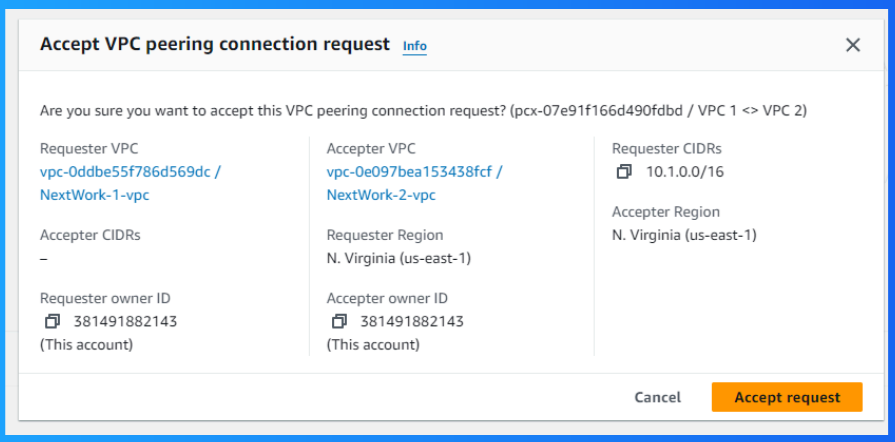
NextWork.org

VPC Peering

Sai Prasanth Reddy



Sai Prasanth Reddy
NextWork Student

Introducing Today's Project!

What is Amazon VPC?

Amazon VPC is a service that allows you to create isolated cloud networks within AWS. It's useful because it provides control over network configurations, enhances security, and allows you to manage resources within a secure, scalable environment.

How I used Amazon VPC in this project

I used Amazon VPC in today's project to create two VPCs, set up public subnets, established a VPC peering connection, configured route tables for cross-VPC communication, and tested connectivity between the VPCs using ping and other tools.

One thing I didn't expect in this project was...

One thing I didn't expect in this project was the need to assign Elastic IP addresses to my instances for remote access, as disabling auto-assign IP prevented me from connecting using EC2 Instance Connect initially.

This project took me...

This project took me little over an hour, including setting up the VPCs, configuring subnets, establishing the peering connection, updating security groups, and testing the connectivity between the instances.

Sai Prasanth Reddy
NextWork Student

In the first part of my project...

Step 1 - Set up my VPC

In this step, we're creating two VPCs from scratch! We'll use the visual VPC resource map to quickly set up the VPCs, defining their IP ranges and subnets, so we can later connect them using VPC peering for communication between both networks.

Step 2 - Create a Peering Connection

In this step, we're setting up a connection link between the two VPCs using VPC peering. This will allow the VPCs to communicate with each other, enabling resource sharing and network traffic flow between them.

Step 3 - Update Route Tables

In this step, we're setting up routing so traffic from VPC 1 can reach VPC 2, and vice versa. We'll update the route tables in both VPCs to ensure traffic flows smoothly between them via the peering connection.

Step 4 - Launch EC2 Instances

Launching an EC2 instance in each VPC, so we can use them to test our VPC peering connection later.

