



nextwork.org

VPC Peering

ME

Melvin J Bonner

Accept VPC peering connection request Info X

Are you sure you want to accept this VPC peering connection request? (pcx-0fcdb89bb0b19ae9b2 / VPC 1 <> VPC 2)

Requester VPC vpc-05128a811d7607152 / Next-Work-1-vpc	Acceptor VPC vpc-06c229140ebaa8ada / Nextwork-2-vpc	Requester CIDRs <input checked="" type="checkbox"/> 10.1.0.0/16
Acceptor CIDRs -	Requester Region N. Virginia (us-east-1)	Acceptor Region N. Virginia (us-east-1)
Requester owner ID <input checked="" type="checkbox"/> 812541449234(This account)	Acceptor owner ID <input checked="" type="checkbox"/> 812541449234(This account)	

Cancel Accept request



ME

Melvin J Bonner
NextWork Student

nextwork.org

Introducing Today's Project!

What is Amazon VPC?

Amazon VPC is an isolated section of the AWS cloud where you can launch AWS resources in a virtual network that you define.

How I used Amazon VPC in this project

In this project I used Amazon VPC to demonstrate how peering works.

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

One thing I didn't expect is how well EC2 Instance Connect worked. I expected to run into some of the same issues I ran into using VS Code.

This project took me...

The project took me 3 hours.

In the first part of my project...

Step 1 - Set up my VPC

In this step, I will be creating two VPCs using the VPC wizard.

Step 2 - Create a Peering Connection

In this step I will be bridging the two VPCs together with a peering connection.

Step 3 - Update Route Tables

In this step, I will be connecting the incoming and outgoing traffic between VPC 1 and VPC 2.

Step 4 - Launch EC2 Instances

In this step, I will be launching an EC2 instance in each VPC, in order to test the peering connection between my two VPCs.

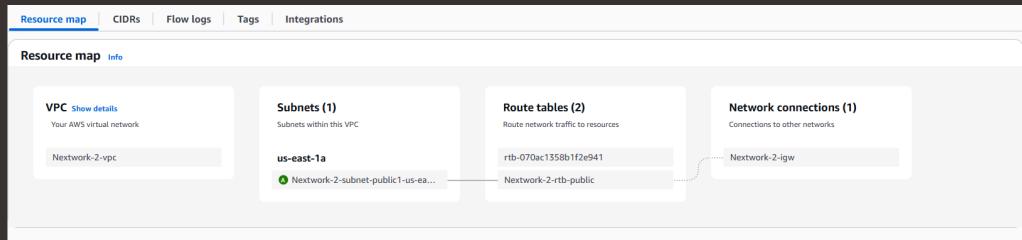
Multi-VPC Architecture

I started my project by launching two VPCs. I created 1 subnet for each VPC.

The CIDR blocks for VPC1 and VPC 2 are 10.1.0.0/16 and 10.2.0.0/16. They have to be unique IPv4 CIDR blocks so the IP addresses of their resources don't overlap. Having overlapping IP addresses could cause routing conflicts and connectivity issues.

I also launched 2 EC2 instances

I didn't set up key pairs for these EC2 instances as AWS will manage the key pairs for me using EC2 Instance Connect.



VPC Peering

A VPC peering connection is a direct connection between two VPCs.

VPCs would use peering connections to route traffic between them using their private IP addresses. This means data can now be transferred between VPCs without going through the public internet.

In a peering connection the requester is the VPC that initiates a peering connection. The requester sends the other VPC an invitation to connect. The accepter is the VPC that receives a peering connection request and the accepter can either accept or decline the invitation.

Select another VPC to peer with

Account
 My account
 Another account

Region
 This Region (us-east-1)
 Another Region

VPC ID (Accepter)
vpc-06c229140ebaa8ada (Nextwork-2-vpc)

VPC CIDRs for vpc-06c229140ebaa8ada (Nextwork-2-vpc)

CIDR	Status	Status reason
10.2.0.0/16	Associated	-

Updating route tables

After accepting a peering connection, my VPCs' route tables needs to be updated because the traffic in VPC 1 won't know how to get to the resources in VPC 2 without a route in my route table. I needed to set up a route that directed traffic bound for VPC 2 to the peering connection I set up.

My VPCs' new routes have a destination of 10.1.0.0/16 and 10.2.0.0/16. The routes' target was the peering connection.

Routes				
Routes (3)		Subnet associations		Edge associations
Destination	Target	Route propagation	Tags	
0.0.0.0/0	igw-0477c2bb6e20e4bd7	Active		No
10.1.0.0/16	pxx-0fd89bb0b19ae9b2	Active		No
10.2.0.0/16	local	Active		No

In the second part of my project...

Step 5 - Use EC2 Instance Connect

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

Step 6 - Connect to EC2 Instance 1

In this next step, I will be connecting to the EC2 instance with Instance Connect.

