

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.

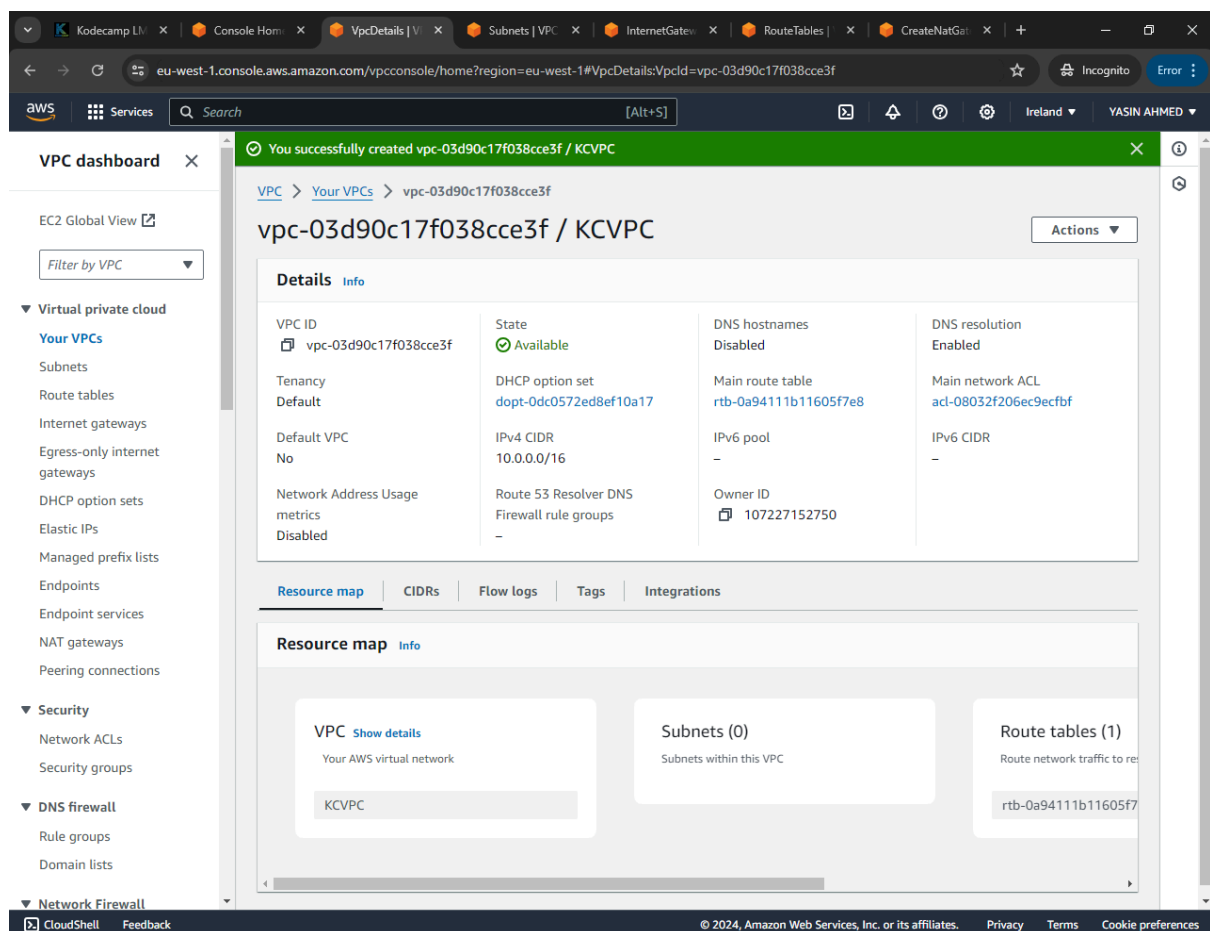
Objective

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".

