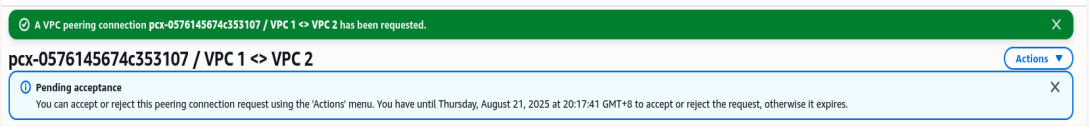
nextwork.org

VPC Peering

Sean Calderon



Introducing Today's Project!

What is Amazon VPC?

Amazon VPC is a wonderful service to create an easily scalable network infrastructure in the cloud. It is useful for engineers if the business requirement requires having different VPCs for peering or simply peering to a friend's VPC, isn't it wonderful?

How I used Amazon VPC in this project

In this project, I used Amazon VPC to setup scalable network infrastructure that allows me to peer two different VPCs using VPC peering connection.

This project took me...

I took this project approximately 50 minutes. I find it challenging for me to troubleshoot instance connection problems when the issue is only the default security group? That, only I didn't expect to be the problem and overlooked. It was rewarding to see when my two instances that reside on different VPC are able to talk to each other.

In the first part of my project...

Step 1 - Set up my VPC

In this step, I will create two VPCs from scratch with VPC wizard for super fast deploying of VPC within a few seconds.

Step 2 - Create a Peering Connection

In this step, I have two non-overlapping blocks of VPCs and I will deploy a VPC peering connection to link both VPCs.

Step 3 - Update Route Tables

In this step, I am going to set up a way for traffic coming from VPC 1 to get to VPC 2. On the other end, I am going to set up a way for traffic coming from VPC 2 to get to VPC 1. This way, the traffic will be allowed going back and forth.

Step 4 - Launch EC2 Instances

In this step, I am going to launch an EC2 instance in each VPC, so I can use them to test reachability between my VPCs with VPC peering connection.

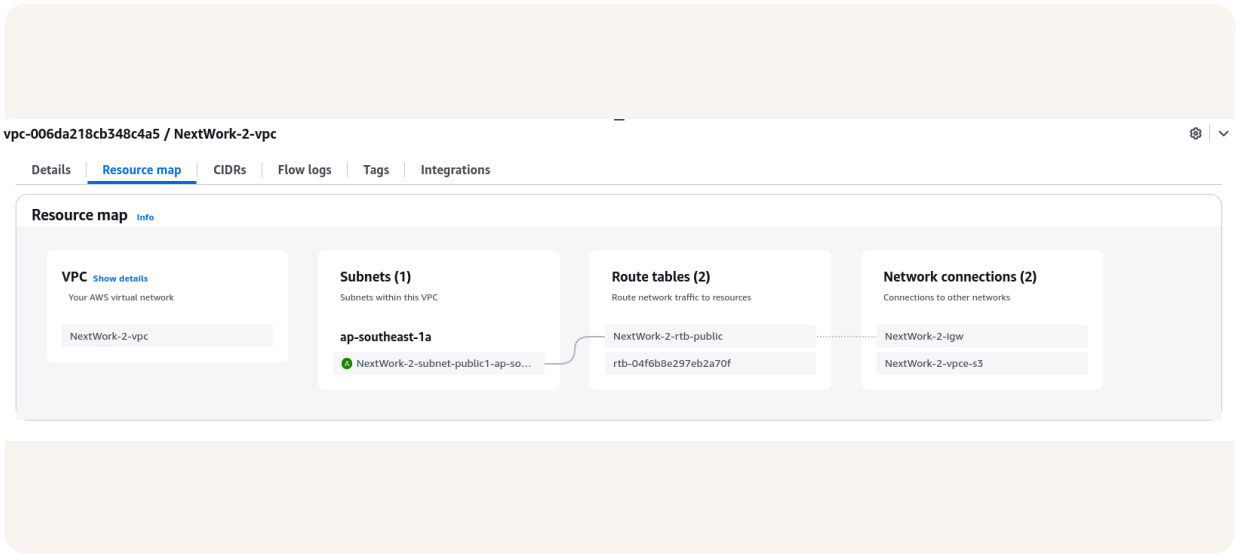
Multi-VPC Architecture

I started my project by launching two VPCs using VPC wizard and configured the IPv4 CIDR block to be non-overlapping of CIDR block of 10.1.0.0/16 and 10.2.0.0/16. Subsequently, I created only one public subnet per VPC.

The CIDR blocks for VPCs 1 and 2 are non-overlapping. They have to be unique because VPC peering is impossible since traffic is only to be routed within the VPC itself if other VPC has the same CIDR block causing conflicts and connectivity issues.

I also launched 2 EC2 instances

I didn't set up key pairs for these EC2 instances as I am going to stick on dedicated EC2 instance connect.



