What Are Networking Commands in Linux & Networking For DevOps

Networking For DevOps

Networking for DevOps

What is a Network?
When two or more computers and computing devices connected together with each other through communication channels, such as cables or wireless media and sharing some files, then it is called a **Network**.

A network is used to:
Allow the connected devices to communicate with each other.

Enable multiple users to share devices over the network, such as music and video servers, printers and scanners.

The Internet is the largest network in the world and can be called "the network of networks".

Types of Networks
There are different types of networks. But the main two are LAN and WAN

1. **LAN** (Local Area Network) - interconnects computer within a limited area, such as residences, schools. e.g.: Wi-Fi, Ethernet
2. **MAN** (Metropolitan area network) - used in metropolitan area (cities).
3. **WAN** (Wide Area Network) - extends LAN over a large geographic area. e.g.: optical fiber cable
4. **SONET** (Synchronous Optical Network) - used in submarine.

Network Components:
1. Switch:
It is a device which connects two or more computers.

2. Router:
It is a device which is actually used to connect one network with another.

3. Modem:
It is also a device used for modulation and Demodulation.

4. Hub:
It is just a power extension dummy device that just broadcast the signals to its connected computers.

5. NIC:
It is known as Network Interface Card which is used to connect your computer with the internet. It is wireless card preinstalled on motherboard now-a-days. It has a MAC (Media Access Control) address.

6. Bridge:
It is also a networking device that connects multiple LANs (local area networks) together to form a larger LAN. It reduces the broadcasting part, and it store the MAC address of the computer but now this device is also obsoleted and replaced by switch.

What is Protocol?
A network protocol is a set of rules which is set up by people that determine how a particular data is transmitted between different devices in the same network. e.g.:
HTTP, TCP, IP, FTP, SMTP etc.

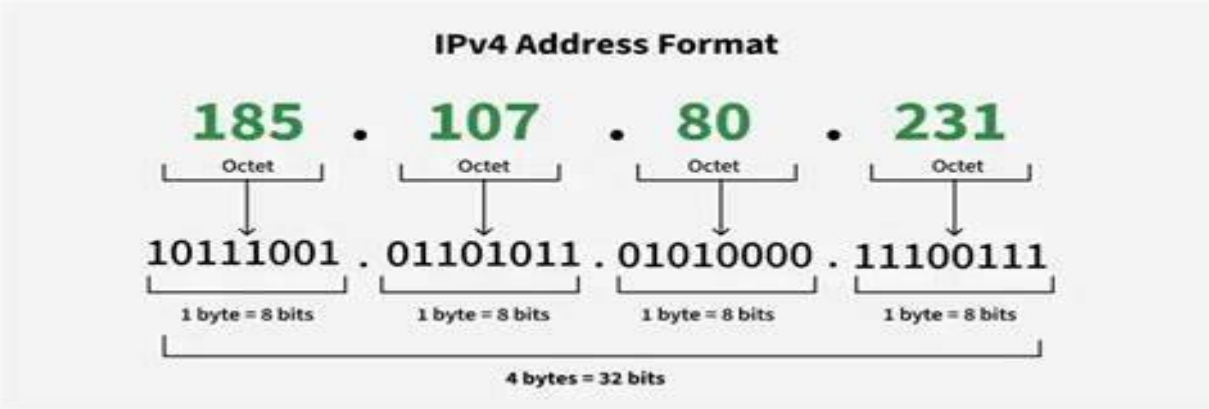
IP Address and its Types and Classes:
IP Address: An IP (Internet Protocol) address is a unique number assigned to each device on a network, allowing them to communicate with each other. It's like a device's"address" on the internet or local network.

Types of IP Addresses

1. **IPv4:**

32-bit address, written as four numbers separated by dots (e.g., 123.89.46.7).
This is a 32-bit IP address, means it contains a combo of 32 (1 and 0's). In this version of IP address there are 4 groups or **Octets** (8 bits), and each octet is represented by a decimal value in the address. It is easy to remember.

IPv4 Address Format (Dotted Decimal Notation)



2. **IPv6:**
   - 128-bit address, written in eight groups of hexadecimal numbers (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334).
   - Provides a vastly larger pool of addresses, designed to replace IPv4 as it runs out

3. **Public IP:**
   - Used to identify devices on the internet.
   - Assigned by ISPs and accessible globally.

4. **Private IP:**
   - Used within private networks (like home or office networks).
   - Not accessible from the internet; usually in ranges like 192.168.NX .

5. **Static IP:**
   - Manually assigned, doesn't change.
   - Often used for servers and devices that need a consistent address.

6. **Dynamic IP:**
   - Automatically assigned by a DHCP (Dynamic Host Configuration Protocol) server.
   - Changes periodically; commonly used for home devices.

IP Address Classes (IPv4 Only)
There is an organization called **IANA** (Internet Assigned Numbers Authority) who divides the IP address into different classes. You have to know about binary to decimal conversion to understand this.\

IPv4 addresses are divided into **five classes** based on the starting number, which determines their usage in networks.

Class	Range	Purpose

A	1.0.0.0 - 126.0.0.0	Large networks, like big organizations.
B	128.0.0.0 - 191.255.0.0	Medium-sized networks.
C	192.0.0.0 - 223.255.255.0	Small networks, like home or business LANs.
D	224.0.0.0 - 239.255.255.255	Reserved for multicasting.
E	240.0.0.0 - 255.255.255.255	Experimental, used for research.

Class	Range (Decimal)	Binary Representation
A	0-127	Starts with 0
B	128-191	Starts with 10
C	192-223	Starts with 110
D	224-239	Starts with 1110
E	240-255	Starts with 1111

Note:

Class A addresses in IPv4 officially start from 1.0.0.0 and go up to 126.0.0.0. The address 0.0.0.0 is not part of the Class A range and has a special purpose in networking.

• 0.0.0.0 is a special address, not part of the usable IP address range in Class A. The 127.0.0.0 to 127.255.255.255 range, especially 127.0.0.1, is reserved for loopback addresses in IPv4.

What is Loopback?

- Loopback address allows a device to communicate with itself.
- It's often used for testing network software on the local machine.

Key Points:

- **127.0.0.1** is commonly known as "localhost."

