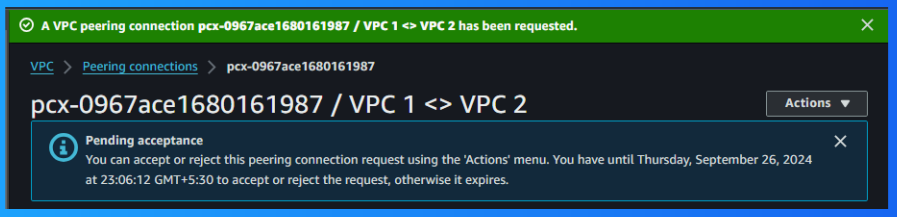
NextWork.org

VPC Peering

sirajudeen athif



S
sirajudeen athif
NextWork Student

Introducing Today's Project!

What is Amazon VPC?

Amazon VPC is a service that allows users to create private networks within the AWS cloud. It provides enhanced security, customizable network configurations and easy scalability.

How I used Amazon VPC in this project

I used Amazon VPC to create two virtual private cloud (VPC) and established a peering connection between them.

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

That it is very simple to use AWS, given you have some practice.

This project took me...

Approximately one and a half hours.

S
sirajudeen athif
NextWork Student

In the first part of my project...

Step 1 - Set up my VPC

We're setting up a VPC! By using the VPC resource map, we can automatically create all the necessary components like subnets, IGs, route tables, etc., making the process much easier and saving us from doing it all manually.

Step 2 - Create a Peering Connection

We're setting up VPC Peering to connect our two VPCs directly. This way, they can communicate securely without relying on the internet, which isn't the safe. By keeping their traffic private, we're making sure our data stays protected and secure.

Step 3 - Update Route Tables

With the peering connection in place, we're ready to create a pathway for traffic to flow between the two VPCs. This will enable smooth communication, allowing data to move freely and securely between them without any issues.

Step 4 - Launch EC2 Instances

In this step, we're launching EC2 instances in each VPC to test the peering connection. This will help us verify that the two VPCs can communicate effectively and ensure everything is working as intended.

S
sirajudeen athif
NextWork Student

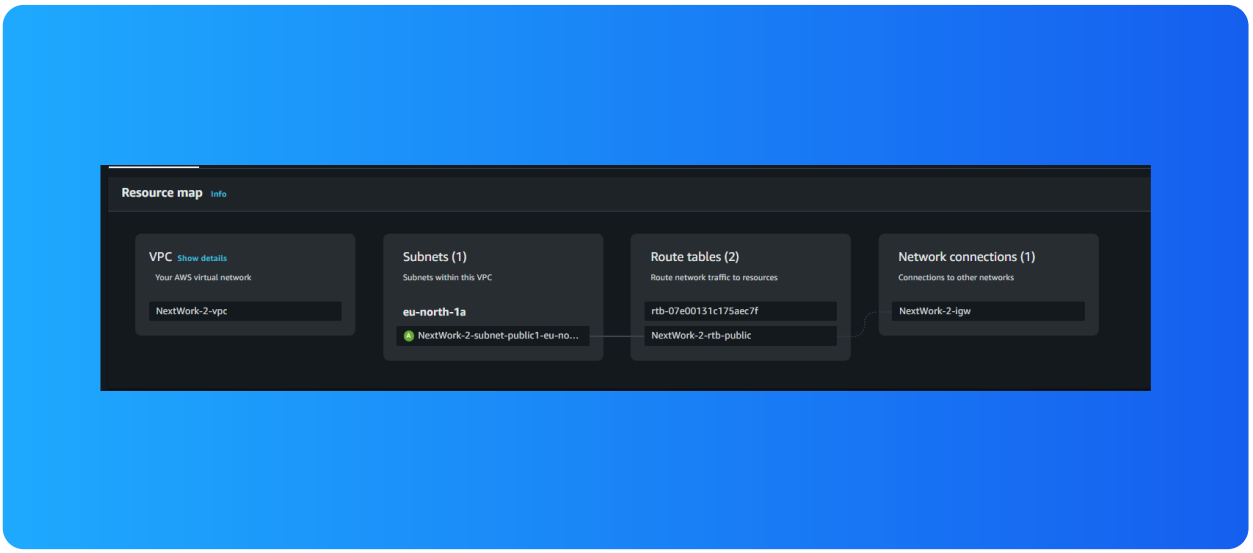
Multi-VPC Architecture

I initiated the project by launching two VPCs: one public and one private. This time, we kept things straightforward by only creating a public subnet in each VPC, simplifying the setup while still maintaining essential connectivity options.

