Setting Up a Secure AWS VPC with Public and Private Subnets: A Step-by-Step Guide

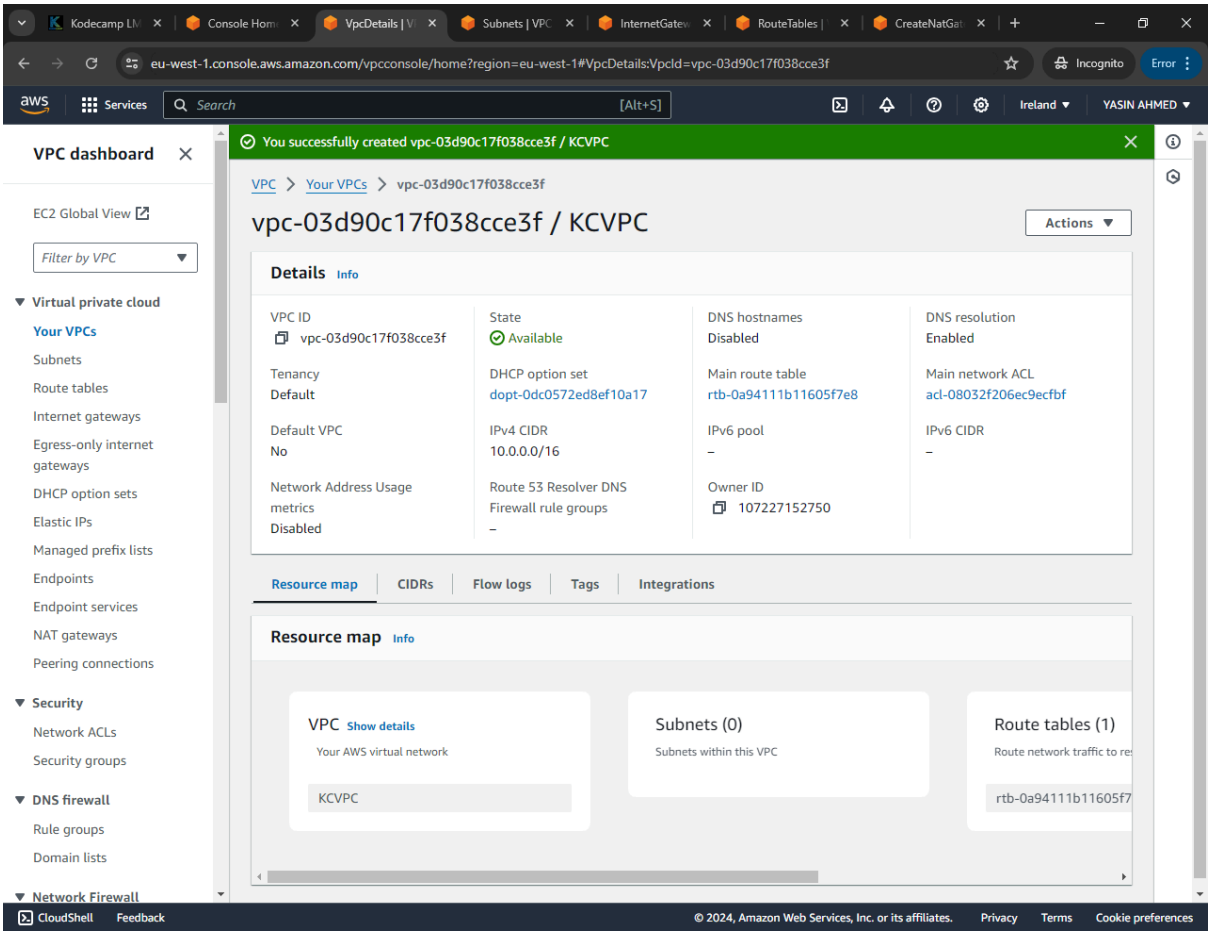
This document outlines the steps to create and configure a Virtual Private Cloud (VPC) with both public and private subnets on AWS, including routing, security groups, and network access control lists (NACLs) to ensure proper communication and security within the VPC.

Objective

Design and set up a Virtual Private Cloud (VPC) with both public and private subnets. Implement routing, security groups, and network access control lists (NACLs) to ensure proper communication and security within the VPC. Work in the AWS EU-West-1 (Ireland) region.