Any IP address in the **127.x.x.x** range will loop back to the same device.
Useful for **testing** networking applications without needing an external network.

IP address - Network ID and Host ID:
There are two parts to an IP address - Network ID and Host ID (Any device which gets the IP address is called a Host).

The **Network ID** portion differs depending on the IP class:

- **Class A**: 1st octet is the Network ID.
- **Class B**: 1st and 2nd octets are the Network ID.
- **Class C**: 1st, 2nd, and 3rd octets are the Network ID.

Direct Connection: Devices with the same Network ID can connect without a router.

Router Requirement: Devices with different Network IDs need a router to connect.

We will try to break it with text based structural diagram for a better understanding:
IP Address Structure and Device Connection

Class	Network ID	Host ID
Class A	First octet (8 bits)	Remaining three octets (24 bits)
Class B	First two octets (16 bits)	Remaining two octets (16 bits)
Class C	First three octets (24 bits)	Last octet (8 bits)

Device Connection Scenario:

1. Device A IP Address: 17.0.0.1 (Class A)
2. Device B IP Address: 17.0.4.2 (Class A)

Network ID for Class A: 17 (1st Octet)
Since both devices have the same Network ID (17), they are in the same network and can connect directly.

Connection Summary:

Device A IP	Device B IP	Connection Type
17.0.0.1	**17.0.4.2**	**Direct Connection**
17.0.0.1	**192.168.1.5**	**Requires a Router/Switch**

Explanation:

- If Device A and Device B are in the same network (same Netwo
- If they are on different networks (different Network IDs), a

Router Usage:

Network Details	Example	Connection Type
Device A Network ID	17 (Class A)	
Device B Network ID	192.168 (Class C)	
Router Needed	Yes	Required

Subnetting:
Divides a network into smaller, more manageable segments.

Example: A network with IP address 192.168.1.0/24 can be divided into subnets like 192.168.1.0/25 and 192.168.1.128/25.

Example of Subnetting:
Given network: 192.168.1.0/24

192.168.1.0/24 is a Class C network.

/24 indicates a subnet mask of 255.255.255.0, meaning there are 8 bits for hosts (32 total bits in IPv4 - 24 bits for the network portion > 8 bits for hosts).

192.168.1.0/24 provides 256 IP addresses (from 192.168.1.0 to 192.168.1.255).

Dividing into Smaller Subnets:
To divide this into two equal subnets, we can use /25 subnet masks, which allocate 7 bits for hosts (32 - 25 > 7 bits for hosts).

1. **Subnet 1: 192.168.1.0/25**

Range: 192.168.1.0 to 192.168.1.127
Subnet Mask: 255.255.255.128
Total IPs: 128 addresses (126 usable for hosts, as the first address is the network address and the last is the broadcast address).

2. **Subnet 2: 192.168.1.128/25**

Range: 192.168.1.128 to 192.168.1.255
Subnet Mask: 255.255.255.128
Total IPs: 128 addresses (126 usable for hosts).

Summary Table

Subnet	Range	Subnet Mask	Total IPs	Usable Host IPs
192.168.1.0/25	192.168.1.0 - 192.168.1.127	255.255.255.128	128	126
192.168.1.128/25	192.168.1.128 - 192.168.1.255	255.255.255.128	128	126

Explanation:
By using a /25 mask instead of /24, we split the network into two subnets with 128 IP addresses each.
This creates smaller segments within the original network, making it easier to manage specific groups of hosts separately.
Benefits of Subnetting:

1. **Improves Network Performance**: Reduces broadcast domains, limiting broadcast traffic to specific subnets.
2. **Enhances Security**: Allows segregation of different departments or functions within an organization.
3. **Efficient IP Usage**: Prevents wasting IP addresses by only allocating what is necessary for each subnet.

CIDR (Classless Inter-Domain Routing):

CIDR (Classless Inter-Domain Routing) is a method for allocating IP addresses and IP routing that replaces the older **classful network** system. It was introduced to improve IP address utilization and simplify routing.

The table below outlines the most common combination of addresses and netmasks and important details about them.

Prefix	Netmask	Number of addresses	Relation to class	Comment
/32	255.255.255.255	1	Class C/256	Single host in a network
/25	255.255.255.128	128	Class C/2	
/24	255.255.255.0	256	Class C	
/23	255.255.254.0	512	Class C*2	
/16	255.255.0.0	65,536	Class C*256 > Class B	

/15	255.254.0.0	131,072	Class B*2	
/8	255.0.0.0	16,777,216	Class B*256 > Class A	
/0	0.0.0.0	4,294,967,296	Class A*256	0.0.0.0/0 means entire internet. Often used in public firewall rules

Network Models

There are mainly two types of network model -

1. OSI Reference Model
2. TCP/IP Model

1. OSI Reference Model:

The **OSI (Open Systems Interconnection)** Model is a set of rules that explains how different computer systems communicate over a network. OSI Model was developed by the **International Organization for Standardization (ISO)**. The OSI Model consists of 7 layers and each layer has specific functions and responsibilities.

1. **Physical Layer**: Handles the physical connection between devices, transmitting raw data as bits over cables, radio signals, etc.
2. **Data Link Layer**: Manages data transfer between directly connected nodes. It handles error detection and flow control. Examples: Ethernet, Wi-Fi.
3. **Network Layer**: Manages packet forwarding and routing through the network. Uses IP addressing. Example: IP (Internet Protocol).
4. **Transport Layer**: Ensures reliable data transfer with error correction and flow control. Examples: TCP, UDP.
5. **Session Layer**: Establishes, maintains, and manages communication sessions between applications.
6. **Presentation Layer**: Translates data formats to ensure compatibility between systems. Handles encryption and compression. Example: SSL/TLS.
7. **Application Layer**: Interfaces directly with the user and provides network services like HTTP, FTP, SMTP.

