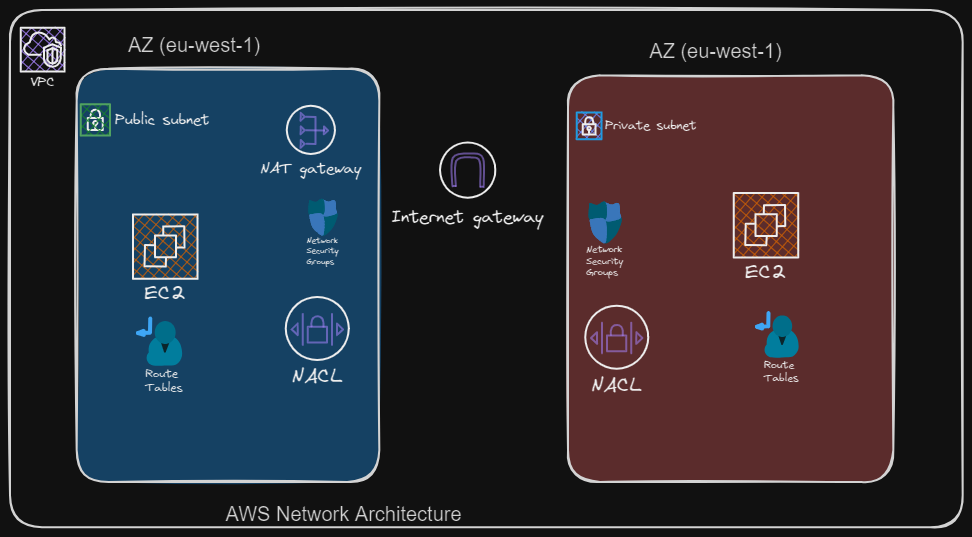
Objective:

To set up a Virtual Private Cloud (VPC) with both public and private subnets in the AWS EU-West-1 (Ireland) region that includes creating and configuring the VPC, subnets, Internet Gateway (IGW), NAT Gateway, route tables, security groups, and Network Access Control Lists (NACLs) as well as deploying instances and verifying their access and connectivity.



