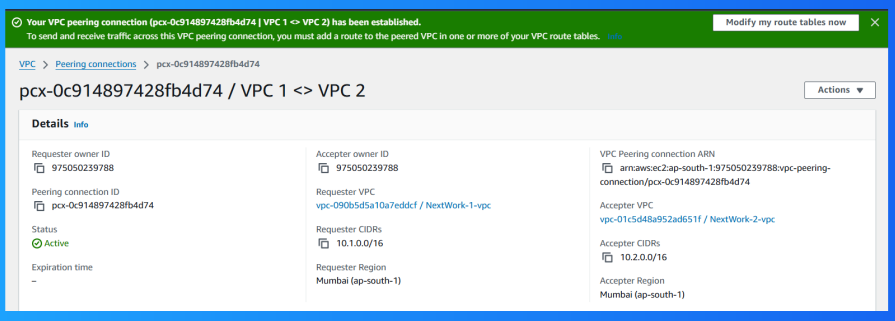
NextWork.org

VPC Peering

SH shravanirg03@gmail.com



SH
shravanirg03@gmail.com
NextWork Student
NextWork.org

Introducing Today's Project!

What is Amazon VPC?

Amazon VPC is a core networking service in AWS that allows us to build secure and scalable network architectures to host and manage resources effectively within the cloud.

How I used Amazon VPC in this project

In today's project, I used Amazon VPC to implement VPC Peering by setting up a multi-VPC architecture and establishing a peering connection between them. I verified the connection using EC2 Instance Connect and updating security group rules.

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

I did not expect that Elastic IPs can assign static IPv4 addresses to resources. I also did not expect that an EC2 instance can have multiple private IPs.

This project took me...

This project took me 2 hours to complete including the time to write documentation.

SH
shravanirg03@gmail.com
NextWork Student
NextWork.org

In the first part of my project...

Step 1 - Set up my VPC

To streamline the VPC creation process, I'm utilizing the VPC resource map/launch wizard to quickly deploy two VPCs with their respective components.

Step 2 - Create a Peering Connection

Setting up a VPC Peering Connection, which is a VPC component designed to direclty connect two VPCs together.

Step 3 - Update Route Tables

In this step, I am going to update the route tables so that the traffic coming from both VPCs can navigate easily.

Step 4 - Launch EC2 Instances

In this step, we are launching EC2 instances in the subnets of each VPC that we set up. These EC2 instances will be used to test the connectivity of the VPC Peering Connection.

SH
shravanirg03@gmail.com
NextWork Student
NextWork.org

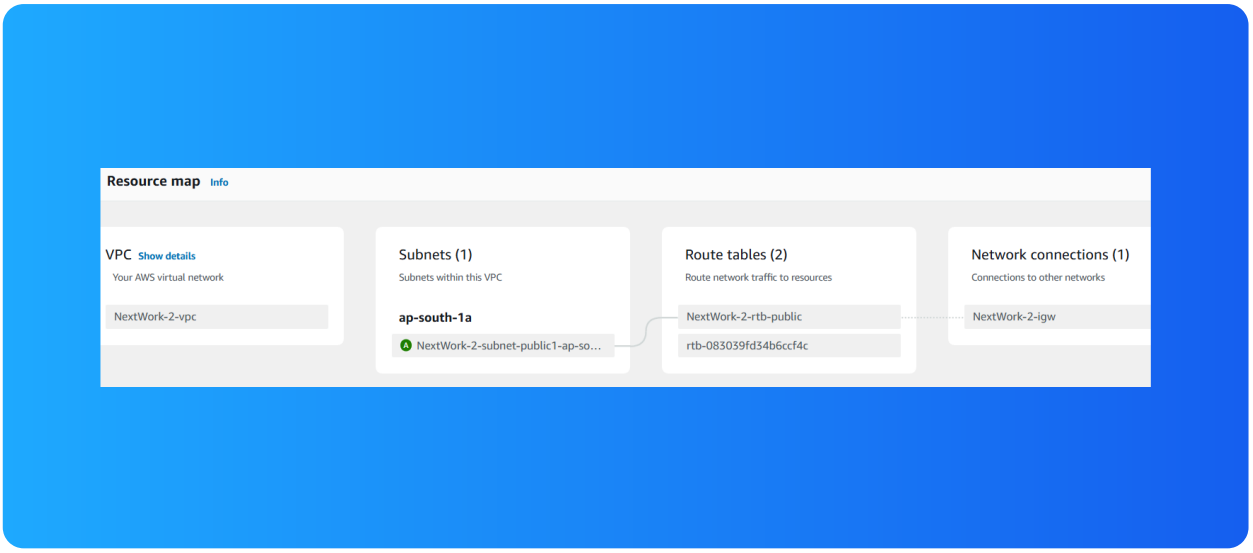
Multi-VPC Architecture

I started my project by launching two VPCs - they have unique CIDR blocks and they each have one public subnet.