Here's a

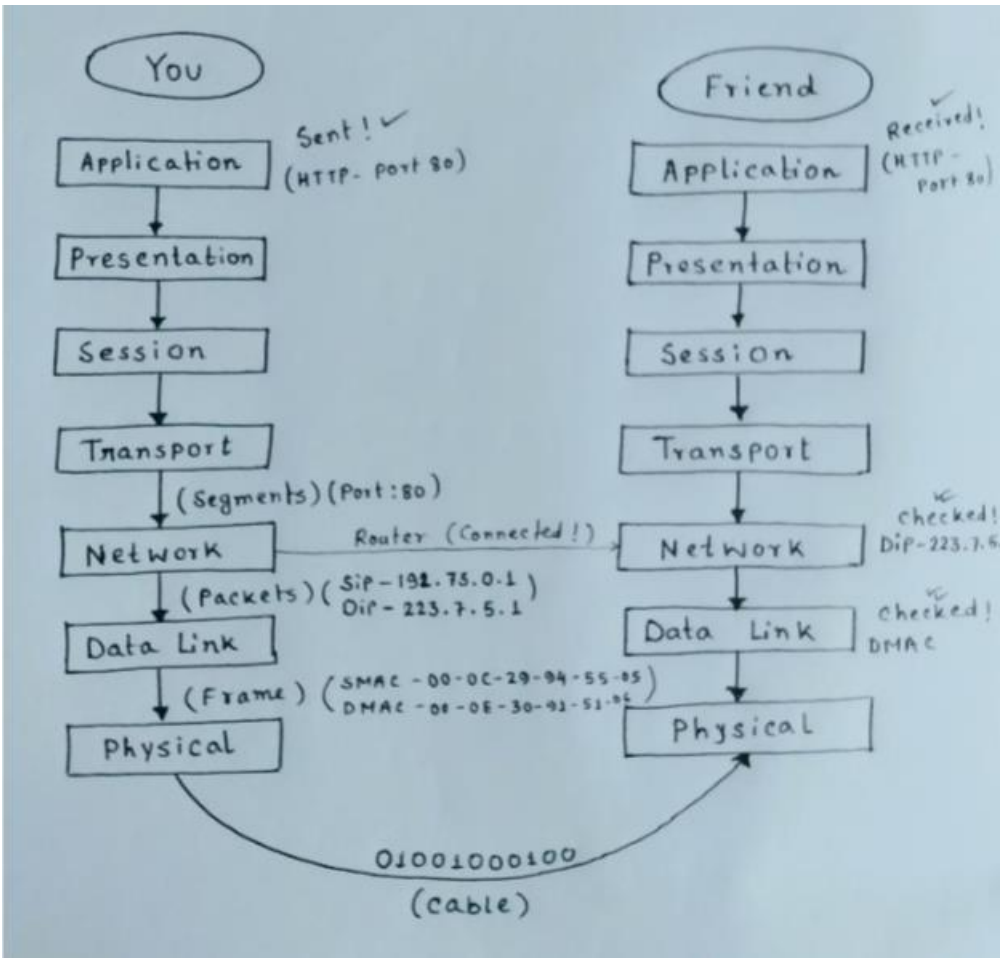
text-based structural diagram that shows the flow through the **OSI model** from **Person X** to **Person Y**:

START: Person X Sends a Message

1. **Application Layer (Layer 7):**
    - Prepare the message using a messaging app protocol (e.g., SMTP, HTTP).
2. **Presentation Layer (Layer 6):**
    - Encode or encrypt the message into a suitable format.
3. **Session Layer (Layer 5):**
    - Establish a session with Person Y's device through the network.
4. **Transport Layer (Layer 4):**
    - Break the message into segments and add TCP/UDP headers for reliable delivery.
5. **Network Layer (Layer 3):**
    - Add source and destination IP addresses for routing across networks.
6. **Data Link Layer (Layer 2):**
    - Convert segments into frames and add MAC addresses for direct device communication.
7. **Physical Layer (Layer 1):**
    - Convert frames into bits (0s and 1s) for transmission.
    - Transmit through the physical medium (e.g., Ethernet cable, Wi-Fi).

Person Y Receives Message

1. **Physical Layer (Layer 1):**
    - Receive bits and reassemble them into frames.
2. **Data Link Layer (Layer 2):**
    - Verify and process frames using MAC addresses.
3. **Network Layer (Layer 3):**
    - Extract source and destination IP addresses for routing.
4. **Transport Layer (Layer 4):**
    - Reassemble segments into the original message using TCP/UDP headers.
5. **Session Layer (Layer 5):**
    - Maintain or terminate the session after receiving the message.
6. **Presentation Layer (Layer 6):**
    - Decode or decrypt the message into a readable format.
7. **Application Layer (Layer 7):**
    - Deliver the message to Person Y through their messaging app.



The 7-Layer OSI Model

No.	Layer	Function	Data unit	Hardware	Protocols
7	Application	Human-computer interaction through applications that access network services	Message/data	Gateway	UPnP, DHCP, DNS, HTTP, HTTPS, NFS, NTP, POP3, SMTP, SNMP, FTP, Telnet, SSH, TFTP, IMAP
6	Presentation	Data formatting and encryption/ decryption	Message/data	Gateway redirector	TLS, SSL, AFP

5	Session	Inter-host communication	Message/data	Gateway	NetBIOS, RPC, SMB, Socks
4	Transport	Data transmission	TCP: segment; UDP: datagram	Gateway	TCP, UDP, SCTP
3	Network	Path determination and logical addressing	Packet, datagram	Router, Brouter	ARP, IP, NAT, ICMP, IPsec, ICMP (ping)
2	Data Link	Physical addressing	Frame, cell	Switch, bridge, NIC	ARP, Ethernet, LZTP, LLDP, MAC, NDP, PPP, PPTP, VTP, VLAN
1	Physical	Binary signal transmission over physical media	Bit, frame	Cables, modem, hub, repeater, NIC, multiplexer	Ethernet, IEEE802.11, ISDN, USB, Bluetooth

Below is the list of protocols in each layer of the **OSI model** along with their **port numbers** (where applicable):

1. Application Layer (Layer 7)

HTTP (Port 80): Web browsing.
HTTPS (Port 443): Secure web browsing.

SMTP (Port 25): Sending email.
FTP (Ports 20, 21): File transfer.
DNS (Port 53): Domain name resolution.
POP3 (Port 110): Receiving email.
IMAP (Port 143): Receiving email.

2. Presentation Layer (Layer 6)

SSL/TLS (Port 443 for HTTPS, also used in other protocols): Encryption for secure data transmission.
MIME: Used for formatting email attachments.
JPEG/PNG: Image formats used to encode multimedia files.

