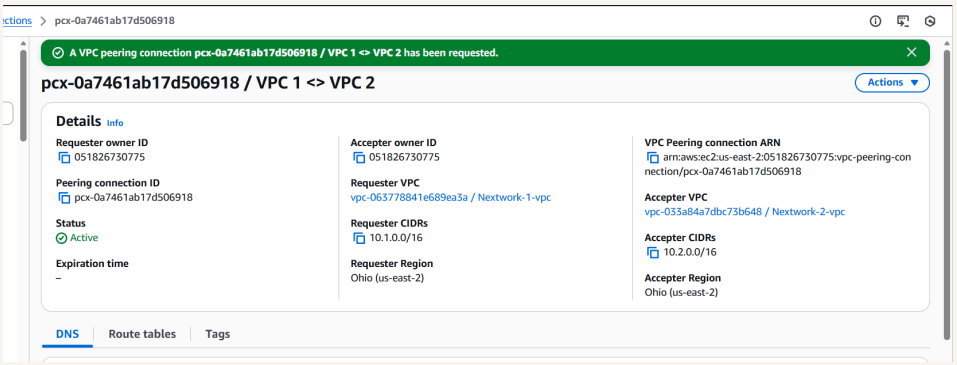
nextwork.org

VPC Peering

H Harshada Kripal



Harshada Kripal
NextWork Student

Introducing Today's Project!

What is Amazon VPC?

Amazon VPC is Virtual Private Cloud which allows us to launch our resources in our VPC. It is a foundational networking service that let's us create our own network so that we can control traffic flow, security and organize resources into public and private subnet.

How I used Amazon VPC in this project

In today's project, I used Amazon VPC to setup a multi VPC architecture (I setup 2 VPCs) create a peering connection between them and update security group rules to run a successful connectivity test to validate my VPC Peering setup.

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

One thing I didn't expect in this project was I didn't expect to need a public IPV4 address for EC2 instance connect to work. Also that elastic IP can assign static IPV4 to resources.

This project took me...

This project took me almost 2 hours to complete.

Harshada Kripal
NextWork Student

In the first part of my project...

Step 1 - Set up my VPC

In this step, I'm using a VPC resource map/launch wizard to create Two VPC's and their components in just few minutes.

Step 2 - Create a Peering Connection

In this step, I'm setting up an VPC peering connection which is a component to directly connect two VPC's together.

Step 3 - Update Route Tables

In this step, I will be setting up route tables to set up a way for traffic coming from VPC1 to get to VPC2 so that traffic can be directed to the peering connection.

Step 4 - Launch EC2 Instances

In this step, I will launch EC2 instances in each of the VPC's because to directlt connect them and use them later to test the VPC peering connection.

Harshada Kripal
NextWork Student

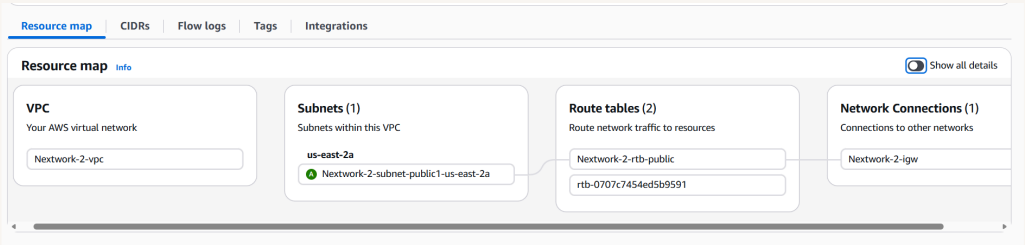
Multi-VPC Architecture

I started my project by launching two VPC's. 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 & 10.2.0.0/16 They have to be unique because once you setup a VPC peering coonection route tables need unique addressess for correct routing across VPC's.

I also launched 2 EC2 instances

I didn't set up key pairs for these EC2 instances as I'm using EC2 instance connect to directly connect with my EC2 instance later in this project which handles key pair creation & management for us.



Harshada Kripal
NextWork Student

VPC Peering

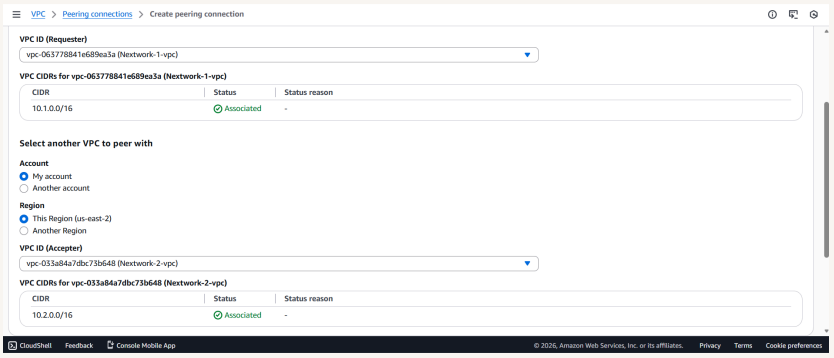
A VPC peering connection lets VPCs and their resources route traffic between them using their private IP addresses. This means data can now be transferred between VPCs without going through the public internet.

