Project: AWS Virtual Private Cloud

Aryan Vij

Date Started: 4/21/2024

Date Completed: 4/21/2024

Table of Contents

EXECUTIVE SUMMARY

The document outlines a series of procedures for setting up a Virtual Private Cloud (VPC) in Amazon Web Services (AWS) and deploying an EC2 instance within it. Beginning with the creation of the VPC and definition of its IPv4 CIDR block, the process includes configuring subnets across multiple availability zones and establishing an internet gateway for external connectivity. Additionally, it involves setting up a NAT gateway in a public subnet to enable internet access for instances in private subnets, achieved through modifications to routing tables. Following this, the document details the launch of an EC2 instance within the configured VPC, accompanied by the configuration of security groups to regulate SSH access. Secure SSH connectivity to the EC2 instance is facilitated through the use of key pairs.

The outlined procedures culminate in a successful demonstration of the setup, with emphasis placed on network isolation, security, and connectivity. Results highlight the effective deployment of a VPC in AWS, complete with proper routing and secure access controls. The project's success is underscored by the operational efficiency achieved through the deployment of an EC2 instance within the VPC environment. Testing confirms the functionality of internet access from the EC2 instance, facilitated by the configured NAT gateway. Overall, the documented procedures provide a comprehensive guide for establishing and operating a secure and well-connected VPC environment in AWS, ensuring the reliability and functionality of cloud-based infrastructure.

PROCEDURES

Sign in to the AWS Management Console

1. Go to the AWS Management Console and sign in with your credentials.

Navigate to VPC Dashboard

1. Once logged in, navigate to the VPC Dashboard by selecting *"Services"* from the top menu
2. Selecting *"VPC"* under the *"Networking & Content Delivery"* section.

Create VPC

1. Click on the *"Create VPC"* button.
2. Specify a name for your VPC.
3. Define the IPv4 CIDR block for your VPC. This determines the range of IP addresses that can be used within your VPC. For example, 10.0.0.0/16.

Create Subnets

1. In the VPC Dashboard, select *"Subnets"* from the left-hand menu.
2. Click on the *"Create subnet"* button.
3. Specify a name for the subnet.
4. Choose the VPC you created in the previous step.
5. Define the IPv4 CIDR block for the subnet. Ensure it falls within the CIDR block of your VPC.
6. Repeat this step to create additional subnets for different availability zones if needed.

Create Internet Gateway (IGW)

1. In the VPC Dashboard, select *"Internet Gateways"* from the left-hand menu.
2. Click on the *"Create internet gateway"* button.
3. Give the internet gateway a name.
4. Select the newly created internet gateway and attach it to your VPC.

Modify Route Tables

1. In the VPC Dashboard, select *"Route Tables"* from the left-hand menu.
2. Identify the route table associated with your VPC (usually named main).
3. Edit the route table and add a route to the internet gateway (0.0.0.0/0) to enable internet access from your subnets.

Create NAT Gateway

1. In the VPC Dashboard, select *"NAT Gateways"* from the left-hand menu.
2. Click on the *"Create NAT Gateway"* button.
3. Choose the subnet where you want to place the NAT Gateway. This subnet should be a public subnet.
4. Select an Elastic IP address for the NAT Gateway.
5. Create the NAT Gateway.

Modify Route Tables for Private Subnets

1. For each private subnet, modify the associated route table.
2. Add a route to the NAT Gateway (0.0.0.0/0) to enable internet access for instances in the private subnet.

Security Groups and Network ACLs

1. Optionally, configure security groups and network ACLs to control inbound and outbound traffic to your instances.

Review and Test

1. Review the configuration to ensure everything is set up correctly.
2. Test connectivity from instances in your subnets to the internet and other resources within and outside of your VPC.

Navigate to EC2 Dashboard

1. Once logged in, navigate to the EC2 Dashboard by selecting *"Services"* from the top menu
2. Then select *"EC2"* under the *"Compute"* section

Launch EC2 Instance

1. Click on the *"Launch Instance"* button.
2. Choose an Amazon Machine Image (AMI) based on your requirements (e.g., Amazon Linux 2, Ubuntu, etc.).
3. Select an instance type based on your workload needs. Click *"Next: Configure Instance Details"*.

Configure Instance Details

1. Choose the VPC you created earlier from the *"Network"* dropdown menu.
2. Choose the subnet within the VPC where you want to launch the instance.
3. Optionally, configure additional settings such as IAM role, shutdown behavior, etc.
4. Click *"Next: Add Storage"*.

Add Storage

1. Configure the size and type of the root volume (usually EBS).
2. Optionally, add additional volumes if needed.
3. Click *"Next: Add Tags"*.

Add Tags (Optional)

1. Add any tags that are helpful for identifying your instance.
2. Click "*Next: Configure Security Group"*.

Configure Security Group

1. Create a new security group or select an existing one.
2. Configure the inbound rules to allow SSH access (port 22) from your IP address or IP range.
3. Optionally, configure additional rules for other protocols and ports as needed.

4. Click *"Review and Launch"*.

Review and Launch

1. Review the configuration to ensure everything is set up correctly.
2. Click *"Launch"*.

Select Key Pair

1. Choose an existing key pair or create a new one.
2. Download the private key file (.pem) to your local machine.
3. Click *"Launch Instances"*.

SSH into the Instance

1. Once the instance is launched and running, note its public IP address or DNS name from the EC2 Dashboard.
2. Open a terminal or SSH client on your local machine.
3. Change the permissions of the private key file with the command: `chmod 400 /path/to/your-key.pem`.
4. SSH into the instance using the public IP address or DNS name and the private key:
   a. `ssh -i /path/to/your-key.pem ec2-user@<public_ip_address_or_dns_name>`
5. If you're using a different AMI, replace ec2-user with the appropriate username (e.g., ubuntu for Ubuntu AMIs, centos for CentOS AMIs, etc.).

RESULTS

In this project, we successfully created a VPC in AWS and connected it to an EC2 instance. The key steps and outcomes include:

VPC Creation: Defined a VPC with a specified IPv4 CIDR block. Created subnets within the VPC across multiple availability zones. Established an internet gateway and attached it to the VPC for internet access. Configured routing tables to direct traffic within the VPC and to the internet gateway.

NAT Gateway Setup: Deployed a NAT gateway in a public subnet to allow instances in private subnets to access the internet while remaining secure. Modified routing tables for private subnets to route traffic through the NAT gateway.

EC2 Instance Launch: Launched an EC2 instance within the VPC. Configured security groups to allow SSH access (port 22) from specific IP addresses. Established a secure SSH connection to the EC2 instance using a key pair.

Connectivity Testing: Verified connectivity by SSHing into the EC2 instance remotely. Tested internet access from the EC2 instance to ensure proper NAT gateway functionality. Ensured that the EC2 instance was securely accessible and operational within the VPC environment.

Overall, the project successfully demonstrated the setup of a VPC in AWS and the deployment of an EC2 instance within that VPC, showcasing effective network isolation, security, and connectivity for cloud-based infrastructure.

