1. Please explain private IP ranges in WAN, LAN, and MAN. What are the differences between them and what are the use cases of each?

Private IP ranges (e.g., 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16) are used in LANs (local networks like homes/offices), MANs (city-wide networks), and sometimes WANs (wide-area networks) for internal communication. LANs use private IPs for devices in a single network (e.g., office Wi-Fi), MANs for larger networks connecting multiple LANs (e.g., campus networks), and WANs rarely use private IPs, preferring public IPs for global connectivity. Use cases: LANs for device communication, MANs for organizational networks, WANs for internet-facing services.

2. How do we map between private and public networks? What is this translation display called? Why do we use this translation and why not assign public networks to all sites?

Mapping private to public networks uses Network Address Translation (NAT), displayed as a translation table in routers. NAT allows private IPs to access public networks by mapping them to a public IP. We use NAT to conserve public IPv4 addresses, enhance security by hiding internal IPs, and enable multiple devices to share one public IP. Assigning public IPs to all sites is impractical due to IPv4 scarcity and increased exposure to attacks.

3. Imagine you are a sysadmin or DevOps engineer building a new infrastructure with at least 15 separate products, each requiring at least 500 IP addresses initially and up to 2000 later. How would you design private CIDR blocks to support this?

I will use the 10.0.0.0/8 range (16M addresses) with CIDR blocks. Then I assign each of 15 products a /22 block (1024 IPs, e.g., 10.0.0.0/22 to 10.0.56.0/22), providing 500 IPs initially and room for 2000. Reserve additional /22 blocks (e.g., 10.0.60.0/22) for future expansion. Use VLANs to segment products and NAT for public access.

4. Please explain the TCP three-way handshake.

- Client sends a SYN packet to the server.
- Server responds with a SYN-ACK packet, acknowledging the request.
- Client sends an ACK packet to confirm, completing the connection setup.

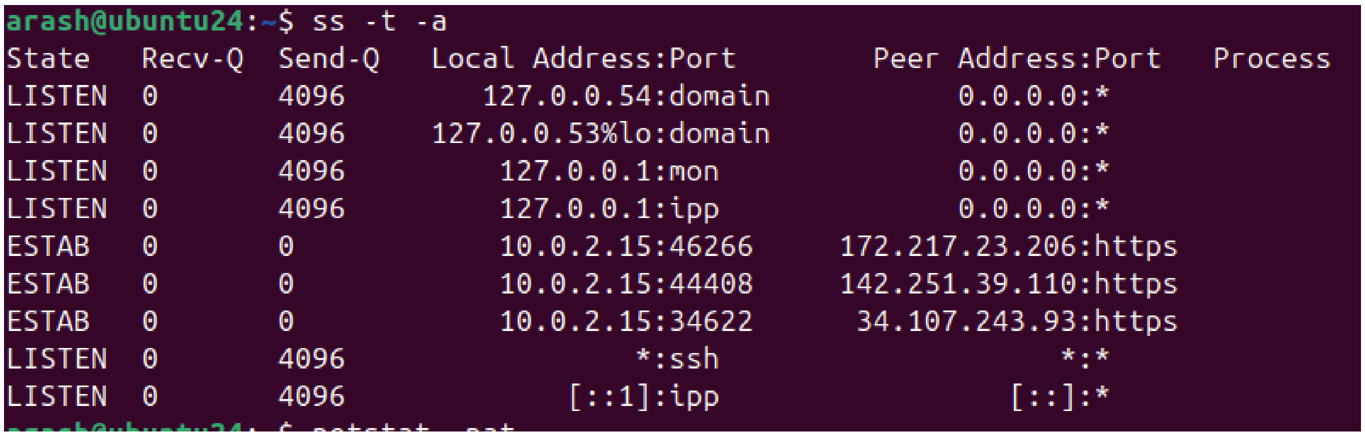
5. Why don't we use only two steps for the handshake?

A two-step handshake (SYN, ACK) doesn't confirm bidirectional communication, risking unverified connections. The three-way handshake ensures both client and server agree on sequence numbers and are ready, preventing issues like packet loss or spoofing.

6. Why do we have two FIN-WAIT states in TCP? What is their purpose?

it ensures reliable connection termination, confirming both sides have sent and received FIN packets, preventing data loss.

