DevOps

[VPC 2](#_Toc194042920)

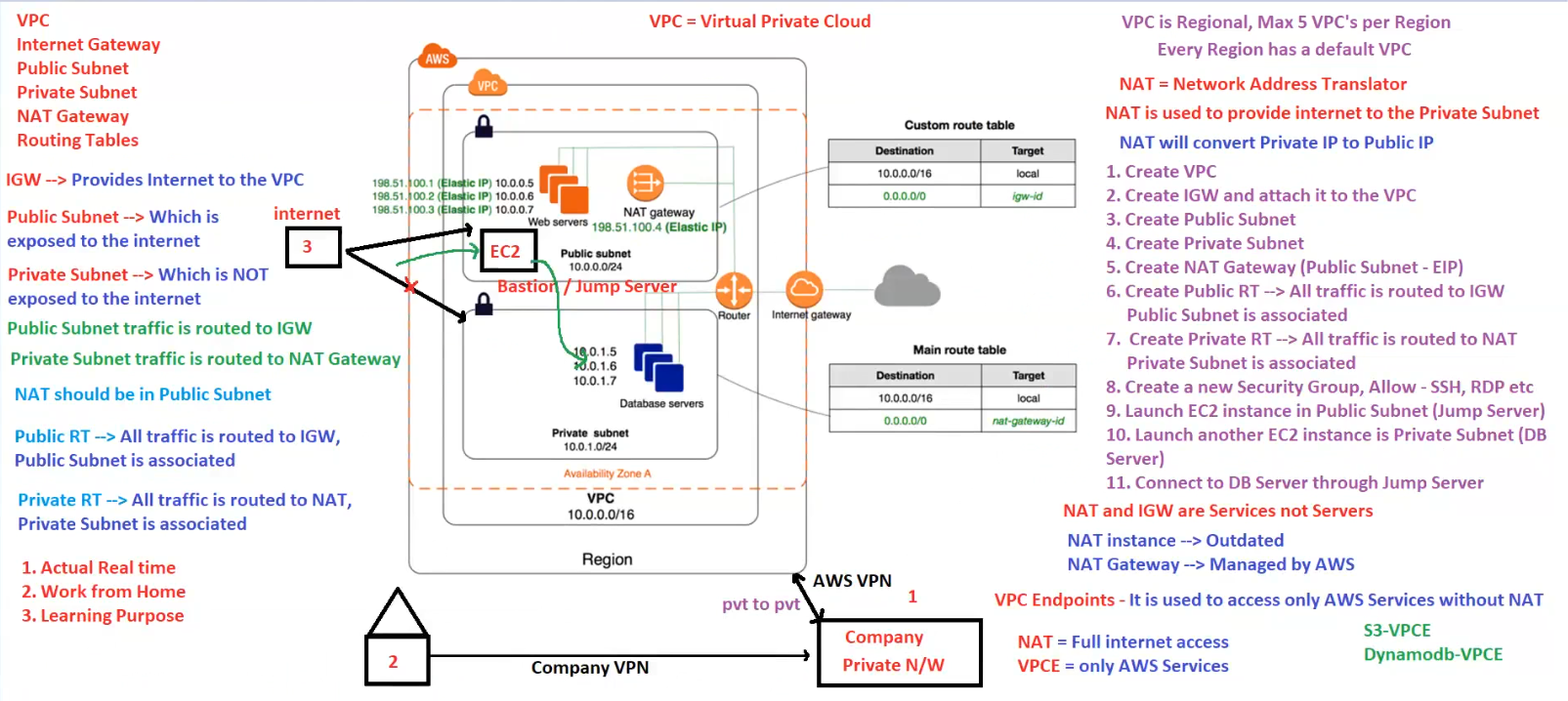
[Architecture 2](#_Toc194042921)

[CIDR (Classless Inter-Domain Routing) 4](#_Toc194042922)

[VPC Peering and VPN Connectivity 6](#_Toc194042923)

# VPC

## Architecture



**1. VPC and Subnets**

* **VPC (Virtual Private Cloud)**: A logically isolated network in AWS.
* **Public Subnet**:
  + **Exposed to the internet** via **Internet Gateway (IGW)**.
  + Typically hosts web servers, bastion/jump servers, or NAT Gateways.
* **Private Subnet**:
  + **Not directly exposed** to the internet.
  + Hosts **database servers or application servers** that should not be publicly accessible.
  + Access to the internet is provided via the **NAT Gateway** in the **public subnet**.

**2. Internet Gateway (IGW)**

* Provides **internet access** to resources in the **public subnet**.
* Routes public traffic via the **public route table (RT)**.