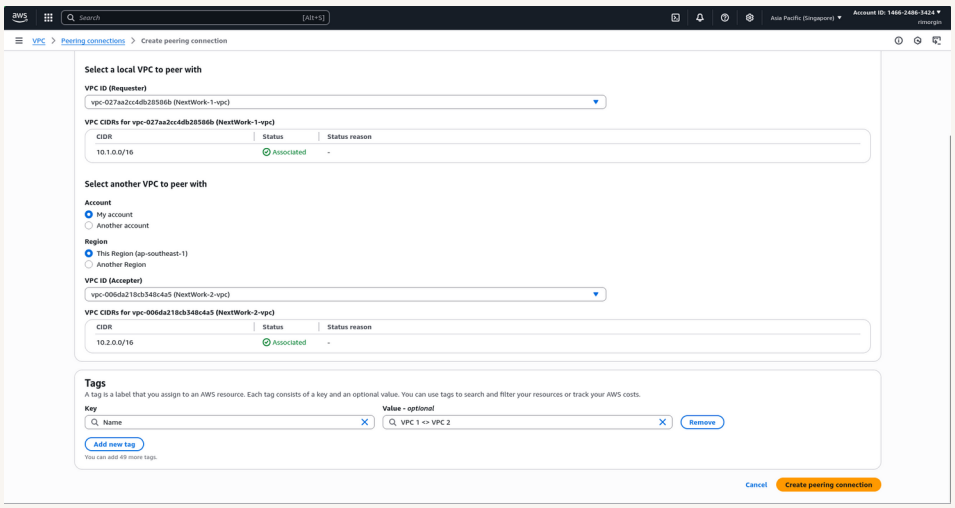

Sean Calderon
NextWork Student

VPC Peering

A VPC peering connection is a direct connection between two VPCs! A peering connection lets VPCs and their resources route traffic between them using their private IP addresses and without going out of the internet.

VPC peering connections eliminates traffic going out to the internet and use dedicated route instead to route traffic directly to the destination VPC, which reduces latency.

The difference between a Requester and an Accepter in a peering connection is Requester is the one the initiates connection and invites to be peer while Accepter is the one that accepts an invitation of peering connection.

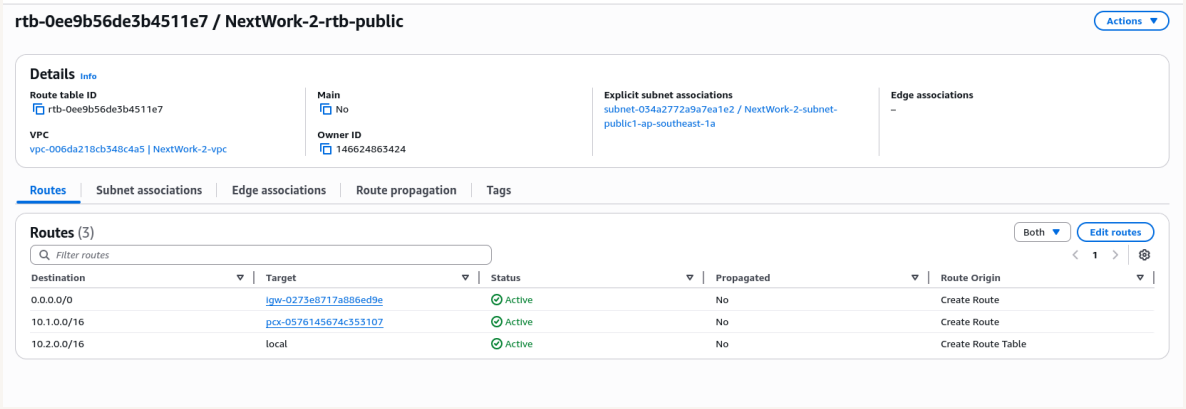



Sean Calderon
NextWork Student

Updating route tables

After accepting a peering connection, my VPCs' route tables need to be updated because there is still no instruction in the route table where to route traffic destined to each VPC's CIDR block.

My VPCs' new routes have a destination of 10.1.0.0/16 for my NextWork-2 route table and 10.2.0.0/16 for my NextWork-1 route table. The routes' target was of course the peering connection I made.



In the second part of my project...

Step 5 - Use EC2 Instance Connect

In this step, I will use EC2 Instance Connect to connect to my first EC2 instance.

Step 6 - Connect to EC2 Instance 1

In this step, I will use EC2 Instance Connect to connect to Instance 1 once again.

