AWS VPC Setup: Configuring Public and Private Subnets Securely

Introduction to VPC

What is a VPC?

A Virtual Private Cloud (VPC) is an isolated network environment within AWS that allows you to launch AWS resources in a logically defined virtual network. It enables better security, control, and scalability for cloud infrastructure.

Key Components of a VPC:

- **CIDR Block:** Defines the range of IP addresses within the VPC.
- **Subnets:** Smaller network segments within the VPC.
- **Route Tables:** Define how traffic is routed within the VPC.
- **Internet Gateway (IGW):** Enables public internet access.
- **NAT Gateway:** Allows private subnets to access the internet securely.
- **Security Groups & Network ACLs:** Control inbound and outbound traffic.

SUBNET

A **subnet (subnetwork)** is a logically defined segment of a network that divides a larger network (such as a **Class A, B, or C network**) into smaller, more manageable parts. Subnetting helps improve **network efficiency, security, and scalability** by reducing congestion and isolating traffic.

Each subnet has:

- A **Network Address** (identifies the subnet)
- A **Subnet Mask** (defines the range of IPs in the subnet)
- A **Broadcast Address** (used to communicate with all devices in the subnet)
- **Usable IP Addresses** (assigned to hosts like servers, computers, and network devices)

Types of Subnets

Based on Accessibility

Public Subnet

- Contains resources **accessible from the internet**.
- Typically used for web servers, application servers, and API gateways.
- Requires **public IP addresses**.

Example:

- `192.168.1.0/26` (Public Subnet) with a **Public IP assigned to a server**.

Private Subnet

- Contains internal resources **not directly accessible from the internet**.
- Used for databases, backend applications, and internal services.
- Uses **private IP addresses** (e.g., `192.168.x.x`, `10.x.x.x`, `172.16.x.x`).
- Can access the internet via **NAT (Network Address Translation)**.

Example:

- `192.168.2.0/26` (Private Subnet) with a **Private IP assigned to a database**.

1. Create a VPC with 10.0.0.0/25 IPv4 CIDR in AWS

1. Log in to AWS Console

- **Go to AWS Management Console.**
- **Navigate to VPC under Services.**

2. Create a New VPC

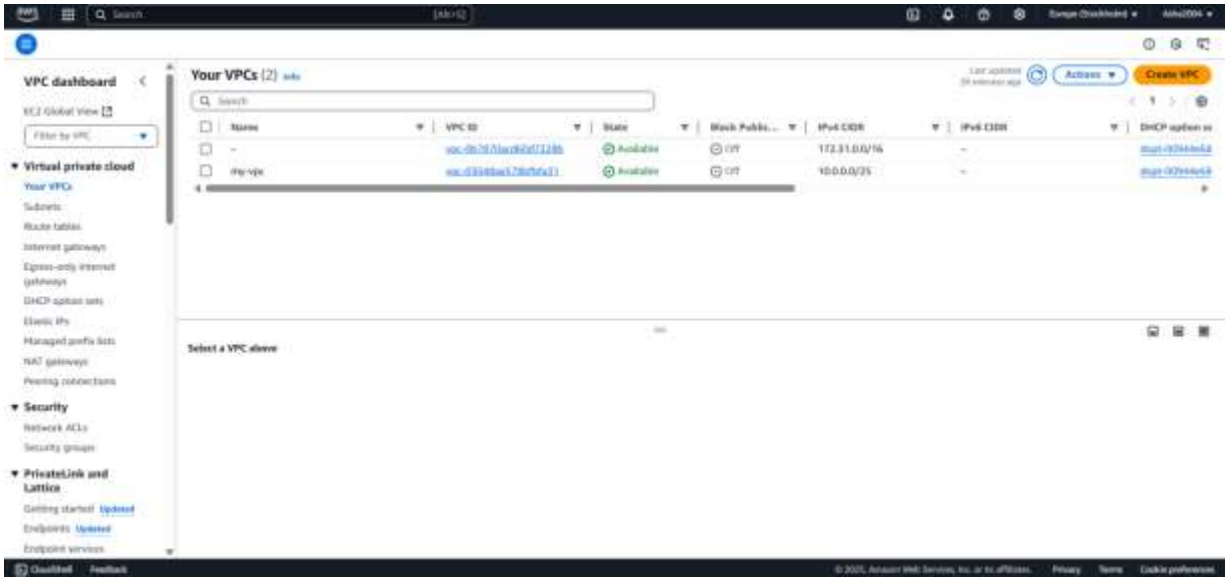
1. **Click Create VPC.**
2. **VPC Settings:**

- **Name:** `my-vpc` (change as needed).
- **IPv4 CIDR Block:** `10.0.0.0/25`.
- **IPv6 CIDR Block:** None (default).
- **Tenancy:** Default (keep for cost efficiency).

