AWS Global Infrastructure: Overview

AWS is built on a **multi-region, multi-availability-zone** architecture.

Structure:

```
AWS Global Infrastructure
│
├── Regions (Geographic areas)
│    ├── Availability Zones (Datacenter clusters)
│    │    └── Data Centers (actual physical buildings)
│    └── Local Zones / Wavelength / Outposts (extensions)
└── Edge Locations (CloudFront, Route53)
```

Region: The Geographic Boundary

A **Region** is a **physically isolated geographic area** where AWS clusters multiple data centers.

Each region:

- Has **independent power, cooling, and network connectivity**
- Contains **2 or more Availability Zones**
- Operates **independently of other regions**

Examples:

Region Name	Region Code	Location
US East (N. Virginia)	`us-east-1`	Virginia, USA
Asia Pacific (Mumbai)	`ap-south-1`	Mumbai, India
Europe (Frankfurt)	`eu-central-1`	Germany
South America (São Paulo)	`sa-east-1`	Brazil

Analogy:

> Think of **a Region as a "country"** — it has its own set of "states" (Availability Zones).
> Your AWS resources like EC2, RDS, and S3 live **inside** a region.

Regions Are Isolated for:

- **Fault tolerance** — if one region fails (e.g., natural disaster), others continue.
- **Data residency** — you can choose to store data in specific countries (e.g., GDPR in EU).
- **Latency optimization** — deploy workloads close to your users.
- **Pricing differences** — costs vary by region (e.g., `us-east-1` is cheaper than `ap-south-1`).

Availability Zone (AZ): The Resiliency Building Block

An **Availability Zone** is **one or more discrete data centers** with:

- **Independent power**, **cooling**, and **networking**
- **High-speed, low-latency fiber connectivity** to other AZs in the same region

AWS ensures that AZs in a region are **physically separated (miles apart)** to avoid single points of failure.

Example:

In `us-east-1`, there are 6 AZs:

```
us-east-1a
us-east-1b
us-east-1c
us-east-1d
us-east-1e
us-east-1f
```

Virtual Private Cloud in AWS

What is a VPC?

A logically isolated section of the AWS Cloud in which we can launch AWS resources in a virtual network that you define.

Description:

Key Features of Amazon VPC

1. **Complete Network Control**
    - Configure
        - VPC's IP address range
        - Create subnets

        - Configure route tables
        - network gateways
2. **Enhanced Security**
    - Security groups and network ACLs
        - allow you to filter inbound and outbound traffic to your instances
3. **Connectivity Options**
    - Connect your VPC to your corporate data center
    - Connect with other VPCs
    - Connect directly to the internet

VPC Architecture

- Subnetting:
- Do you know how to break a network
    - by slash notain, CIDR

The 3 private IP series is:

- 10.X.X.X
- 172.31.X.X
- 192.168.X.X

Any VPC should be created by these 3 private IP ranges only.

Now the X lies in the range 0.0.0.0 to 255.255.255.255
So, if we take 10.X.X.X, then it will be 10.0.0.0 to 10.255.255.255

Now, this 10.0.0.0/8 represents the IP address where 8 refers to the first 8 bits are reserved for the network ID.

Let’s do an exercise of creating 3 subnets with in a network of CIDR - 192.168.0.0/24

We will refer to a table below for doing this

Network	1	2	4	8	16	32	64	128	256
Host	256	128	64	32	16	8	4	2	1
Subnet	/24	/25	/26	/27	/28	/29	/30	/31	/32

We need to create 3 subnets as below

1. for engineering

2. HR
3. reception

	IP Range	Network ID	Broadcast IP	Representation
Engineering	192.168.0.0 - 63	192.168.0.0	192.168.0.63	192.168.0.0/26
HR	192.168.0.64 - 127	192.168.0.64	192.168.0.127	192.168.0.64/26
Reception	192.168.0.128 - 191	192.168.0.128	192.168.0.191	192.168.0.128/26

According in the default VPC we noted that it has an IP range as below

VPC: 10.0.0.0/16

	IP Range	Network ID	Broadcast IP	Representation
Public subnet1	10.0.0.0 - 10.0.15.255	10.0.0.0	10.0.15.255	10.0.0.0/20
HR	10.0.16.0 - 10.0.31.255	10.0.16.0	10.0.31.255	10.0.16.0/20
Reception	10.0.32.0 - 10.0.47.255	10.0.32.0	10.0.47.255	10.0.32.0/20

Detailed tutorial

What is a VPC?

VPC (Virtual Private Cloud) = your own **isolated private network inside AWS**.

Think of AWS as a huge city, and your **VPC is your private gated community** within that city.
You control:

- Who can enter (security groups, NACLs)
- How big it is (CIDR range)
- How traffic moves (through route tables, gateways)
- Where the buildings (subnets) are placed (AZs)

---

2. Key VPC Concepts

Concept	Description	Example
CIDR Block	Defines IP address range for your VPC	`10.0.0.0/16` (≈ 65,536 IPs)
Subnet	Subdivision of VPC network	`10.0.1.0/24`
Route Table	Decides where packets go	Routes traffic to Internet or Private Gateway
Internet	Enables Internet access for public	

Gateway (IGW)	subnets	Attached to VPC
NAT Gateway	Allows private subnets to access Internet *outbound only*	For software updates, API calls
Security Group	Virtual firewall for instances	Controls inbound/outbound traffic
Network ACL (NACL)	Firewall for subnets	Controls packet-level rules
VPC Peering	Connects two VPCs	Enables private communication
VPC Endpoint	Private connection to AWS services	S3, DynamoDB without Internet