The CIDR blocks for VPCs 1 and 2 are 10.1.0.0/16 and 10.2.0.0/16, respectively. These CIDR blocks must be unique to ensure correct routing across VPCs after establishing a VPC peering connection.

I also launched 2 EC2 instances

I didn't set up key pairs for these EC2 instances as it was not necessary in this project. The previous projects covered the working of SSH and EC2 Instance Connect using which AWS manages the key-pairs for the user.



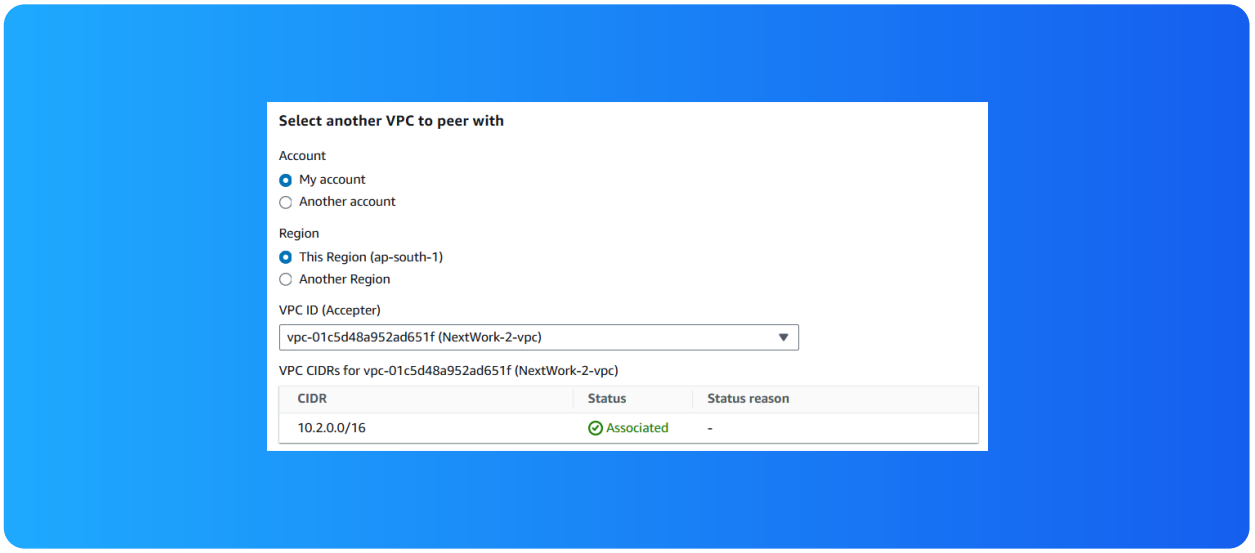
SH
shravanirg03@gmail.com
NextWork Student

VPC Peering

A VPC peering connection creates a link between two VPCs that enables securely routing traffic between them using private addresses.

VPCs would use peering connections to communicate with each other without traversing the public internet. This provides them with a secure and efficient way to communicate.

A Requester is a VPC that initiates a peering connection by sending an invitation to connect. An Accepter is the VPC that receives a peering connection request. It can either accept or decline the invitation.