3. **Click Create VPC.**

3. Verify VPC Creation

- **Navigate to VPCs in the AWS Console.**
- **Ensure `MyCustomVPC` is listed with the correct `10.0.0.0/25` CIDR.**

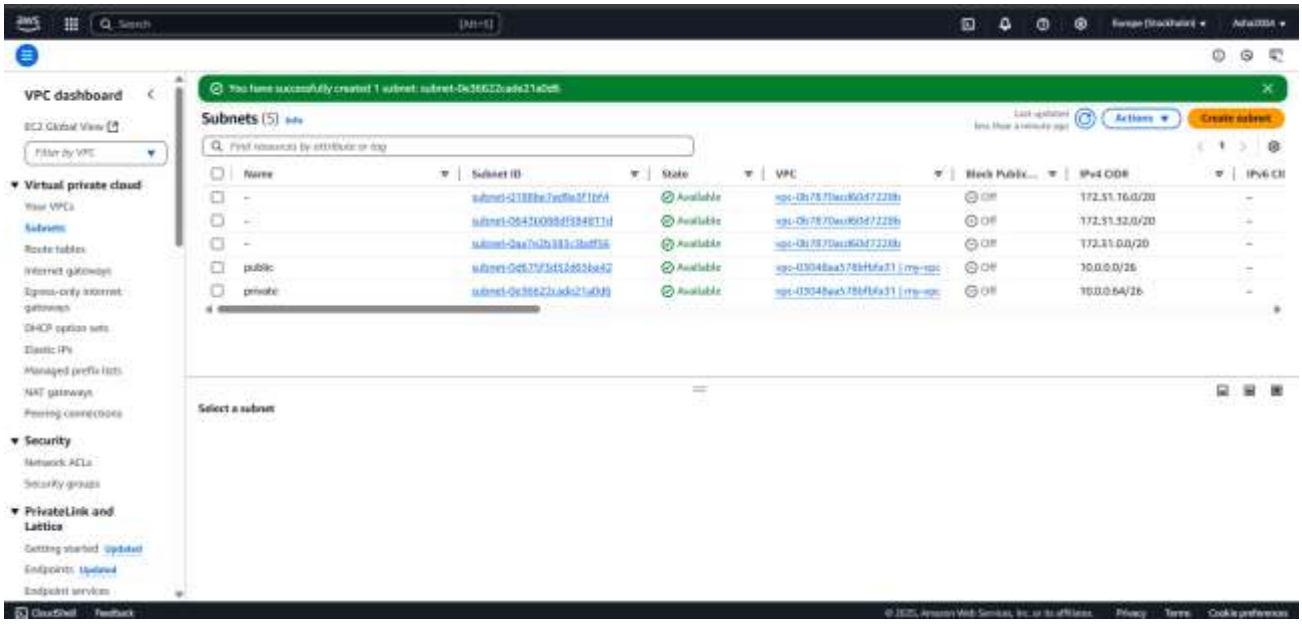


2. Create the Public Subnet

1. **Click Subnets in the left menu.**
2. **Click Create Subnet.**
3. **Configure Public Subnet:**
   - **Name:** `public`
   - **VPC ID:** Select the previously created `my-vpc` (10.0.0.0/25).
   - **Availability Zone:** Choose any (default).
   - **IPv4 CIDR Block:** `10.0.0.0/25`.
   - **Keep all other settings default.**
4. **Click Create Subnet.**

3. Create the Private Subnet

1. **Click Create Subnet again.**
2. **Configure Private Subnet:**
    - **Name:** `private`
    - **VPC ID:** Select the previously created `my-vpc` (10.0.0.0/25).
    - **Availability Zone**: **Choose the same as the public subnet (optional).**
    - **IPv4 CIDR Block:** `10.0.0.64/26`**.**
    - **Keep all other settings default.**
3. **Click Create Subnet.**