Steps and Screenshots

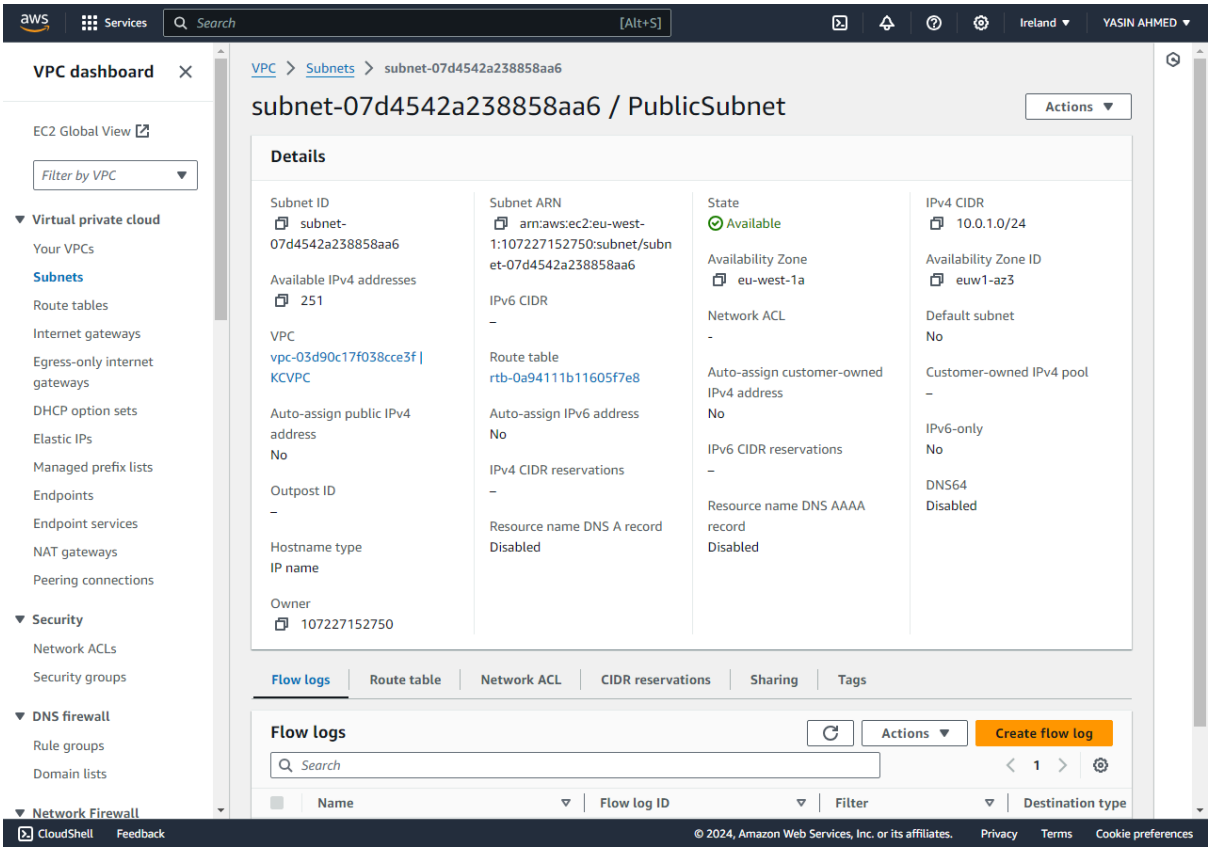
1. Create VPC
1. Go to the VPC Dashboard in the AWS Management Console.
2. Click on "Create VPC".
3. Enter the name "KCVPC" and the IPv4 CIDR block "10.0.0.0/16".
4. Click "Create".



2. Create Subnets

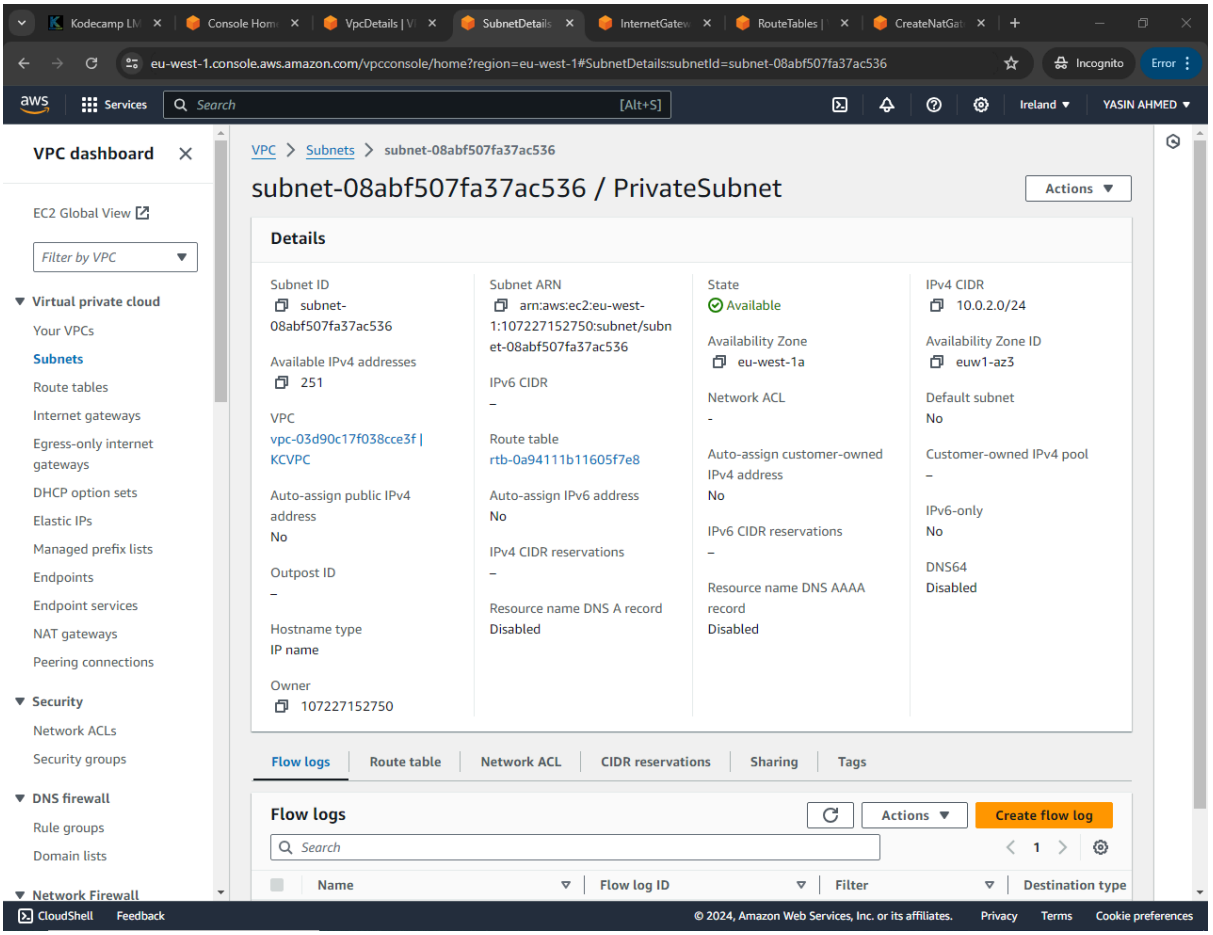
Public Subnet

1. In the VPC Dashboard, click on "Subnets" and then "Create Subnet".
2. Select the VPC "KCVPC".
3. Enter the name "PublicSubnet" and the IPv4 CIDR block "10.0.1.0/24".
4. Select an availability zone (e.g., eu-west-1a).
5. Click "Create".



