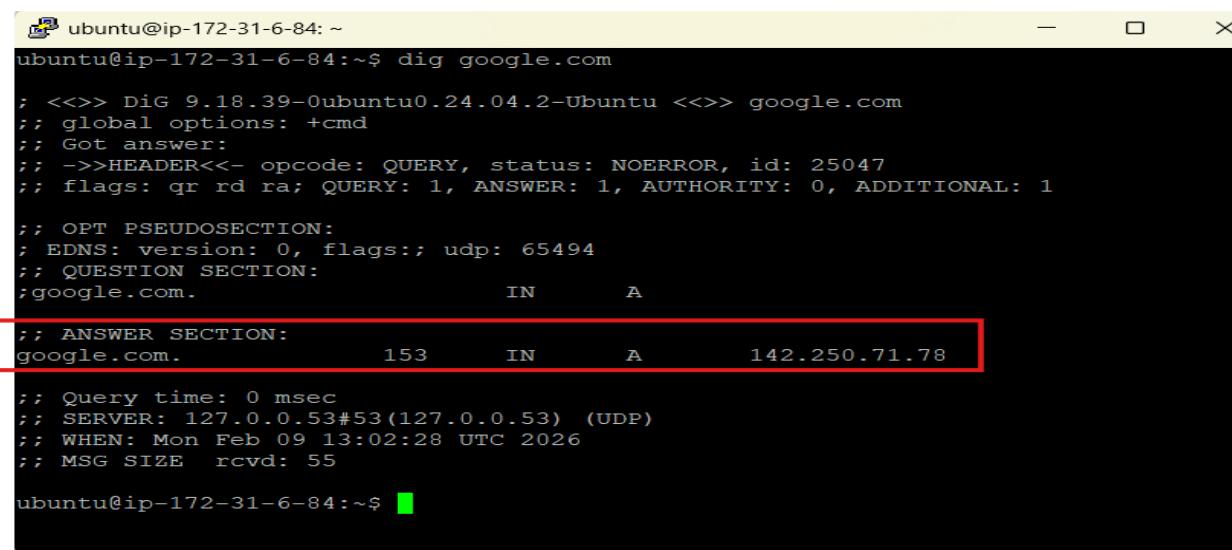
The resolver checks cache first; if not found, it queries root → TLD → authoritative DNS servers. The final IP address is returned to the browser, which then connects to that IP using HTTP/HTTPS.

#### 2-DNS Record Types

- **A** – Maps a domain name to an IPv4 address
- **AAAA** – Maps a domain name to an IPv6 address
- **CNAME** – Alias of one domain name to another
- **MX** – Mail server responsible for receiving emails
- **NS** – Specifies authoritative name servers for a domain

#### Command:

```
# dig google.com
```



```
ubuntu@ip-172-31-6-84: ~
ubuntu@ip-172-31-6-84:~$ dig google.com

; <>> DiG 9.18.39-0ubuntu0.24.04.2-Ubuntu <>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 25047
;; 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.        153     IN      A       142.250.71.78

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Mon Feb 09 13:02:28 UTC 2026
;; MSG SIZE  rcvd: 55

ubuntu@ip-172-31-6-84:~$
```

## Task 2: IP Addressing

### 1-What is an IPv4 address?

An IPv4 address is a 32-bit numeric identifier assigned to a device on a network.

It is written in dotted-decimal format like 192.168.1.10 (4 octets, each 0–255).

#### ◆ Public vs Private IP

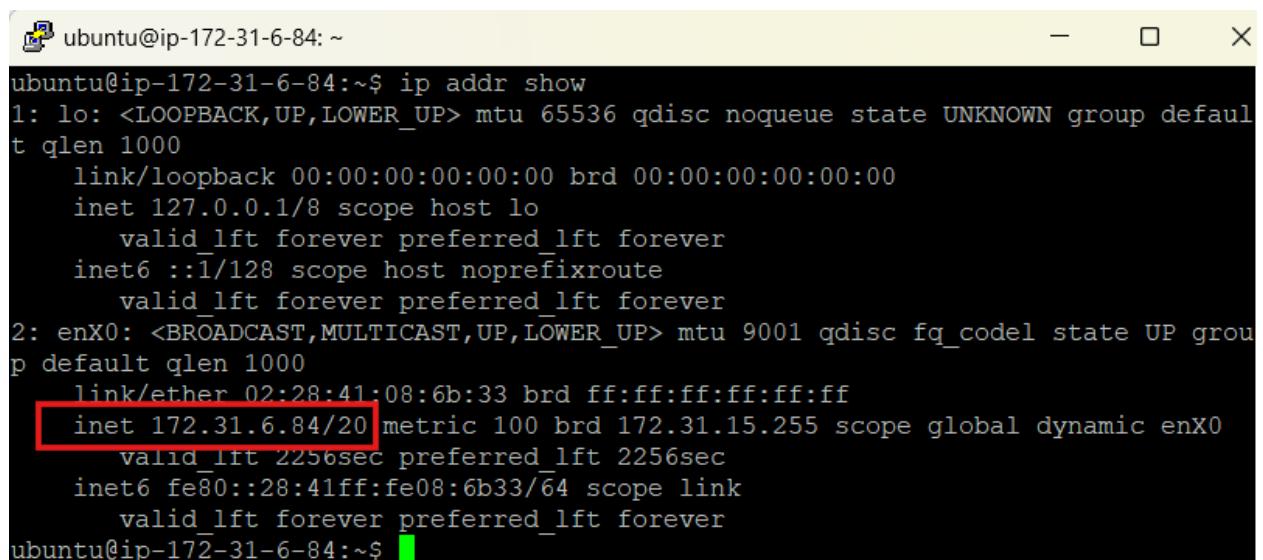
- **Public IP:** Routable on the internet (e.g., 8.8.8.8)
- **Private IP:** Used inside internal networks (e.g., 192.168.1.10)