VPCs would use peering connections to let their resources route traffic between them using their private IP addresses. Transfering data between VPCs would use resources' public address - meaning VPCs have to communicate over the public internet.

The difference between a Requester and an Accepter in a peering connection is 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. The Accepter can either accept or decline the invitation. This means the peering connection isn't actually made until the other VPC also agrees to it.

Harshada Kripal
NextWork Student

nextwork.org

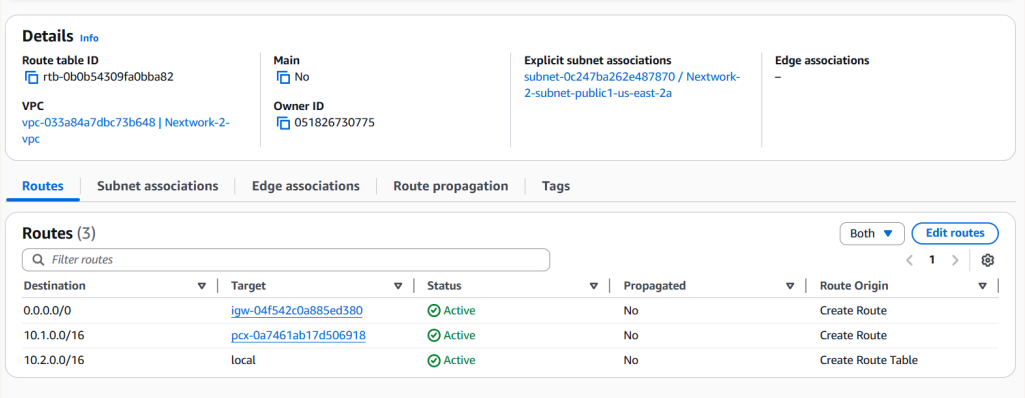


Harshada Kripal
NextWork Student

Updating route tables

After accepting a peering connection, my VPCs' route tables need to be updated because the default route table doesn't have a peering connection yet it needs to be setup so that the resources can be directed to the peering connection when trying to reach the other VPC.

My VPCs' new routes have a destination og the other VPC's CIDR block. The routes' target was the peering connection setup.



Harshada Kripal
NextWork Student

In the second part of my project...

Step 5 - Use EC2 Instance Connect

In this step, I will be using EC2 instance to connect directly with my first EC2 instance because i need to use my EC2 instance for connectivity test later in this project.

Step 6 - Connect to EC2 Instance 1

In this step, I will be reattempting my connection to Instance Nextwork-vpc-1 and resolving another error preventing from using EC2 instance connect to directly connect to the instance.

Step 7 - Test VPC Peering

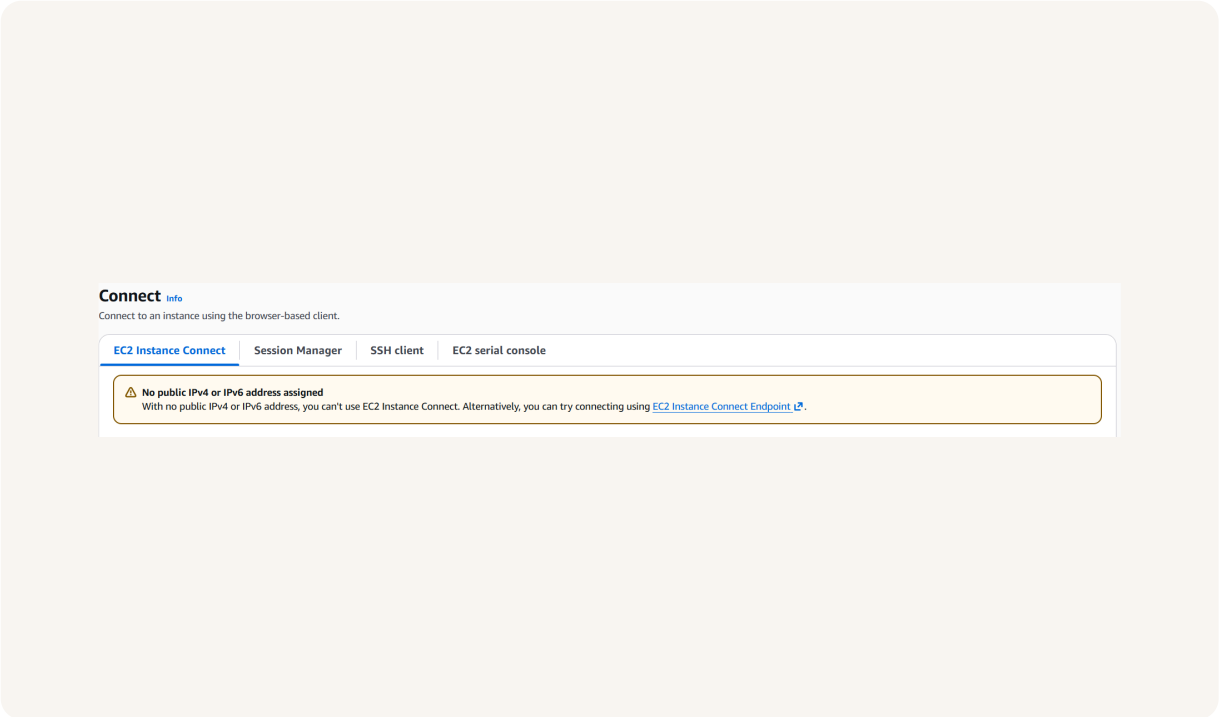
In this step, I will be using the instance Nextwork-vpc-1 to attempt a direct connection with Nextwork-vpc-2 instance because I can validate that my peering is setup properly.