BIBLIOGRAPHY

App Store. (2016, December 5). *Termius - SSH & SFTP client*. Mac App Store. https://apps.apple.com/us/app/termius-ssh-sftp-client/id1176074088?mt=12

Carl Oliver. (2013, September 24). *Subnetting made simple* [Video]. YouTube. https://www.youtube.com/watch?v=nFYilGQ-p-8

CJ Saathoff. (2024, February). *Troubleshooting SSH Authentication: Understanding 'Permission Denied (publickey,gssapi-keyex,gssapi-with-mic).'* NameHero. Retrieved April 21, 2024, from https://www.namehero.com/blog/troubleshooting-ssh-authentication-understanding-permission-denied-publickeygssapi-keyexgssapi-with-mic/

Connect to your Linux instance from Linux or macOS using SSH - Amazon Elastic Compute Cloud. (n.d.-a). https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/connect-linux-inst-ssh.html#ssh-prereqs-linux-from-linux-macos

Connect to your Linux instance from Linux or macOS using SSH - Amazon Elastic Compute Cloud. (n.d.-b). https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/connect-linux-inst-ssh.html#connect-linux-inst-sshClient

How to Enable SSH on a Mac from the Command Line. (2016, August 16). OS X Daily. https://osxdaily.com/2016/08/16/enable-ssh-mac-command-line/

How to fix SSH not working on MacOS Ventura / Sonoma. (2023, November 13). OS X Daily. https://osxdaily.com/2022/12/22/fix-ssh-not-working-macos-rsa-issue/

Restart SSH on Mac Terminal (High Sierra). (n.d.). Gist. https://gist.github.com/influx6/46c39709a67f09908cc7542ca444fca2

Stephane Maarek. (2019, June 11). *Amazon EC2 Basics & Instances Tutorial* [Video]. YouTube. https://www.youtube.com/watch?v=iHX-jtKIVNA

Stephane Maarek. (2021, February 25). *SSH to EC2 Instances using Linux or Mac Tutorial* [Video]. YouTube. https://www.youtube.com/watch?v=8UqtMcX_kg0

Terminate Amazon EC2 instances - Amazon Elastic Compute Cloud. (n.d.). https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/terminating-instances.html

Tony Teaches Tech. (2020, September 6). *How to Use SSH on Your Mac with Terminal* [Video]. YouTube. https://www.youtube.com/watch?v=SfTSBbaFN8Y

Tutorial: Get started with Amazon EC2 Linux instances - Amazon Elastic Compute Cloud. (n.d.). https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html

APPENDICES

Appendix A: Visual Documentation

These visuals offer a detailed representation of the tools used, commands executed, and findings discovered during the assessment. By including screenshots, readers can gain a clear understanding of the methodologies employed and the outcomes achieved during the testing phase.

