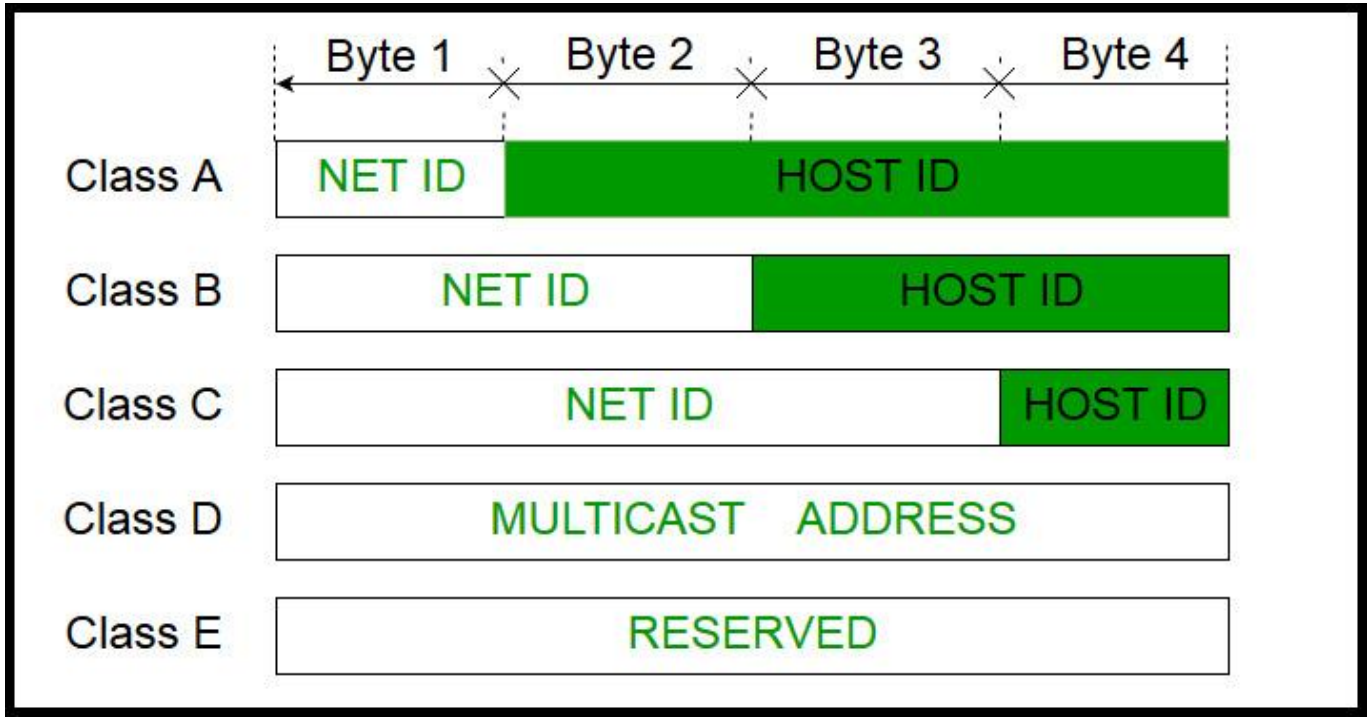
IP Adressing

Subnets and Supernets

Classful IP Address



Key Facts for Classes A, B, and C

	Class A	Class B	Class C
First octet range	1 – 126	128 – 191	192 – 223
Valid network numbers	1.0.0.0 – 126.0.0.0	128.0.0.0 – 191.255.0.0	192.0.0.0 – 223.255.255.0
Total networks	$2^7 - 2 = 126$	$2^{14} - 2 = 16{,}384$	$2^{21} - 2 = 2{,}097{,}152$
Hosts per network	$2^{24} - 2$	$2^{16} - 2$	$2^8 - 2$
Octets (bits) in network part	1 (8)	2 (16)	3 (24)
Octets (bits) in host part	3 (24)	2 (16)	1 (8)
Default mask	255.0.0.0	255.255.0.0	255.255.255.0

Classless IP Address

Classless IP addressing is a modern method of assigning IP addresses that **doesn't rely on predefined classes (A, B, C)**. Instead, it uses **CIDR (Classless Inter-Domain Routing)** notation to allocate IPs based on actual need.