3. Session Layer (Layer 5)

PPTP (Port 1723): Tunneling protocol for VPNs.
NetBIOS (Ports 137, 138, 139): Establishes sessions for network communications.

4. Transport Layer (Layer 4)

TCP: Reliable data transmission with acknowledgment.
UDP: Fast, connectionless data transmission without acknowledgment.
SCTP: Used for applications that require multiple data streams.

5. Network Layer (Layer 3)

IP (IPv4/IPv6): Routing packets between source and destination.
ICMP: Error messaging and diagnostics (e.g., ping).
IGRP: Routing protocol used for sharing routing information.

6. Data Link Layer (Layer 2)

Ethernet: Defines physical addressing and channel access.
PPP: Used for point-to-point connections.
HDLC: For framing and error control on point-to-point links.
ARP: Resolves IP addresses to MAC addresses.

7. Physical Layer (Layer 1)

Ethernet (Physical signaling): Specifies electrical signals, cabling, etc.
USB: Used to physically connect devices.
Summary with Ports

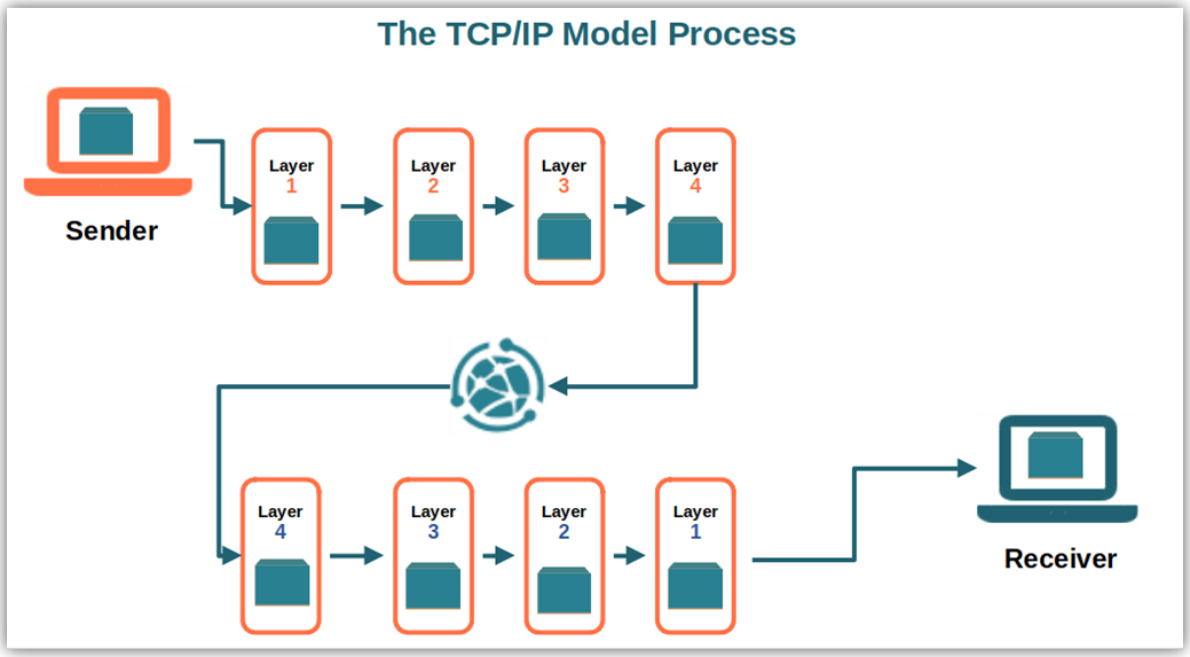
Application Layer (Layer 7) contains the most recognizable protocols with specific **port numbers** for communication (e.g., HTTP - Port 80, HTTPS - Port 443, SMTP - Port 25).
Layers 2 to 6 typically deal with specific network management functions and do not use port numbers as these layers are responsible for connections and managing data formats.
Port numbers are used primarily in the Application and Transport Layers to ensure data is delivered to the correct services and applications running on a computer.

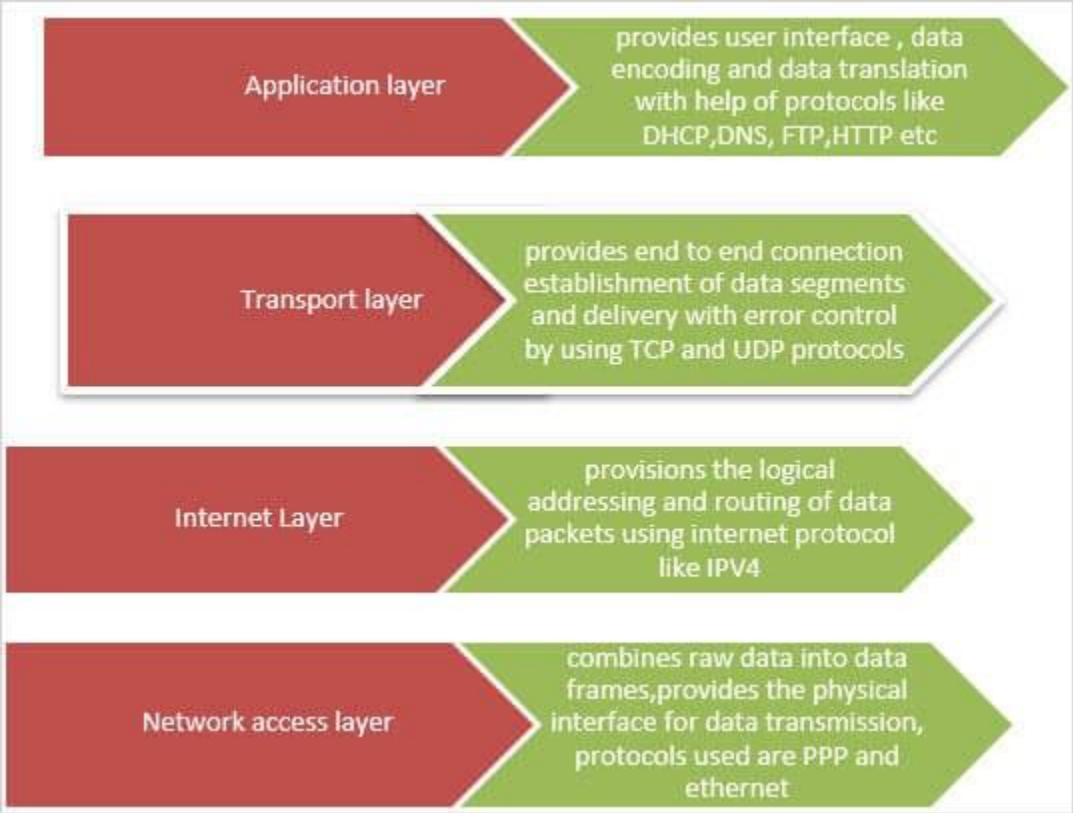
2. TCP/IP Model:

The **TCP/IP model**, also known as the Internet Protocol Suite, is a simplified version of the OSI model with only 4 layers instead of 7.

This model is a real model which actually works in real. This model consists of 4 layers.

1. Application Layer = (Application Layer + Presentation Layer + Session Layer) of OSI model
2. Transport Layer
3. Network Layer
4. Network Interface Layer = (Data Link Layer + Physical Layer) of OSI





Ports and Protocols:

1. **HTTP (Hypertext Transfer Protocol):**

It is a client server stateless (means it never stores any data of client) protocol, and it tells us how it requests any data from the server and also tells us how the server will send the data back to the client.

-
    - When a client makes a request - HTTP request
-
    - When server sends response to client - HTTP response -- Some HTTP methods used to make any request

GET :- Get some data from server
POST :- Post some form/data to server
PUT :- Put some data
DELETE :- Delete some data in server

- Status Codes

Status codes are issued by a server in response to a client's request made to the server.
There are 4 categories of HTTP responses:

1. 200s: Successful responses
2. 300s: Redirects
3. 400s: Client errors
4. 500s: Server errors

Take a look at some of the most common response codes:

Code	What It's Telling	What it Means
200	OK	Request succeeded.
302, 307	Found, Temporary Redirect	The URI of the requested resource has been changed temporarily.
301, 308	Moved Permanently, Permanent Redirect	The URI of the requested resource has been changed permanently.
400	Bad Request	The server can’t understand the request being sent.
401	**Unauthorized**	**The client must authenticate itself before sending the request.**
403	Forbidden	The client does not have enough permission to access the content.
404	Not Found	The server can’t find the requested resource.
408	Request Timeout	The response was sent to an idle connection, and the server wants to terminate it.
500	Internal Server Error	The server does not know how to handle a request.
502	Bad Gateway	The server you are trying to access is a gateway or a reverse proxy (it sits between the client and an actual server that serves the page). You get this error when the gateway gets an incorrect response from a source server.
503	Service Unavailable	The server can’t process the request. This usually happens when a server is down or overloaded.
504	Gateway timeout	Similar to 502, the gateway can’t get a response in time.

2. **SMTP/POP (Simple Mail Transfer Protocol and Post Office Protocol):**

SMTP is used in sending and receiving any email from senders SMTP server to Receiver's SMTP server
POP is used to download any email from POP server

3. **FTP (File Transfer Protocol):**

FTP is used to download, upload and transfer files from one host to another host.

4. **Secure shell (SSH):**

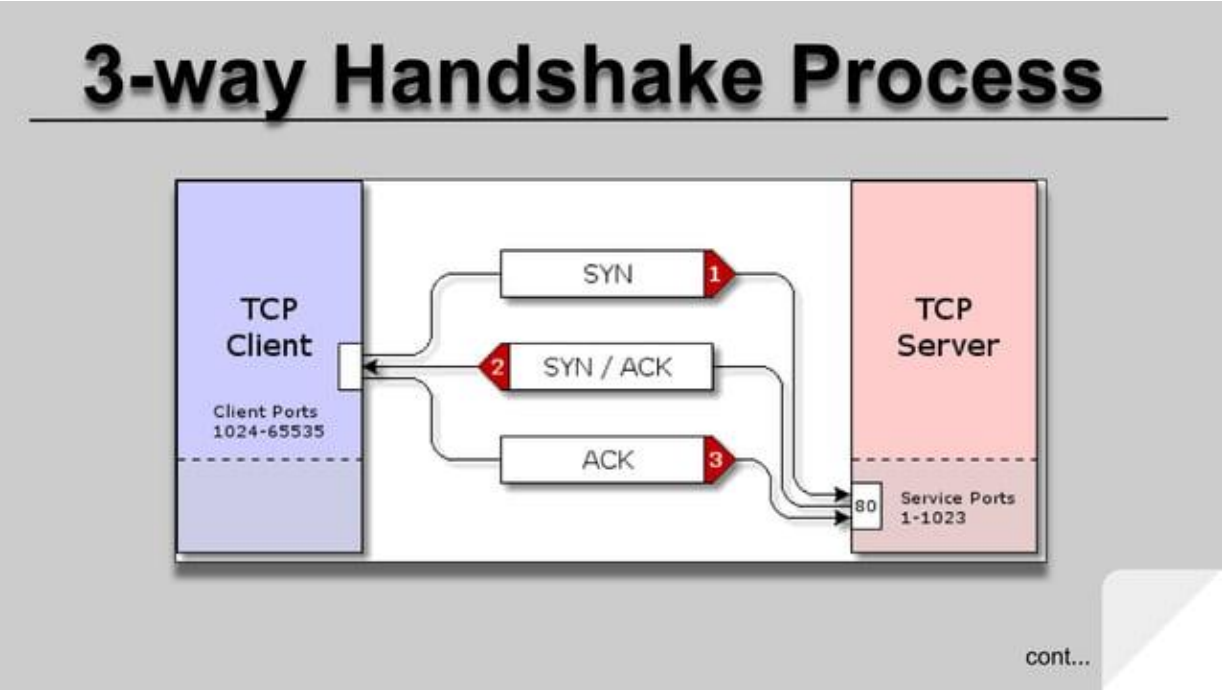
Similar to Telnet. It’s used by system administrators to securely access to access a computer over an insecure network.

5. **TCP (Transmission Control Protocol):**

Reliable and connection-oriented protocol.