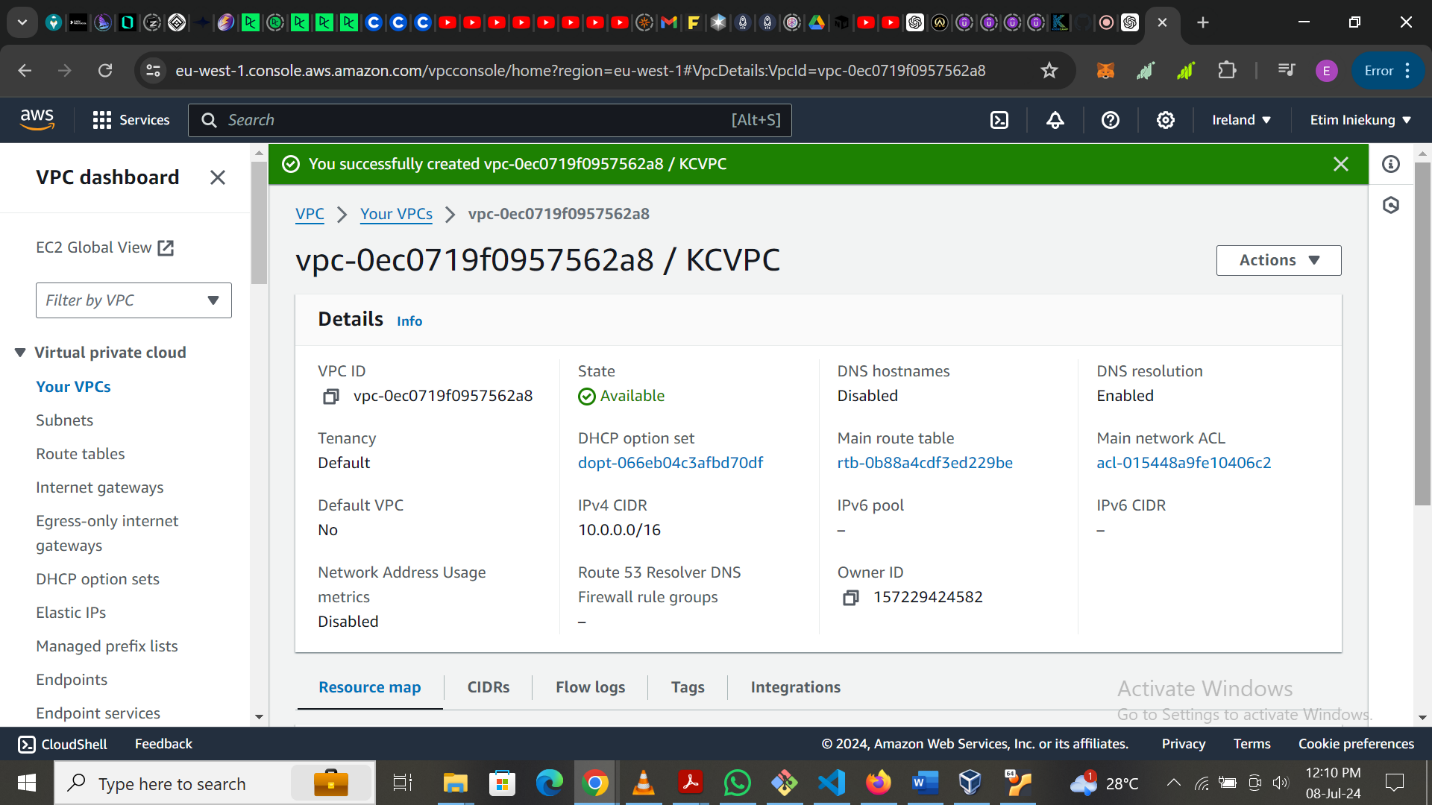
### Step 1: Create a VPC

1. Log in to the AWS Management Console and navigate to the VPC Dashboard.

2. Create VPC:

- Name: `KCVPC`

- IPv4 CIDR block: `10.0.0.0/16`



### Step 2: Create Subnets

1. Create Public Subnet:

- Name: `PublicSubnet`

- VPC: `KCVPC`

- IPv4 CIDR block: `10.0.1.0/24`

- Availability Zone: Select any one from EU-West-1 (e.g., `eu-west-1a`)

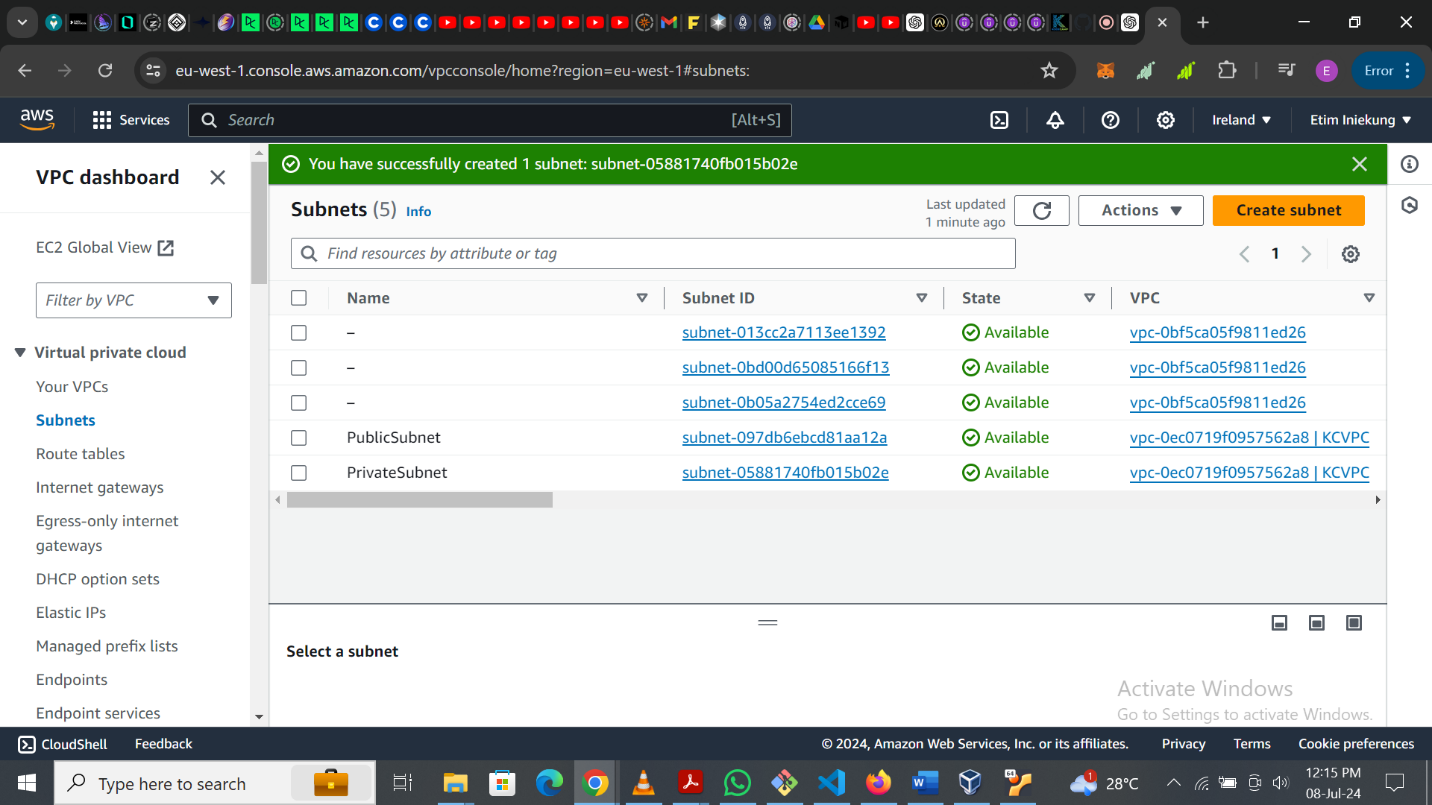
2. Create Private Subnet:

- Name: `PrivateSubnet`

- VPC: `KCVPC`

- IPv4 CIDR block: `10.0.2.0/24`

- Availability Zone: Select the same as Public Subnet (e.g., `eu-west-1a`)