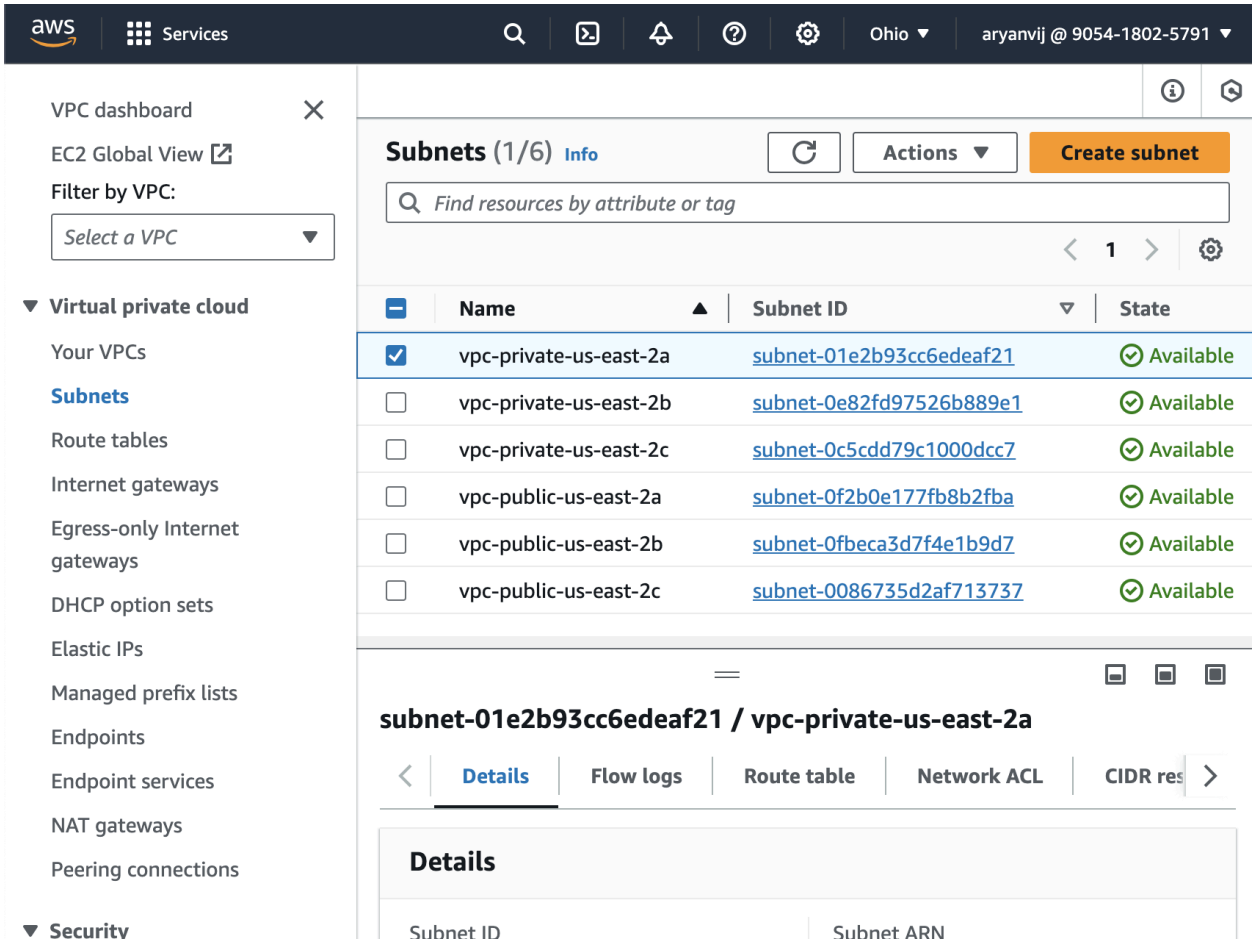

Figure 1: Configuring public and private subnets

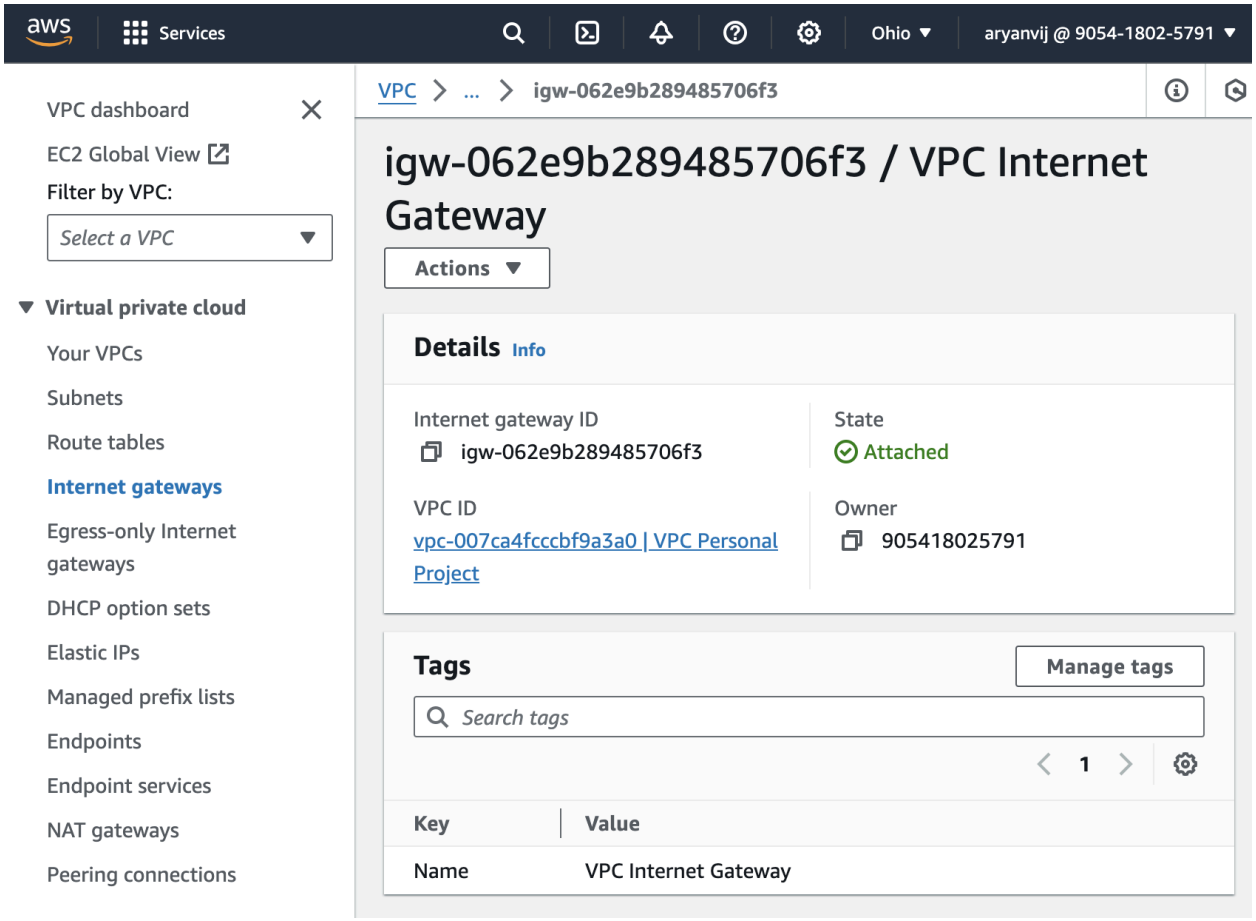

Figure 2: Setting up internet gateway

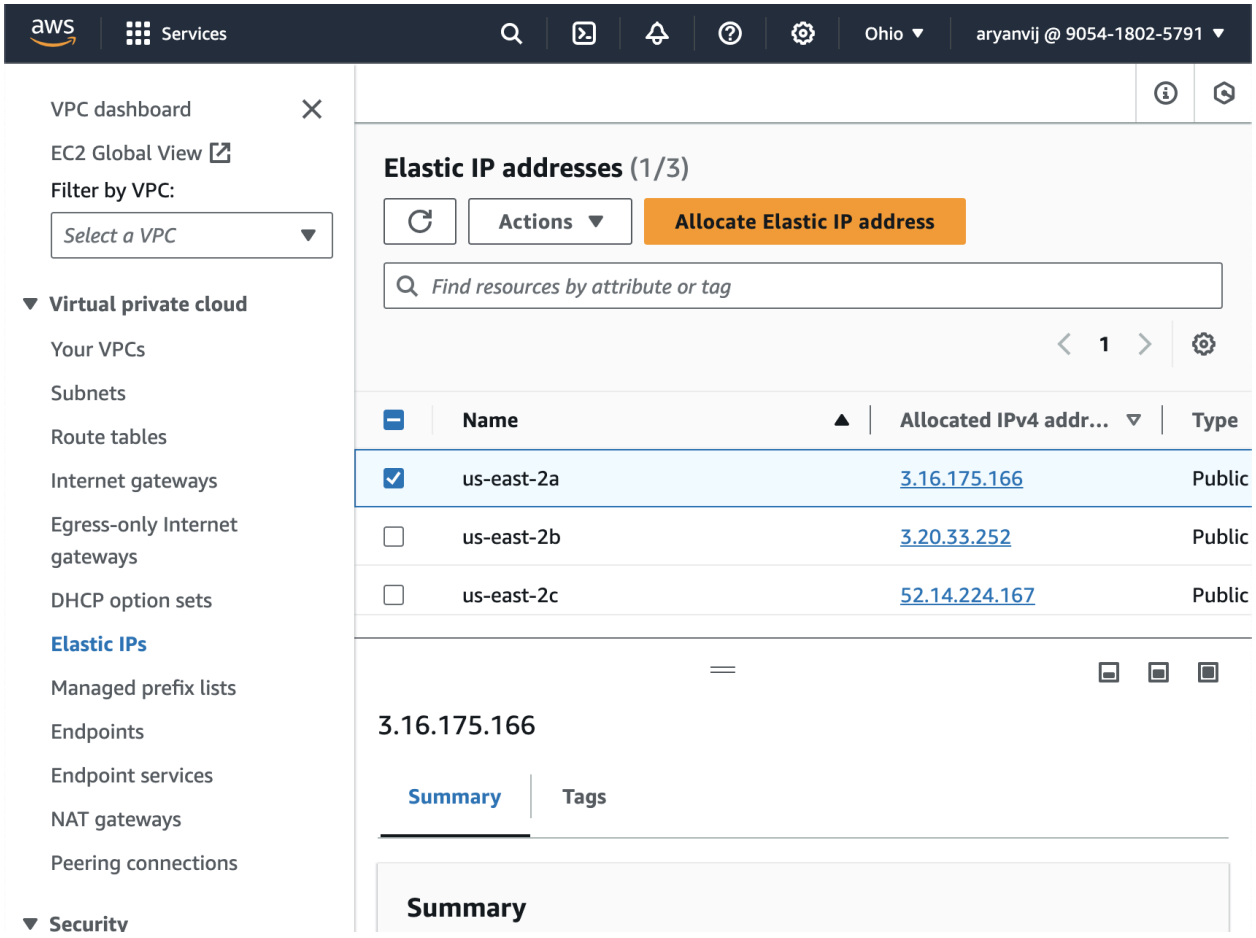