Ensures data is delivered successfully with acknowledgements and retransmission if needed.
Used for applications where data integrity is crucial.
Example: HTTP (port 80), SQL (port 1433).

TCP and the Three-Way Handshake:



How TCP Three-Way Handshake Works
Client

1. The client sends a SYN packet to the server asking him to open a connection (SYN)
2. The server responds acknowledging receipt of the packet and confirming that it's ready to connect. (SYN-ACK)
3. The client responds to validate the connection and the transmission of packets starts. (ACK) Server

6. **UDP (User Datagram Protocol):**

- Unreliable and connectionless protocol.
- No acknowledgements, no retransmission; faster but no guarantee of delivery.
- Used for applications needing speed over reliability.
- Example: Video Streaming or DNS queries.

Routing:
So how do we get a packet of information from a host on one network to a host in another? In one word: Routing.
We use tables to help us determine the routes we want to take. This screenshot demonstrates a typical route table in AWS:

Destination	Target
10.21.0.0/16	local
10.0.0.0/8	tgw-02b7d78edbc6fa7b3
0.0.0.0/0	nat-Oa9a7a4b5947eOaa5

When making a routing decision, more narrow rules are evaluated first:
If a packet destination is in a range of 10.21.0.0/16 – it will remain in a local network (your neighborhood).

If a packet destination is in a range of 10.0.0.0/8 – it will be sent to the transit gateway (TGW) interface (your state highways).

If a packet destination does not fall in any of these ranges, the widest one is evaluated which is 0.0.0.0/0 which means it is internet traffic. And the packet will be redirected to the Network Address Translation (NAT) interface.

DNS (Domain Name System):

DNS (Domain Name System) translates human-readable domain names (e.g., **www.example.com**) into IP addresses (for example, 192.0.2.44).

Root DNS Server stores all the Top-level domain e.g : - .com, .in, .org, .io etc.
DNS works like the **phonebook** of the internet, allowing humans to use readable names while machines use numerical addresses.

How DNS Works

- When you type a **website address** (e.g., www.example.com) into your browser, it needs to know the **IP address** of that server.

The DNS process involves looking up the domain name and finding the corresponding IP address through multiple DNS servers.

Example: Visiting a Website

1. **User Request**:
   You type **www.example.com** into your web browser.

2. **DNS Query**:

Your browser sends a request to a DNS server to get the IP address of www.example.com .

3. **DNS Resolution:**

The DNS server checks if it has the IP address cached. If not, it contacts other DNS servers (root, TLD, and authoritative servers) to find the IP address.

4. **IP Address Found:**

Once the IP address (e.g., 93.184.216.34) is found, the DNS server sends it back to your browser.

5. **Connecting to the Website:**

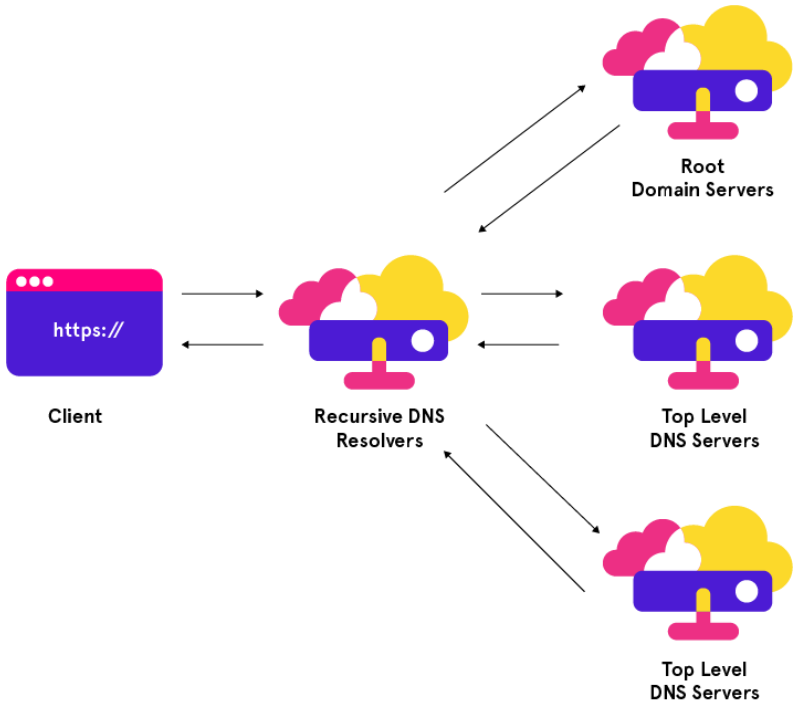
The browser uses this IP address to connect to the web server, and the website loads on your screen.

Domains, Zones, and Delegation:

1. **Domains**: Domains are like branches in a tree-like structure of the internet. The root domain is the highest level, followed by top-level domains (TLDs) like .com /.org , etc. Subdomains (e.g., example.com) branch off from TLDs.
2. **Zones**: A zone is a portion of the domain that is managed by a specific organization. For instance, .com is a zone controlled by Verisign. ICANN manages the root zone at the top of the DNS tree, while different organizations manage subdomains.
3. **Delegation**: Delegation allows one organization to hand over control of part of its domain to another organization. This is done using Nameserver (NS) records.

- For example, ICANN controls the root domain and delegates .com to Verisign.
- Verisign can then delegate control over example.com to "Example Ltd" by adding an NS record those points to their nameserver.

The NS records direct traffic to the appropriate nameserver that manages a domain, allowing different parts of the DNS tree to be managed independently by different organizations.'



Below are the most commonly used types of DNS records and their meaning:

Type	Name	Description
A	Host address	The most basic and the most commonly used DNS record. It translates human-friendly domain names into computer-friendly IP addresses.
AAAA	IPv6 host address	Same as A but for IPv6 (a host address that can have more than one IP address).
CNAME	Canonical name for an alias	Maps a name to another name. It should only be used when there are no other records on that name.
ALIAS	Auto resolved alias	Maps a name to another name but can coexist with other records on that name.
MX	Mail eXchange	Specifies the e-mail server(s) responsible for a domain name.
NS	Name Server	Identifies the DNS servers responsible for a zone. One NS record for each DNS server in a zone.
TXT	Descriptive Text	Holds general information about a domain name such as who is hosting it, contact person, phone numbers, etc. Widely used for domain ownership verification.