The CIDR blocks for VPCs 1 and 2 are 10.1.0.0/16 and 10.2.0.0/16, respectively. They have to be unique so as to ensure that the IP addresses of their resources do not overlap with each other.

I also launched 2 EC2 instances

I didn't set up key pairs for these EC2 instances as we used EC2 Instance Connect, which will manage a key pair for us. We don't need to manage key pairs ourselves.



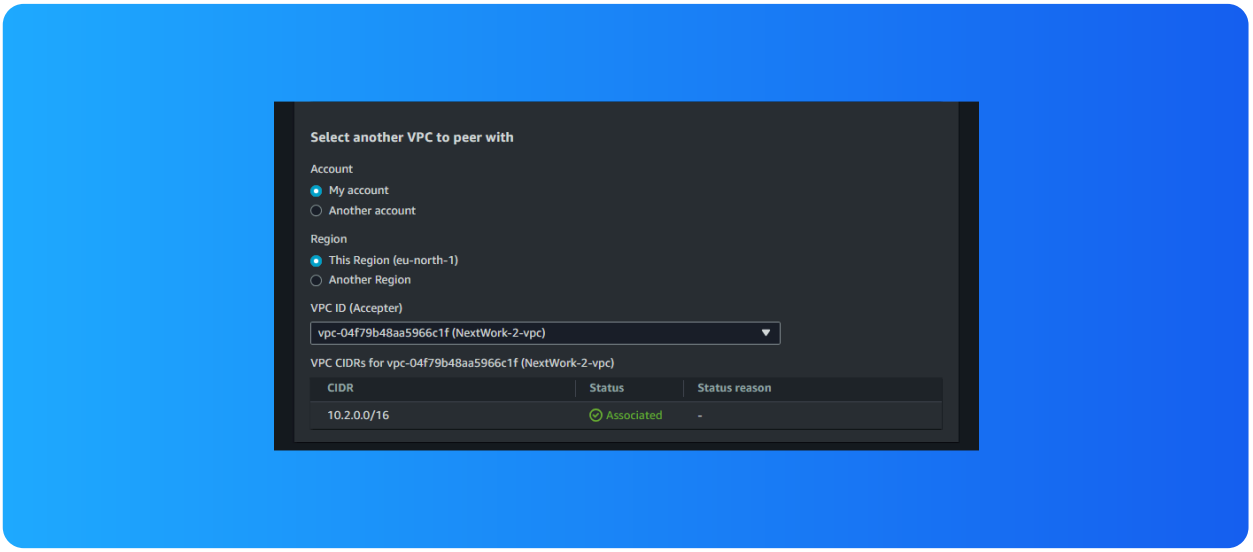
S
sirajudeen athif
NextWork Student

VPC Peering

A VPC peering connection is like a private link between two Virtual Private Clouds (VPCs). It lets them communicate directly using private IP addresses, ensuring secure & efficient traffic flow. This way, they can share resources without the internet

VPCs use peering connections to communicate securely & directly with each other, making it easy to share resources & access data This setup allows applications to work together without going through the internet, boosting both security & performance

In VPC Peering, the Requester is the VPC that initiates a peering connection. As the requester, they will be sending the other VPC an invitation to connect. The Accepter is the VPC that receives a peering connection request.