Figure 3: Configuring elastic IPs

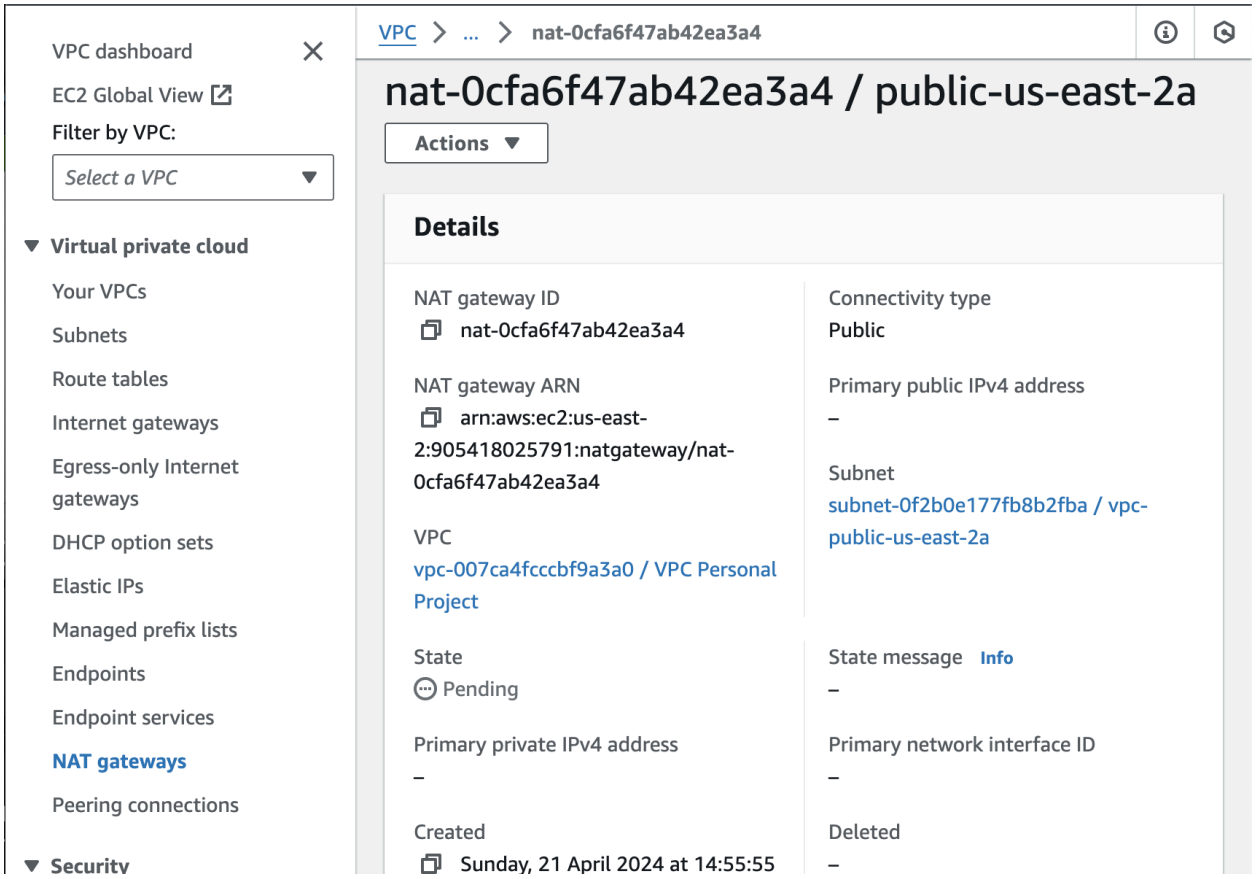

Figure 4: Setting up NAT gateways

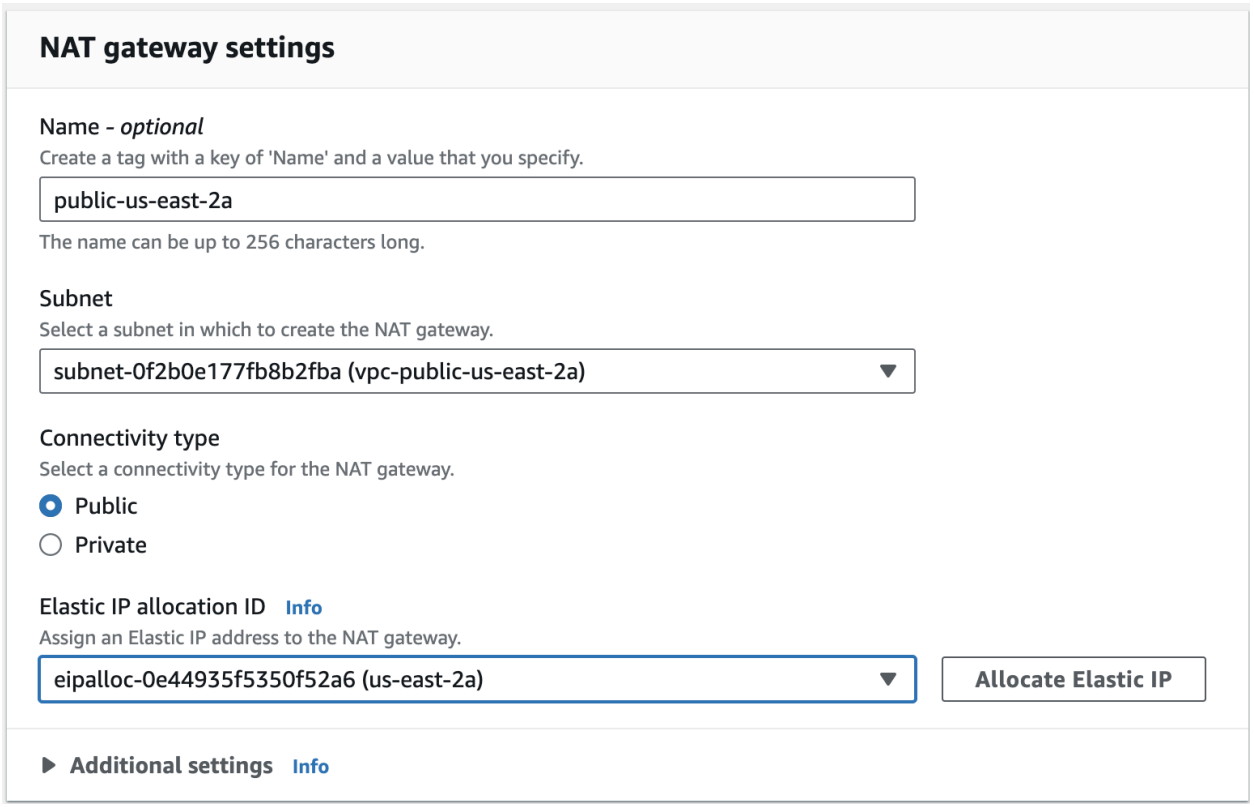

Figure 5: NAT gateway configuration

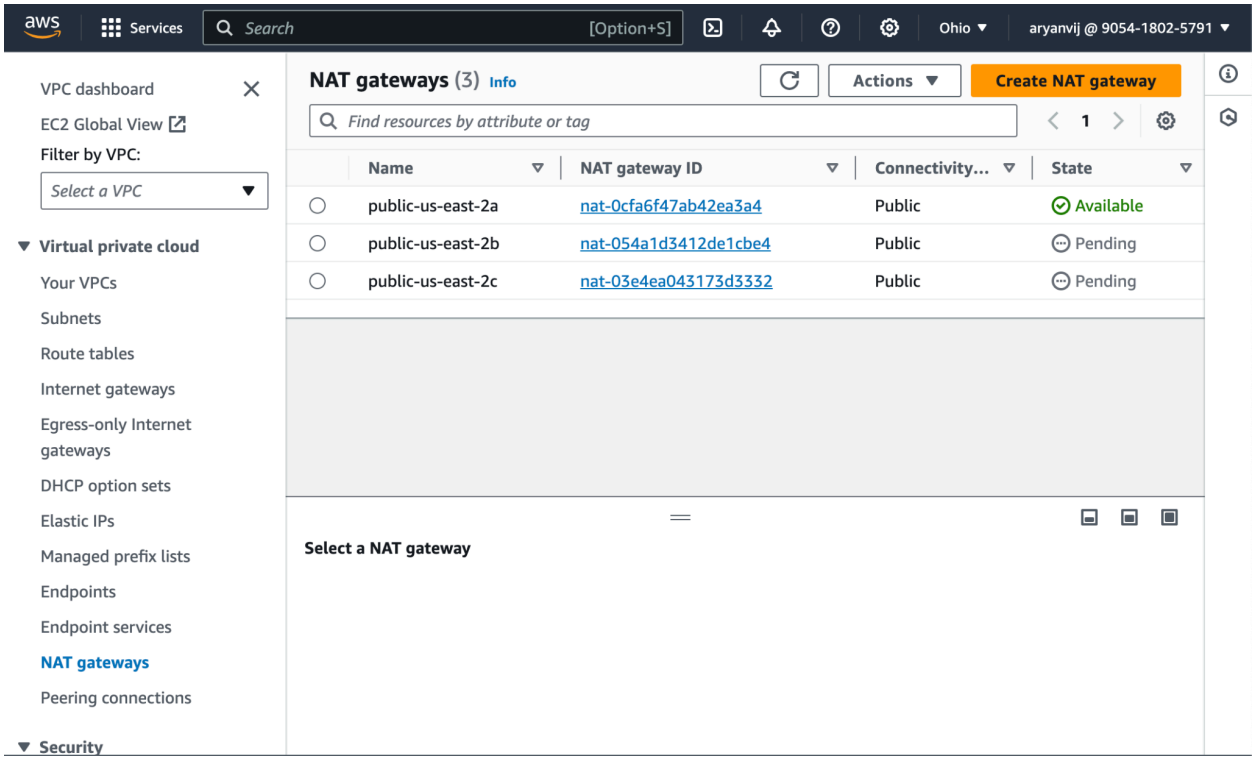

