You're deploying a web application on AWS. It has:

* A public-facing frontend (React app)
* A backend API (Node.js)
* A database (RDS)

**1. Subnet**

A **subnet** is a range of IP addresses within a VPC (Virtual Private Cloud). It helps segment your network.

* **Public Subnet**: Resources that need internet access (e.g., frontend EC2).
* **Private Subnet**: Resources that should not be directly accessible from the internet (e.g., backend API, DB).

🧾 **Example**:

* VPC CIDR: 10.0.0.0/16
  + Public Subnet: 10.0.1.0/24
  + Private Subnet: 10.0.2.0/24

**2. Security Group**

A **security group** acts as a virtual firewall for your EC2 instances to control inbound and outbound traffic.

* **Stateful**: If you allow inbound traffic, the response is automatically allowed outbound.
* You define rules based on **protocol**, **port**, and **source/destination IP**.

🧾 **Example**:

* Frontend EC2 SG:
  + Inbound: Allow TCP 80/443 from 0.0.0.0/0
  + Outbound: Allow all
* Backend EC2 SG:
  + Inbound: Allow TCP 443 from frontend SG
  + Outbound: Allow DB port to RDS SG

**3. Internet Gateway (IGW)**

An **Internet Gateway** allows resources in your VPC to connect to the internet.

* Must be attached to the VPC.
* Route table must direct traffic (0.0.0.0/0) to the IGW.

🧾 **Example**:

* Public Subnet Route Table:
  + 0.0.0.0/0 → IGW

**4. NAT Gateway**

A **NAT Gateway** allows instances in a **private subnet** to access the internet **without being exposed** to it.

* Used for downloading updates, accessing external APIs, etc.
* Placed in a public subnet.

🧾 **Example**:

* Private Subnet Route Table:
  + 0.0.0.0/0 → NAT Gateway (in public subnet)

**5. Route Table**

A **route table** contains rules (routes) that determine where network traffic is directed.

* Each subnet is associated with a route table.
* Routes define destination CIDR blocks and target (IGW, NAT, local, etc.)

🧾 **Example**:

* Public Subnet Route Table:
  + 10.0.0.0/16 → local
  + 0.0.0.0/0 → IGW
* Private Subnet Route Table:
  + 10.0.0.0/16 → local
  + 0.0.0.0/0 → NAT Gateway

**6. Network ACL (NACL)**

A **Network Access Control List** is a stateless firewall at the subnet level.

* Controls inbound and outbound traffic.
* Rules are evaluated in order.
* Unlike SGs, NACLs are **stateless**: you must allow both inbound and outbound explicitly.

🧾 **Example**:

* Public Subnet NACL:
  + Inbound: Allow TCP 80/443 from 0.0.0.0/0
  + Outbound: Allow TCP 1024-65535 to 0.0.0.0/0
* Private Subnet NACL:
  + Inbound: Allow TCP 443 from public subnet
  + Outbound: Allow TCP 3306 to DB subnet

**🔁 Summary Table**

| **Component** | **Purpose** | **Example Use Case** |
| --- | --- | --- |
| Subnet | IP range within VPC | Public for frontend, private for backend |
| Security Group | Instance-level firewall (stateful) | Allow HTTP to frontend, HTTPS to backend |
| Internet Gateway | Internet access for public subnets | Frontend EC2 access |
| NAT Gateway | Internet access for private subnets | Backend EC2 downloading packages |
| Route Table | Routing rules for subnets | Direct traffic to IGW or NAT |
| NACL | Subnet-level firewall (stateless) | Extra layer of security |