Sai Prasanth Reddy
NextWork Student

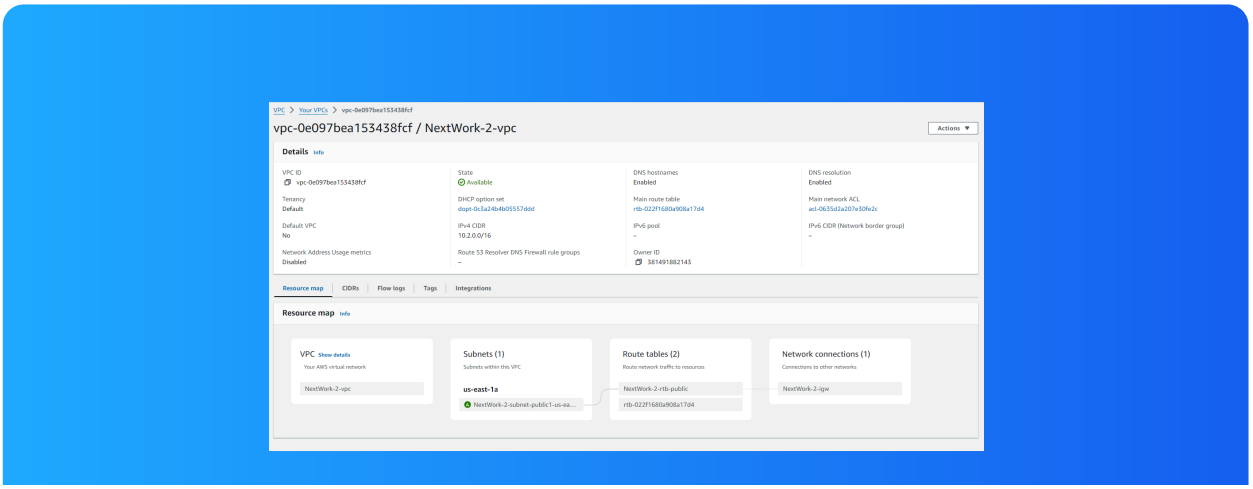
Multi-VPC Architecture

I started my project by launching two VPCs, both with public subnets using the "VPC and more" option. This setup ensures easy and quick configuration of the VPCs, allowing us to proceed with VPC peering and connectivity testing.

The CIDR blocks for VPCs 1 and 2 are unique. They have to be unique because overlapping IP ranges would cause routing conflicts, making it impossible for the two VPCs to communicate with each other through VPC peering.

I also launched 2 EC2 instances

I didn't set up key pairs for these EC2 instances as with EC2 Instance Connect, AWS actually manages a key pair for us! We don't need to manage key pairs ourselves, making the connection process simpler and more secure.



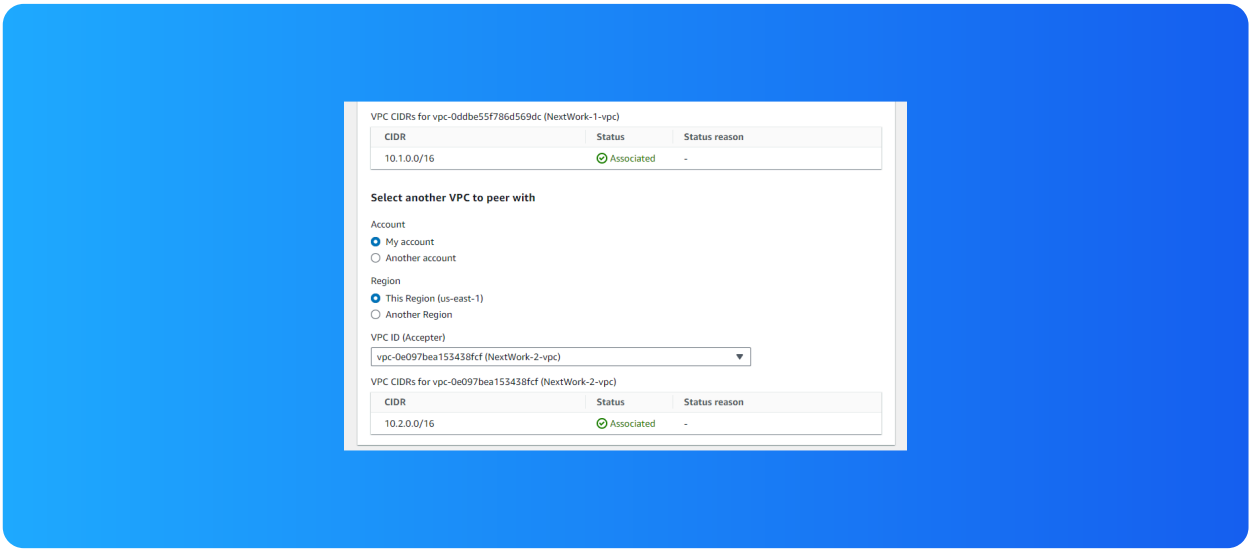
Sai Prasanth Reddy
NextWork Student

VPC Peering

A VPC peering connection is a networking link between two VPCs that allows them to route traffic to each other privately, without using the internet. It enables secure communication between resources in different VPCs.

VPCs would use peering connections to enable secure, private communication between resources in different VPCs without routing traffic through the internet, allowing efficient collaboration and data sharing across isolated networks.

The difference between a Requester and an Accepter in a peering connection is that the Requester initiates the peering request between two VPCs, while the Accepter is the VPC that receives and approves the connection request, establishing the link.