DHCP

DHCP (Dynamic Host Configuration Protocol) is a network management protocol that **automatically assigns IP addresses** and other network configurations (such as **subnet mask**, **gateway**, **DNS servers**) to devices on a network.

Example:
When you connect your laptop to a Wi-Fi network, a **DHCP server** assigns it an **IP address** automatically, allowing it to communicate with other devices on the network without manual configuration.

Network Components and Services

Routers and Switches

Routers: Connect different networks and direct data packets between them.

Switches: Connect devices within the same network and use MAC addresses to forward data to the correct device.

Firewalls
Firewalls control incoming and outgoing network traffic based on predetermined security rules.

Load Balancers
Load balancers distribute incoming network traffic across multiple servers to ensure no single server becomes overwhelmed.

VPN
VPN (Virtual Private Network) provides a secure connection between remote users and the corporate network over the internet.

Network Troubleshooting Tools:

1. ping
Purpose: Test internet network connections.
How It Works: Uses the **ICMP ECHO_REQUEST** to get an **ICMP ECHO_RESPONSE** from a remote host.
Usage: For basic troubleshooting, you can run ping www.google.com to check network connectivity and see response times and packet loss.

2. traceroute (or tracert on Windows)
Purpose: Track the route packets take to reach their destination.
How It Works: Sends **UDP probes** with increasing TTL values, showing each router along the route and the delay in reaching it.
Usage: Helps find which gateway is causing a delay by showing response times and where packets fail (indicated by).

3. telnet
Purpose: Test network connections and protocols.
How It Works: Attempts to establish a connection to a specified IP and port.
Usage: Test if a specific service is reachable, e.g., telnet google.com 443 .

4. curl

- **Purpose:** Transfer data using multiple protocols, often for HTTP requests.
- **Usage:**
  - Basic GET request: [curl http://example.com] .
  - Check headers: [curl -I http://example.com] .
  - POST request: [curl -X POST http://example.com] .
  - Save response to file: [curl http://example.com/file -o output_file] .
  -

5. dig (Domain Information Groper)

- **Purpose:** Troubleshoot DNS problems and verify DNS records How It Works: Performs DNS lookups and displays information such as IP addresses.
• Usage: dig-google.com to get information like IP addresses, TTL, and DNS record types.

6. **netstat**

Purpose: Show network connections and port listening information.
Usage:
o netstat -ip : List listening servers and their program names. netstat -a :
o Show all active ports.
o netstat-r : Show routing table.

7. **nmap (Network Mapper)**

Purpose: Discover hosts and services on a network.
How It Works: Sends raw packets to identify hosts, services, and operating systems.
Usage:
o Discover hosts: nnap-sn_172.31.44.35/20 .
o Scan ports on a host: nnap-A_172.31.36.237 .

8. **ssh (Secure Socket Shell)**
   • Purpose: Securely connect to remote machines to execute commands.

• Usage:
o Connect to a server: ssh-username@hostname .
o Secure and encrypted, used for remote management and file transfers.

9. scp (Secure Copy Protocol)

Purpose: Securely copy files between local and remote hosts.
Usage:
o Copy file to a remote server: scp_localfile.txt-user@remote:/path/to/destination .
These tools are invaluable for network diagnostics, troubleshooting, and secure communications, which are critical skills for any DevOps engineer.

Linux Networking Commands Cheat Sheet

This list of Linux networking commands will help you work with network connections. You can use these commands to set up the networks, fix connection problems, and see the information about the network. If you are an experienced network admin or just starting to learn the Linux networking commands this command list will be a useful tool for you.

Network Configuration Commands

Command	Description	Example
hostname	Shows the Hostname of the current computer system.	**hostname** Displays the hostname of the current system.
ip	Displays and allows you to configure network interfaces, routing, and tunnels.	**ip addr show** Shows the IP addresses assigned to network interfaces.

iwconfig	Shows and lets you set up wireless network interfaces.	**iwconfig wlan0 essid "WiFiName"** Connects the wlan0 interface to the WiFi network named "WiFiName".

Command	Description	Example
netplan	Sets up network interfaces using YAML configuration files.	**netplan apply** Applies network settings from the YAML files.
ifconfig	Shows and configures the network interfaces on the current system.	**ifconfig eth0** Displays information about the eth0 interface.
ifquery	Checks the current setup and settings of network interfaces.	**ifquery --list** Lists all network interfaces and their settings.

nmcli	Reports the network status, manages and configures the network connections and controls the NetworkManager tool.	**nmcli dev wifi list** Lists available WiFi networks.
netplan	This Defines the network interfaces, IP addresses, gateways, DNS servers, and the other network related settings in the YAML config file.	(Example covered in description)
Command	**Description**	**Example**
ifup	This will Activates a network interface that is currently inactive or down.	**ifup eth0** Activates the eth0 network interface.
ifdown	Deactivates a network interface that is currently active or up.	**ifdown wlan0** Deactivates the wlan0 wireless interface.
dhcclient	Automatically gets an IP address and other network configuration details from a DHCP server.	**dhclient eth0** Requests an IP address for the eth0 interface from a DHCP server.

Network Connectivity Commands

Command	Description	Example
arp	Maps IP addresses to MAC addresses on a local network and manages the ARP cache.	**arp** Shows the current ARP cache entries.
arp -a	Prints the ARP table entries.	(Example covered in description)
arp -a -d	Removes all entries from the ARP table.	**arp -d 192.168.1.100** Deletes the ARP entry for IP 192.168.1.100.