3. How a VPC is Structured

```
Region (e.g., us-east-1)
│
└── VPC (10.0.0.0/16)
    ├── Subnet A (Public) 10.0.1.0/24  → Route to IGW
    ├── Subnet B (Private) 10.0.2.0/24 → Route to NAT Gateway
    └── Subnet C (DB) 10.0.3.0/24      → No Internet Access
```

- Each **subnet lives in one Availability Zone** (AZ)
- **Public Subnets** → have a route to Internet Gateway
- **Private Subnets** → no direct route to Internet Gateway
- **Database Subnets** → often (by default) isolated with no outbound routes

4. CIDR and IP Allocation

CIDR = Classless Inter-Domain Routing

CIDR block defines how many IPs you can use:

CIDR	# of IPs	Common Use
/16	65,536	Whole VPC
/24	256	One subnet (small app)
/28	16	Test subnet

AWS *reserves 5 IPs* in every subnet:

- `.0` → Network address
- `.1` → AWS VPC router
- `.2` → DNS
- `.3` → Reserved for future use
- `.255` → Broadcast address

So a `/24` subnet gives you **251 usable IPs**, not 256.

---

5. Subnets – Public vs Private

Type	Has route to IGW?	Can host public apps?	Typical Usage
Public Subnet	Yes	Yes	Web servers, Load balancers
Private Subnet	No (via NAT)	No	App servers, internal services
Database Subnet	No	No	RDS, ElastiCache

Example:

```
VPC CIDR: 10.0.0.0/16
│
├── Public Subnet 1: 10.0.1.0/24 → Route: IGW
├── Private Subnet 1: 10.0.2.0/24 → Route: NAT Gateway
└── DB Subnet: 10.0.3.0/24 → No Internet Route
```

---

6. Route Table Example

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-xxxx (for public subnet)

For private subnets:

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	nat-xxxx

---

7. Security Groups vs NACLs

Feature	Security Group	Network ACL
Level	Instance level	Subnet level
Stateful	Yes	No
Rules	Allow only	Allow + Deny
Use case	Control EC2 traffic	Extra layer at subnet boundary

---

8. Internet Gateway & NAT Gateway

Component	Purpose	Attached To

Internet Gateway	Enables Internet access	VPC
NAT Gateway	Allows private subnet to access Internet *outbound*	Public Subnet

Example flow:

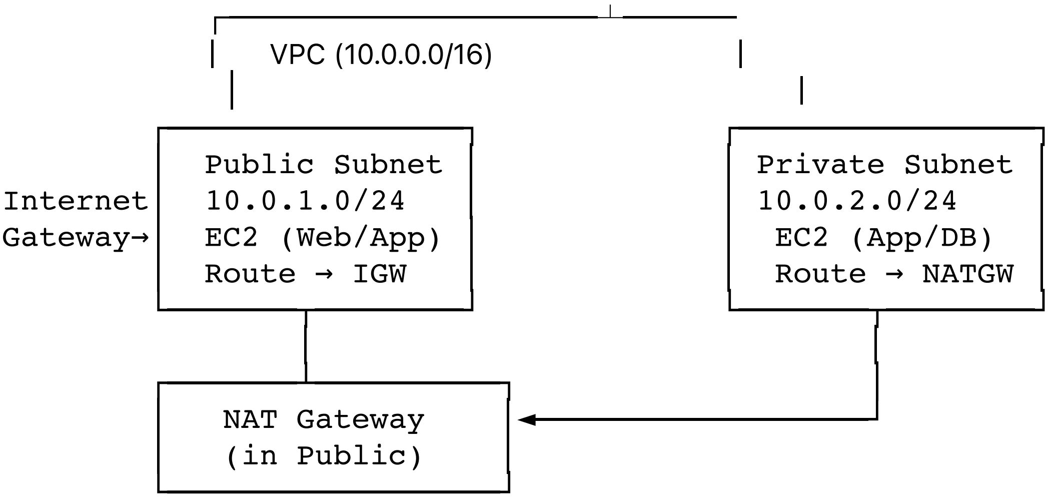
- EC2 in **private subnet** → NAT → IGW → Internet

9. VPC Endpoint Example

Allows **private access to AWS services** without public Internet.

for example to access internet without the traffic going to the internet

10. Visual Summary



11. Common Pitfalls

Issue	Reason	Fix
Can’t SSH to EC2	Security group or subnet missing IGW	Check SG + Route
Private EC2 can’t update packages	No NAT Gateway	Add NAT Gateway
Overlapping CIDRs	Two VPCs use same CIDR	Redesign
No Internet in VPC	IGW not attached	Attach and update route

VPC Peering

A VPC peering connection

networking connection between two VPCs

route traffic between them using

private IPv4 addresses or IPv6 addresses.

Key Features:

- Direct network route between two VPCs
- No gateway or VPN connection required
- No single point of failure or bandwidth bottleneck
- Traffic stays on the AWS global network

Comparison Table

Feature	Security Groups	Network ACLs
Level of operation	Instance level	Subnet level
State	Stateful (return traffic automatically allowed)	Stateless (return traffic must be explicitly allowed)
Rule evaluation	All rules are evaluated before deciding to allow traffic	Rules are evaluated in order (lowest to highest)
Default behavior	Deny all inbound, allow all outbound	Allow all inbound, allow all outbound
Rule types	Allow rules only	Allow and deny rules