The screenshot displays the AWS VPC console interface. The left-hand navigation pane shows the 'VPC dashboard' with a sidebar menu containing categories like 'Virtual private cloud', 'Security', 'DNS firewall', and 'Network Firewall'. The main content area is titled 'subnet-07d4542a238858aa6 / PublicSubnet'. Below the title, a 'Details' section provides a comprehensive overview of the subnet's configuration, including its ID, ARN, state (Available), availability zone (eu-west-1a), and associated VPC (KCVPC). A 'Flow logs' section at the bottom offers options to view logs, route tables, network ACLs, and other related resources. The footer of the console includes a copyright notice for Amazon Web Services, Inc. and links to privacy, terms, and cookie preferences.

Details			
Subnet ID subnet-07d4542a238858aa6	Subnet ARN arn:aws:ec2:eu-west-1:107227152750:subnet/subnet-07d4542a238858aa6	State Available	IPv4 CIDR 10.0.1.0/24
Available IPv4 addresses 251	IPv6 CIDR -	Availability Zone eu-west-1a	Availability Zone ID euw1-az3
VPC vpc-03d90c17f038cce3f KCVPC	Route table rtb-0a94111b11605f7e8	Network ACL -	Default subnet No
Auto-assign public IPv4 address No	Auto-assign IPv6 address No	Auto-assign customer-owned IPv4 address No	Customer-owned IPv4 pool -
Outpost ID -	IPv4 CIDR reservations -	IPv6 CIDR reservations -	IPv6-only No
Hostname type IP name	Resource name DNS A record Disabled	Resource name DNS AAAA record Disabled	DNS64 Disabled
Owner 107227152750			

Flow logs | Route table | Network ACL | CIDR reservations | Sharing | Tags

Flow logs

Search

1

Name | Flow log ID | Filter | Destination type

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

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".

The screenshot displays the AWS Management Console interface for a VPC dashboard. The main content area shows the details for the subnet 'subnet-08abf507fa37ac536 / PrivateSubnet'. The details are organized into a grid with the following information:

Details			
Subnet ID subnet-08abf507fa37ac536	Subnet ARN arn:aws:ec2:eu-west-1:107227152750:subnet/subnet-08abf507fa37ac536	State Available	IPv4 CIDR 10.0.2.0/24
Available IPv4 addresses 251	IPv6 CIDR -	Availability Zone eu-west-1a	Availability Zone ID euw1-az3
VPC vpc-03d90c17f038cce3f KCVPC	Route table rtb-0a94111b11605f7e8	Network ACL -	Default subnet No
Auto-assign public IPv4 address No	Auto-assign IPv6 address No	Auto-assign customer-owned IPv4 address No	Customer-owned IPv4 pool -
Outpost ID -	IPv4 CIDR reservations -	IPv6 CIDR reservations -	IPv6-only No
Hostname type IP name	Resource name DNS A record Disabled	Resource name DNS AAAA record Disabled	DNS64 Disabled
Owner 107227152750			

Below the details, there are tabs for 'Flow logs', 'Route table', 'Network ACL', 'CIDR reservations', 'Sharing', and 'Tags'. The 'Flow logs' tab is active, showing a search bar and a 'Create flow log' button. At the bottom, there is a table header for flow logs with columns for Name, Flow log ID, Filter, and Destination type.

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".

The screenshot displays the AWS VPC dashboard. The left sidebar shows the navigation menu with categories like Virtual private cloud, Security, DNS firewall, and Network Firewall. The main content area shows the details for the Internet Gateway **igw-041b274ffa1aefd77 / KCIGW**. The details section includes the Internet gateway ID, State (Attached), VPC ID (vpc-03d90c17f038cce3f1), and Owner (107227152750). The Tags section shows a single tag with Key 'Name' and Value 'KCIGW'.