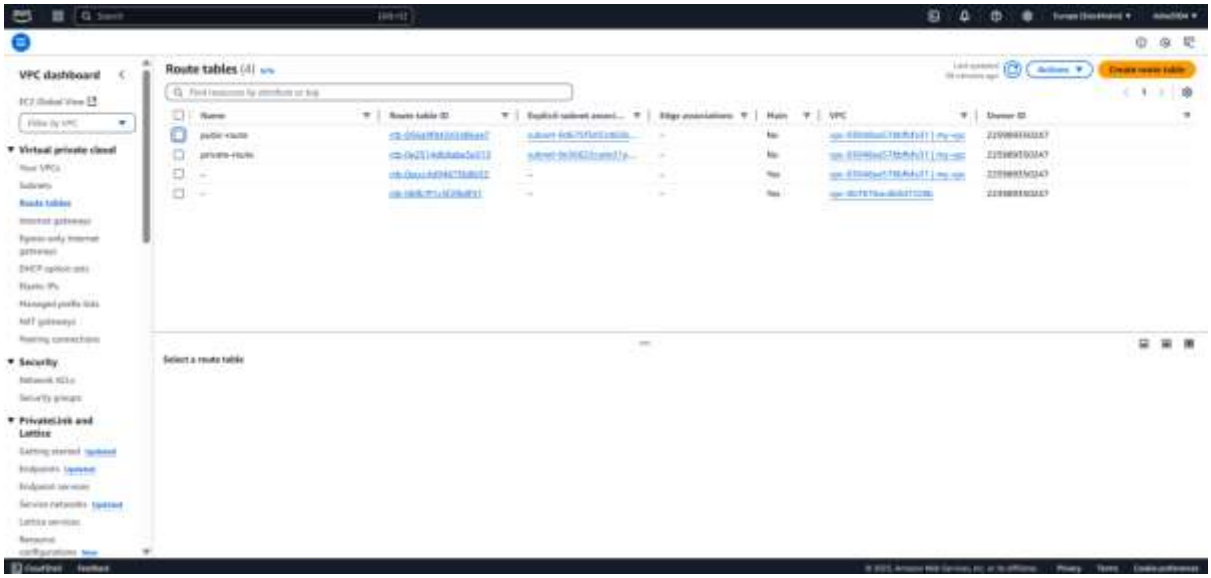
4. Create a Route Table for the Public Subnet

1. **Click Route Tables in the left menu.**
2. **Click Create Route Table.**
3. **Configure Public Route Table:**
    - **Name:** `public-subnet`
    - **VPC ID:** Select the previously created `my-vpc`.
    - **Keep all other settings default.**

4.Click Create Route Table.

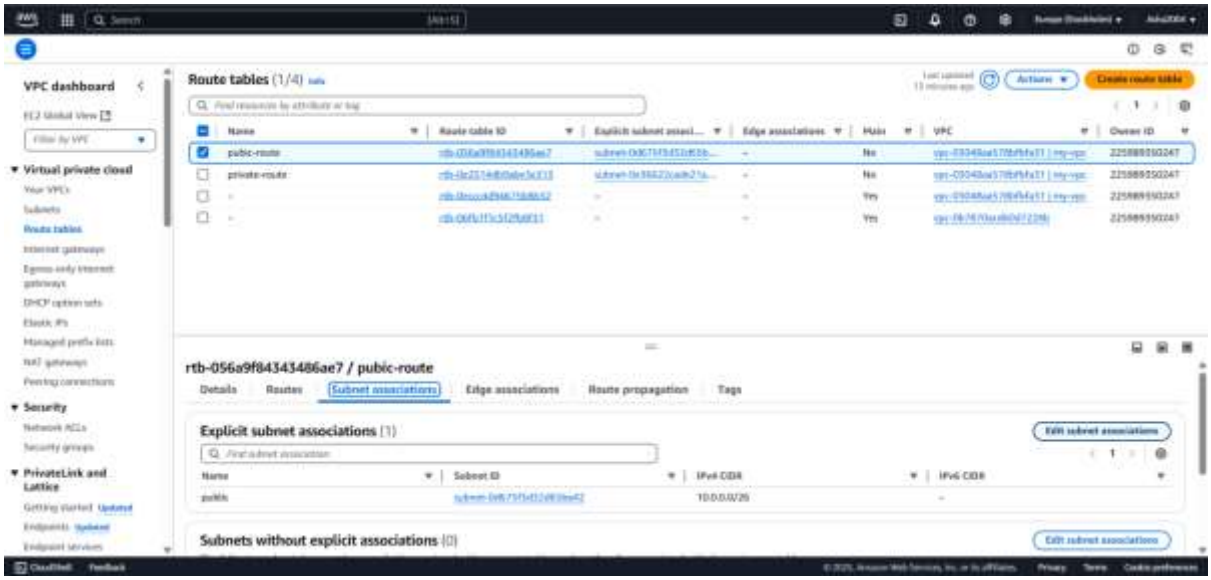
5. Create a Route Table for the Private Subnet