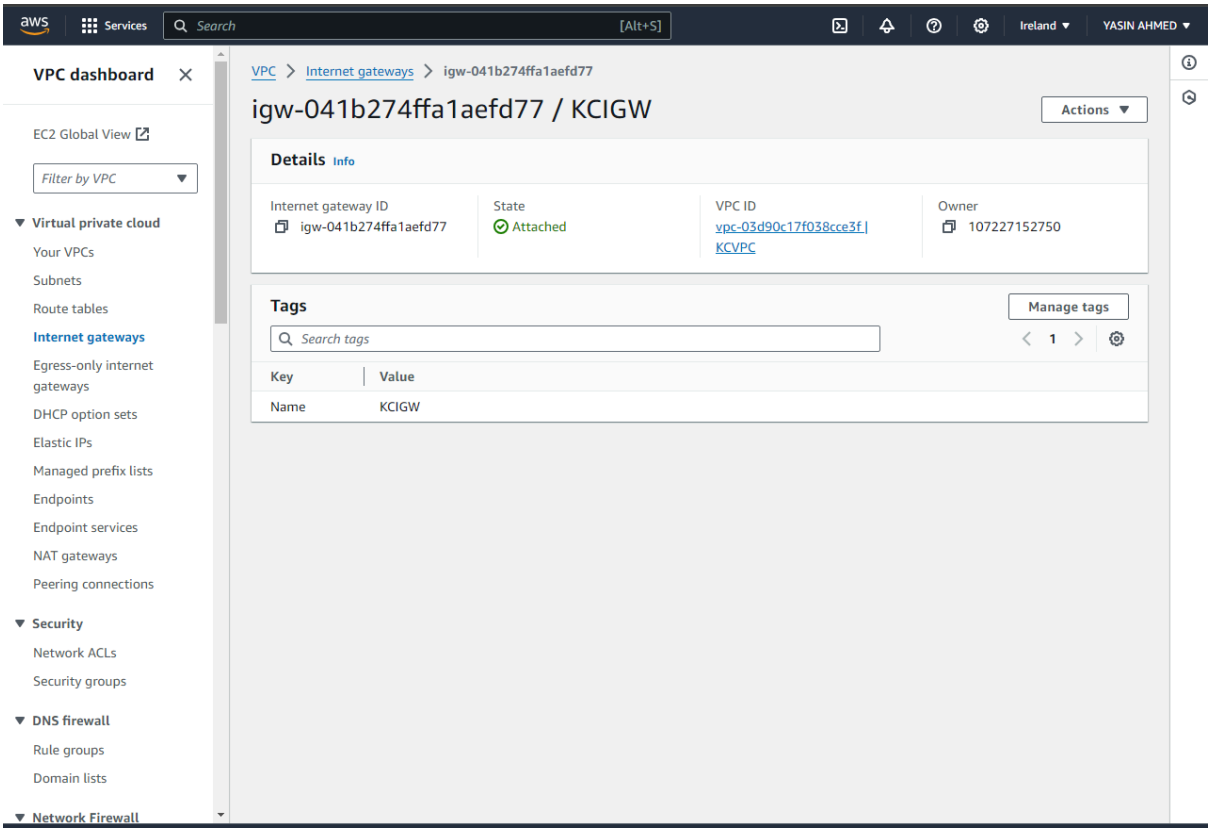
Private Subnet

1. Repeat the above steps to create the private subnet with the name "PrivateSubnet" and the IPv4 CIDR block "10.0.2.0/24".



3. Configure Internet Gateway

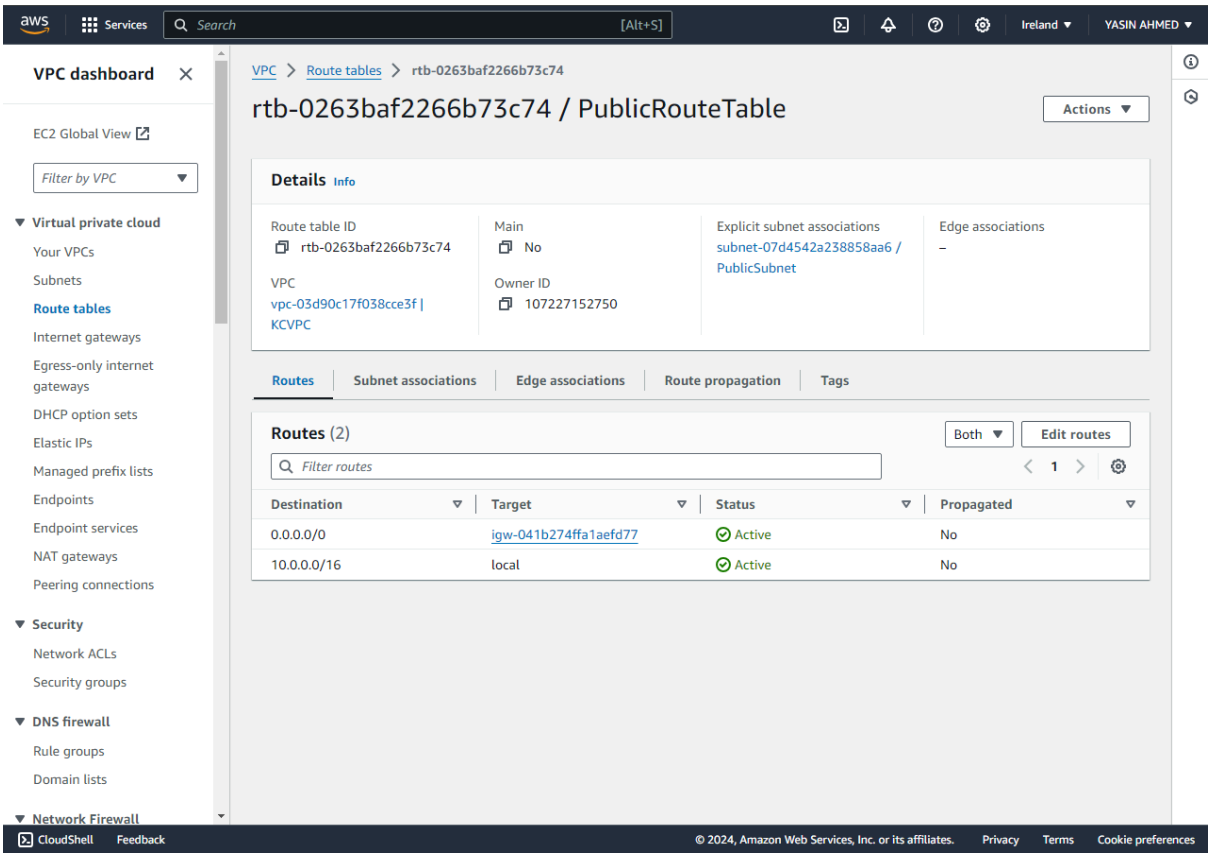
1. In the VPC Dashboard, click on "Internet Gateways" and then "Create Internet Gateway".
2. Enter the name "KCIGW" and click "Create".
3. Select the created IGW and click "Actions" -> "Attach to VPC".
4. Select the VPC "KCVPC" and click "Attach".