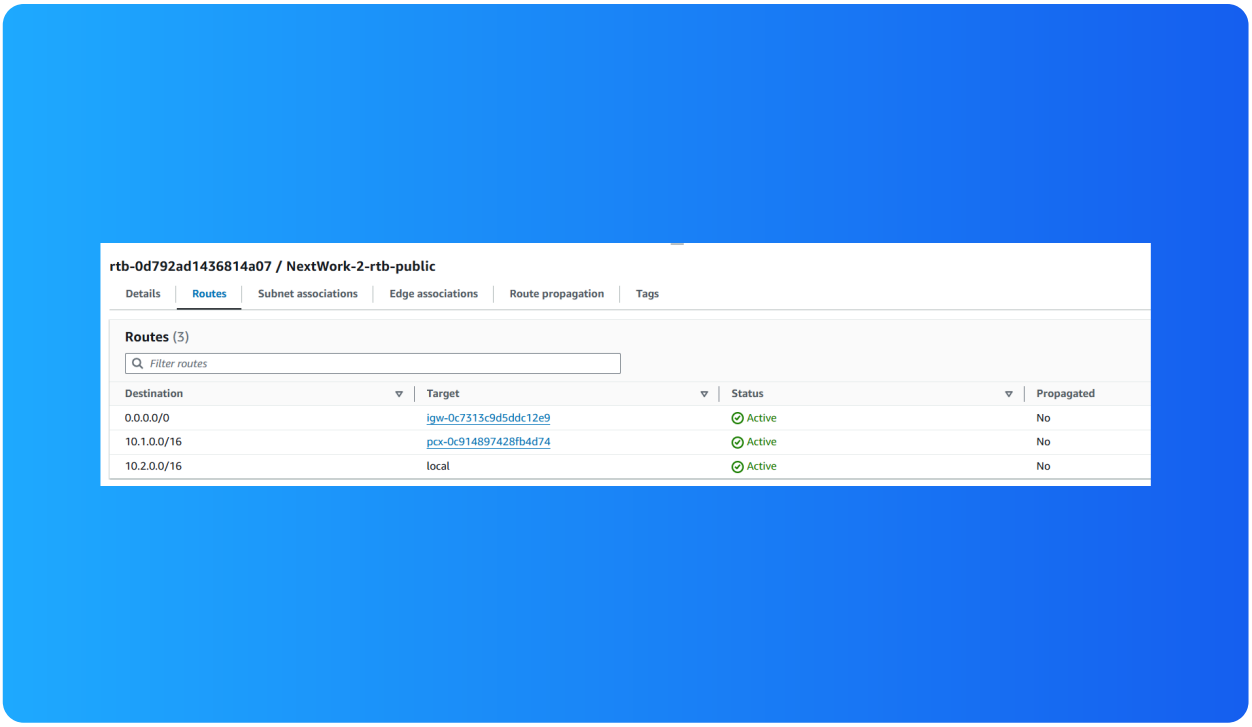


SH
shravanirg03@gmail.com
NextWork Student

Updating route tables

The traffic from the VPCs need directions from route table while navigating. There doesn't exist a route that directs the traffic to the peering connection. Thus, route tables need to be updated for traffic from one VPC to reach the other VPC.

My VPCs' new routes have a destination of the other VPC's CIDR block. The routes' target was the Peering Connection that I set up.



SH
shravanirg03@gmail.com
NextWork Student
NextWork.org

In the second part of my project...

Step 5 - Use EC2 Instance Connect

In this step, we are using EC2 Instance Connect to connect to the first EC2 instance while fixing any connection errors that might occur in the process.

Step 6 - Connect to EC2 Instance 1

Reattempting to connect to Instance - NextWork VPC 1, and resolving another error preventing us from using EC2 Instance Connect to direclty connect with the instance.