Key Features:

- Uses **prefix length** (e.g., `/27`) to define the network portion
- Supports **Variable Length Subnet Masking (VLSM)**
- Enables **efficient IP allocation** and **route aggregation**

CIDR notation looks like this: `192.168.1.0/27`

- `192.168.1.0` → Network address
- `/27` → First 27 bits are the **network prefix**, remaining 5 bits are for **hosts**
- Total IPs = 25=322^5 = 32 (30 usable after subtracting network & broadcast)

Scenario: You Need 300 IP Addresses

Classful Addressing

- You'd be forced to choose a **Class B** block:
  - Range: `128.0.0.0 – 191.255.255.255`
  - Subnet Mask: `255.255.0.0`
  - Total IPs: **65,536**
- You get your 300 IPs...
- But you waste **65,236 IPs** that you don't need!
- Class C networks cannot be used because Class C has only 8 bits for the host **Total IPs**: 2^8=256

Classless Addressing

CIDR lets you go beyond rigid class boundaries:

- `/23` subnet so 9 host bits and 23 network bits
- Number of host IPs = 2^9=512
- Usable IPs = 510
- Wasted IP = 210
- Example: `192.168.1.0/23` → Covers `192.168.1.0` to `192.168.2.255`

CIDR gives you just enough IPs with minimal waste and better routing

Subnetting

- You **borrow bits** from the **host portion** of the IP address.
- This creates **multiple subnets** within the same class.

Example:

- Class C network: `192.168.1.0/24`
- Subnet into:
  - `192.168.1.0/26` → 64 IPs
  - `192.168.1.64/26` → 64 IPs
  - `192.168.1.128/26` → 64 IPs
  - `192.168.1.192/26` → 64 IPs

Subnet	IP Range Start	IP Range End	Usable IPs	Broadcast Address
`192.168.1.0/26`	`192.168.1.0`	`192.168.1.63`	`192.168.1.1 – 192.168.1.62`	`192.168.1.63`
`192.168.1.64/26`	`192.168.1.64`	`192.168.1.127`	`192.168.1.65 – 192.168.1.126`	`192.168.1.127`
`192.168.1.128/26`	`192.168.1.128`	`192.168.1.191`	`192.168.1.129 – 192.168.1.190`	`192.168.1.191`
`192.168.1.192/26`	`192.168.1.192`	`192.168.1.255`	`192.168.1.193 – 192.168.1.254`	`192.168.1.255`

What Is a Subnet Mask?

A **subnet mask** is a 32-bit number that tells devices which part of an IP address is the **network** and which part is the **host**.

- It works like a filter:
  - `1s` in the mask = network bits
  - `0s` in the mask = host bits

Example:

Let's say we have:

- **IP Address**: `192.168.1.10`
- **Subnet Mask**: `255.255.255.0` → `/24` in CIDR notation

IP Address: 11000000.10101000.00000001.00001010
Subnet Mask: 11111111.11111111.11111111.00000000

Given an IP Address and Subnet Mask, You Can Find:

1. Network Address

- It’s the starting point of the subnet.
- Use **bitwise AND** between IP and subnet mask.

Example: IP: `192.168.1.10` Subnet Mask: `255.255.255.0` → Network Address: `192.168.1.0`

2. Broadcast Address

- It’s the last address in the subnet.
- Used to send messages to all hosts in the subnet.

Formula: Broadcast = Network Address **OR** Inverse Subnet Mask → `192.168.1.255`

Given an IP Address and Subnet Mask, You Can Find:

3. Usable Host IP Range

- These are IPs that can be assigned to devices.
- Range: From **Network Address + 1** to **Broadcast Address – 1**

→ `192.168.1.1` to `192.168.1.254`

4. Total IPs in Subnet

- Calculated using host bits:

2n2^n where n = number of host bits

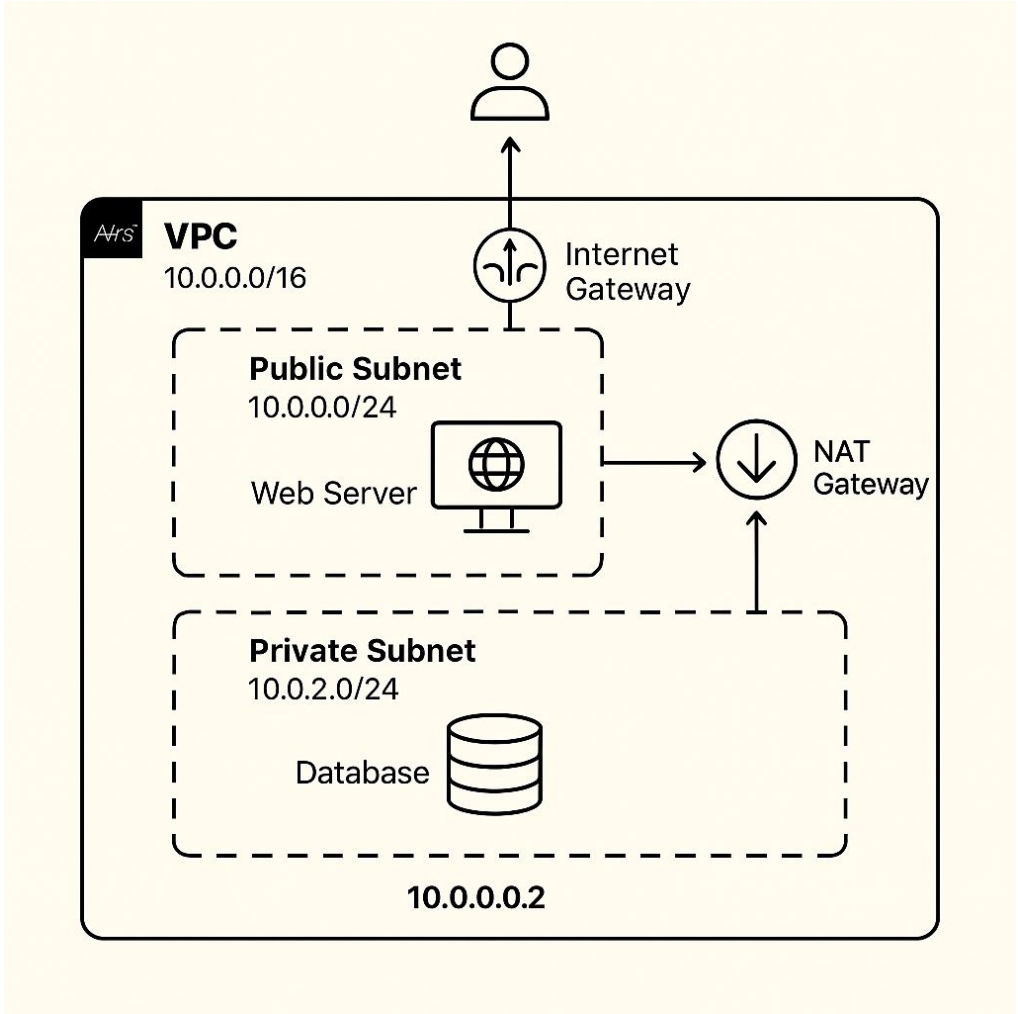
For `/24` (8 host bits): → 2^8=256 total IPs → 254 usable (excluding network & broadcast)

VPC

A **VPC** is a logically isolated section of a cloud provider's network (like AWS, Azure, or GCP) where you can launch and manage resources such as virtual machines, databases, and storage.

Key Features:

- Custom IP address ranges (CIDR blocks)
- Public and private subnets
- Route tables and internet gateways
- Security groups and network ACLs
- VPN or Direct Connect options


VPC
10.0.0.0/16
Internet
Gateway
Public Subnet
10.0.0.0/24
Web Server
NAT
Gateway
Private Subnet
10.0.2.0/24
Database
10.0.0.0.2

How to Set up VPC

CIDR Block: `10.0.0.0/16` → defines the IP range
Subnets:

- `10.0.1.0/24` → Public Subnet (for web server)
- `10.0.2.0/24` → Private Subnet (for database)

Internet Gateway: Attached to the VPC for public access

NAT Gateway: Allows private subnet to access the internet securely
Security Groups:

- Web server allows HTTP/HTTPS from anywhere
- Database only allows access from the web server