4. Configure Route Tables

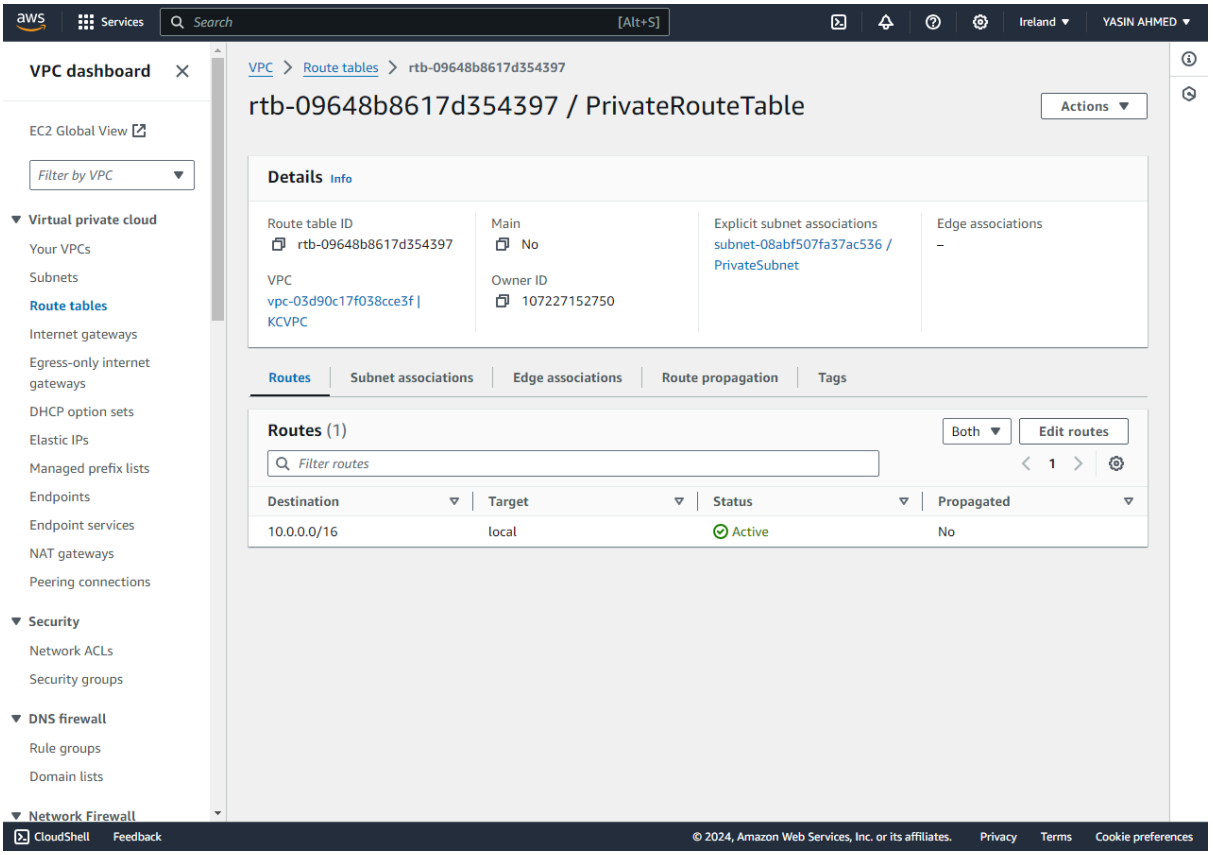
Public Route Table

1. In the VPC Dashboard, click on "Route Tables" and then "Create Route Table".
2. Enter the name "PublicRouteTable" and select the VPC "KCVPC".
3. Click "Create".
4. Select the created route table and click "Actions" -> "Edit routes".
5. Add a route with Destination "0.0.0.0/0" and Target as the created IGW.
6. Click "Save routes".
7. Click "Actions" -> "Edit subnet associations".
8. Associate the PublicSubnet.