Figure 6: Configured NAT

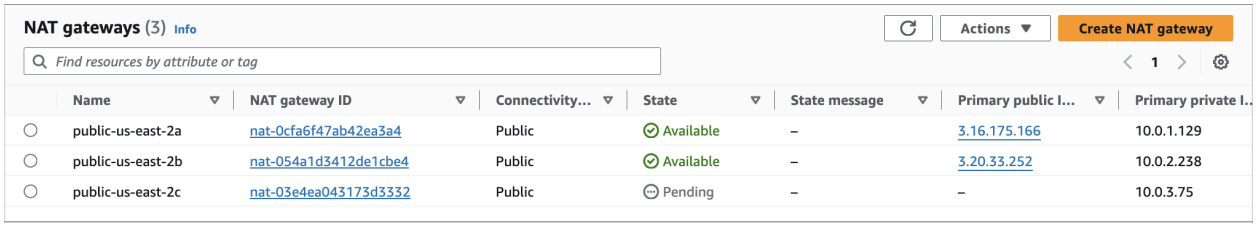

Figure 7: Public and Private IP addresses

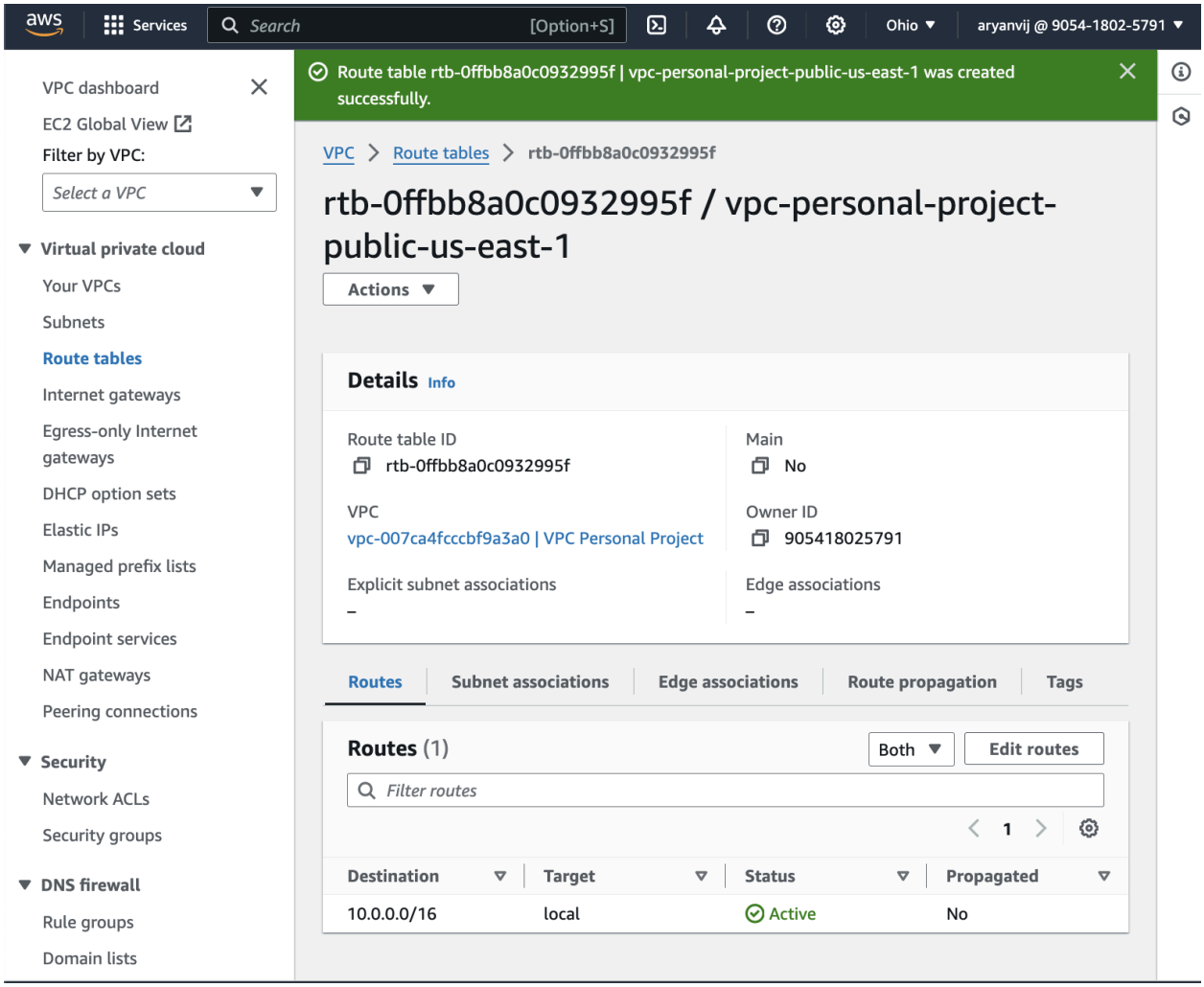

Figure 8: Configured route tables