#### ◆ Private IP Ranges

- 10.0.0.0 – 10.255.255.255
- 172.16.0.0 – 172.31.255.255
- 192.168.0.0 – 192.168.255.255

**Command:**

```
# ip addr show
```



```
ubuntu@ip-172-31-6-84: ~
ubuntu@ip-172-31-6-84:~$ ip addr show
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 noprefixroute
                valid_lft forever preferred_lft forever
2: enX0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc fq_codel state UP group default qlen 1000
        link/ether 02:28:41:08:6b:33 brd ff:ff:ff:ff:ff:ff
        inet 172.31.6.84/20 metric 100 brd 172.31.15.255 scope global dynamic enX0
            valid_lft 2256sec preferred_lft 2256sec
            inet6 fe80::228:41ff:fe08:6b33/64 scope link
                valid_lft forever preferred_lft forever
ubuntu@ip-172-31-6-84:~$
```

## Task 3: CIDR & Subnetting

- **What does /24 mean in 192.168.1.0/24?**

/24 means the first 24 bits are used for the network portion, leaving 8 bits for hosts.

- **Usable Hosts**

- **/24:** 254 usable hosts
- **/16:** 65,534 usable hosts
- **/28:** 14 usable hosts

- **Why do we subnet?**

Subnetting divides large networks into smaller ones to improve performance, security, and IP management.

CIDR	Subnet Mask	Total IPs	Usable Hosts
/24	255.255.255.0	256	254
/16	255.255.0.0	65,536	65,534
/28	255.255.255.240	16	14

## Task 4: Ports – The Doors to Services

### What is a port?

A port is a logical communication endpoint that identifies a specific service running on a system.

### Why do we need ports?

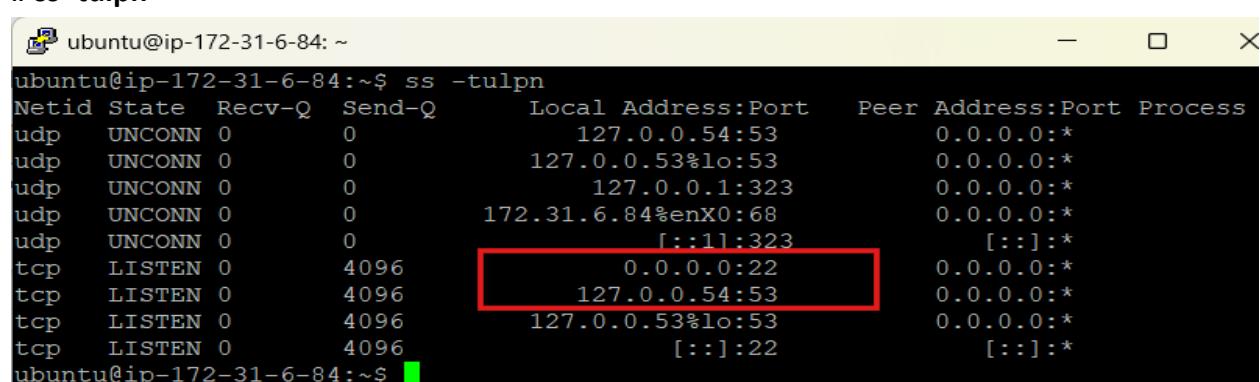
→ Ports allow multiple services (web, database, SSH, etc.) to run on the same IP without conflict.

### Common Ports

Port	Service
22	SSH
80	HTTP
443	HTTPS
53	DNS
3306	MySQL
6379	Redis
27017	MongoDB

### Command:

```
# ss -tulpn
```



```
ubuntu@ip-172-31-6-84:~$ ss -tulpn
Netid State Recv-Q Send-Q          Local Address:Port      Peer Address:Port Process
udp  UNCONN 0        0              127.0.0.54:53        0.0.0.0:*
udp  UNCONN 0        0              127.0.0.53%lo:53      0.0.0.0:*
udp  UNCONN 0        0              127.0.0.1:323        0.0.0.0:*
udp  UNCONN 0        0              172.31.6.84%enX0:68    0.0.0.0:*
udp  UNCONN 0        0              [::]:323            [::]:*
tcp  LISTEN 0       4096           0.0.0.0:22          0.0.0.0:*
tcp  LISTEN 0       4096           127.0.0.54:53        0.0.0.0:*
tcp  LISTEN 0       4096           127.0.0.53%lo:53      0.0.0.0:*
tcp  LISTEN 0       4096           [::]:22            [::]:*
```

Example matches: 0.0.0.0:22 → SSH

## **Task 5: Putting It Together**

### **1-curl <http://myapp.com:8080> — what concepts are involved?**

→DNS resolves myapp.com to an IP address, then TCP connects to port 8080 on that IP using HTTP protocol.

### **2- App can't reach DB at 10.0.1.50:3306 — what to check first?**

→Check network connectivity (ping), verify the database service is running on port 3306, and confirm firewall/security group rules allow access.

## **What I Learned – Day 15 (Networking Concepts)**

1-I learned how DNS resolves domain names into IP addresses and the purpose of common DNS records.

2-I understood IPv4 addressing, the difference between public and private IPs, and their reserved ranges.

3-I learned CIDR and subnetting to calculate usable hosts and why networks are divided into smaller subnets.

4-I also learned how ports identify services and how DNS, IPs, subnets, and ports work together in real connections.