Step 7 - Test VPC Peering

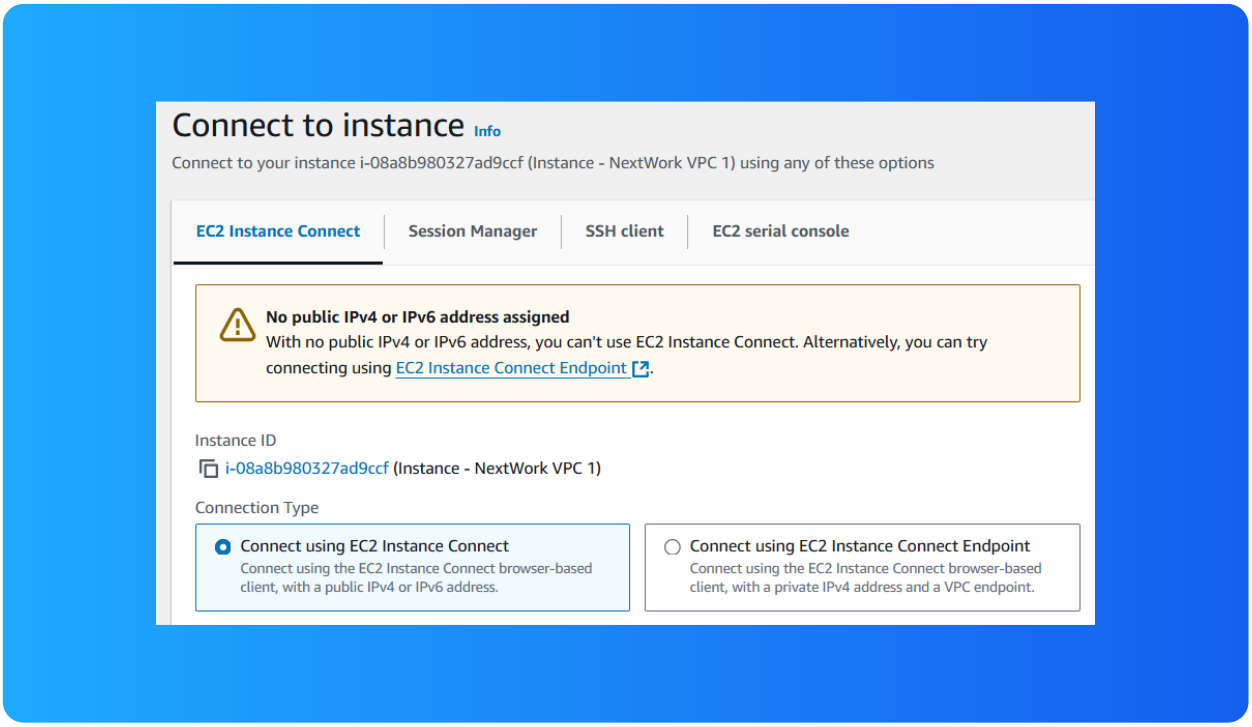
We are going to send a message to Instance 2 from Instance 1 and solve all the connection errors that arise in the process until Instance 2 responds to the message.

SH
shravanirg03@gmail.com
NextWork Student

Troubleshooting Instance Connect

Next, I used EC2 Instance Connect to directly connect with Instance - NextWork VPC 1 just by using the AWS Management Console.

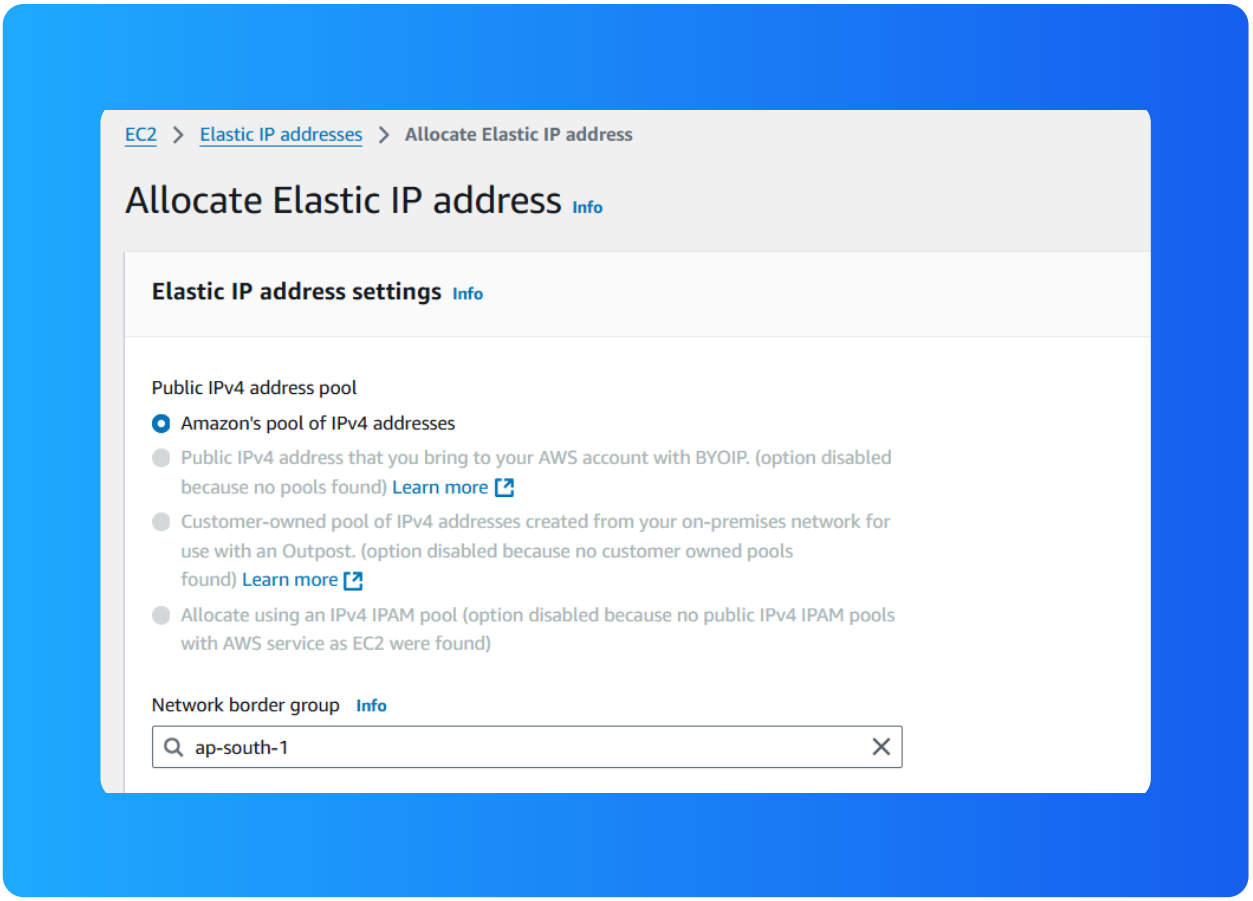
I was stopped from using EC2 Instance Connect as the instance did not have a public IPv4 address. In order for EC2 Instance Connect to work, the EC2 instance must have a public IPv4 address and be in a public subnet.