Harshada Kripal
NextWork Student

Troubleshooting Instance Connect

Next, I used EC2 Instance Connect to directly connect with my instance Nextwork-vpc -1 just by using AWS managemnet console.

I was stopped from using EC2 Instance Connect as my instance did not have an public IPV4 address.

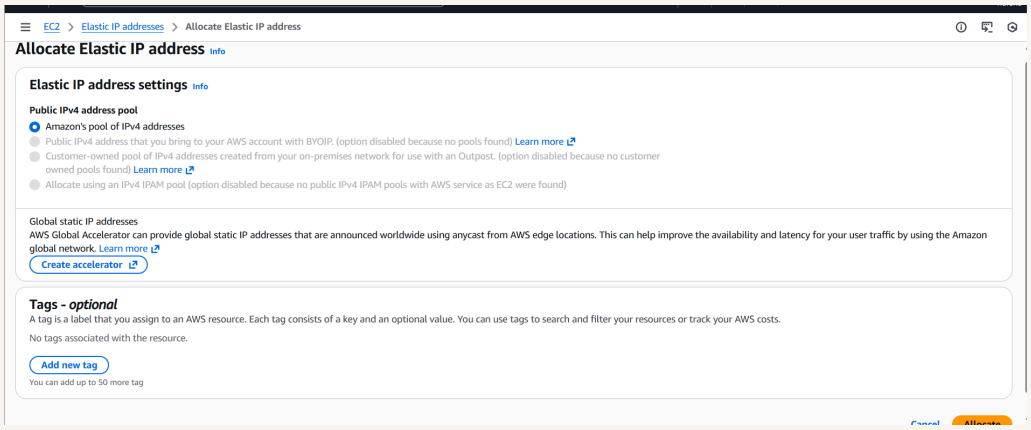


Harshada Kripal
NextWork Student

Elastic IP addresses

To resolve this error, I set up Elastic IP addresses. Elastic IP addresses are public IPV4 addresses that we can request for our AWS account and then deligate to spcific resources that will use this ip address.

Associating an Elastic IP address resolved the error because it gives my EC2 instance a public IP address.



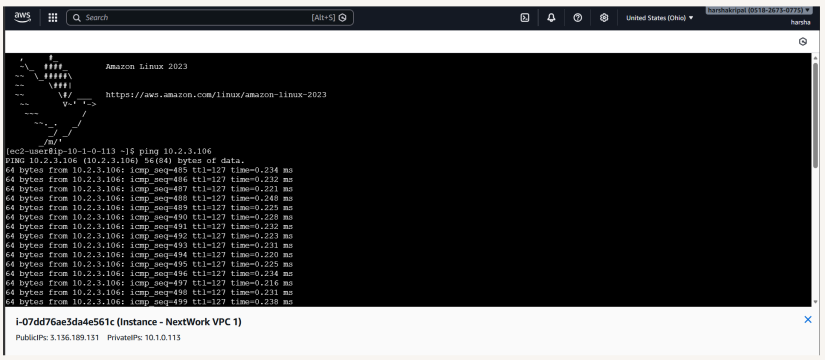
Harshada Kripal
NextWork Student

Troubleshooting ping issues

To test VPC peering, I ran the command $ping 10.2.X.XXX this is the private IPV4 addrss of other EC2 instance in VPC2.

A successful ping test would validate my VPC peering connection because this ping test would not get any replies from the other EC2 instance if the peering connection did not successfully connect to our two VPC's. Getting Ping replies equals hearing connection was setp up properly.

I had to update my second EC2 instance's security group because it was not letting in ICMP traffic which is the traffic type for ping message. I added a new rule that allows ICMP traffic coming in from any resource in VPC2.



The place to learn & showcase your skills

Check out nextwork.org for more projects