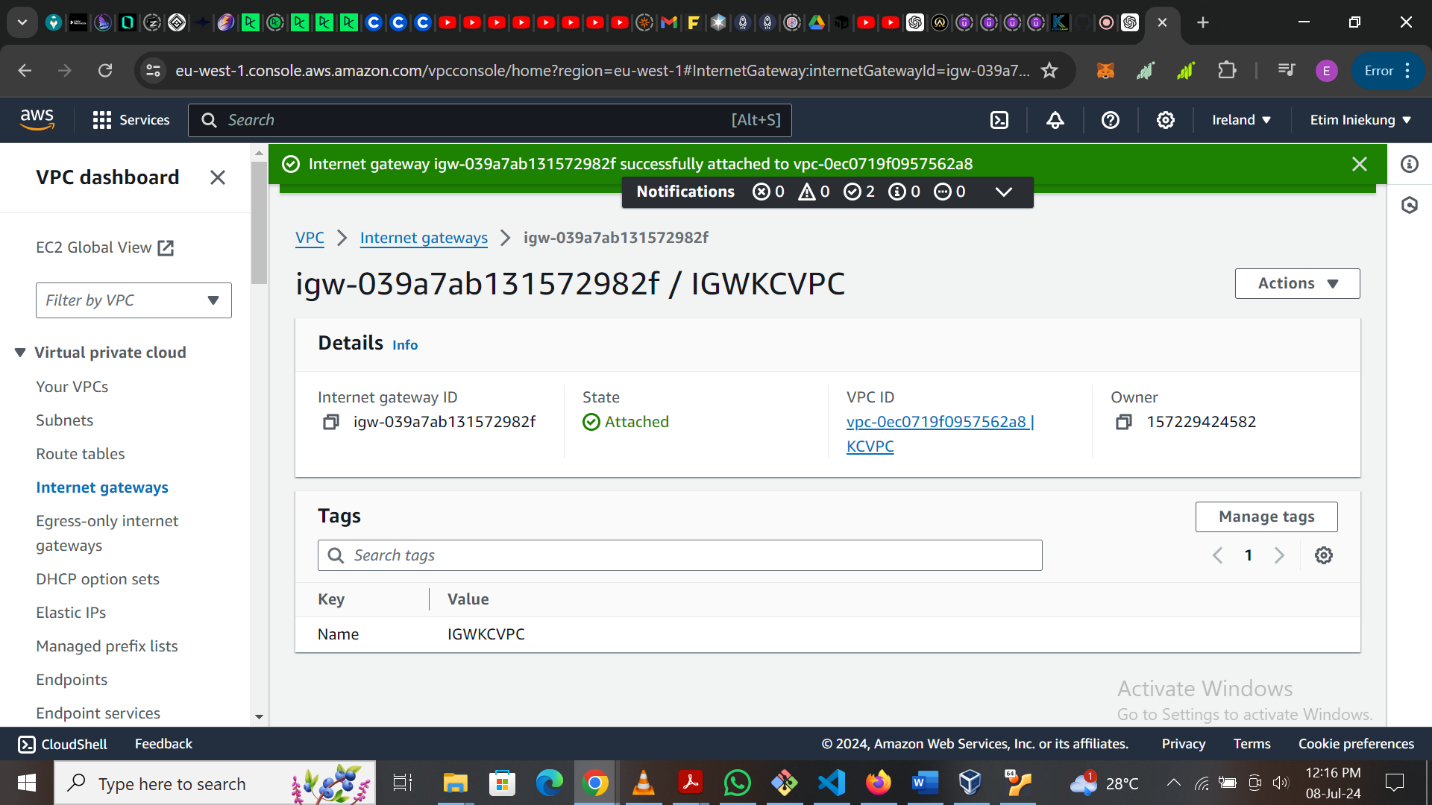


### Step 3: Configure an Internet Gateway (IGW)

1. Create and attach an IGW:

- Name: `KCVPC-IGW`

- Attach it to `KCVPC`.



### Step 4: Configure Route Tables

1. Create Public Route Table:

- Name: `PublicRouteTable`

- VPC: `KCVPC`

- Associate `PublicSubnet` with this route table.

- Add a route to the IGW: `0.0.0.0/0 -> KCVPC-IGW`.

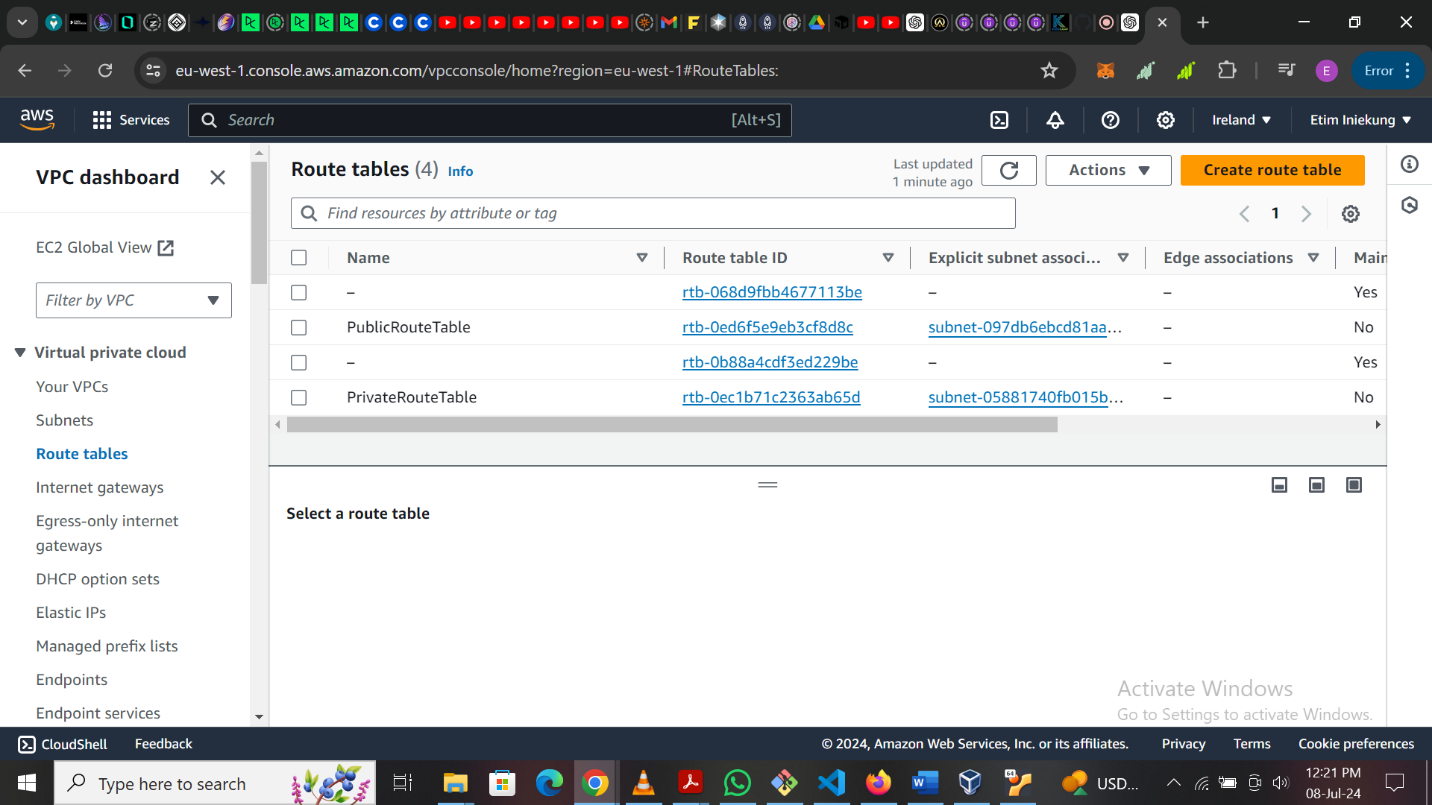
2. Create Private Route Table:

- Name: `PrivateRouteTable`

- VPC: `KCVPC`

- Associate `PrivateSubnet` with this route table.

- No direct route to the internet initially.



### Step 5: Configure NAT Gateway

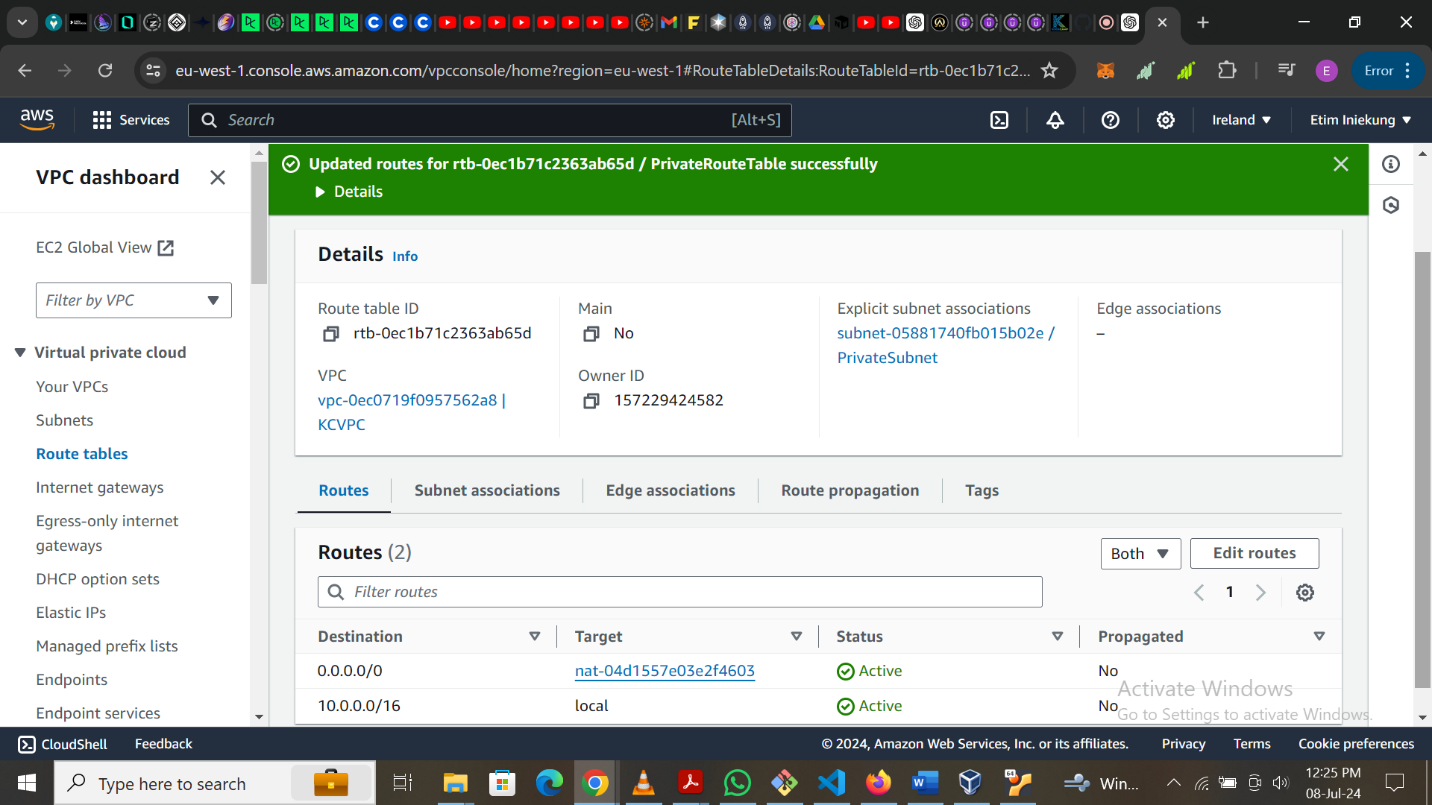
1. Create a NAT Gateway:

- Subnet: `PublicSubnet`

- Allocate a new Elastic IP for the NAT Gateway.