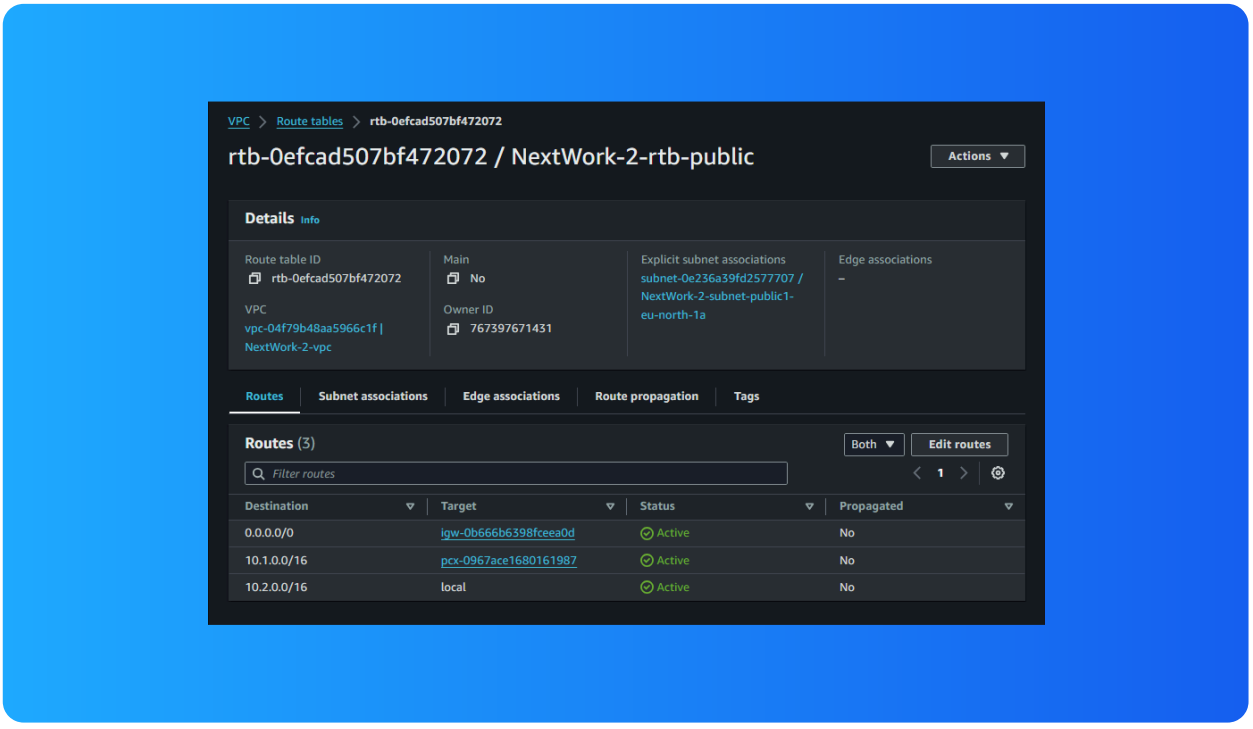


S
sirajudeen athif
NextWork Student

Updating route tables

After accepting a peering connection, my VPCs' route tables need to be updated because even if your peering connection has been accepted, traffic in one VPC won't know how to get to resources in the other VPC without a route in the route table.

The new routes for my VPCs are set to destinations 10.1.0.0/16 and 10.2.0.0/16, with the target being the established peering connection. This setup ensures that traffic can flow smoothly between the two networks.



S
sirajudeen athif
NextWork Student

In the second part of my project...

Step 5 - Use EC2 Instance Connect

We are trying to use EC2 Instance Connect to test out the VPC peering connection. In other words, we will need to get one of our EC2 instances to try to talk with the other.

Step 6 - Connect to EC2 Instance 1

Here, we will be connecting one EC2 instance with the other (after resolving all the errors) using EC2 Instance Connect.

Step 7 - Test VPC Peering

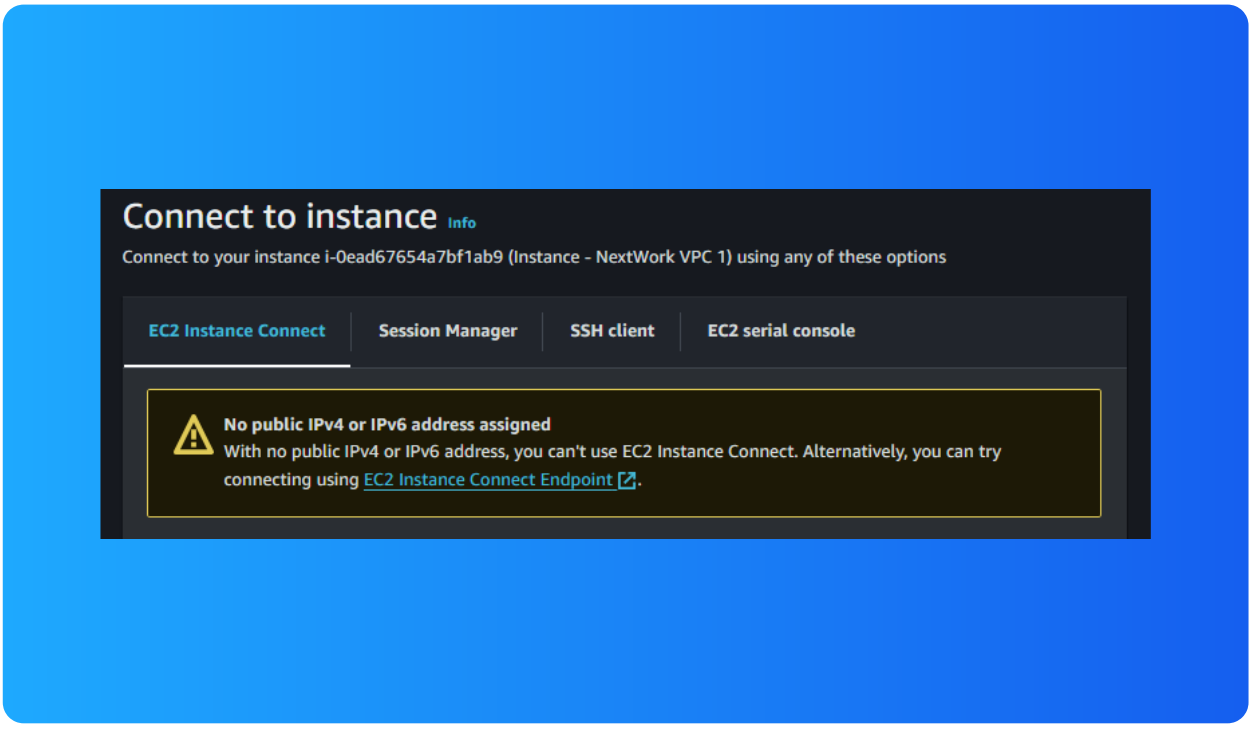
After resolving all the connection errors, we have finally established a connection from one instance to another using VPC peering.