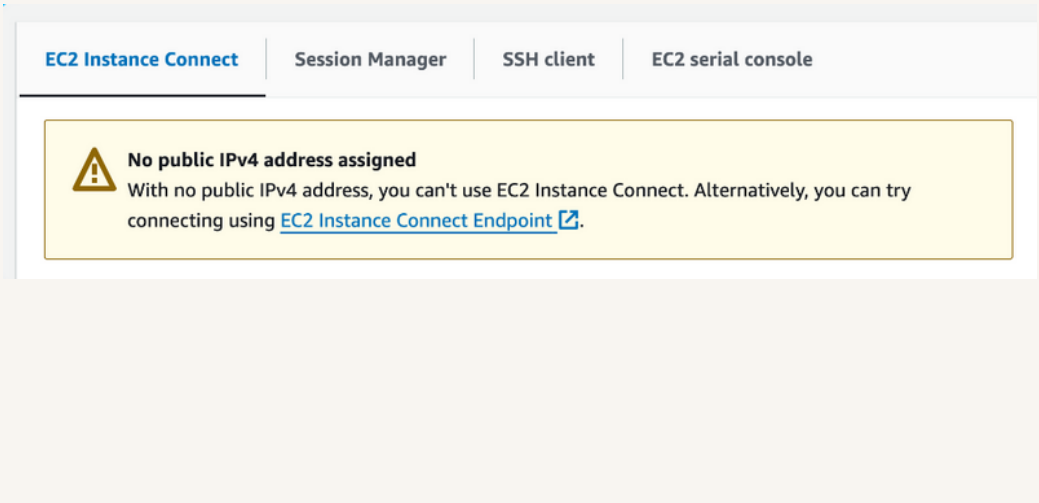
Step 7 - Test VPC Peering

In this step, I will have my Instance 1 to send test messages to Instance 2.

Troubleshooting Instance Connect

Next, I used EC2 Instance Connect to connect to my instance.

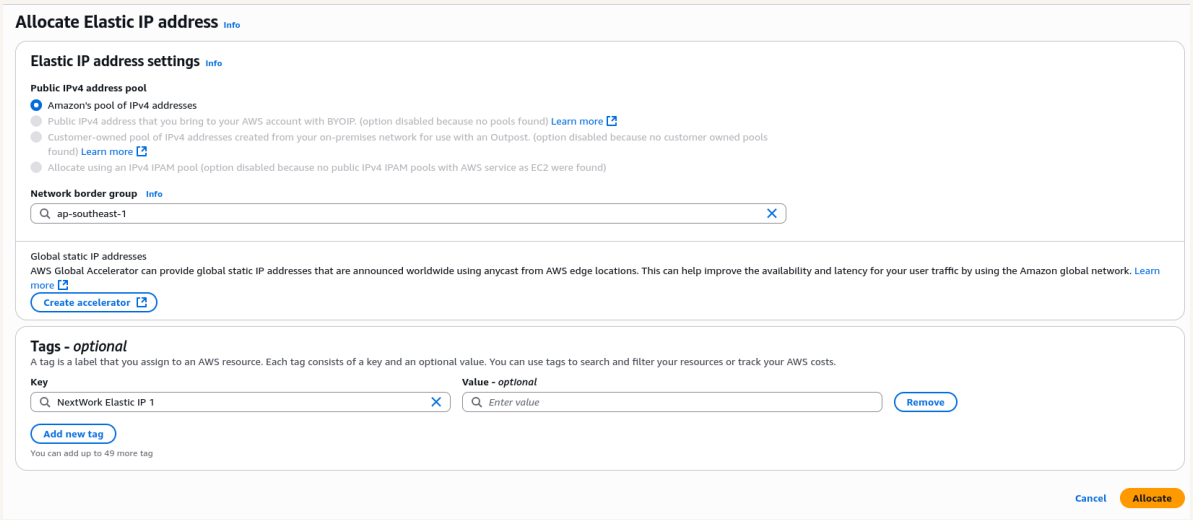
I was stopped from using EC2 Instance Connect as the instance does not have its public IP address assigned yet.



Elastic IP addresses

To resolve this error, I set up Elastic IP addresses. Elastic IP addresses are fixed allocation of IP address attached to the instance.

Associating an Elastic IP address resolved the error because it is impossible for an instance to be accessible over the internet if its only accessible inside the VPC subnet since it has only private IP address.

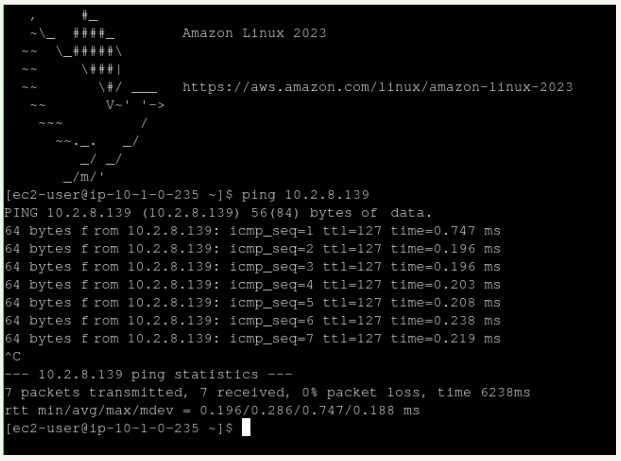


Troubleshooting ping issues

To test VPC peering, I ran the command ping to test end-to-end connectivity of my VPCs.

A successful ping test would validate my VPC peering connection because ping test actually verify connectivity going to the destination and to the destination coming back to the source.

I had to update my second EC2 instance's security group because the default only allows inbound for VPC only so I added a new rule that accepts traffic coming to the peered VPC.



The place to learn & showcase your skills

Check out nextwork.org for more projects