Day 10-Practical Networking for DevOps

TCP/IP Model Overview

The TCP/IP model is the foundation of network communication in modern systems. As a DevOps engineer, it's important to understand this model to troubleshoot connectivity, deploy services, and manage infrastructure effectively.

The TCP/IP model has 4 layers:

1. Application Layer
2. Transport Layer
3. Internet Layer
4. Network Access (Link) Layer

Each layer handles specific functions and interacts with the layers above and below it.

Layer 1: Application Layer

- Interfaces directly with user applications (e.g., browsers, curl, APIs).
- Protocols: HTTP, HTTPS, FTP, SSH, DNS, SMTP
- DevOps Relevance:
    - Understanding service ports and protocols (e.g., NGINX on port 80/443, SSH on port 22)
    - Monitoring with tools like curl, telnet, dig, or nc
    - Configuring reverse proxies, load balancers, and APIs

Layer 2: Transport Layer

- Responsible for end-to-end communication
- Protocols: TCP (connection-oriented), UDP (connectionless)
- DevOps Relevance:
    - Troubleshooting connectivity (e.g., TCP port checks)
    - Choosing protocols for services (e.g., DNS uses UDP, HTTP uses TCP)
    - Using tools like netstat, ss, or nmap to verify open ports

Layer 3: Internet Layer

- Handles addressing and routing between networks
- Protocols: IP, ICMP
- DevOps Relevance:
    - Configuring static IPs, CIDR blocks, default gateways
    - Debugging with ping, traceroute, and ip route
    - Working with cloud VPCs and CIDR ranges

Layer 4: Network Access (Link) Layer

- Manages communication between devices on the same network (physical + data link)
- Protocols: Ethernet, ARP
- DevOps Relevance:
    - Understanding MAC addresses, switches, and ARP resolution
    - Diagnosing network hardware or cable issues

  - Using tools like ip link, arp, and ethtool

IP Addressing

IP addressing is essential to networking. As a DevOps engineer, understanding how IP addresses are assigned, structured, and used helps correctly configure servers, containers, networks, and routing.

- What is an IP Address?
  - An IP address is a unique identifier for a device on a network.
- Two types of IP versions:
  - IPv4: 32-bit address (e.g., 192.168.1.10)
  - IPv6: 128-bit address (e.g., 2001:0db8:85a3::8a2e:0370:7334)

IPv4 Address Structure

- Written in dotted decimal format: w.x.y.z
- Each part ranges from 0 to 255 (e.g., 192.168.1.100)
- Classes:
  - Class A: 0.0.0.0 – 127.255.255.255 (Large networks)
  - Class B: 128.0.0.0 – 191.255.255.255 (Medium networks)
  - Class C: 192.0.0.0 – 223.255.255.255 (Small networks)
- Private IP Ranges (commonly used in internal networks):
  - 10.0.0.0/8
  - 172.16.0.0/12
  - 192.168.0.0/16

CIDR Notation

- CIDR = Classless Inter-Domain Routing
- Format: IP/Prefix (e.g., 192.168.1.10/24)
- The /24 means the first 24 bits are the network portion
- DevOps Relevance:
  - CIDR is used in configuring subnets, firewalls, and cloud VPCs

Subnetting

- Subnetting divides a large network into smaller ones
- Helps organize and secure infrastructure (e.g., frontend and backend separated)
- For example:

  -
    ```
    ipcalc <IP>/<CIDR>       # Show network, broadcast, usable IPs (requires ipcalc)
    ```

Loopback and Special IPs

- 127.0.0.1: Loopback (localhost)
- 0.0.0.0: All interfaces/default route
- 255.255.255.255: Broadcast
- Link-local (IPv6): fe80::/10

DNS Basics

Basic Connectivity

- You can use `ping <IP>` to check network connectivity with another host.
- For example:
  ```
  ping 8.8.8.8
  ```
- Instead of using an IP address, you can assign a user-friendly hostname by adding an entry in the /etc/hosts file:
  ```
   192.168.1.100  myhost
  ```
- For example:
  ```
  ping myhost
  ```
- The `/etc/hosts` file is the local source of truth for hostname-to-IP mapping.
  This process is known as name resolution.

DNS – Domain Name System

- On small systems, /etc/hosts works fine, but at scale it's inefficient to manage.
- To centralize hostname resolution, we use a DNS server.
- DNS settings are configured in /etc/resolv.conf as:
  ```
  nameserver 192.168.1.100
  ```
- For example:
  ```
  cat /etc/resolv.conf
  ```
- Entries in /etc/hosts are checked before DNS servers. This order is defined in /etc/nsswitch.conf:
  ```
  hosts: files dns
  ```

  You can modify this to give DNS preference:
  ```
  hosts: dns files
  ```

DNS Search Domains

- Add a search domain to /etc/resolv.conf:
  ```
    search myfoodcompany.com
  ```
- For example:
  ```
   ping food      # Will try to resolve food.myfoodcompany.com
  ```
- You can specify multiple search domains.
- For example:
  ```
  search myfoodcompany.com mydevlab.internal
  ```

Common DNS Record Types

A → Maps hostname to an IPv4 address
AAAA → Maps hostname to an IPv6 address
CNAME → Maps an alias to another hostname

DNS Query Tools

nslookup

- For example:
  ```
    nslookup http://google.com
  ```
  ```
   nslookup myhost.local
  ```

dig

- For example:

```
dig http://github.com
```

Interfaces, IP Addressing, and Routing

Viewing Network Interfaces

- Use the following command to see the network interfaces available on a host:

  ```
  ip link
  ```

  Example:

  ```
  ip link show
  ```

Assigning IP Addresses

- To assign an IP address to an interface:

  ```
  ip addr add <IP/CIDR> dev <interface>
  ```

  Example:

  ```
  ip addr add 192.168.1.100/24 dev eth0
  ```

Switches and Routing

- Switches enable communication within a network (Layer 2).
- Routing is used to enable communication across networks (Layer 3).
- A router has one IP address per connected network interface.

Gateway

- A gateway is a device that routes traffic from a local network to other networks.
- It acts as a "door" between networks.

Routing Configuration

- To view current routing table:

  ```
  route
  ```

  or (modern command):

  ```
  ip route show
  ```

- To add a static route:

  ```
  ip route add <network/CIDR> via <gateway_IP>
  ```

  Example:

  ```
  ip route add 192.168.10.0/24 via 192.168.1.1
  ```

- To set a default gateway:

  ```
  ip route add default via 192.168.10.1
  ```

  Alternatively:

  ```
  ip route add 0.0.0.0/0 via 192.168.10.1
  ```

Persisting Changes

- To make routing or interface changes persistent, add them to:

  ```
  /etc/network/interfaces
  ```

Managing Network Interfaces

- To bring an interface up:

  ```
  ip link set dev <interface> up
  ```

  Example:

  ```
    ip link set dev eth0 up
  ```

- To bring an interface down:

  ```
  ip link set dev <interface> down
  ```

  Example:

  ```
    ip link set dev eth0 down
  ```