VPC dashboard ×

EC2 Global View

Filter by VPC ▼

▼ Virtual private cloud

- Your VPCs
- Subnets
- Route tables
- Internet gateways**
- Egress-only internet gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists
- Endpoints
- Endpoint services
- NAT gateways
- Peering connections

▼ Security

- Network ACLs
- Security groups

▼ DNS firewall

- Rule groups
- Domain lists

▼ Network Firewall

VPC > Internet gateways > igw-041b274ffa1aefd77

igw-041b274ffa1aefd77 / KCIGW Actions ▼

Details Info

Internet gateway ID igw-041b274ffa1aefd77	State Attached	VPC ID vpc-03d90c17f038cce3f1 KCVPC	Owner 107227152750
----------------------------------------------	-------------------	---------------------------------------------------------------------------	-----------------------

Tags Manage tags

Search tags

Key	Value
Name	KCIGW

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.

The screenshot displays the AWS VPC console interface. The left sidebar shows the navigation menu with categories like Virtual private cloud, Security, DNS firewall, and Network Firewall. The main content area is titled "rtb-0263baf2266b73c74 / PublicRouteTable". Below the title, there are tabs for "Details" and "Info". The "Details" tab is active, showing a table with four columns: Route table ID, Main, Explicit subnet associations, and Edge associations. The "Routes" tab is also visible, showing a table with two columns: Destination and Target. The "Routes" table lists two routes: one for destination 0.0.0.0/0 with target igw-041b274ffa1aefd77, and another for destination 10.0.0.0/16 with target local. Both routes are marked as "Active".

Details

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-0263baf2266b73c74	No	subnet-07d4542a238858aa6 / PublicSubnet	-

Routes (2)

Destination	Target	Status	Propagated
0.0.0.0/0	igw-041b274ffa1aefd77	Active	No
10.0.0.0/16	local	Active	No

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.

aws

Services

Search

[Alt+S]

Ireland

YASIN AHMED

VPC dashboard

EC2 Global View

Filter by VPC

Virtual private cloud

Your VPCs

Subnets

Route tables

Internet gateways

Egress-only internet gateways

DHCP option sets

Elastic IPs

Managed prefix lists

Endpoints

Endpoint services

NAT gateways

Peering connections

Security

Network ACLs

Security groups

DNS firewall

Rule groups

Domain lists

Network Firewall

VPC > Route tables > rtb-09648b8617d354397

rtb-09648b8617d354397 / PrivateRouteTable

Actions

Details

Info

Route table ID

rtb-09648b8617d354397

Main

No

Explicit subnet associations

subnet-08abf507fa37ac536 / PrivateSubnet

Edge associations

-

VPC

vpc-03d90c17f038cce3f | KCVPC

Owner ID

107227152750

Routes

Subnet associations

Edge associations

Route propagation

Tags

Routes (1)

Both

Edit routes

Filter routes

< 1 >

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

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.

The screenshot shows the AWS VPC console interface. At the top, a green notification bar states: "NAT gateway nat-08baff7609a8076b0 | KCNatGateway was created successfully." The main content area displays the details for the NAT gateway "nat-08baff7609a8076b0 / KCNatGateway".

Details:

NAT gateway ID nat-08baff7609a8076b0	Connectivity type Public	State Pending	State message -
NAT gateway ARN arn:aws:ec2:eu-west-1:107227152750:natgateway/nat-08baff7609a8076b0	Primary public IPv4 address -	Primary private IPv4 address -	Primary network interface ID -
VPC vpc-03d90c17f038cce3f / KCVPC	Subnet subnet-07d4542a238858aa6 / PublicSubnet	Created Thursday, July 4, 2024 at 00:36:00 PDT	Deleted -