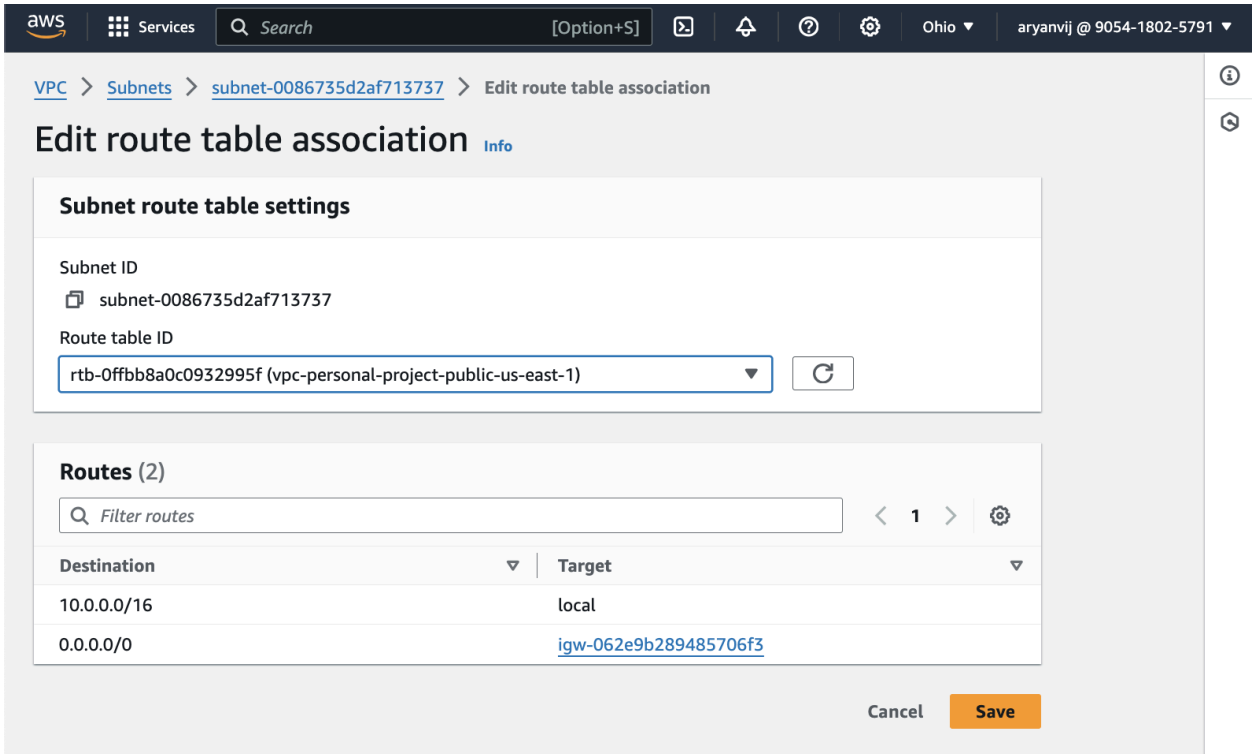

Figure 9: route table association setup

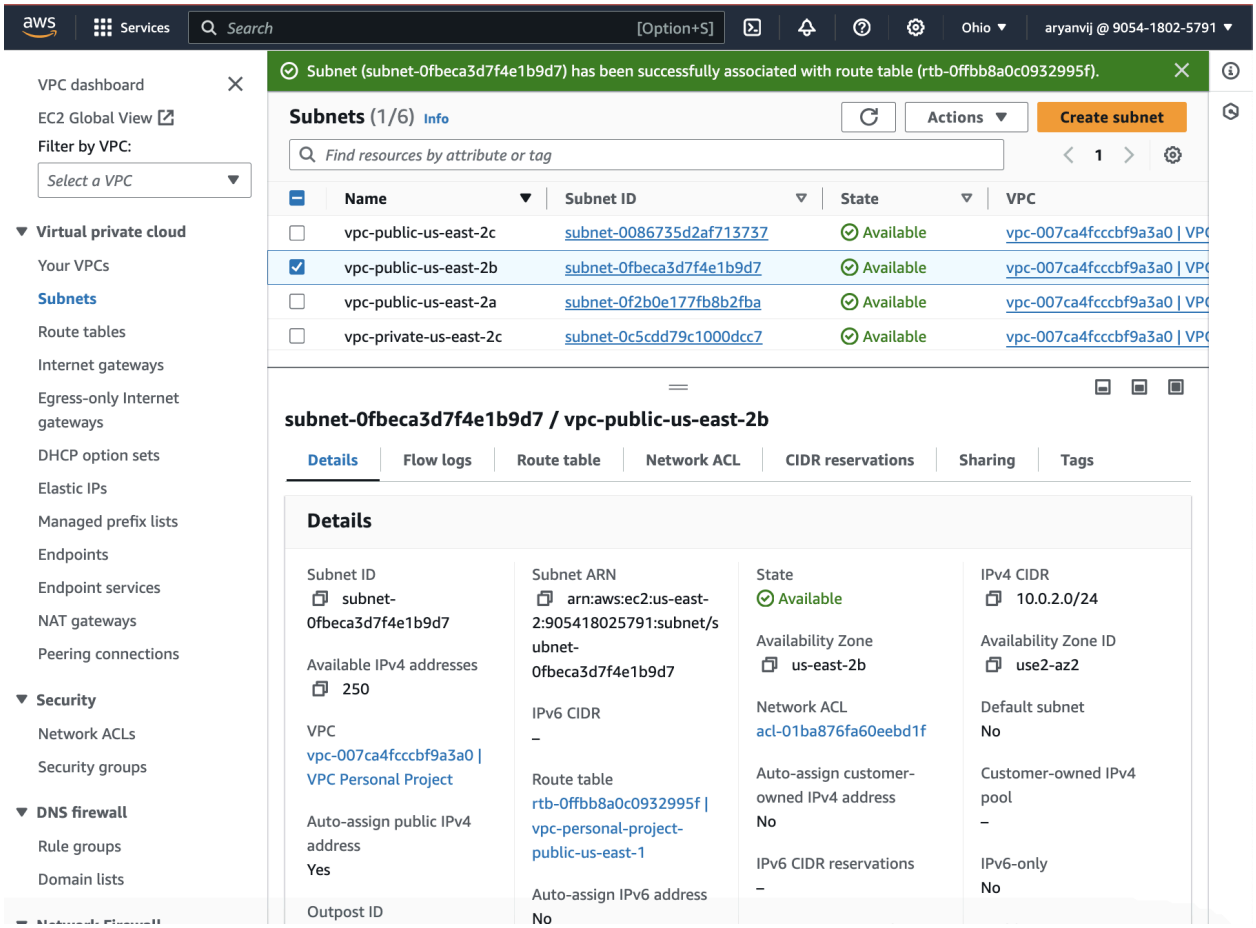

Figure 10: Assigning subnets to route tables

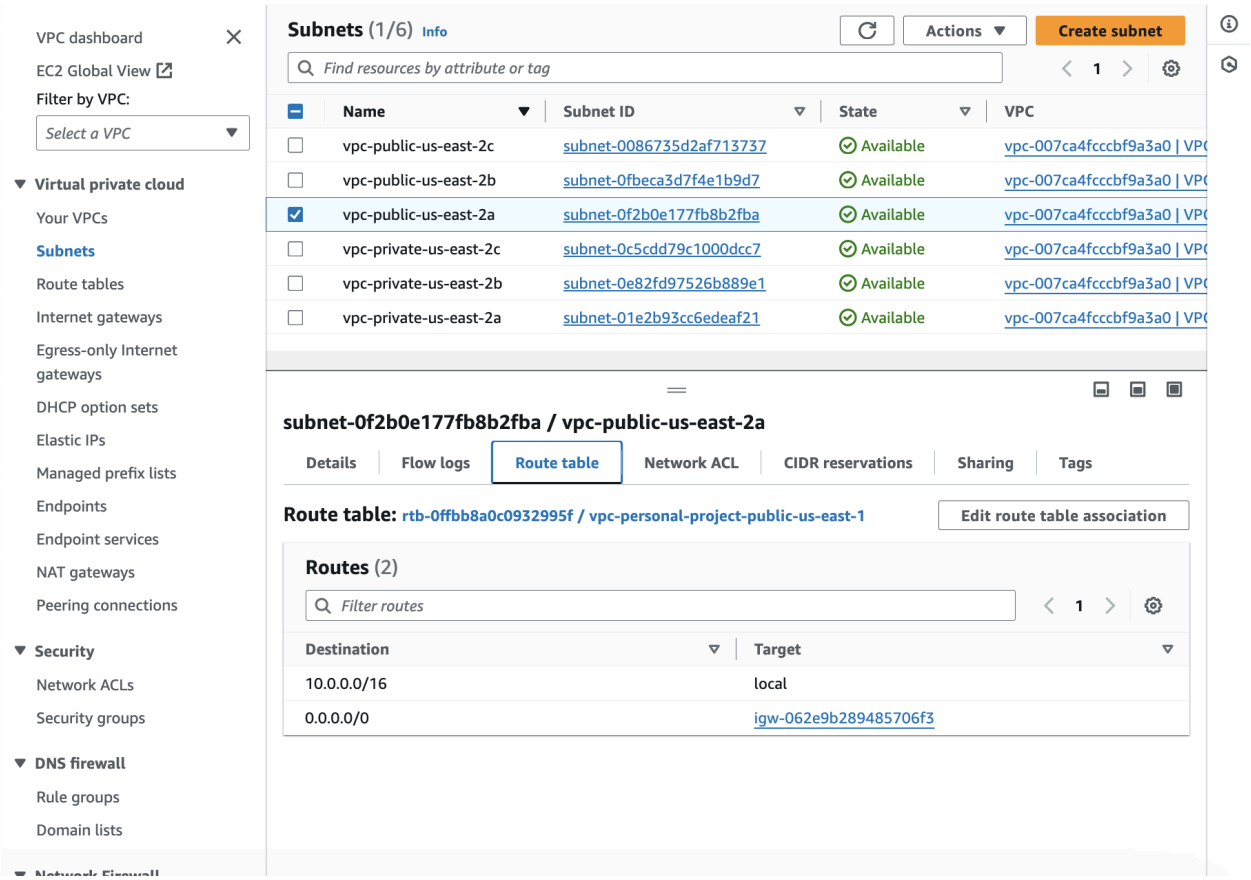

Figure 11: Route tables for subnets

