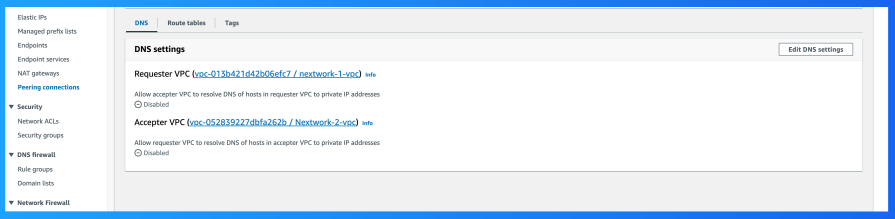
NextWork.org

VPC Peering

AL alexten3517@gmail.com



AL
alexten3517@gmail.com
NextWork Student
NextWork.org

Introducing Today's Project!

What is Amazon VPC?

Amazon VPC is AWS's foundational networking service that lets us create a our own networks,control traffic flow and security and organize our resources into public / private subnets.

How I used Amazon VPC in this project

We used Amazon VPC to set up a multi-VPC architecture (we set up two VPC's),create a peering connection between them and upadate security group rules to run a successful connectivity test to validate our VPC setup.

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

I did not expect to troubleshoot our peering connection by evaluating our subnet, network ACL and our security groups to see exactly where the bottleneck was which was holding back our connection.

This project took me...

I spent a total of 2 hours learning, evaluating and creating this project.

AL
alexten3517@gmail.com
NextWork Student
NextWork.org

In the first part of my project...

Step 1 - Set up my VPC

In this step , we are using the VPC resource map/launch wizard to create two VPCs and their components in just minutes.

Step 2 - Create a Peering Connection

In this step we are setting up a VPC Peering Connection, which is a VPC component designed to directly connect two VPC's together.

Step 3 - Update Route Tables

In this step we have are creating the component that allows a connection(peering connection) between VPC1 and VPC2 to communicate with each other.

Step 4 - Launch EC2 Instances

We are launching an EC2 insstance in each of the VPC's so that we can directly connect with an EC2 our instances later and test our vpc peering connection.

AL
alexten3517@gmail.com
NextWork Student
NextWork.org

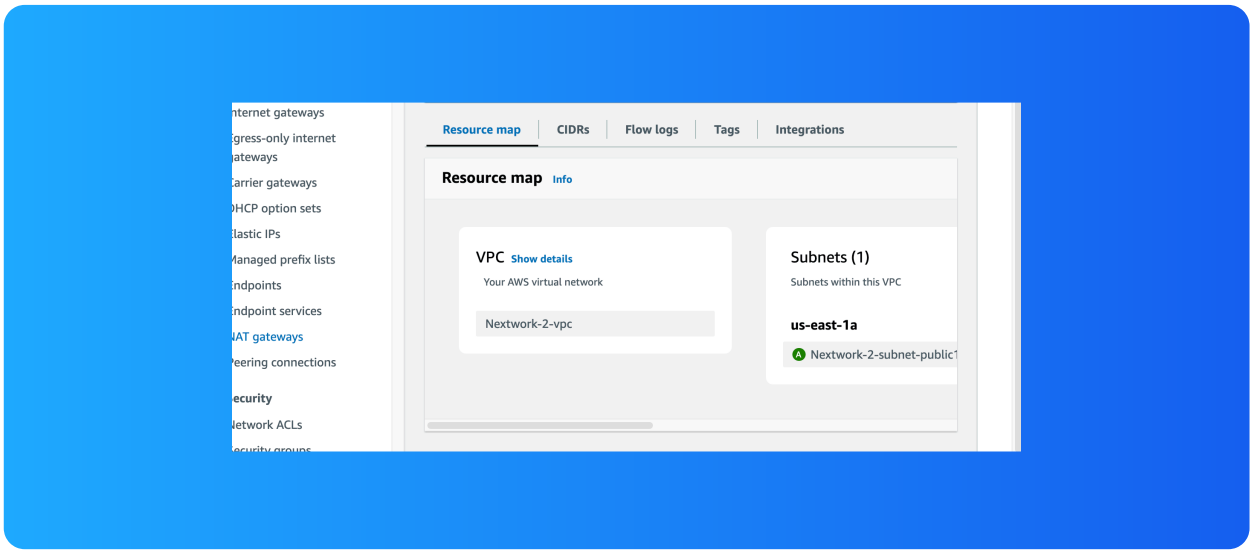
Multi-VPC Architecture

I started my project by launching two VPC's. They each have a unique CIDR block and they each have 1 public subnet.

