Project Title: Creating a VPC with Public and Private Subnets

Objective: To design and deploy a secure and organized network infrastructure in AWS by creating a Virtual Private Cloud (VPC) that includes both public and private subnets, an Internet Gateway, a NAT Gateway, and route tables to manage internet access.

Step-by-Step Process:

1. Created a VPC: - **Name:** MyCustomVPC - **CIDR Block:** 10.0.0.0/16

2. Created Two Subnets: - **Public Subnet:** - Name: Public-Subnet - CIDR Block: 10.0.1.0/24 - Availability Zone: ap-south-1a

- **Private Subnet:**
- Name: Private-Subnet
- CIDR Block: 10.0.2.0/24
- Availability Zone: ap-south-1a

3. Created an Internet Gateway (IGW): - Created and attached to MyCustomVPC

4. Created Route Tables: - **Public Route Table:** - Route to 0.0.0.0/0 via Internet Gateway - Associated with Public Subnet

- **Private Route Table:**
- Route to 0.0.0.0/0 via NAT Gateway (after it was created)
- Associated with Private Subnet

5. Created and Allocated an Elastic IP: - Reserved an Elastic IP for use with the NAT Gateway

6. Created a NAT Gateway: - Placed inside the Public Subnet - Attached the allocated Elastic IP

7. Launched EC2 Instances (Optional Test): - Public EC2 Instance: - With Public IP - Verified internet connectivity (ping google.com)

- Private EC2 Instance:
- Without Public IP
- Verified outbound internet via NAT Gateway

8. Took Screenshots: - VPC configuration - Subnets - Route Tables - IGW - NAT Gateway - Elastic IP - EC2 instances (optional)

Conclusion: This project demonstrates a standard AWS VPC setup with a clear separation of public and private resources. Internet access is enabled securely using an Internet Gateway for public resources and a NAT Gateway for private instances, following AWS best practices.

Tools Used: - AWS Management Console - VPC, EC2, and Networking services

Next Steps (Optional): - Deploy a web server in the public subnet - Connect a database in the private subnet - Implement security groups and NACLs for more control