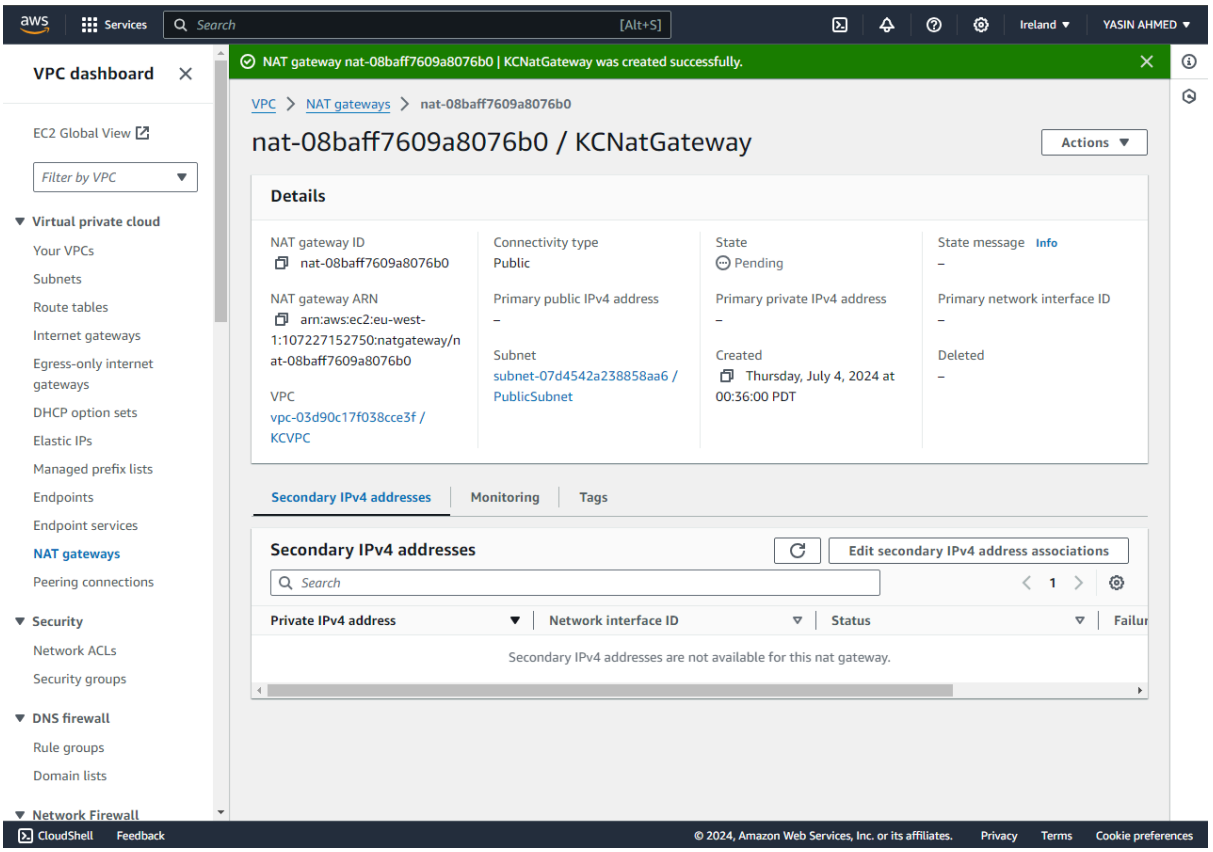


Private Route Table

1. Repeat the above steps to create the private route table with the name "PrivateRouteTable".
2. Ensure there is no direct route to the internet.



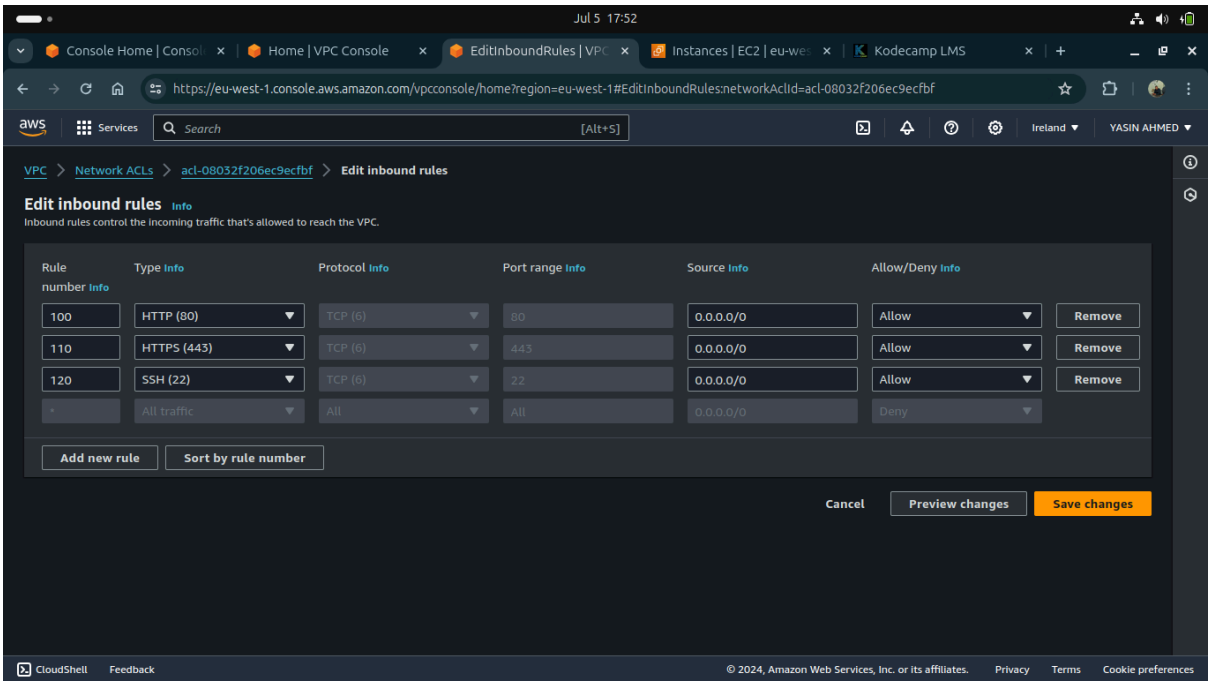
5. Configure NAT Gateway
1. In the VPC Dashboard, click on "NAT Gateways" and then "Create NAT Gateway".
2. Select the PublicSubnet and allocate an Elastic IP.
3. Click "Create".
4. Go to the PrivateRouteTable and edit the routes.
5. Add a route with Destination "0.0.0.0/0" and Target as the created NAT Gateway.