1. **Click Create Route Table again.**
2. **Configure Private Route Table:**
    - **Name:** `private-subnet`
    - **VPC ID:** Select the previously created `my-vpc`**.**
    - **Keep all other settings default.**
3. **Click Create Route Table**



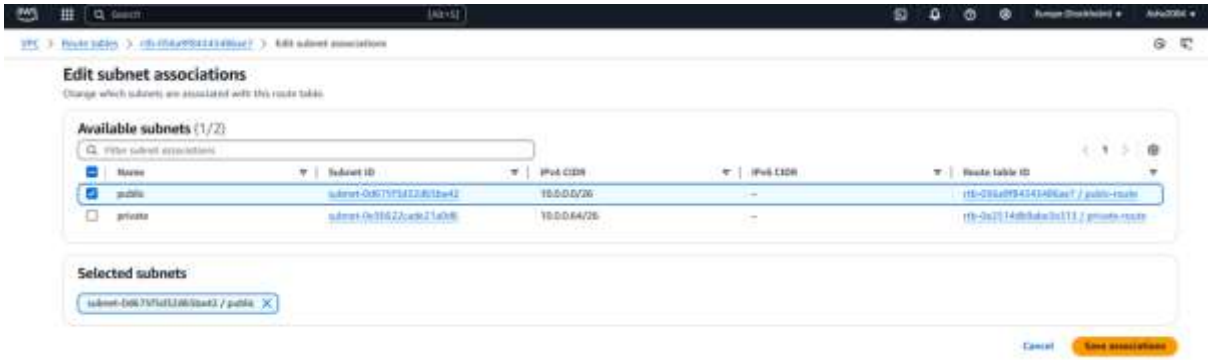
6. Associate the Public Route Table with the Subnet

1. **Click Route Tables in the left menu.**
2. **Select the `public-subnet` route table.**
3. **Go to the Subnet Associations tab.**
4. **Click Edit Subnet Associations.**



5. Select the previously created `public` subnet (10.0.0.0/25)`.

6. Click Save Changes.



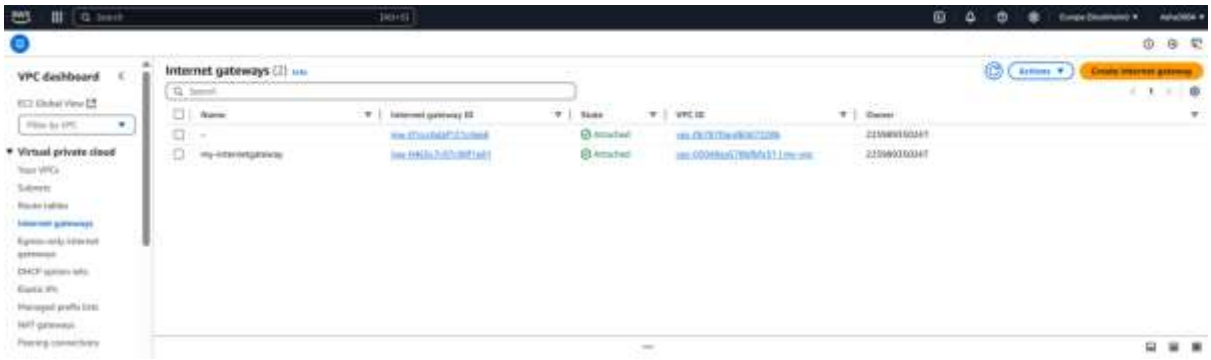
➢ **Repeat the same process for private subnet**

7. Create an Internet Gateway (IGW)

1. **Click Internet Gateways in the left menu.**
2. **Click Create Internet Gateway.**
3. **Configure IGW:**
    - **Name:** `my-internetgateway` (or any name you prefer).
    - **Keep all other settings default.**
4. **Click Create Internet Gateway.**