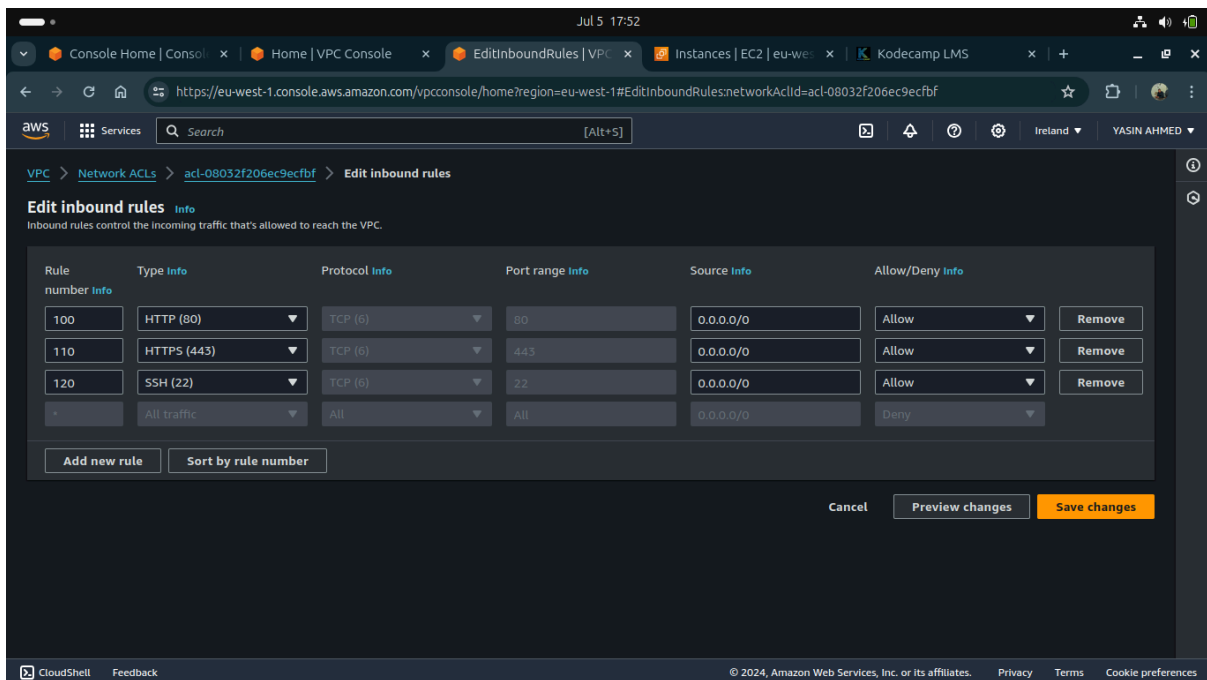
Secondary IPv4 addresses:

Secondary IPv4 addresses are not available for this nat gateway.