The CIDR blocks for VPC's 1 and 2 are 10.1.0.0/16 and the second one is 10.2.0.0/16 to avoid overlapping of ip addresses. Once you setup a VPC peering connection, route tables need unique addresses for correct routing accross VPC's.

I also launched 2 EC2 instances

We did not choose to create a set up key pairs because we are using EC2 instance connect to directly connect with our EC2 instance later in this project, which handles key pair creation and management for us.



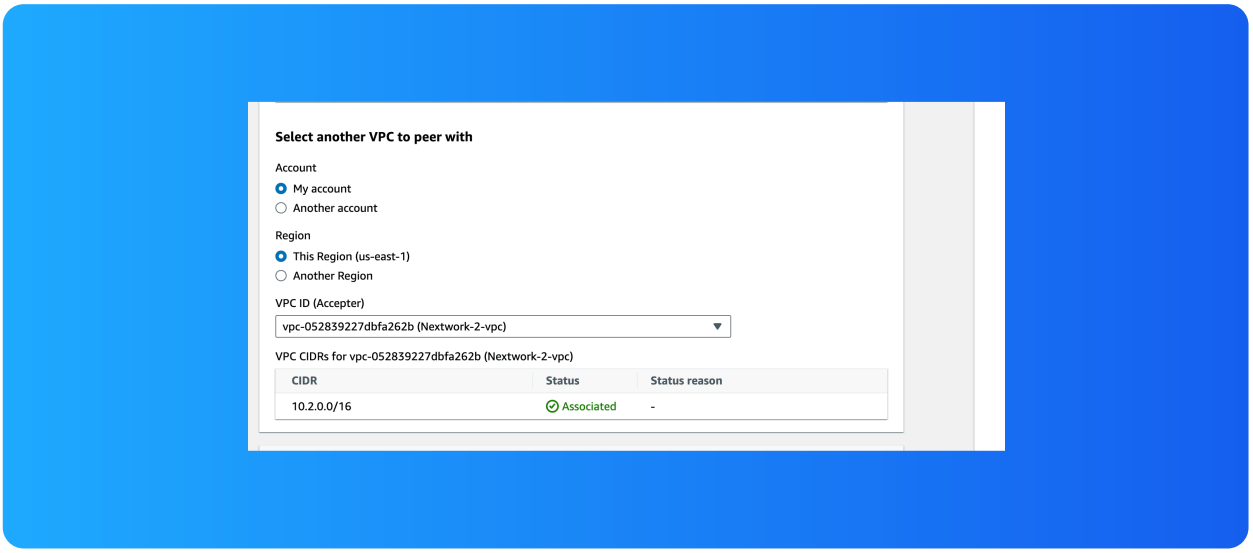
AL
alexten3517@gmail.com
NextWork Student

VPC Peering

A peering connection would be when two VPC's are able to communicate with each other privately without the use of the internet. This connection is one that allows for data transfer between both VPC's in a more secure and robust way.

VPCs would use peering connections to allow the communication to held privately. No longer would we need to send data throught the public internet to communicate with another VPC - with a peering connection we can connect VPC 1 to VPC 2 privately.

The difference between a Requester and an Accepter in a peering connection is VPC 1 requesting permission to connect to VPC 2.