7. Please capture a screenshot and explain the live TCP connection states on your own device.



```
arash@ubuntu24:~$ ss -t -a
State    Recv-Q   Send-Q    Local Address:Port          Peer Address:Port    Process
LISTEN   0        4096        127.0.0.54:domain              0.0.0.0:*
LISTEN   0        4096      127.0.0.53%lo:domain             0.0.0.0:*
LISTEN   0        4096         127.0.0.1:mon                 0.0.0.0:*
LISTEN   0        4096         127.0.0.1:ipp                 0.0.0.0:*
ESTAB    0        0            10.0.2.15:46266     172.217.23.206:https
ESTAB    0        0            10.0.2.15:44408     142.251.39.110:https
ESTAB    0        0            10.0.2.15:34622      34.107.243.93:https
LISTEN   0        4096                 *:ssh                       *:*
LISTEN   0        4096             [::1]:ipp                   [::]:*
```

- **State** → The socket state (`LISTEN` , `ESTAB` = established).
- **Recv-Q / Send-Q** → Data waiting in the receive/send queue.
- **Local Address:Port** → My system's IP and port.
- **Peer Address:Port** → The remote system's IP and port.
- **Process** → Shows process using the socket

8. Please compare TCP and UDP.

TCP is connection-oriented, ensuring reliable, ordered data delivery with error checking, used for web browsing (HTTP). UDP is connectionless, faster but unreliable, used for streaming or DNS queries where speed trumps reliability.

9. What is the main concern of TCP?

TCP's main concern is reliable data transmission, ensuring packets are delivered in order, without loss, using acknowledgments and retransmissions.

10. When do we use UDP?

UDP is used for low-latency applications like video streaming, online gaming, VoIP, or DNS queries, where occasional packet loss is acceptable for speed.

11. Why is TCP very popular?

TCP is popular due to its reliability, ensuring data integrity and order, critical for applications like web browsing, email, and file transfers.

12. Why does IPv6 exist? Does IPv6 have any translation mechanism, and if so, why is it needed?

IPv6 exists to address IPv4’s address exhaustion with a 128-bit address space. It uses **NAT64** or **NPTv6** for translation to communicate with IPv4 networks, needed for compatibility during the transition to IPv6-only networks.

13. What do TX and RX mean in the output of `ip -c a` or `ifconfig`?

TX (transmitted) shows bytes/packets sent, and RX (received) shows bytes/packets received.

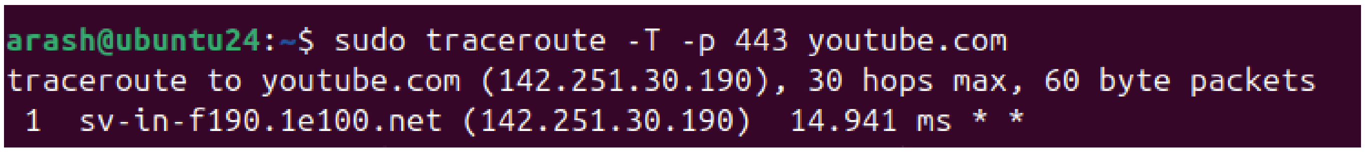
14. What do these interface flags mean: <UP, BROADCAST, RUNNING, MULTICAST>?

UP: Interface is active.
BROADCAST: Supports broadcast packets.
RUNNING: Interface is operational.
MULTICAST: Supports multicast traffic.

15. What is a DNS server? Does DNS use UDP or TCP? What are the important DNS servers?

A DNS server resolves domain names to IP addresses. DNS primarily uses UDP for fast queries but TCP for large responses or zone transfers. Important DNS servers include Google (8.8.8.8) and Cloudflare (1.1.1.1)

16. Use `tracepath` or other tools to analyze the traffic path to youtube.com with and without a proxy.



```
arash@ubuntu24:~$ sudo traceroute -T -p 443 youtube.com
traceroute to youtube.com (142.251.30.190), 30 hops max, 60 byte packets
 1  sv-in-f190.1e100.net (142.251.30.190)  14.941 ms * *
```

I could not test it with proxy!