6. Set Up Security Groups

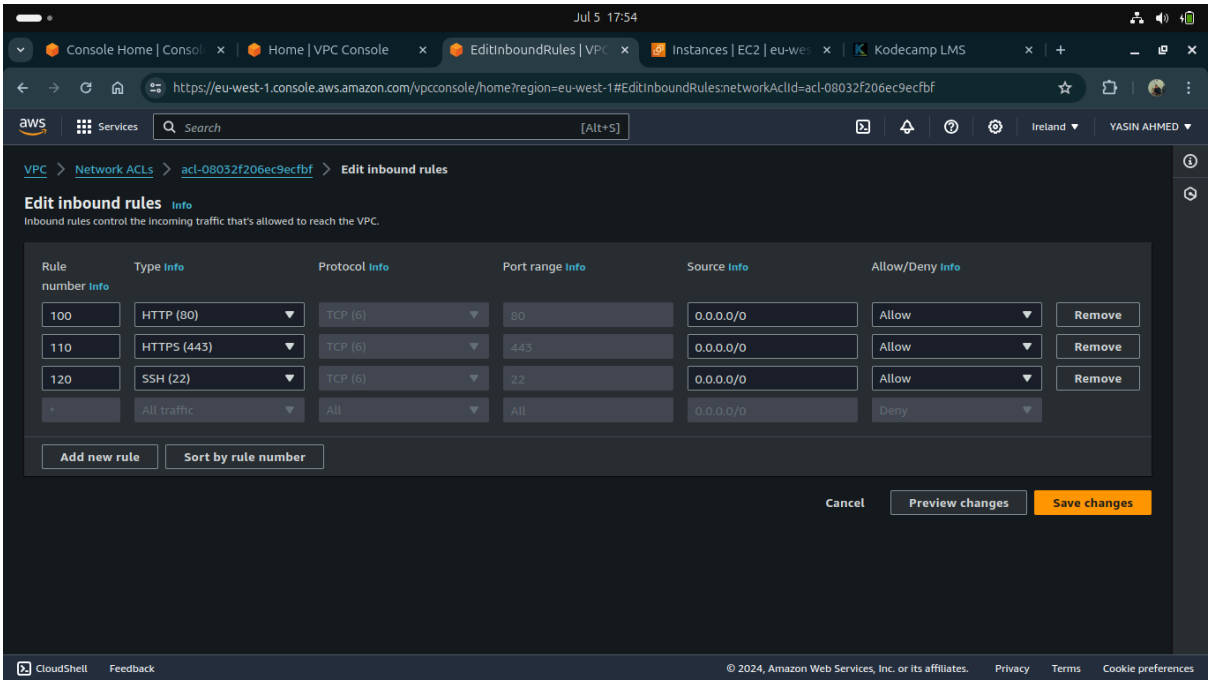
Public Security Group

1. In the EC2 Dashboard, click on "Security Groups" and then "Create Security Group".
2. Enter the name "PublicSecurityGroup", description, and select the VPC "KCVPC".
3. Add the following inbound rules:
   - HTTP (port 80) from 0.0.0.0/0
   - HTTPS (port 443) from 0.0.0.0/0
   - SSH (port 22) from my local IP
4. Allow all outbound traffic.
5. Click "Create".



Private Security Group

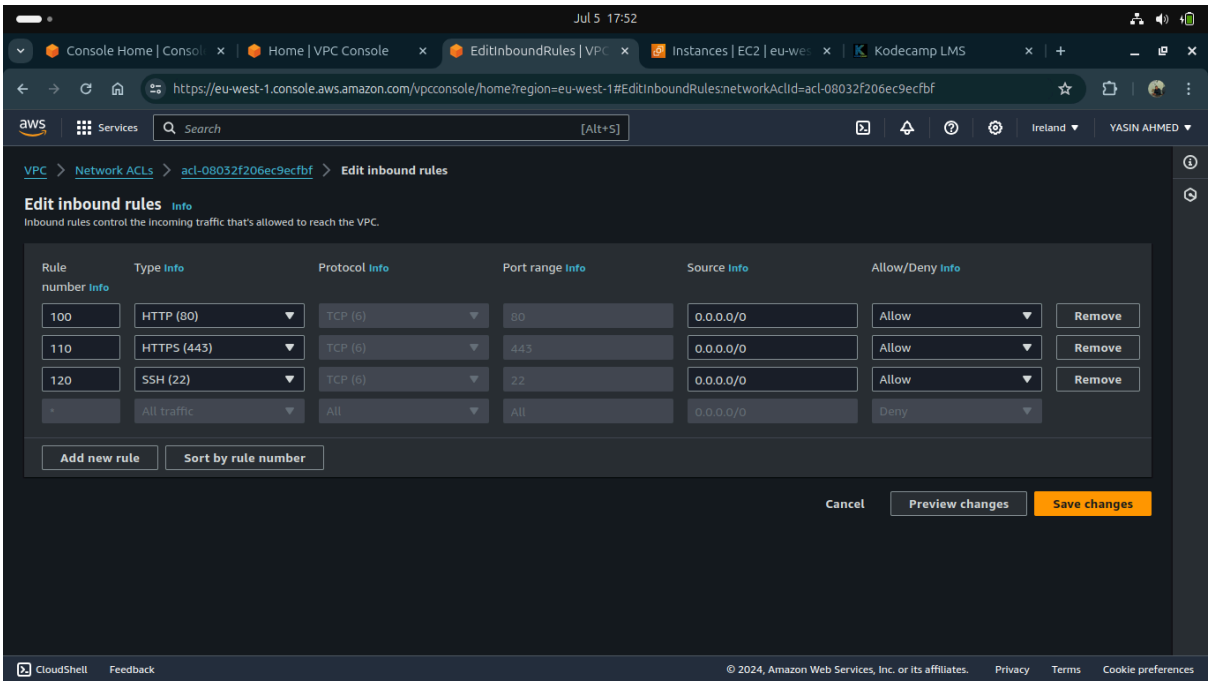
1. Repeat the above steps to create the private security group with the name "PrivateSecurityGroup".
2. Allow inbound traffic from the PublicSubnet on the required ports (e.g., MySQL port 3306).
3. Allow all outbound traffic.