6. Set Up Security Groups

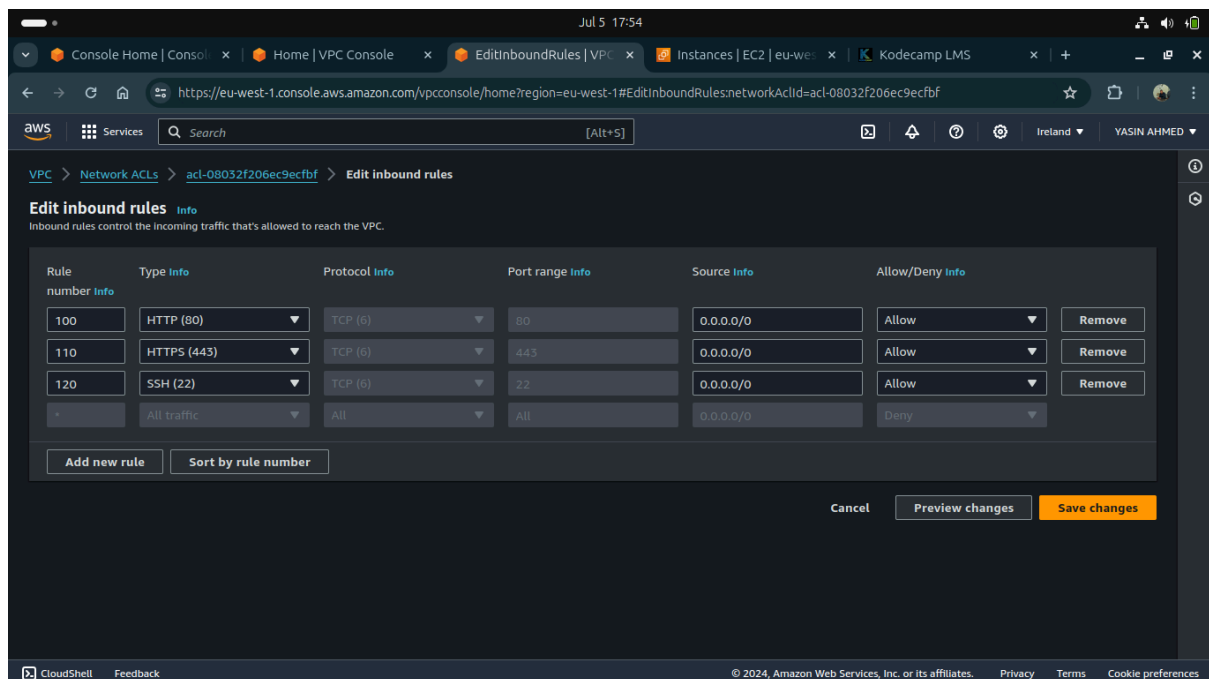
Public Security Group

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.

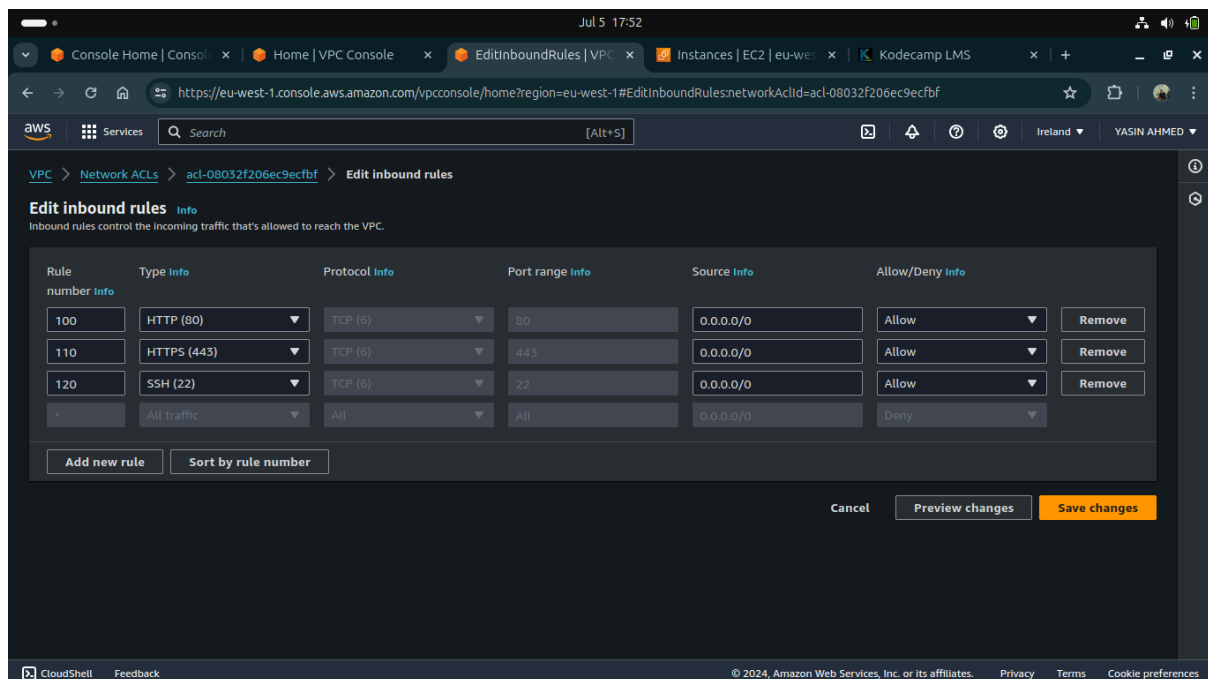
The screenshot shows the AWS Management Console interface for editing inbound rules on a Network ACL. The breadcrumb navigation indicates the path: VPC > Network ACLs > acl-08032f206ec9ecfbf > Edit inbound rules. The main heading is 'Edit inbound rules' with an 'Info' link. Below the heading, a note states: 'Inbound rules control the incoming traffic that's allowed to reach the VPC.'