2. Update the PrivateRouteTable:

- Add a route to the NAT Gateway: `0.0.0.0/0 -> NAT Gateway`.



### Step 6: Set Up Security Groups

1. Create a Security Group for Public Instances:

- Name: `PublicSG`

- VPC: `KCVPC`

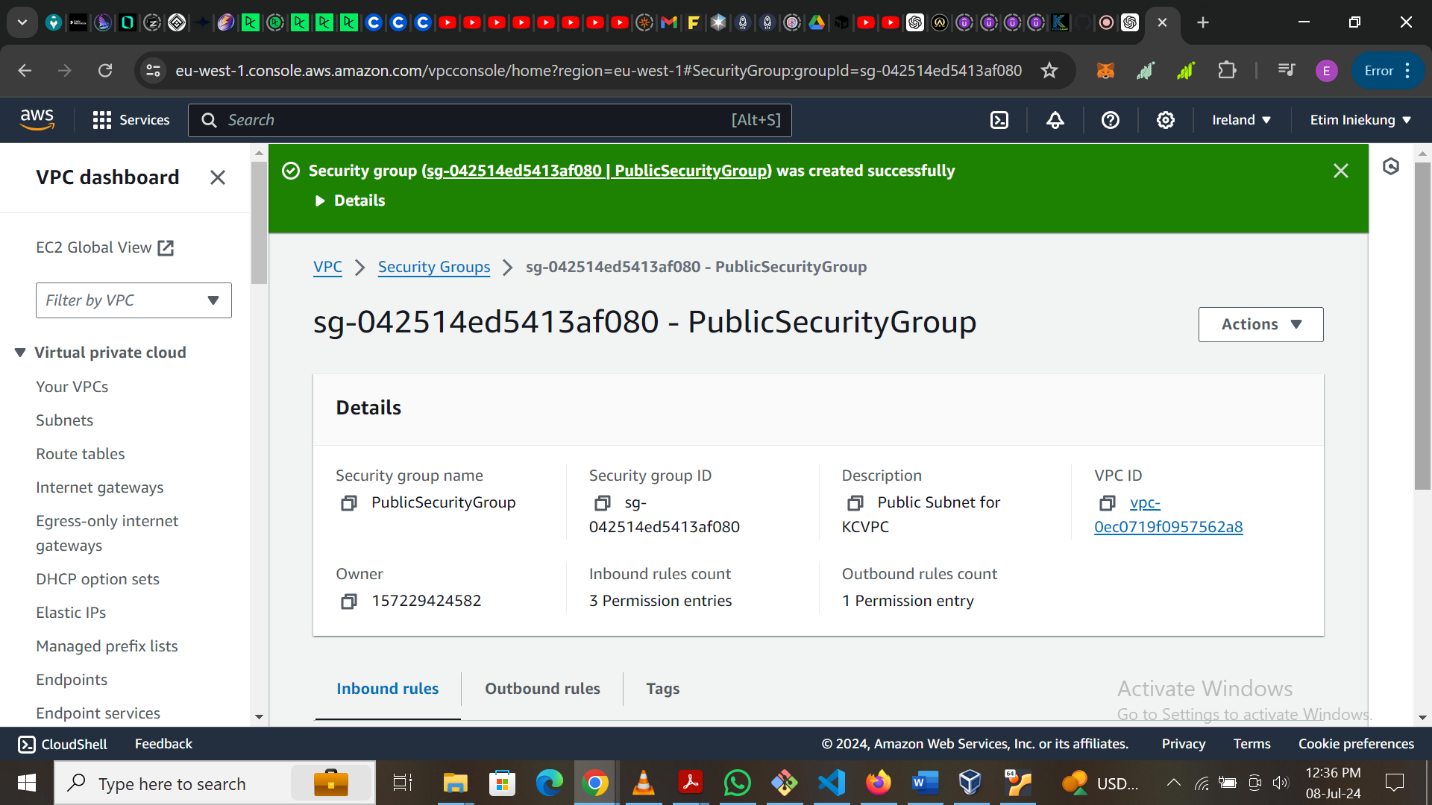
- Inbound rules:

- HTTP (port 80): Source `0.0.0.0/0`

- HTTPS (port 443): Source `0.0.0.0/0`

- SSH (port 22): Source `<your-local-IP>/32` (find your IP from [whatismyip.com](https://www.whatismyip.com/))

- Outbound rules: Allow all traffic.