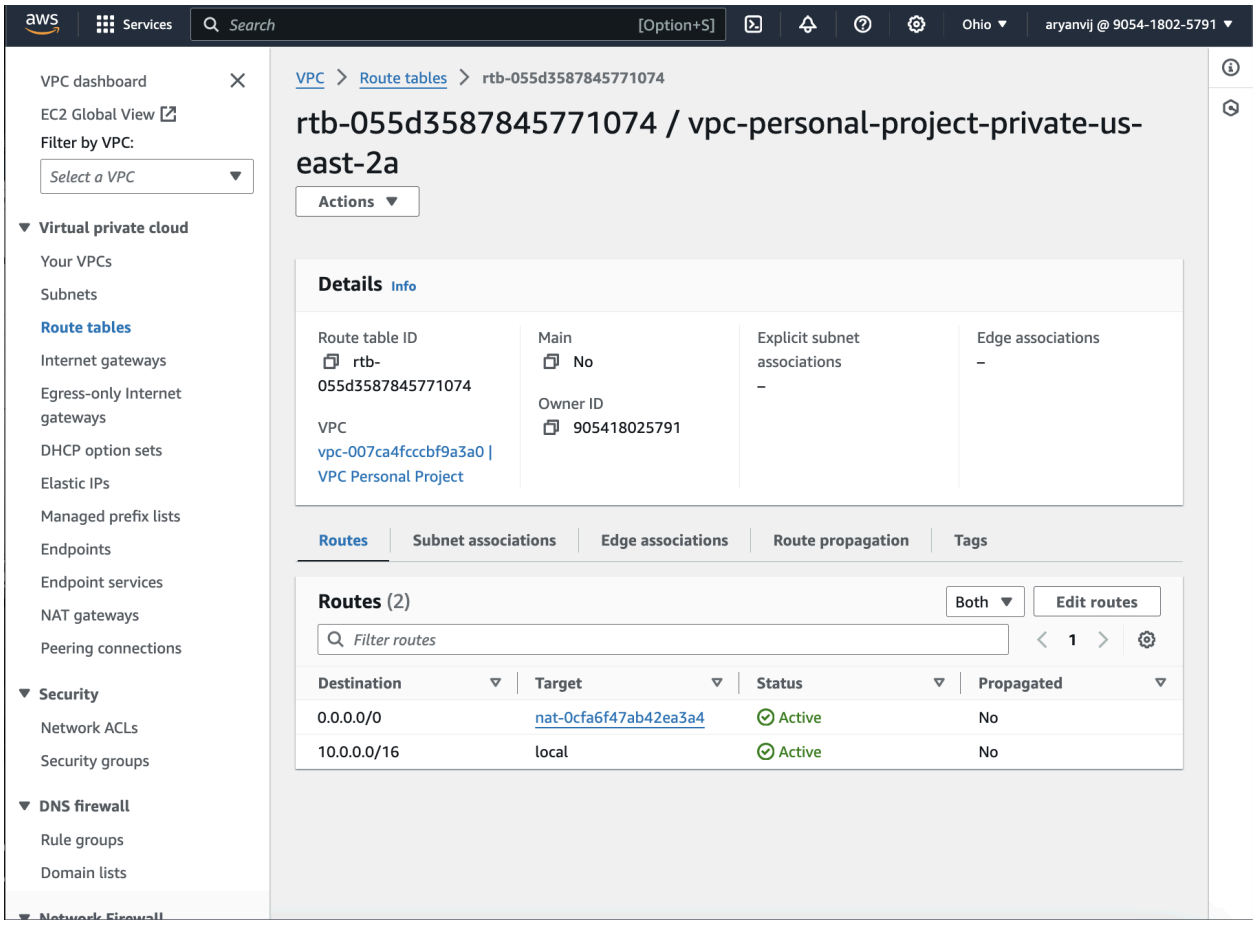

Figure 12: Assigning route tables to subnets

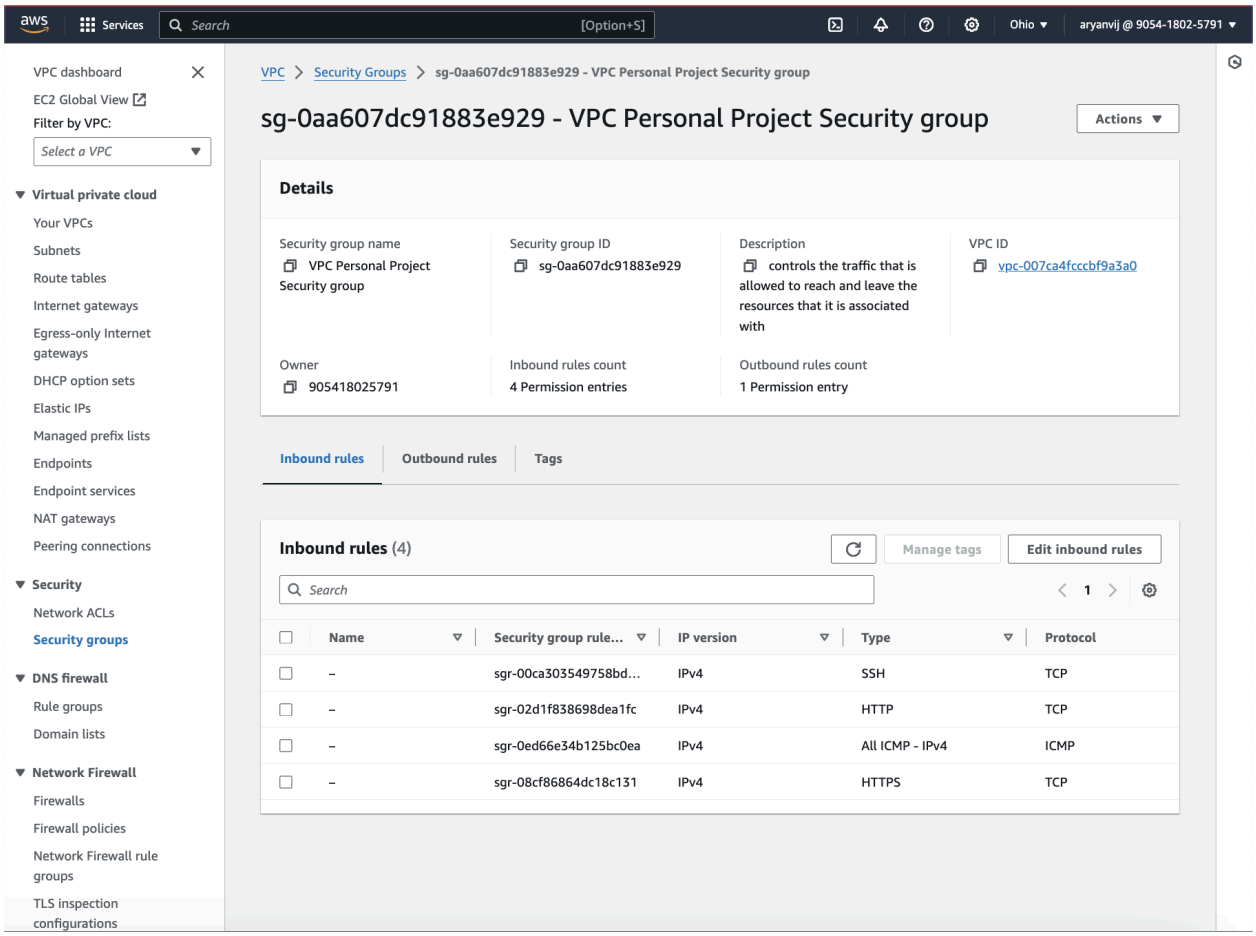

Figure 13: Security group setup

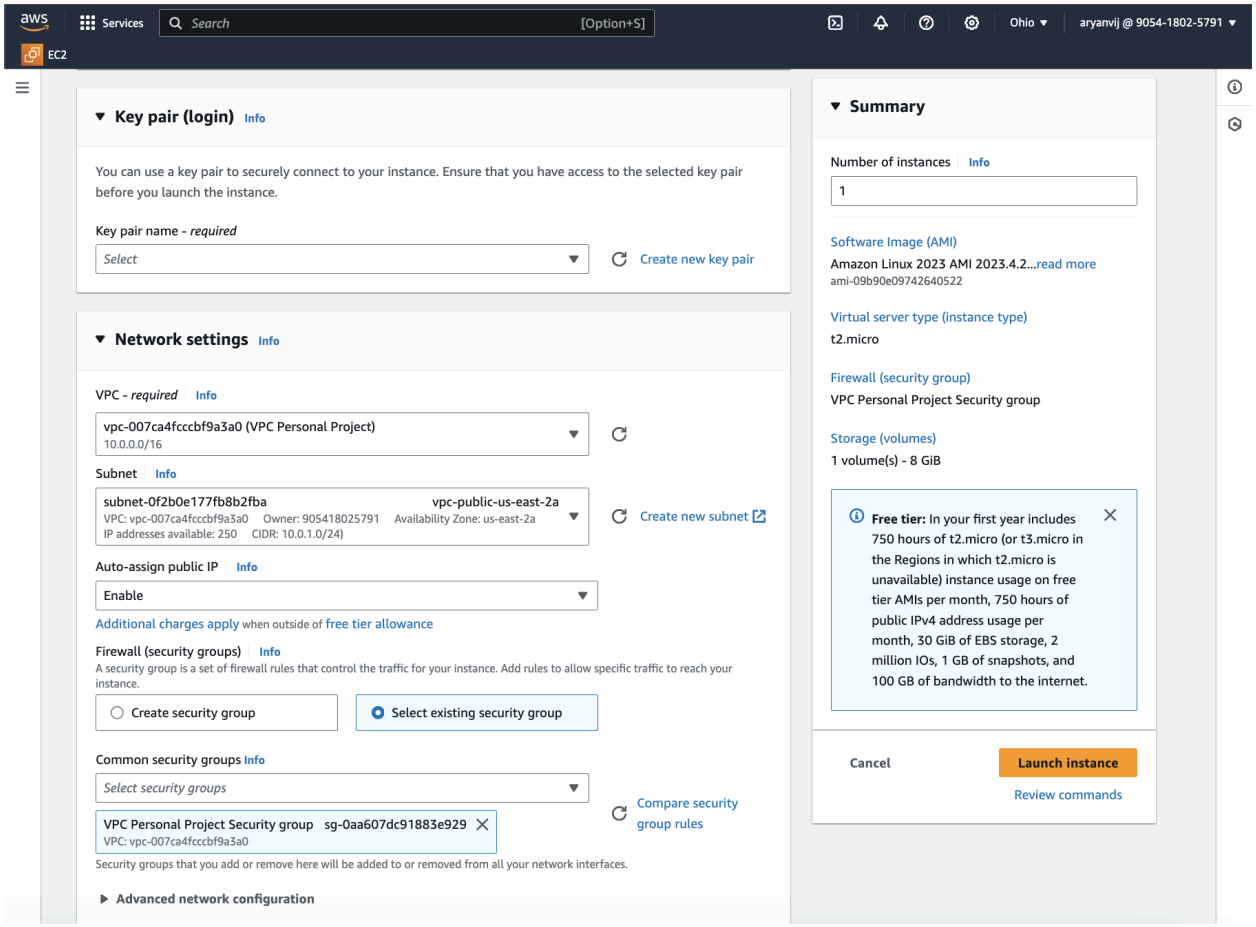