7. Configure Network ACLs

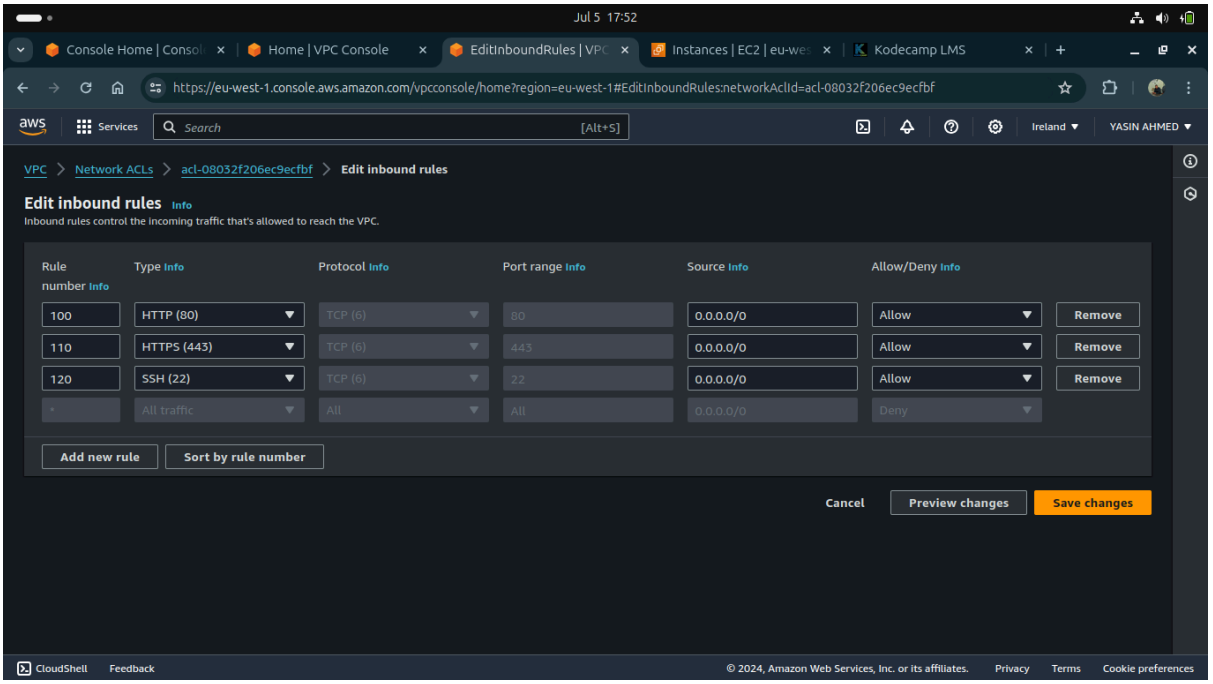
Public Subnet NACL

1. In the VPC Dashboard, click on "Network ACLs" and then "Create Network ACL".
2. Enter the name "PublicSubnetNACL" and select the VPC "KCVPC".
3. Add the following inbound rules:
   - HTTP (port 80) from 0.0.0.0/0
   - HTTPS (port 443) from 0.0.0.0/0
   - SSH (port 22) from your local IP
4. Allow all outbound traffic.
5. Associate it with the PublicSubnet.



Private Subnet NACL

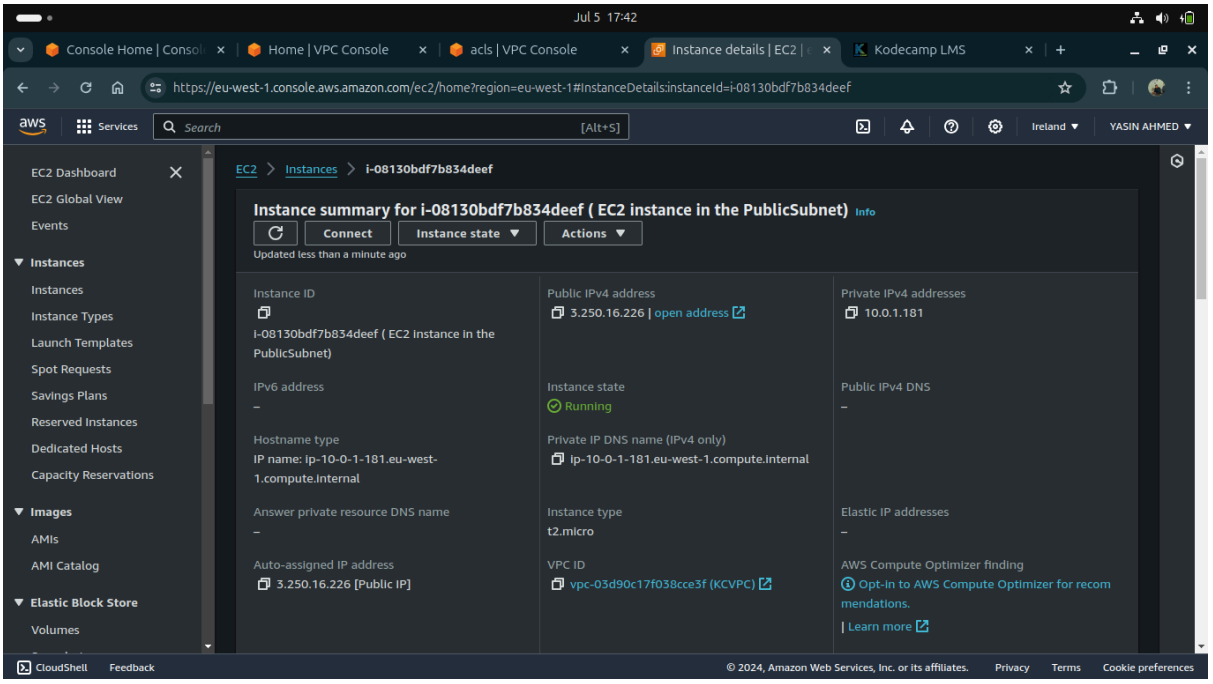
1. Repeat the above steps to create the private subnet NACL with the name "PrivateSubnetNACL".
2. Allow inbound traffic from the PublicSubnet.
3. Allow outbound traffic to the PublicSubnet and internet.