2. Create a Security Group for Private Instances:

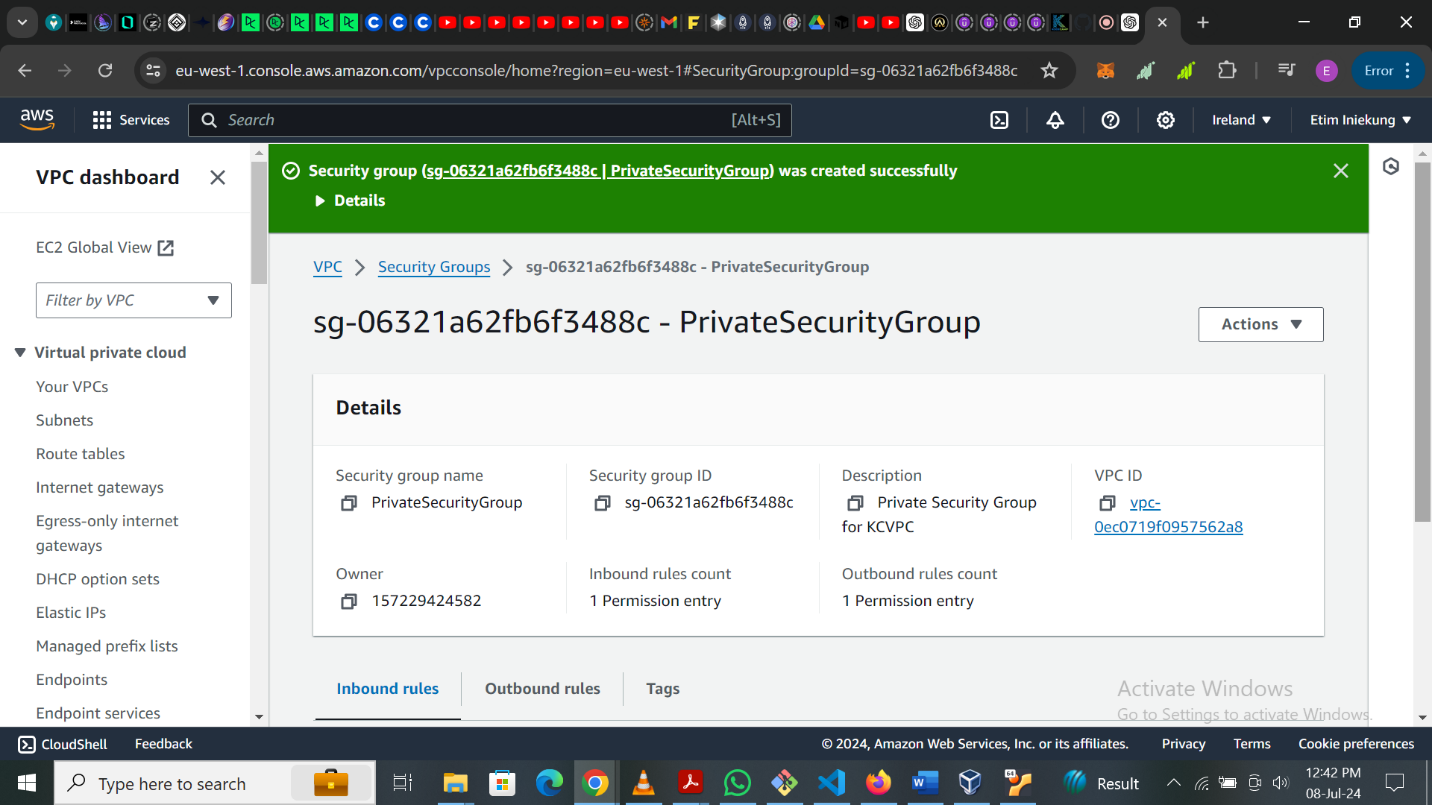
- Name: `PrivateSG`

- VPC: `KCVPC`

- Inbound rules:

- MySQL (port 3306): Source `PublicSubnet` CIDR block `10.0.1.0/24`

- Outbound rules: Allow all traffic.



### Step 7: Configure Network ACLs

1. Public Subnet NACL:

- Inbound rules:

- HTTP (port 80): Source `0.0.0.0/0`

- HTTPS (port 443): Source `0.0.0.0/0`

- SSH (port 22): Source `<your-local-IP>/32`

- Outbound rules: Allow all traffic.