Figure 14: Launching EC2 instance network configuration

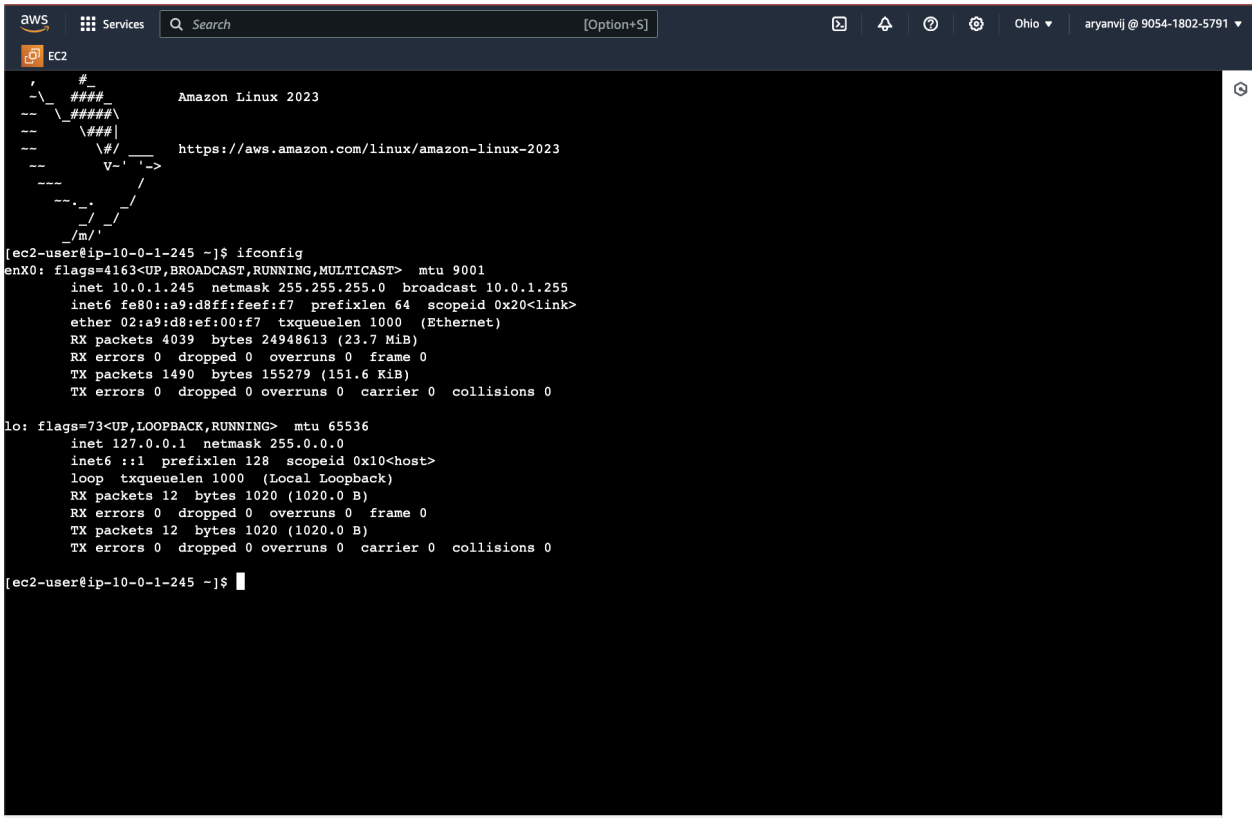

Figure 15: Launching EC2 instance locally

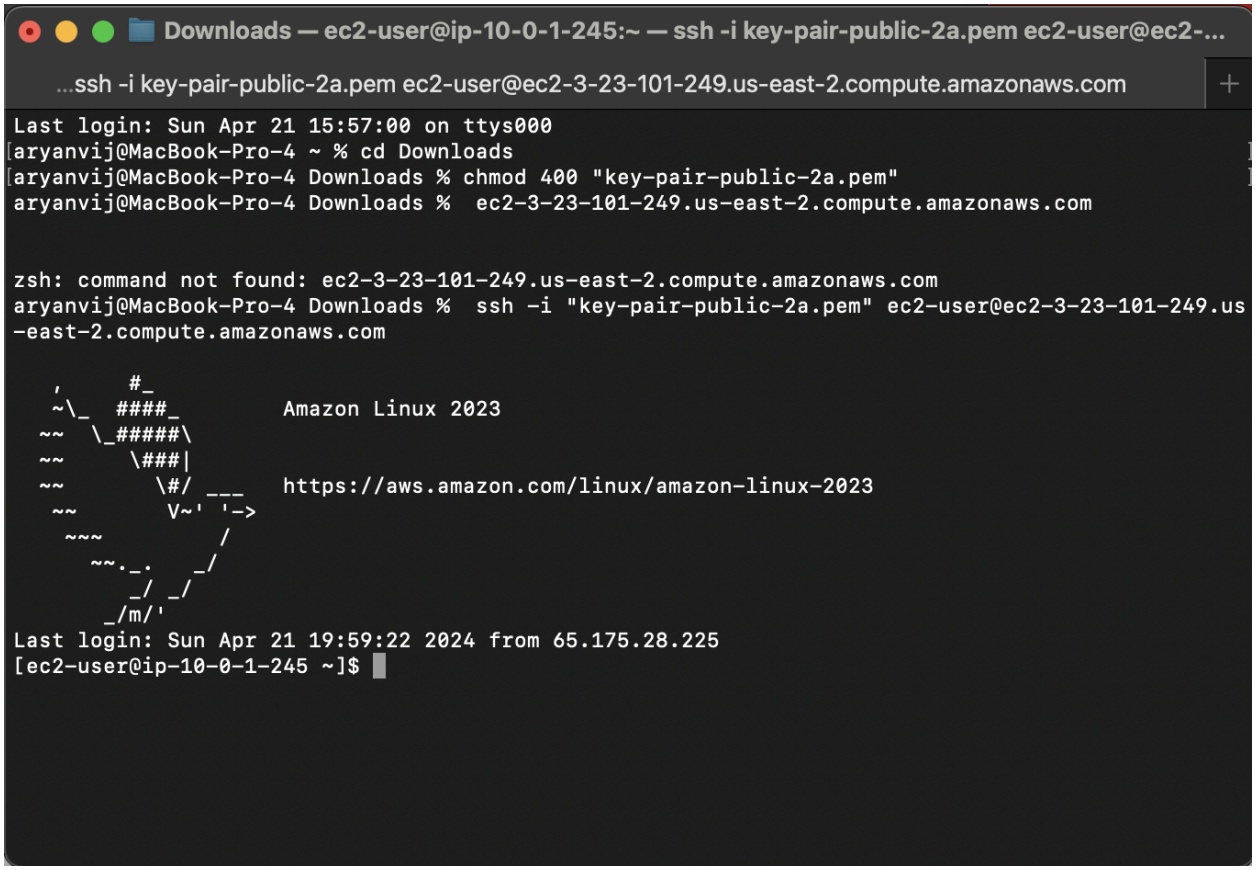

Figure 16: Remote SSH into EC2 instance