17. How does the ICMP protocol work?

ICMP (Internet Control Message Protocol) sends control and error messages (e.g., ping, destination unreachable). It operates at the network layer, using packets with type/code fields to report issues or test connectivity, like `ping` using echo requests/replies.

18. What is the nc command and what can we use it for?

The `nc` (netcat) command creates TCP/UDP connections, acting as a client or server. Use it for port scanning, file transfer, chatting, or debugging network services.

19. Please create both TCP and UDP connections with `nc`.

TCP Server: `nc -l 12345`
TCP Client: `nc localhost 12345`
UDP Server: `nc -u -l 12345`
UDP Client: `nc -u localhost 12345`

20. Explain the output of `dig youtube.com` vs `dig varzesh3.com` and how DNS resolution works.

```
dig youtube.com



; <<>> DiG 9.10.6 <<>> youtube.com

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63870

;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1



;; OPT PSEUDOSECTION:

; EDNS: version: 0, flags:; udp: 4096

;; QUESTION SECTION:

;youtube.com. IN A

```

```


;; ANSWER SECTION:

youtube.com. 75 IN A 142.250.179.142



;; Query time: 69 msec

;; SERVER: 192.168.1.1#53(192.168.1.1)

;; WHEN: Fri Aug 22 11:47:26 CEST 2025

;; MSG SIZE  rcvd: 56




```

```
dig varzesh3.com



; <<>> DiG 9.10.6 <<>> varzesh3.com

;; global options: +cmd

;; Got answer:

;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7739

;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1



;; OPT PSEUDOSECTION:

; EDNS: version: 0, flags:; udp: 4096

;; QUESTION SECTION:

;varzesh3.com. IN A



```

```
;; ANSWER SECTION:

varzesh3.com. 3600 IN A 185.143.235.201

varzesh3.com. 3600 IN A 185.143.232.201



;; Query time: 191 msec

;; SERVER: 192.168.1.1#53(192.168.1.1)

;; WHEN: Fri Aug 22 11:47:59 CEST 2025

;; MSG SIZE  rcvd: 73
```

- `->>HEADER<<-`: This section contains metadata about the DNS response.
  - `opcode: QUERY`: The type of DNS operation, which in this case is a standard query.
  - `status: NOERROR`: The query was successful. The DNS server was able to find the requested information.
  - `id: A unique query ID.
  - `flags: qr rd ra`: The flags indicate the behavior of the query and response. `qr` means it's a query response, `rd` means recursion was desired, and `ra` means recursion was available.
  - `QUERY: 1, ANSWER: 1` (or `2`), `AUTHORITY: 0`, `ADDITIONAL: 1`: This shows how many records are in each section of the DNS packet. The key one here is `ANSWER`, which is the number of records returned for your query.
- `OPT PSEUDOSECTION`: This section relates to **EDNS** (Extension Mechanisms for DNS), a protocol that allows for larger DNS messages and more features. The `udp: 4096` indicates the maximum size of the UDP packet that the client can handle.
- `QUESTION SECTION`: This is a direct echo of the query.
  - `;youtube.com. IN A` or `;varzesh3.com. IN A`: This asks the DNS server for an **A record** (which stands for Address) for the domain `youtube.com` or `varzesh3.com`.
- `ANSWER SECTION`
  - This is where the unique information for each query is found.
    - `youtube.com.`: The domain name that was resolved.
    - `75/3600`: This is the **TTL**
    - `IN A`: The type of record is an **IPv4 Address**.
    - The actual IP address returned for `youtube.com`/`varezesh3.com`
    - `varzesh3.com` returns **two A records**. This is a common form of **DNS Round Robin load balancing**. Instead of returning a single IP, the DNS server provides

multiple IPs, and the client can choose one.

- `Query time`: The time it took in milliseconds for the DNS server to respond.
- `SERVER`: The IP address and port of the DNS server that provided the answer. In both cases, this is `192.168.1.1`, which is an IP for a local router acting as a DNS resolver.
- `WHEN`: The date and time the query was performed.
- `MSG SIZE`: The size of the DNS response packet in bytes.

21. Please explain your thoughts about the `lsof` command in the context of the Linux philosophy that "everything is a file."

In Linux, files, directories, sockets, pipes, devices, and even network connections are represented as files in the file system. `lsof` uses this to display all open files associated with processes, revealing how processes interact with these resources.