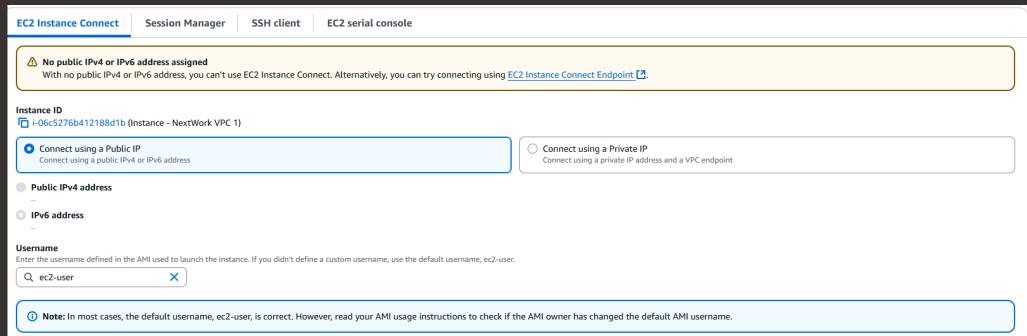
Step 7 - Test VPC Peering

In this next step, I will try to connect instance 1 to instance 2. I will be sending messages from instance 2 and have instance 2 send messages back.

Troubleshooting Instance Connect

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

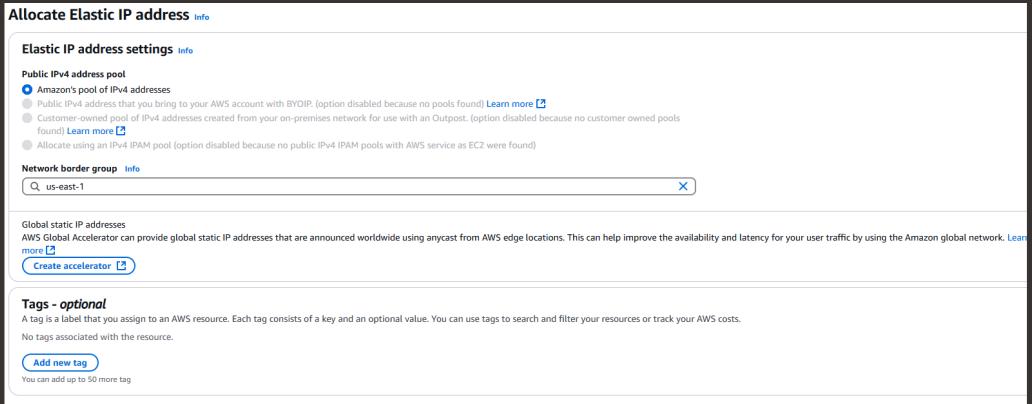
I was stopped from using EC2 Instance Connect originally, because there was no public IPv4 address assigned to the EC2 instance.



Elastic IP addresses

To resolve this error, I set up an Elastic IP address. The IP addresses are static, meaning that they remain the same even after the EC2 is restarted.

Associating an Elastic IP address resolved the error because the EC2 is now assigned a public IPv4 address.



Troubleshooting ping issues

To test VPC peering, I ran the ping command from EC2 1 to EC2 2.

A successful ping test would validate my EC2 peering connection because it shows replies from the ping message I sent. Traffic is successfully reaching EC2 instance 2.

I had to update my second EC2 instance's security group because ICMP traffic was being blocked and ping messages are ICMP traffic. I added a new rule that allowed ICMP traffic.



```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-10-1-7-147 ~]$ ping 10.2.8.45
PING 10.2.8.45 (10.2.8.45) 56(84) bytes of data.
64 bytes from 10.2.8.45: icmp_seq=559 ttl=127 time=0.480 ms
64 bytes from 10.2.8.45: icmp_seq=560 ttl=127 time=0.494 ms
64 bytes from 10.2.8.45: icmp_seq=561 ttl=127 time=0.454 ms
64 bytes from 10.2.8.45: icmp_seq=562 ttl=127 time=0.455 ms
64 bytes from 10.2.8.45: icmp_seq=563 ttl=127 time=0.459 ms
64 bytes from 10.2.8.45: icmp_seq=564 ttl=127 time=0.440 ms
64 bytes from 10.2.8.45: icmp_seq=565 ttl=127 time=0.450 ms
64 bytes from 10.2.8.45: icmp_seq=566 ttl=127 time=0.516 ms
64 bytes from 10.2.8.45: icmp_seq=567 ttl=127 time=0.436 ms
64 bytes from 10.2.8.45: icmp_seq=568 ttl=127 time=0.450 ms
64 bytes from 10.2.8.45: icmp_seq=569 ttl=127 time=0.523 ms
64 bytes from 10.2.8.45: icmp_seq=570 ttl=127 time=0.497 ms
64 bytes from 10.2.8.45: icmp_seq=571 ttl=127 time=0.450 ms
64 bytes from 10.2.8.45: icmp_seq=572 ttl=127 time=0.498 ms
64 bytes from 10.2.8.45: icmp_seq=573 ttl=127 time=0.450 ms
64 bytes from 10.2.8.45: icmp_seq=574 ttl=127 time=0.470 ms
64 bytes from 10.2.8.45: icmp_seq=575 ttl=127 time=0.488 ms
64 bytes from 10.2.8.45: icmp_seq=576 ttl=127 time=0.450 ms
64 bytes from 10.2.8.45: icmp_seq=577 ttl=127 time=0.459 ms
64 bytes from 10.2.8.45: icmp_seq=578 ttl=127 time=0.464 ms
64 bytes from 10.2.8.45: icmp_seq=579 ttl=127 time=0.455 ms
64 bytes from 10.2.8.45: icmp_seq=580 ttl=127 time=0.436 ms
64 bytes from 10.2.8.45: icmp_seq=581 ttl=127 time=0.447 ms
64 bytes from 10.2.8.45: icmp_seq=582 ttl=127 time=0.399 ms
```



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