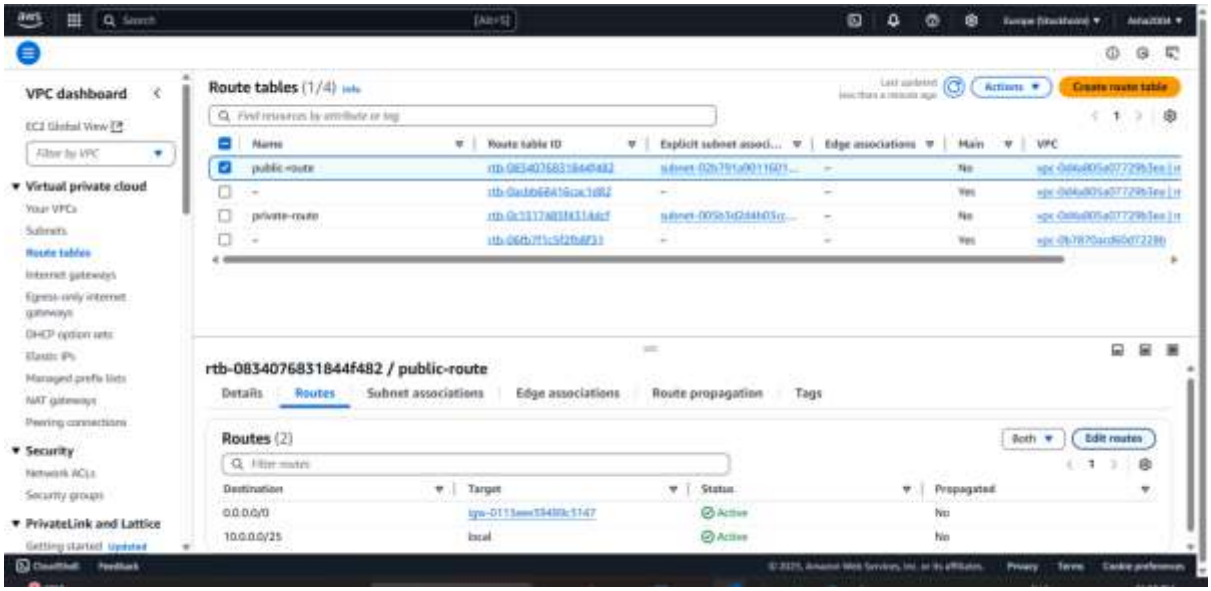
8. Attach the Internet Gateway to the VPC

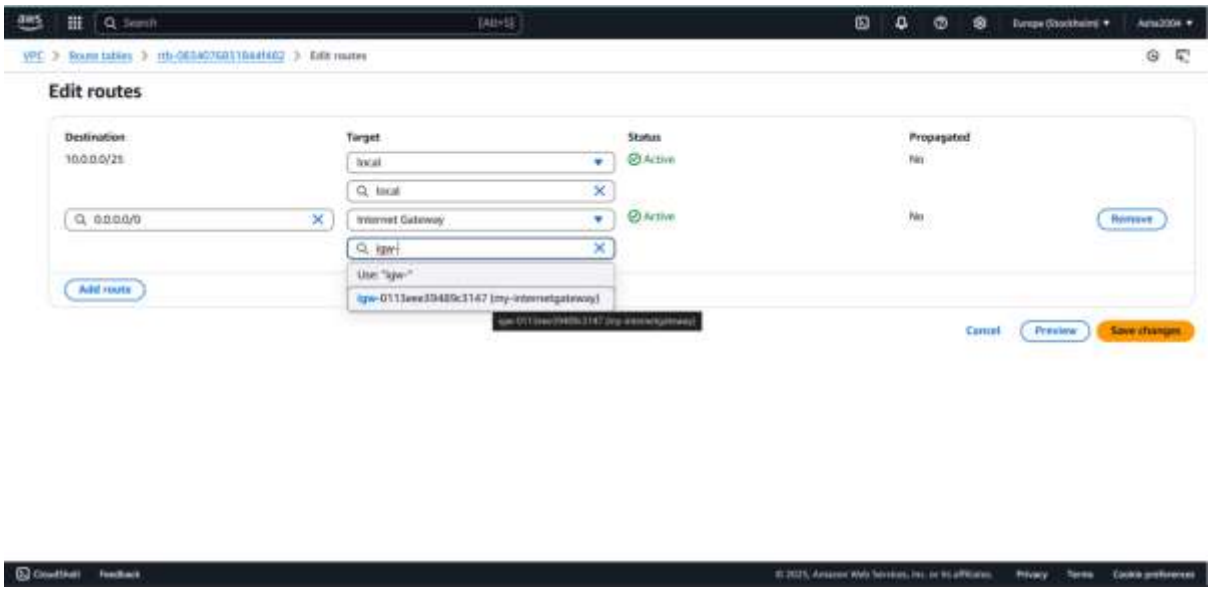
1. **Select the newly created `my-internetgateway`.**
2. **Click Actions** → Attach to VPC.
3. **Select `my-vpc` (the previously created VPC).**
4. **Click Attach Internet Gateway.**