S
sirajudeen athif
NextWork Student

Troubleshooting Instance Connect

Next, I used EC2 Instance Connect to link the two VPCs. This makes things much easier since it handles key pair management for us, so we don't have to worry about creating and managing them manually. It really streamlines the process.

I was stopped from using EC2 Instance Connect as by default, If you want to connect to your instance over EC2 Instance Connect, then your instance must have a public IP address and be in a public subnet.

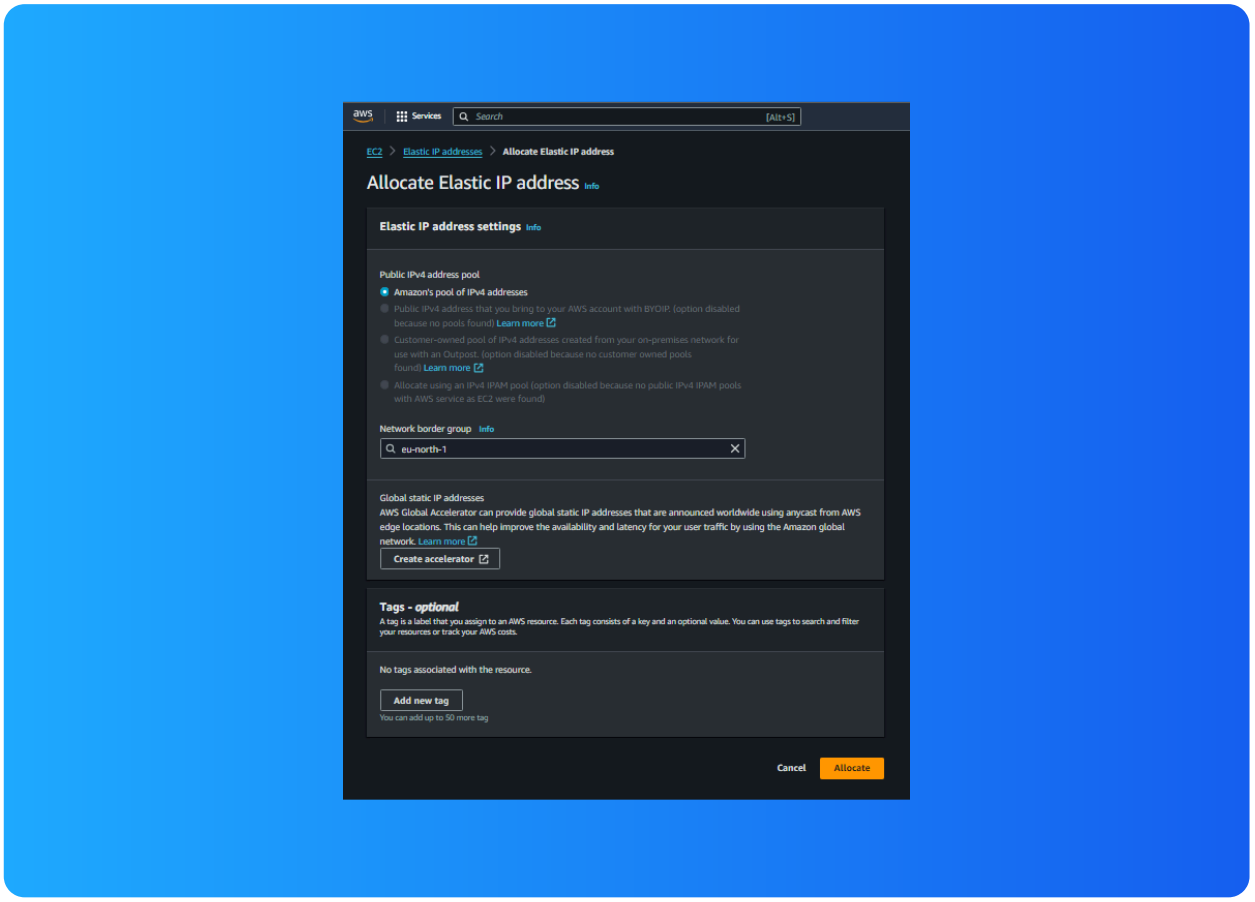


S
sirajudeen athif
NextWork Student

Elastic IP addresses

To resolve this error, I set up Elastic IP addresses. Elastic IP addresses are static IPv4 addresses that get allocated to your AWS account, and is yours to delegate to an EC2 instance

Associating an Elastic IP address resolved the error because it provided the necessary public IP address for the subnet.