✅ **Best for**: Allowing public-facing instances (e.g., web servers) to communicate with the internet.

**3. NAT Gateway (Network Address Translation)**

* **Allows private subnet instances to access the internet** while keeping them private.
* **Converts private IPs to public IPs** for outbound traffic.
* Placed in the **public subnet**.

✅ **Best for**:

* Allowing private instances to download software updates.
* Ensuring private instances **can access the internet** but remain inaccessible from the internet.

**4. Routing Tables**

* **Public Route Table (Public RT)** → Routes all traffic **via IGW**.
* **Private Route Table (Private RT)** → Routes all traffic **via NAT Gateway**.

✅ **Best for**: Controlling traffic flow between public and private subnets.

**5. Bastion / Jump Server**

* A **secure EC2 instance** in the **public subnet** used to **connect to private instances**.
* Acts as an **intermediary for SSH/RDP access**.
* Private instances **are never directly exposed**.

✅ **Best for**: Securely accessing private resources **without exposing them to the internet**.

**6. VPN and Private Connectivity**

* **Company VPN**: Connects an on-premises private network to AWS.
* **AWS VPN**: Secure communication between private AWS resources across regions.

✅ **Best for**:

* Work-from-home employees accessing AWS private resources securely.
* Connecting **on-premise data centers** to AWS VPC.

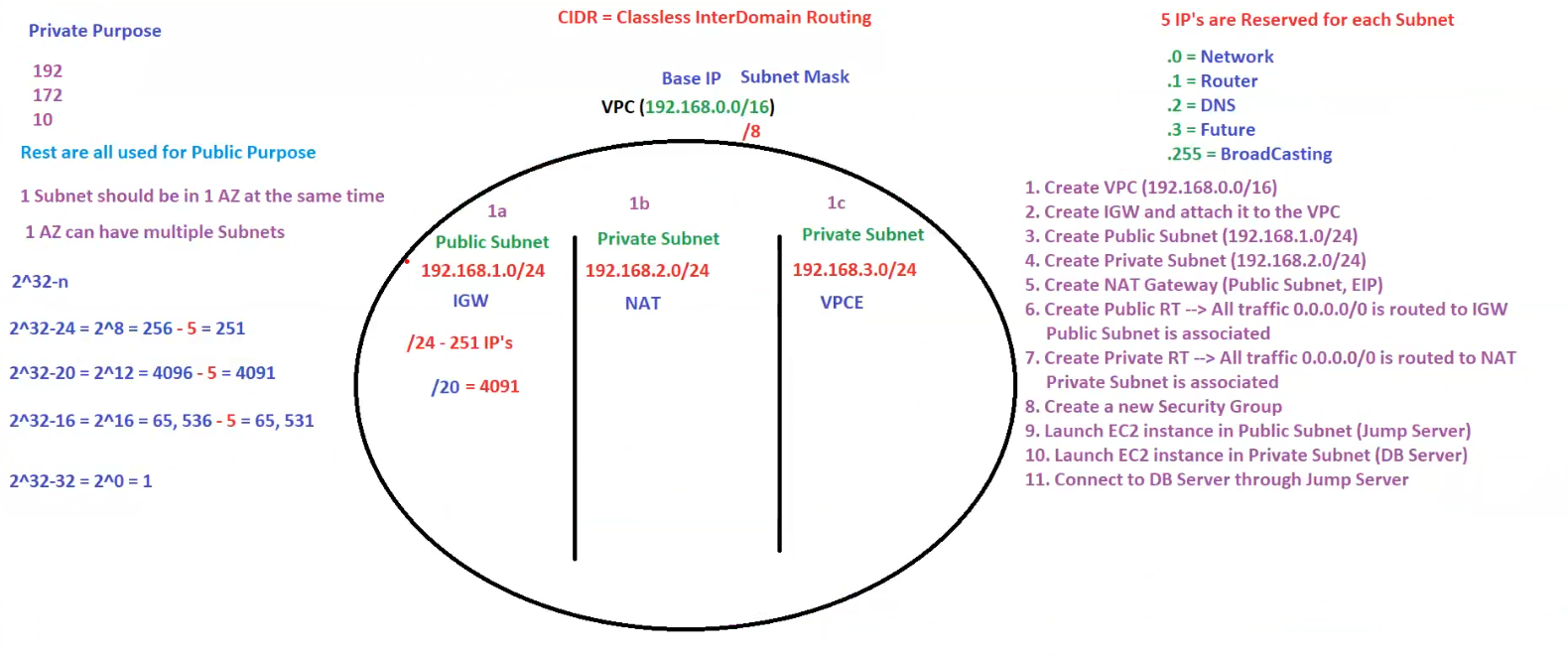
**7. Key Points in the Image**

* **NAT and IGW are services, not servers.**
* **NAT Gateway (Managed by AWS) is preferred over NAT instances (outdated).**
* **VPC Endpoints (VPCE) allow access to AWS services like S3 & DynamoDB without NAT.**
* **Steps to set up the VPC** are provided in the right-hand side of the image.