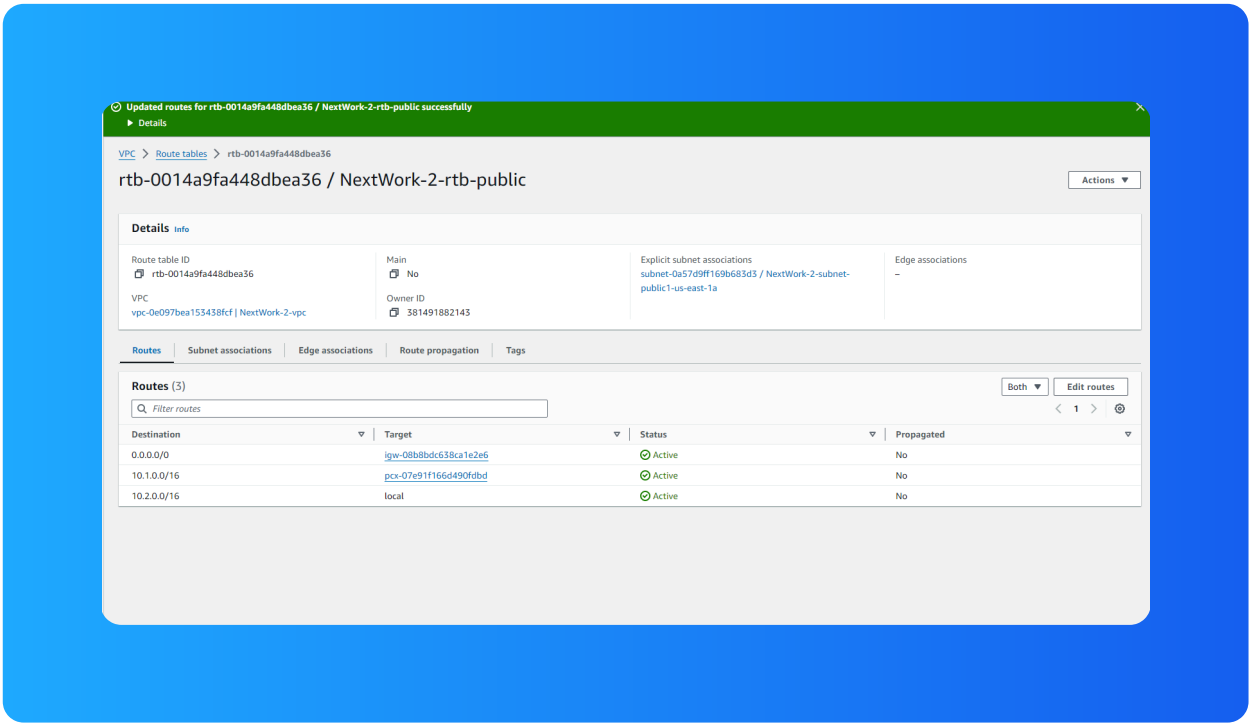


Sai Prasanth Reddy
NextWork Student

Updating route tables

After accepting a peering connection, my VPCs' route tables need to be updated because traffic between the two VPCs requires explicit routes, allowing each VPC to direct traffic through the peering connection to reach the other VPC.

My VPCs' new routes have a destination of the CIDR block of the opposite VPC. The routes' target was the VPC peering connection, enabling traffic between the two VPCs.



Sai Prasanth Reddy
NextWork Student

In the second part of my project...

Step 5 - Use EC2 Instance Connect

Using EC2 Instance Connect to connect to out first EC2 instance and fix a connection error.

Step 6 - Connect to EC2 Instance 1

Using EC2 Instance Connect to connect to Instance 1 and fix another error.