2. Private Subnet NACL:

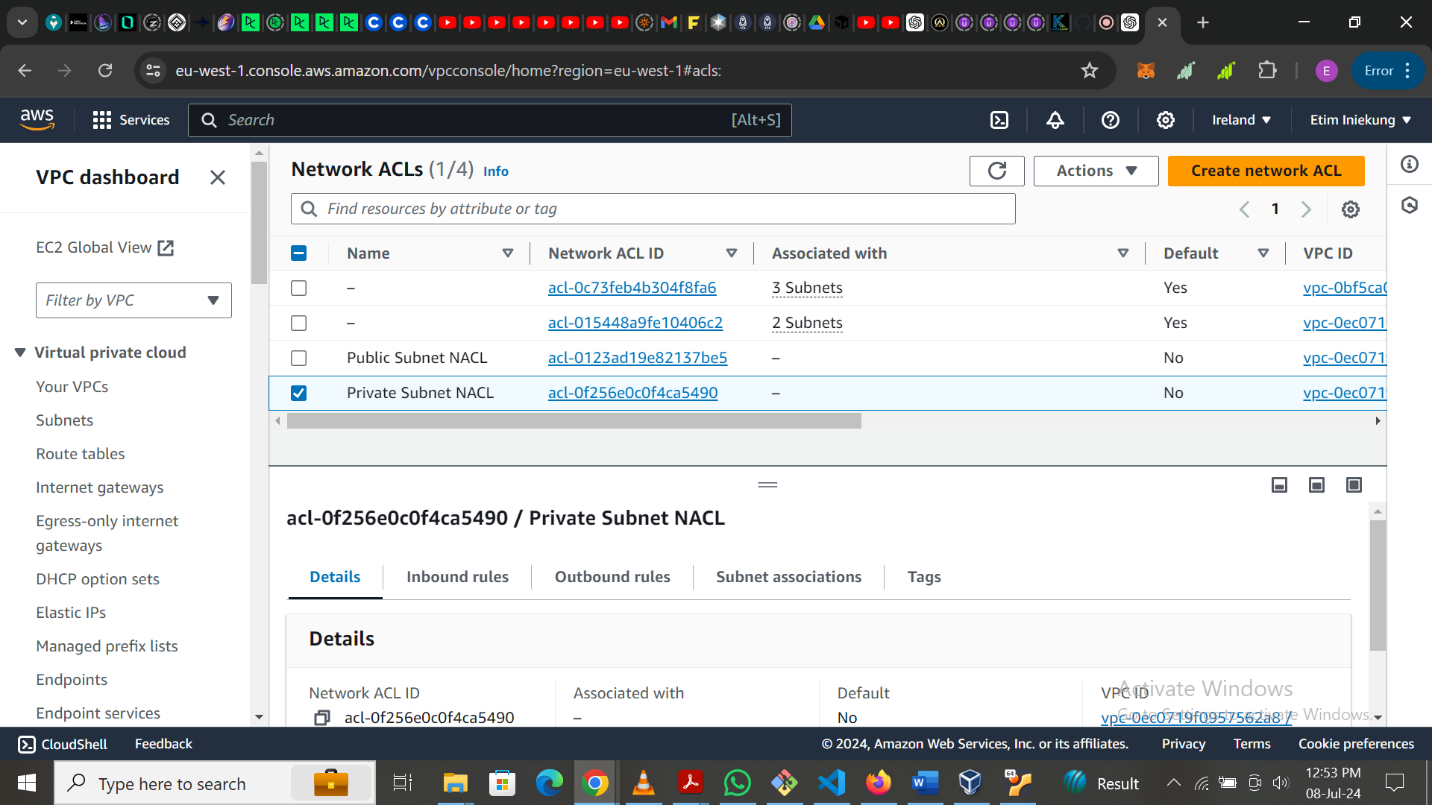
- Inbound rules:

- Allow traffic from `PublicSubnet` CIDR block `10.0.1.0/24`

- Outbound rules:

- Allow traffic to `PublicSubnet` CIDR block `10.0.1.0/24`

- Allow traffic to the internet `0.0.0.0/0`



### Step 8: Deploy Instances

1. Launch an EC2 instance in the PublicSubnet:

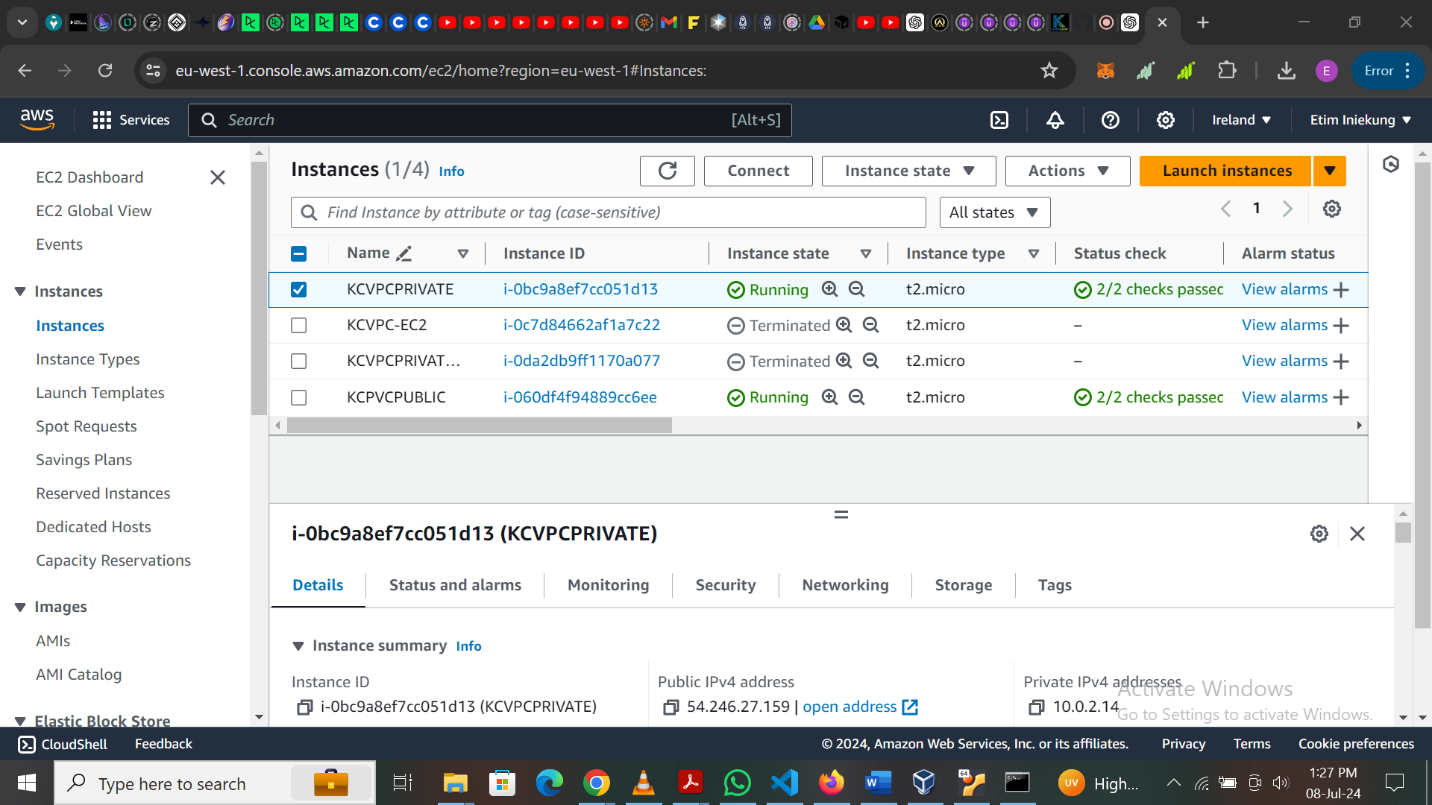
- Use the public security group `PublicSG`.