Rule number	Type	Protocol	Port range	Source	Allow/Deny	Remove
100	HTTP (80)	TCP (6)	80	0.0.0.0/0	Allow	Remove
110	HTTPS (443)	TCP (6)	443	0.0.0.0/0	Allow	Remove
120	SSH (22)	TCP (6)	22	0.0.0.0/0	Allow	Remove
+	All traffic	All	All	0.0.0.0/0	Deny	

At the bottom of the table, there are two buttons: 'Add new rule' and 'Sort by rule number'. Below the table, there are three buttons: 'Cancel', 'Preview changes', and 'Save changes'.

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

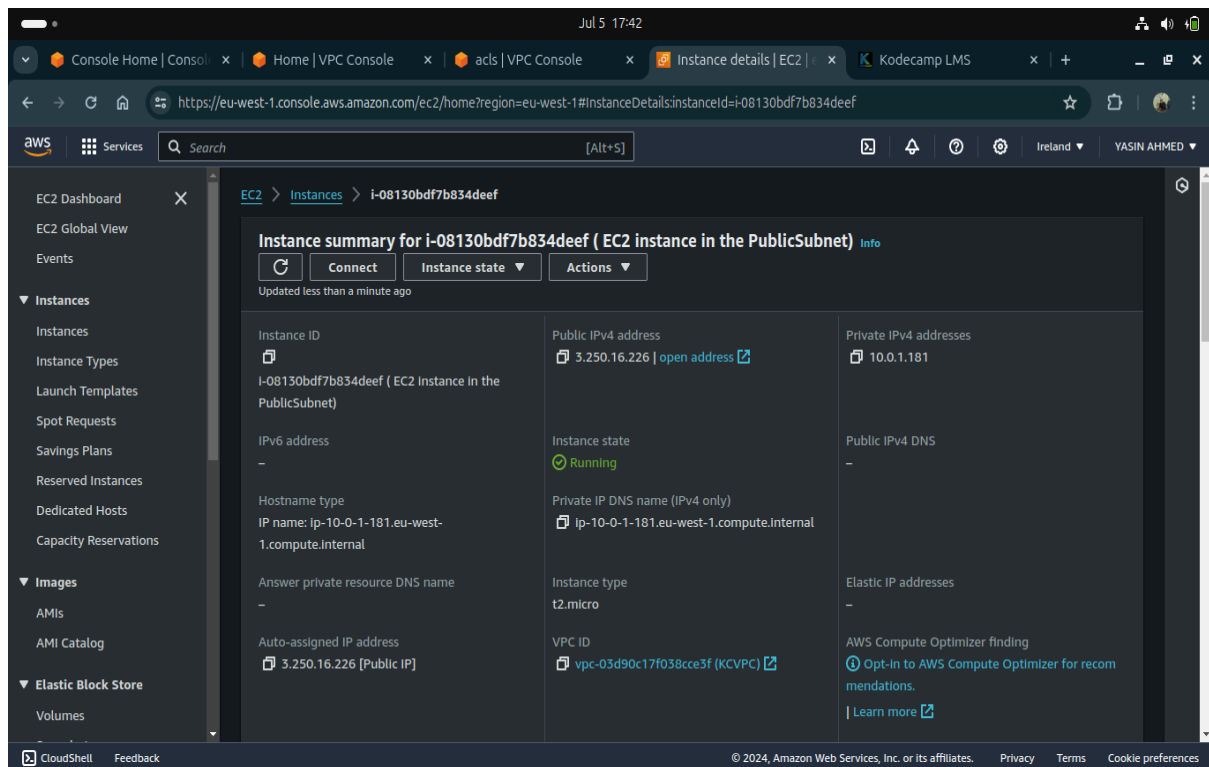
PublicSubnet EC2 Instance

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.

```
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)
```

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.



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.