Elastic IP addresses

To resolve this error, I set up Elastic IP addresses. Elastic IP addresses are static public IPv4 addresses that we can request for our AWS account and then delegate to specific resources.

Associating an Elastic IP address resolved the error because it assigned the EC2 instance with a public address, fulfilling the requirements for Instance Connect to work.



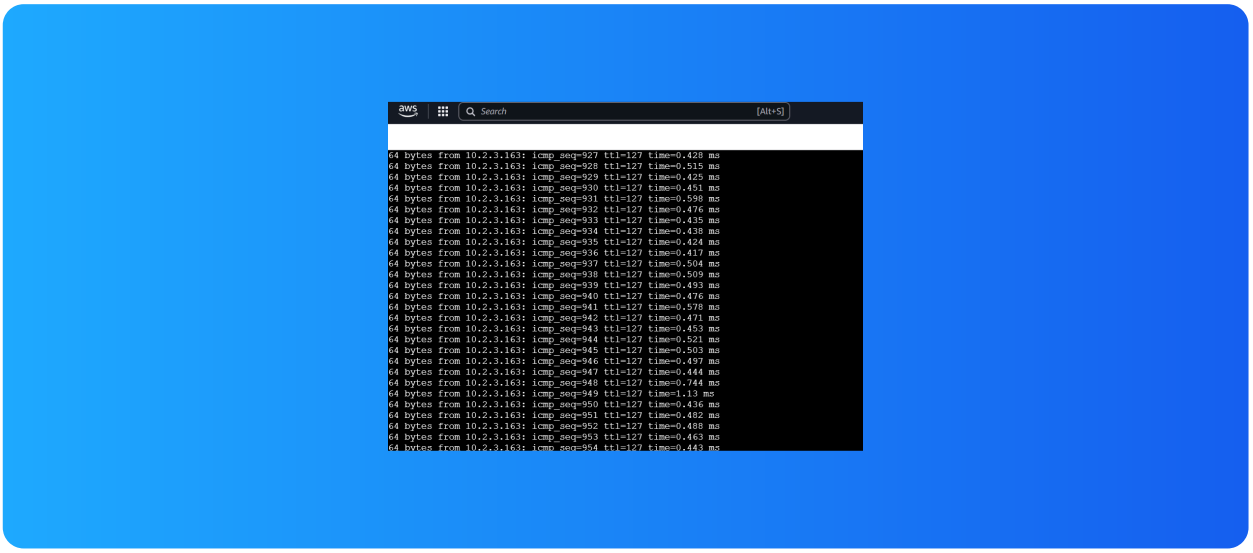
SH
shravanirg03@gmail.com
NextWork Student

Troubleshooting ping issues

To test VPC peering, I ran the command 'ping 10.2.3.163'. Here, '10.2.3.163' is the private IPv4 address of the other instance which is in VPC 2.

A successful ping test would validate my VPC peering connection because the destination instance in the ping command would respond to the message sent by an instance if the peering connection between the two VPCs was successful.

I had to update my second EC2 instance's security group because it was not letting in ICMP traffic - which is the traffic type of a ping message. I added a new rule that allows ICMP traffic coming in from any resource in VPC 2.



Everyone should be in a job they love.

Check out nextwork.org for more projects