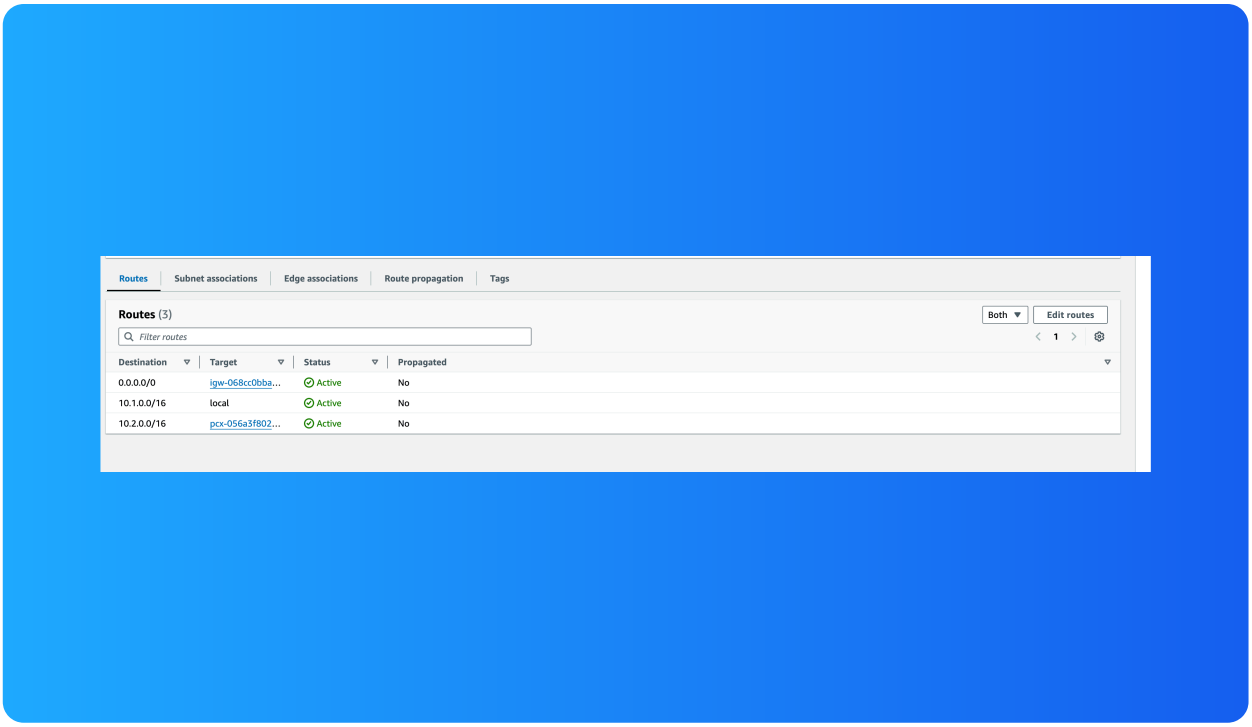


AL
alexten3517@gmail.com
NextWork Student

Updating route tables

After accepting a peering connection, my VPCs' route tables need to be updated because only public traffic was accepted and we needed a way to have our routing table set up to create effective communication with our VPC2.

The new route that was added was a route that will allow for intercommunication between VPC's in a more direct way without the any other traffic interferance.



AL
alexten3517@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 directly with our first EC2 instance. We need to use our EC2 instance for connectivity test later in this project.

Step 6 - Connect to EC2 Instance 1

We are reattempting our connection to Instance VPC 1, and resolving another error preventing us from using EC2 Instance connect to directly connect with the Instance.

Step 7 - Test VPC Peering

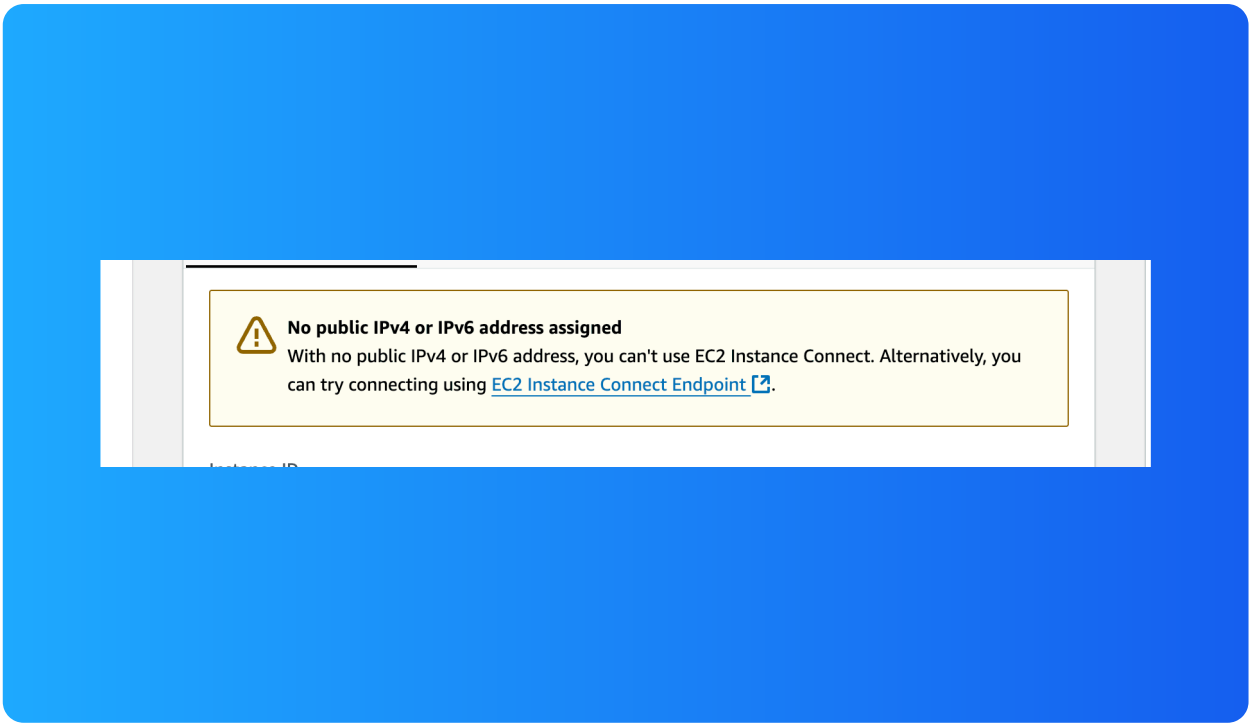
In this step we are going to create a connectivity test to see if we are able to pass our test which will give us a direct connection from VPC1 to VPC2.