9. Configure Public Route Table for Internet Access

1. **Click Route Tables in the left menu.**
2. **Select `public-route-table`.**
3. **Click Routes** → Edit Routes**.**
4. **Click Add Route.**
5. **Enter the following:**
   - **Destination:** `0.0.0.0/0`
   - **Target:** Select previously created **Internet Gateway** (`my-internet-gateway`)
   - **VPC:** Select `my-vpc`
6. **Click Save Changes.**





10. Launch a Windows EC2 Instance(public)

1. **Click Launch Instance.**
2. **Configure Instance Settings:**
    - **Name:** `Public-Server`.
    - **AMI (Amazon Machine Image):** Select Windows Server 2019/2022 Base.
    - **Instance Type:** `t2.micro` (Free Tier) or higher as needed.
3. **Network Settings:**
    - **VPC:** Select `my-vpc` (previously created).
    - **Subnet:** Select `public-subnet (10.0.0.0/26)`.

- **Auto-assign Public IP:** □ Enabled.

4. **Configure Security Group:**
   - **Click Create a new security group.**
   - **Name:** `All-Traffic`.
   - **Rules:**
     - **Type:** All traffic
     - **Protocol:** All
     - **Port Range:** All
     - **Source:** Anywhere (`0.0.0.0/0` and `::/0`)
5. **Create or Use a Key Pair:**
   - **Select "Create a new key pair" or use an existing one.**
   - **If creating a new key pair:**
     - **Name:** `MyWindowsKeyPair`.
     - **Key Type:** RSA.
     - **Click Download Key Pair (save `.pem` file).**
6. **Click "Launch Instance"**.

11. Launch a Windows EC2 Instance(private)

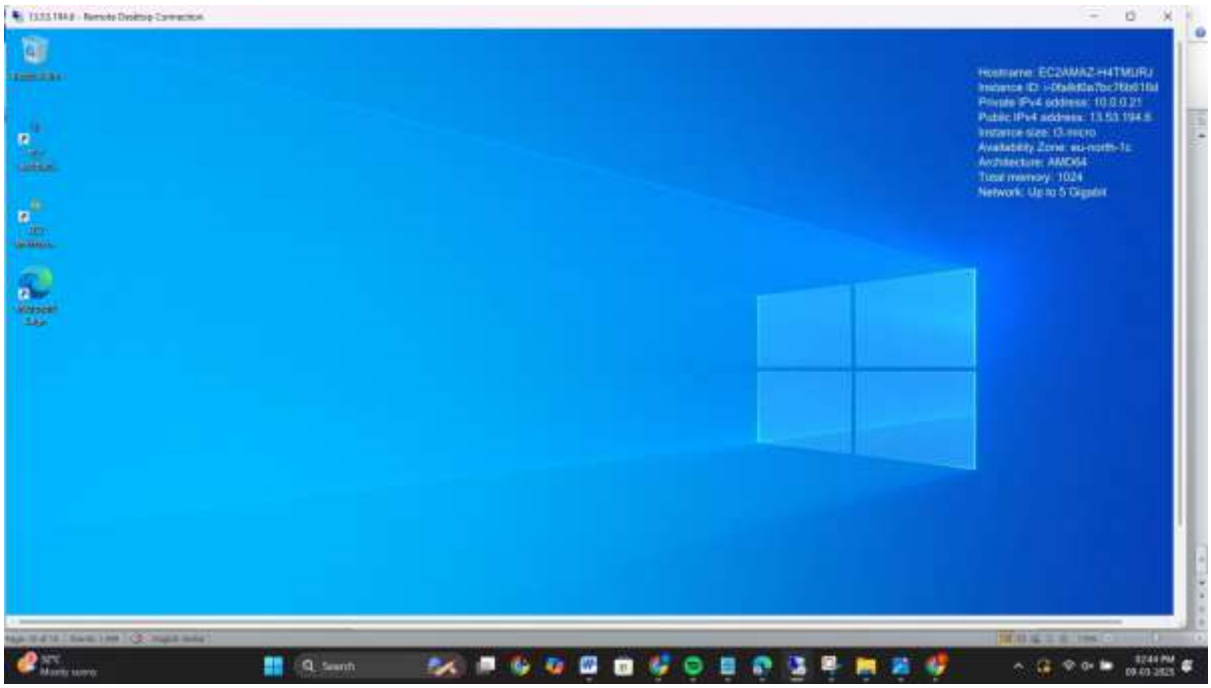
1. **Click Launch Instance.**
2. **Configure Instance Settings:**
   - **Name:** `Private-Server`.
   - **AMI (Amazon Machine Image):** Select Windows Server 2019/2022 Base.
   - **Instance Type:** `t2.micro` (Free Tier) or higher as needed.
3. **Network Settings:**
   - **VPC:** Select `my-vpc`(previously created).
   - **Subnet:**Select `private-subnet (10.0.0.64/26)`.
   - **Auto-assign Public IP: □ Disabled (since it's a private server).**
4. **Configure Security Group:**
   - **Click Create a new security group.**
   - **Name**: `Private-Server-SG`.
   - **Rules:**
     - **Allow RDP (3389) only from the Public Server's Private IP.**