Step 7 - Test VPC Peering

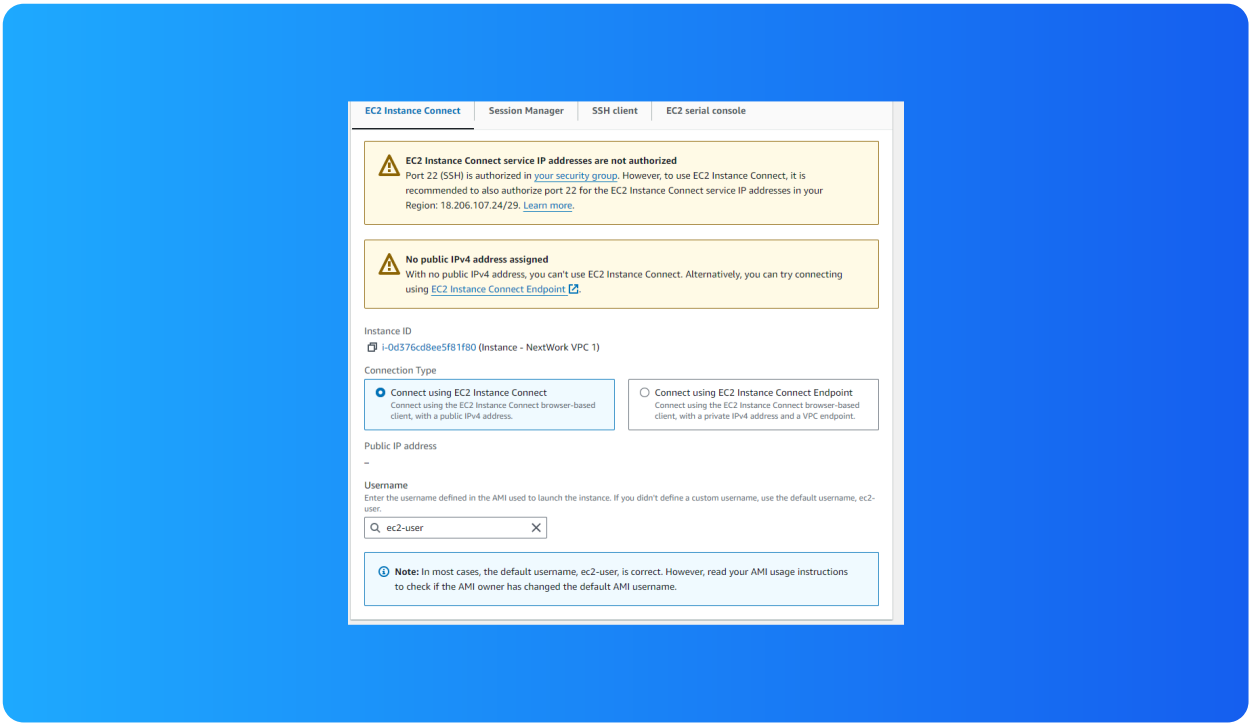
Getting Instance 1 to send test messages to Instance 2 and to solve connection errors until Instance 2 is able to send messages back.

Sai Prasanth Reddy
NextWork Student

Troubleshooting Instance Connect

Next, I used EC2 Instance Connect to securely access my EC2 instances without needing to manage SSH key pairs, allowing me to quickly test and configure my instances through a browser-based connection.

I was stopped from using EC2 Instance Connect because auto-assign IP was disabled, and therefore there was no public IP for the instance, preventing me from connecting to it remotely.