**🚀 Summary**

* **Public Subnet** → IGW → Internet Access.
* **Private Subnet** → NAT Gateway → Indirect Internet Access.
* **Bastion Host** → Secure access to private instances.
* **Routing Tables** → Control traffic direction.
* **VPN & VPCE** → Secure connectivity.

## CIDR (Classless Inter-Domain Routing)



**1. CIDR (Classless Inter-Domain Routing) Overview**

* **VPC CIDR:** 192.168.0.0/16
  + This means the network has **65,536 (2^16) IPs**.
  + It is a **private range** (as per RFC 1918).
* **Subnet Masking:**
  + 192.168.1.0/24 → Public Subnet (251 usable IPs).
  + 192.168.2.0/24 → Private Subnet with NAT.
  + 192.168.3.0/24 → Private Subnet with VPCE.

**2. Subnet Allocation in Availability Zones (AZs)**

* **1a (Public Subnet):**
  + 192.168.1.0/24
  + Has an **Internet Gateway (IGW)** for direct internet access.
* **1b (Private Subnet with NAT):**
  + 192.168.2.0/24
  + Uses a **NAT Gateway** (in the public subnet) for outbound internet access.
* **1c (Private Subnet with VPCE):**
  + 192.168.3.0/24
  + Uses **VPC Endpoints (VPCE)** for AWS services access (e.g., S3, DynamoDB) without needing NAT.

**3. Reserved IPs Per Subnet**

Each subnet reserves **5 IPs**:

1. .0 → Network Address
2. .1 → Router
3. .2 → AWS DNS
4. .3 → Reserved (Future use)
5. .255 → Broadcast Address

Thus, in a **/24 subnet (256 IPs), only 251 are usable**.