- **Type:** RDP
- **Protocol:** TCP
- **Port Range:** 3389
- **Source:**Public Server's Private IP (`10.0.0.x/32`)

5. **Create or Use a Key Pair:**
    - **Select "Use existing key pair" (Use the same key pair as the Public Server, e.g., `MyWindowsKeyPair`).**
6. **Click "Launch Instance".**

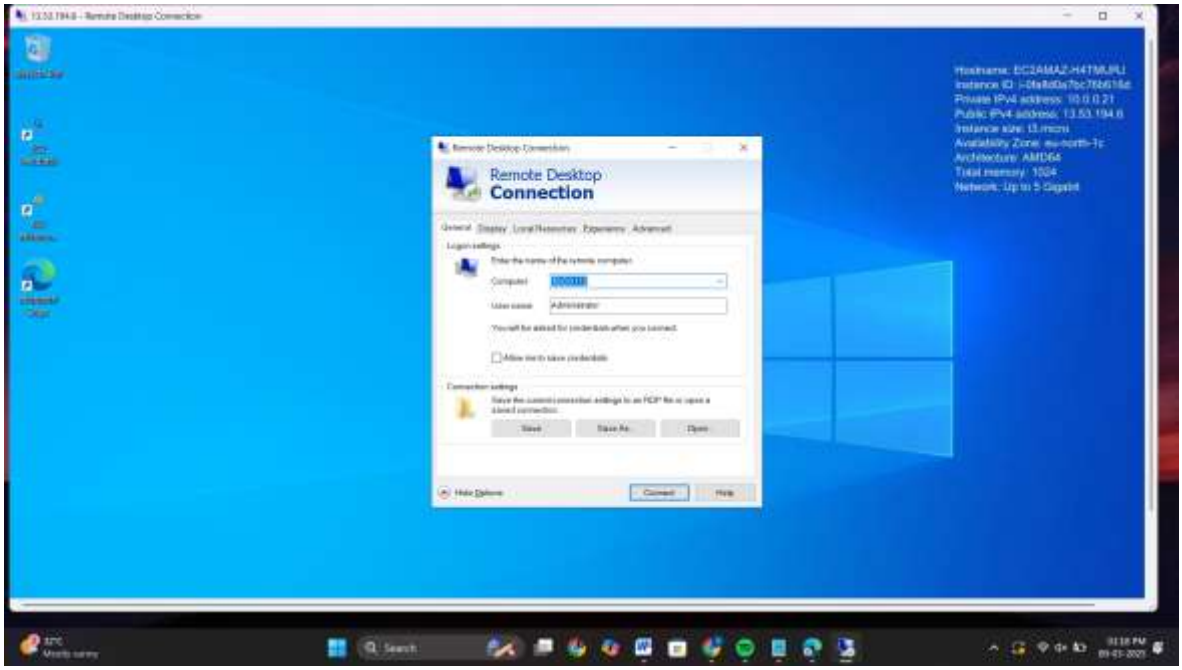
12. Connect to the Public Server Using RDP