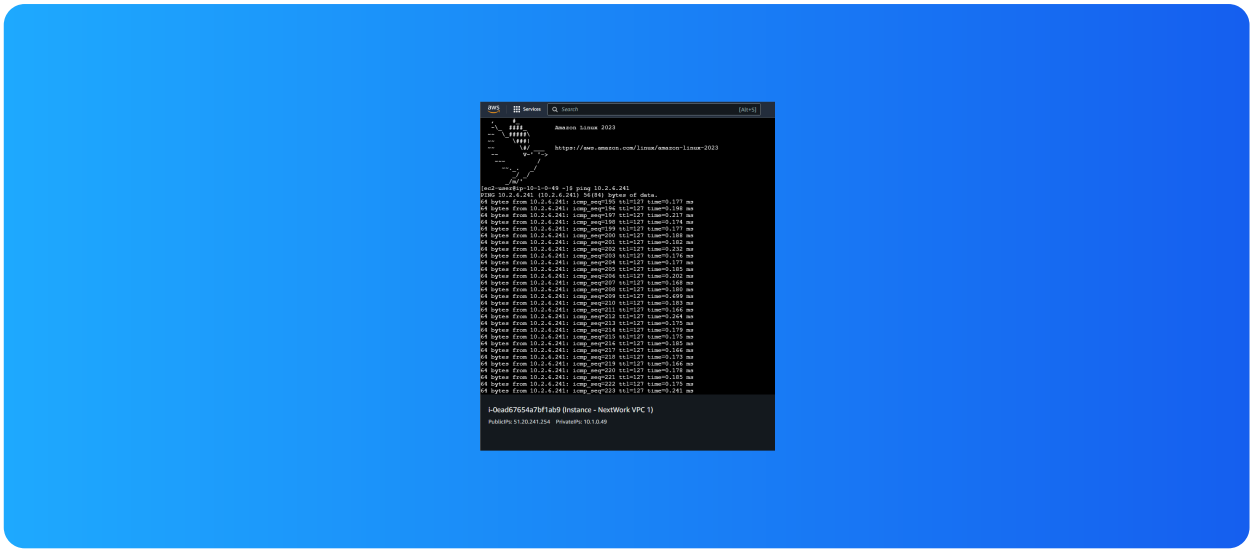
S
sirajudeen athif
NextWork Student

Troubleshooting ping issues

To test VPC peering, I ran the command 'ping'. Ping is a common computer network tool used to check whether your computer can communicate with another computer or device on a network.

A successful ping test would validate my VPC peering connection because Ping will tell you whether you get a response back and how long it took to get a response from the other end.

I had to update my second EC2 instance's security group because the associated security group was blocking the type of messages used in ping, which is known as ICMP. I added a new rule that allowed all types of ICMP messages for IPv4 addresses.



Everyone should be in a job they love.

Check out nextwork.org for more projects