Command	Description	Example
arp -s	Adds a new entry to the ARP table.	**arp -s 192.168.1.10 00:11:22:33:44:55** Adds a static ARP entry for IP 192.168.1.10 and MAC 00:11:22:33:44:55.
netstat	Shows active network connections, routing information, and other network statistics.	**netstat -antp** Lists all active network connections and the processes using them.

ping	Tests network connectivity between a host/client and server by sending data packets and measuring the response time.	**ping 8.8.8.8** Tests connectivity to the Google DNS server at 8.8.8.8.
route	Configures the kernel's routing table entries.	**route add default gw 192.168.1.1** Sets the default gateway to 192.168.1.1.
route flush	Removes all routes from the routing table.	(Example covered in description)
traceroute	Displays the path and hops that packets take to reach a	**traceroute**

Command	Description	Example
	remote server.	**www.example.com** Traces the route taken to reach www.example.com.
telnet	Establishes the connection to the remote server or system through a TCP/IP network using the Telnet protocol.	**telnet 192.168.1.10 22** Connects to the SSH server on 192.168.1.10 via Telnet.

w	This will Shows the information about users currently logged into the system.	(Example covered in description)
mail	Sends and receives email messages via the command line interface.	**mail -s "Hello" user@example.com** Sends an email with subject "Hello" to user@example.com.
ngrep	Displays and filters network packet data based on a given regular expression pattern.	**ngrep -W byline "^GET"** Captures and prints HTTP GET requests.
iw	Displays and configures wireless network interface settings.	**iw dev wlan0 scan** Scans for available wireless networks on the wlan0 interface.

DNS and Name Resolution Commands

Command	Description	Example
nslookup	Performs DNS lookups to resolve hostnames, IP addresses, mappings, or any specific DNS record.	**nslookup example.com** Resolves the IP address for example.com

host	Performs DNS lookups to resolve hostnames or IP addresses.	**host google.com** Resolves the IP address for google.com
dig	Performs DNS lookups to query information about DNS name servers.	**dig @8.8.8.8 example.org** Queries Google's DNS server for information about example.org

Network Information Commands

Command	Description	Example
finger	Displays user login information on a remote system.	**finger user@example.com** Shows login information for user on example.com
Command	**Description**	**Example**
whois	Retrieves information about domain ownership from various WHOIS servers.	**whois example.org** Shows ownership details for example.org

Network Testing and Monitoring Commands

Command	Description	Example

hping	Assembles and analyzes TCP/IP packets, sends packets to a remote host, and examines the responses.	**hping3 -S -p 80 example.com** Sends SYN packets to port 80 on example.com
mtr	Combines functionalities of traceroute and ping commands, continuously sends packets and shows the ping time for each hop.	**mtr google.com** Traces the route to google.com and shows ping times
nc / netcat / ncat	Allows reading and writing data across network connections.	**nc -l -p 8080** Listens on port 8080 for incoming connections

Command	Description	Example
smokeping	Measures network latency and packet loss between two hosts, performs checks at set intervals.	**smokeping example.com** Monitors latency and packet loss to example.com

socat	Facilitates data transfer between two bidirectional byte streams.	**socat TCP4-LISTEN:8080,fork EXEC:/bin/bash** Opens a remote shell on port 8080
speedometer	Displays bandwidth usage in real-time.	**speedometer eth0** Shows bandwidth usage on eth0 interface
speedtest-cli	Measures network performance by conducting a speed test.	**speedtest-cli** Runs a speed test to measure download/upload speeds
ss	Shows active network connections, socket statistics, and other network-related information.	**ss -tulpn** Lists all listening TCP/UDP sockets with process information
tracepath	Traces the route packets take, discovering the MTU	**tracepath**
Command	**Description**	**Example**
	along the path.	**example.net** Traces the path to example.net, showing MTU

Network Analysis and Monitoring Commands

Command	Description	Example
tcpdump	Captures and inspects network traffic in real-time for analysis.	**tcpdump -i eth0 port 80** Captures traffic on eth0 interface for port 80
nmap	Discovers active hosts and services on a network, useful for security audits.	**nmap -sV example.com** Scans example.com for open ports and services
bmon	Displays real-time bandwidth consumption for individual network interfaces.	**bmon** Shows bandwidth usage per interface
bwm-ng	Monitors the current bandwidth utilization across multiple network interfaces.	**bwm-ng** Displays bandwidth usage for all interfaces
Command	**Description**	**Example**

iftop	Shows real-time bandwidth usage broken down by individual connections.	**lsof -i** Lists all open network connections
iperf	Measures the network performance and throughput between two systems.	**iperf -c example.com** Tests network performance to example.com
iptraf-ng	Monitors and displays network traffic flows in real-time.	**iptraf-ng** Shows real-time network traffic statistics
nethogs	Tracks network traffic and bandwidth usage per process or application.	**nethogs** Displays bandwidth usage per process/application
vmstat	Keeps historical records of network traffic statistics for selected interfaces, displaying hourly, daily, and monthly summaries.	**vnstat -i eth0** Shows traffic statistics for eth0 interface

Remote Access Commands

Command	Description	Example

ssh	Establishes a secure encrypted shell session with a remote system over the SSH protocol.	**ssh user@example.com** Opens an SSH session to example.com as user
scp	This will Securely copies files between the local and remote systems using the SSH protocol.	**scp file.txt user@example.com:/path/to/dir** Copies file.txt to example.com
sftp	This will Securely transfers files between the hosts using the SFTP (SSH File Transfer Protocol).	**sftp user@example.com** Opens an SFTP session to example.com as user

Security Commands

Command	Description	Example
iptables	The firewall utility that manages the packet filtering and the Network Address Translation (NAT) rules.	**iptables -L** Lists all current iptables rules
Command	**Description**	**Example**

snort	An intrusion detection system that analyzes the network traffic for the suspicious activities.	**snort -dev** Runs Snort in packet logging mode
wireshark	Captures and analyzes network traffic data in a formatted text output.	**wireshark** Starts Wireshark for packet capturing and analysis
ufw	Manages the system's firewall, allowing you to add, delete, modify, or reset packet filtering rules.	**ufw allow 22/tcp** Allows incoming connections on port 22/TCP

Utility Commands

Command	Description	Example
curl	Transfers data to or from a server over various protocols.	**curl https://example.com** Fetches the website example.com
aria2	Supports multiple protocols for downloading files from the internet.	**aria2c https://example.com/file.zip** Downloads file.zip from example.com

Command	Description	Example
ethtool	Modifies and queries settings of network interfaces, such as speed and other parameters.	**ethtool -s eth0 speed 1000** Sets eth0 interface speed to 1000Mbps
wget	A command-line utility for downloading files from the web.	**wget https://example.com/file.zip** Downloads file.zip from example.com