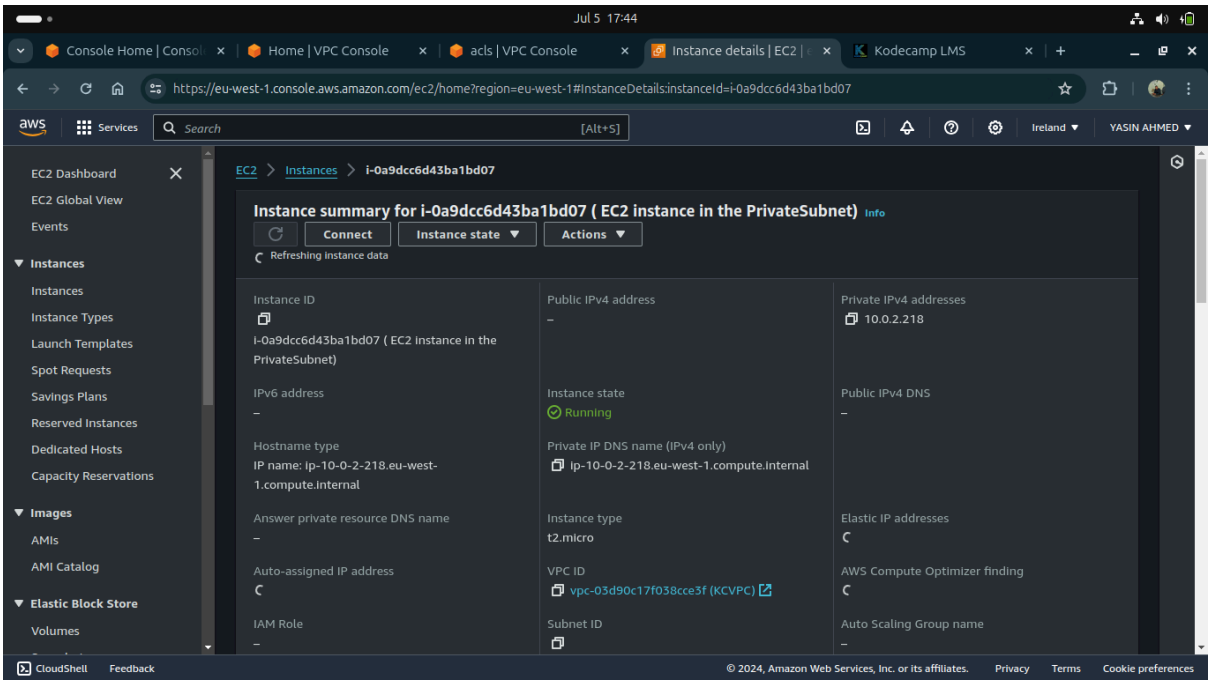
8. Deploy Instances

PublicSubnet EC2 Instance

1. In the EC2 Dashboard, click on "Launch Instance".
2. Select the desired AMI and instance type.
3. Configure the instance details to launch in the PublicSubnet and assign the PublicSecurityGroup.
4. Launch the instance and verify that it can be accessed via the internet.



PrivateSubnet EC2 Instance
1. Repeat the above steps to launch an instance in the PrivateSubnet and assign the PrivateSecurityGroup.
2. Verify that the instance can access the internet through the NAT Gateway and can communicate with the public instance.



Running instance

```
ahmedii@Lenovo:~/Downloads/keypair$ chmod 400 "KcKeypair.pem"
ahmedii@Lenovo:~/Downloads/keypair$ ssh -i "KcKeypair.pem" ubuntu@3.253.97.65
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-1009-aws x86_64)
```

A brief explanation of the purpose and function of each component created in the AWS VPC setup:

Virtual Private Cloud (VPC)
Purpose and Function:

A VPC is a logically isolated section of the AWS cloud where you can launch AWS resources in a virtual network that you define.
It allows for the customization of the network configuration, including selection of IP address ranges, creation of subnets, and configuration of route tables and gateways.

Subnets
Public Subnet:

Purpose: To host resources that need to be accessible from the internet (e.g., web servers).

Function: Provides a range of IP addresses within the VPC. It is associated with a route table that directs internet-bound traffic to the Internet Gateway.

Private Subnet:

Purpose: To host resources that should not be directly accessible from the internet (e.g., databases).
Function: Provides a range of IP addresses within the VPC. It is associated with a route table that does not direct traffic to the Internet Gateway, ensuring the resources remain private.

Internet Gateway (IGW)

Purpose and Function:

An IGW is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in the VPC and the internet.
It provides a target in your VPC route tables for internet-routable traffic and performs network address translation (NAT) for instances that have been assigned public IP addresses.

NAT Gateway

Purpose and Function:

A NAT Gateway allows instances in the private subnet to connect to the internet or other AWS services, but prevents the internet from initiating a connection with those instances.
It is used to provide outbound internet access for instances in a private subnet while keeping them secure from inbound internet traffic.

Route Tables

Public Route Table:

Purpose: To manage the routing of network traffic within the VPC for the public subnet.
Function: Contains routes that direct traffic to the IGW for internet-bound traffic and to other subnets within the VPC.

Private Route Table:

Purpose: To manage the routing of network traffic within the VPC for the private subnet.
Function: Contains routes that direct internet-bound traffic to the NAT Gateway and to other subnets within the VPC, without direct access to the IGW.

Security Groups

Public Security Group:

Purpose: To control the inbound and outbound traffic for instances in the public subnet.
Function: Allows inbound traffic on HTTP (80), HTTPS (443), and SSH (22) ports, and allows all outbound traffic.

Private Security Group:

Purpose: To control the inbound and outbound traffic for instances in the private subnet.

Function: Allows inbound traffic from the public subnet on required ports (e.g., MySQL 3306) and allows all outbound traffic.

Network Access Control Lists (NACLs)
Public Subnet NACL:
Purpose: To provide an additional layer of security at the subnet level by controlling inbound and outbound traffic.
Function: Allows inbound HTTP, HTTPS, and SSH traffic, and all outbound traffic.

Private Subnet NACL:
Purpose: To provide an additional layer of security at the subnet level by controlling inbound and outbound traffic.
Function: Allows inbound traffic from the public subnet and all outbound traffic to the public subnet and the internet.