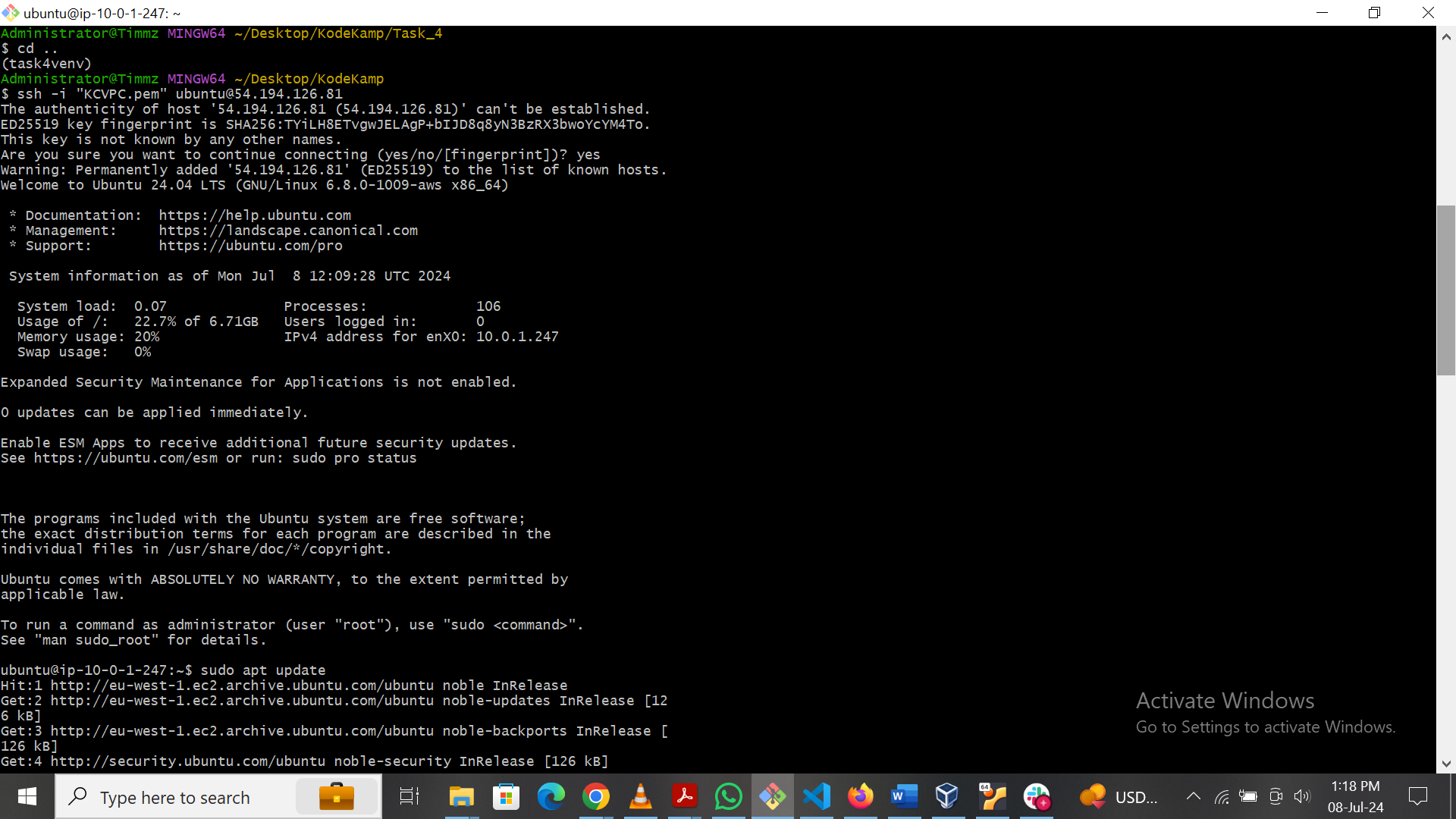
- Verify that the instance can be accessed via the internet (SSH using `<your-local-IP>`).

2. Launch an EC2 instance in the PrivateSubnet:

- Use the private security group `PrivateSG`.

- Verify that the instance can access the internet through the NAT Gateway and can communicate with the public instance.





### Deliverables

Brief Explanation of Each Component:

- VPC: A logically isolated network within AWS where you can launch AWS resources.

- Subnets: Subdivision within a VPC to place resources in different segments (public/private).

- Internet Gateway (IGW): Allows internet access to instances in the public subnet.

- NAT Gateway: Enables instances in the private subnet to access the internet without allowing inbound traffic.

- Route Tables: Define rules for traffic flow within the VPC and to/from the internet.

- Security Groups: Virtual firewalls controlling inbound and outbound traffic for instances.

- Network ACLs: Additional layer of security controlling inbound and outbound traffic at the subnet level.

Once you complete these steps, you'll have a fully functional VPC with public and private subnets, ensuring proper communication and security within the VPC.