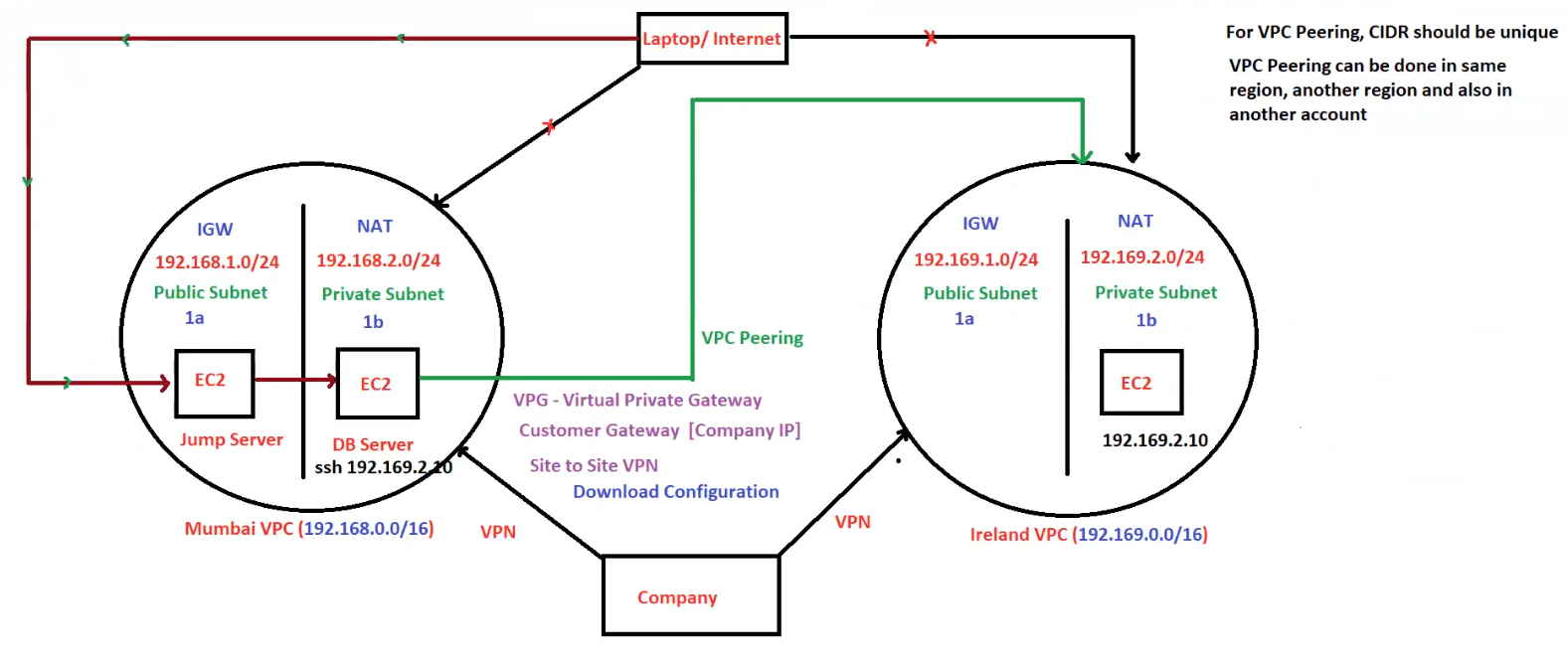
**4. Steps to Create This VPC Setup**

1. **Create a VPC** (192.168.0.0/16).
2. **Attach an IGW** to the VPC.
3. **Create a Public Subnet** (192.168.1.0/24).
4. **Create Private Subnets** (192.168.2.0/24 and 192.168.3.0/24).
5. **Set Up a NAT Gateway** (in the public subnet) for private subnet access.
6. **Configure Route Tables:**
   * Public RT → **Route all traffic to IGW**.
   * Private RT → **Route all traffic to NAT Gateway**.
7. **Create a Security Group** (Allow SSH, RDP, etc.).
8. **Launch a Jump Server (Bastion Host) in the Public Subnet.**
9. **Launch a DB Server in the Private Subnet.**
10. **Connect to the DB Server using the Jump Server.**

**5. Key Learnings from the Diagram**

✅ **Public Subnet → IGW → Full Internet Access**.  
✅ **Private Subnet → NAT Gateway → Secure Internet Access**.  
✅ **VPC Endpoints (VPCE) → AWS Services Access Without NAT**.  
✅ **Subnets are AZ-specific but a single AZ can have multiple subnets**.

## VPC Peering and VPN Connectivity



**VPC Peering and VPN Connectivity - Architecture Breakdown**

**1. Overview of VPCs**

Two VPCs in different regions are connected:

1. **Mumbai VPC (192.168.0.0/16)**
   * Public Subnet (192.168.1.0/24): Jump Server (EC2) with IGW
   * Private Subnet (192.168.2.0/24): DB Server (EC2) with NAT Gateway
2. **Ireland VPC (192.169.0.0/16)**
   * Public Subnet (192.169.1.0/24): IGW for internet access
   * Private Subnet (192.169.2.0/24): EC2 instance

**2. Key Components**

**VPC Peering (Green Arrows)**

* **Direct communication** between Mumbai and Ireland VPCs over AWS's private network.
* CIDR blocks **must be unique** to avoid conflicts.
* **VPC Peering allows private communication**, but routes must be manually updated in each VPC’s route table.

**VPN (Black Lines)**

* **VPN (Virtual Private Network) connects the company network to both VPCs** for secure access.
* **Components involved:**
  + **Virtual Private Gateway (VPG)** on AWS side.
  + **Customer Gateway (CGW)** at the company’s end (with a public IP).
  + **Site-to-Site VPN** connects on-premises resources to AWS VPCs.

**Jump Server in Public Subnet**

* Acts as a **bastion host** to access private subnet resources (DB Server).
* External users **cannot** directly SSH into the DB Server (192.169.2.10).
* Access via Jump Server:
* ssh -J ec2-user@jump-server ec2-user@192.169.2.10

**NAT Gateway (Private Internet Access)**

* **Mumbai Private Subnet (192.168.2.0/24)** cannot access the internet directly.
* NAT Gateway (in the **public subnet**) allows outbound internet traffic while keeping it private.

**3. Traffic Flow**

🔴 **Blocked Traffic:**

* **Direct access from the internet to private subnet instances is restricted.**

🟢 **Allowed Traffic:**

1. **Users connect to the Jump Server** → SSH into private subnet EC2 (DB Server).
2. **Private instances use NAT Gateway** for internet access.
3. **VPC Peering allows private subnet communication** between Mumbai and Ireland VPCs.
4. **Company network accesses AWS resources over VPN.**

**4. Key Takeaways**

✅ **VPC Peering → Low latency, AWS private networking.**  
✅ **VPN → Secure on-premises to AWS connectivity.**  
✅ **Jump Server → Controls access to private instances.**  
✅ **NAT Gateway → Enables outbound internet access for private subnets.**