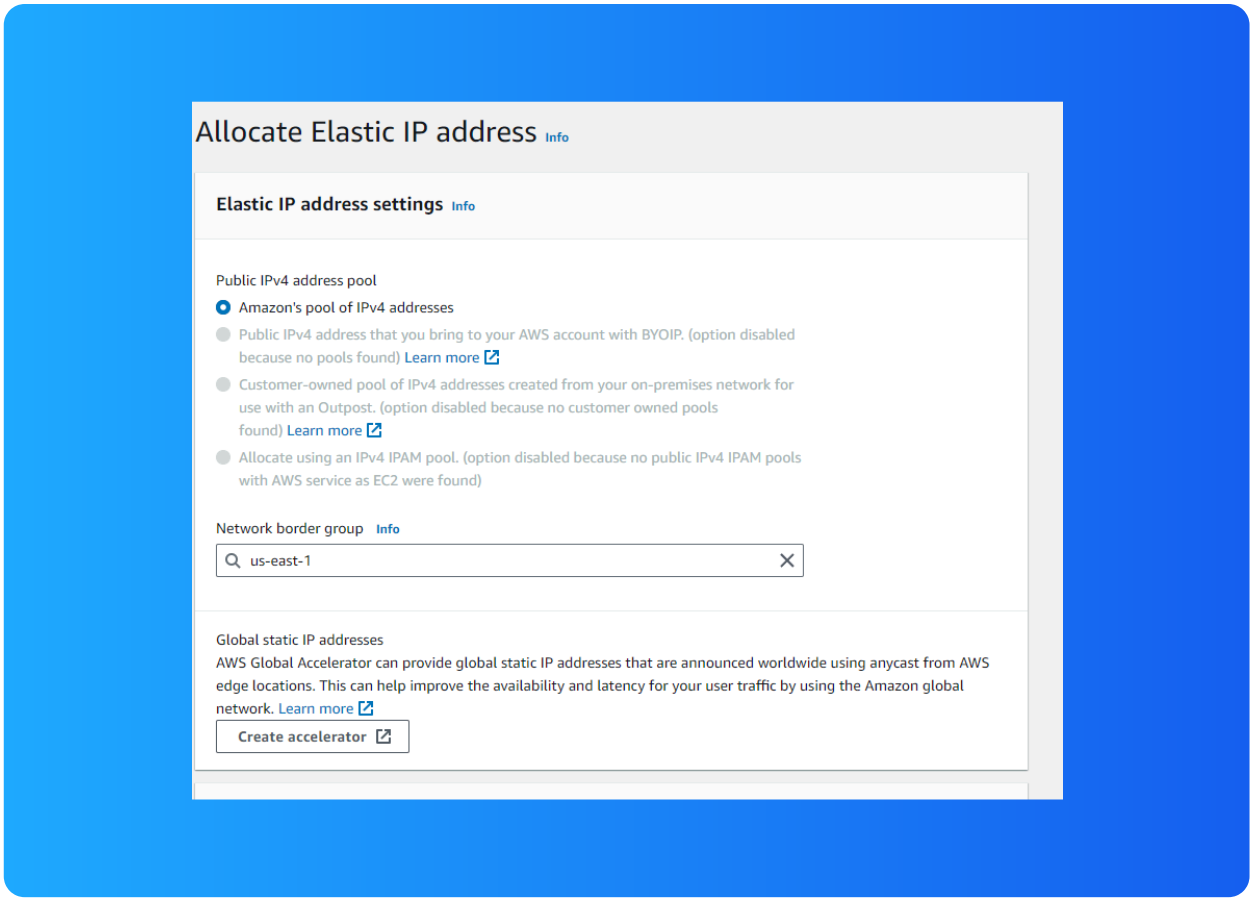


Sai Prasanth Reddy
NextWork Student

Elastic IP addresses

To resolve this error, I set up Elastic IP addresses. Elastic IPs are static, public IPv4 addresses in AWS that give instances a fixed public IP for remote access, ensuring continuity even if the instance is stopped or restarted.

Associating an Elastic IP address resolved the error because it provided the instance with a public IP, enabling remote access through EC2 Instance Connect, which requires a publicly reachable IP address for connectivity.



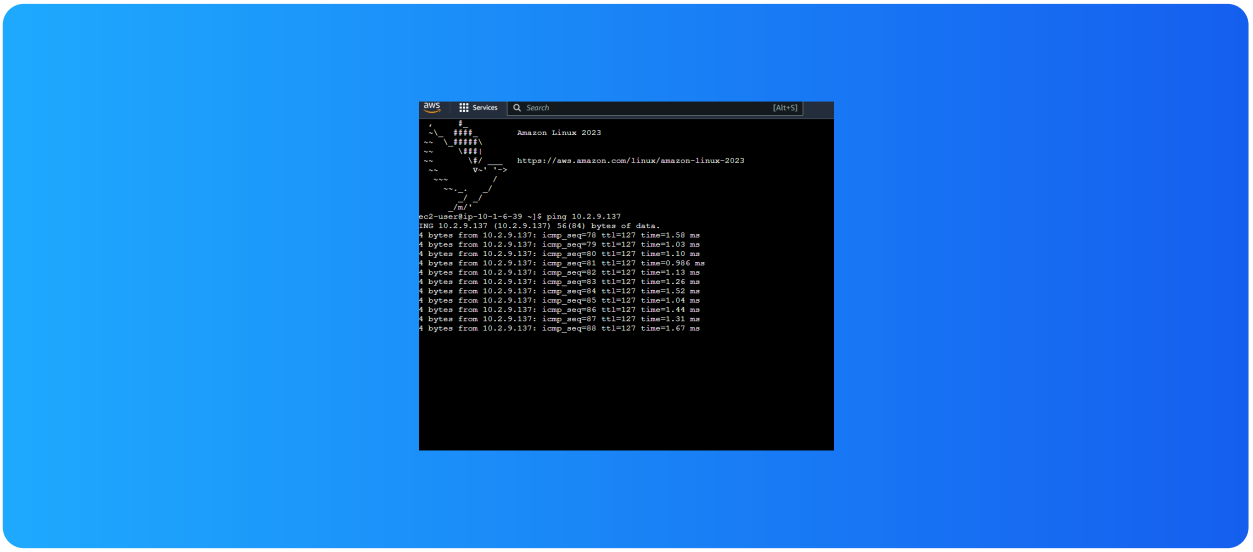
Sai Prasanth Reddy
NextWork Student

Troubleshooting ping issues

To test VPC peering, I ran the command `ping 10.2.9.137`, which checked the connectivity between instances in VPC 1 and VPC 2, confirming that the peering connection was working properly.

A successful ping test would validate my VPC peering connection because it shows that traffic can flow between the two VPCs, confirming that the route tables and peering setup are correctly allowing communication between the instances.

I had to update my second EC2 instance's security group because it wasn't allowing traffic from the first VPC. I added a new rule that permits inbound ICMP (ping) traffic from the private IP range of the first VPC to enable successful communication.



Everyone should be in a job they love.

Check out nextwork.org for more projects