AL
alexten3517@gmail.com
NextWork Student

Troubleshooting Instance Connect

Next, I used EC2 Instance Connect to directly connect with Instance - VPC1 just by using the AWS Management console.

I was stopped from using EC2 Instance Connect as our instance did not hava public PIPv4 address.



AL
alexten3517@gmail.com
NextWork Student
NextWork.org

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 delegate to specific resources.

Associating an Elastic IP address resolved the error because it gives our EC2 instance a public IP address. Fulfilling all the requirements for instance connect to work.

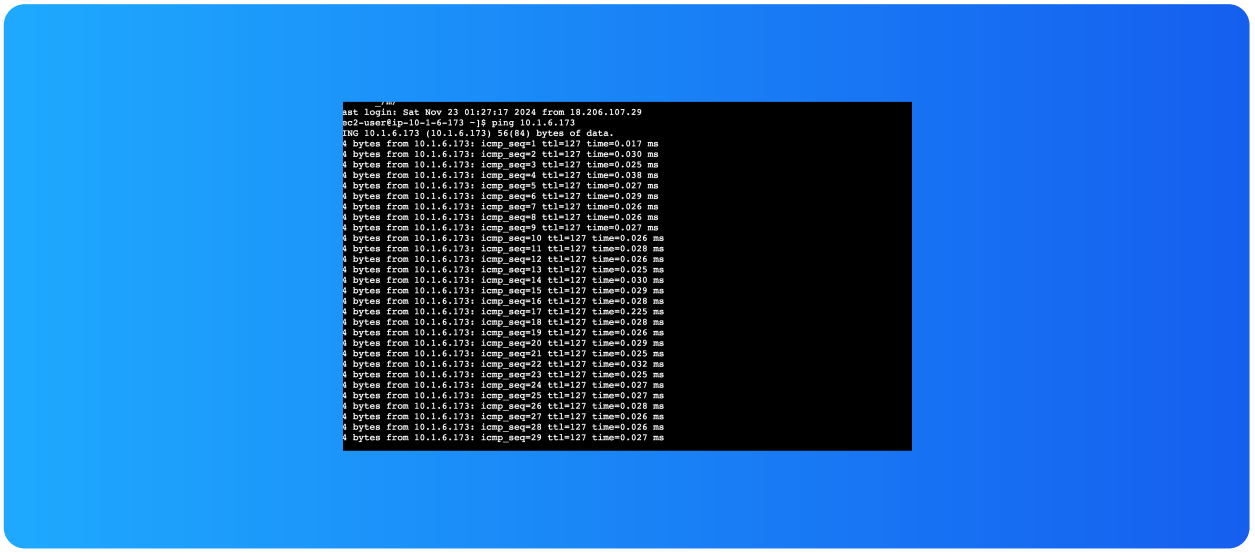
AL
alexten3517@gmail.com
NextWork Student

Troubleshooting ping issues

To test VPC peering, I ran a ping test to see if communication was successful. Using ping and the private ip address to get a connection is what was needed in getting a peering connection.

A successful ping test would validate my VPC peering connection because we were able to get a responce from VPC 2 which means communication between VPC's 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 it from any resource in VPC.



Everyone should be in a job they love.

Check out nextwork.org for more projects