1. **Navigate to** EC2 Instances **in AWS Console.**
2. **Select the** Public Server **instance.**
3. **Click** Connect**.**
4. **Go to the** RDP Client **tab.**
5. **Click** Get Password**.**
6. **Upload the** Key Pair **used during instance creation.**
7. **Copy the decrypted password.**
8. **Open** Remote Desktop Connection **on your local machine.**
9. **Enter the** Public IP **of the public server.**
10. **Enter the** Username **(default: `Administrator`).**
11. **Enter the** Password **from AWS.**
12. **Click** Connect**.**



13.Connect to the Private Server from the Public Server

1. Open the **Remote Desktop Connection** inside the **Public Server**.
2. Enter the **Private IP** of the private server.
3. Use the same **Username** and **Password**.
4. Click **Connect**.
5. The connection to the **Private Server** should now be established.



NOTE: Since the private server does not have a public IP, it can only be accessed from within the VPC. The public server acts as a **jump server (bastion host)**, enabling secure access to internal resources.

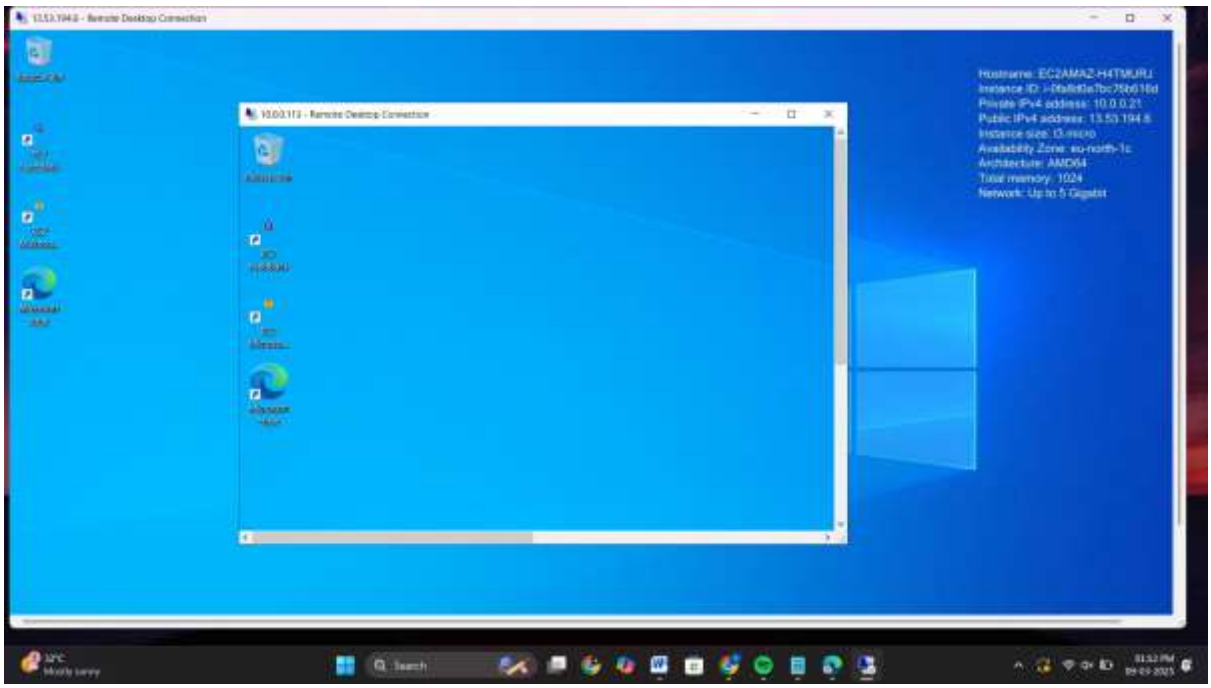
Summary

By following these steps, we have successfully:

- **Created a custom** VPC **(`10.0.0.0/25` CIDR)**
- **Defined** Public **and** Private Subnets

- **Configured** Route Tables **for traffic management**
- **Created and attached an** Internet Gateway **for public access**
- **Configured the** Public Route Table **to allow internet access**
- **Connected to the** Public Server **via RDP**
- **Connected to the** Private Server **through the Public Server using its Private IP**

This setup ensures that public resources have direct internet access, while private resources remain secure, following best networking practices in AWS.



THANK YOU!

By: Kota Asha

Mentor: GuruTeja sir