



VPC Peering



Joba Amunigun

<https://www.linkedin.com/in/dvoice>

The screenshot shows the AWS VPC Peering Connection details page for a connection named 'pcx-092d47768997df777'. A green banner at the top indicates that the connection has been established. The main table displays various connection details:

Details		Info	
Requester owner ID	394971810761	Acceptor owner ID	394971810761
Peering connection ID	pcx-092d47768997df777	Requester VPC	vpc-0e54eddb958fa87a / NextWork-1-vpc
Status	Active	Requester CIDR	10.1.0.0/16
Expiration time	-	Requester Region	Ohio (us-east-2)
VPC Peering connection ARN			
arn:aws:ec2:us-east-2:394971810761:vpc-peering-connection/pcx-092d47768997df777			
Acceptor VPC			
vpc-0dd404db945d06a242 / NextWork-2-vpc			
Acceptor CIDR			
10.2.0.0/16			
Acceptor Region			
Ohio (us-east-2)			



Introducing Today's Project!

What is Amazon VPC?

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

How I used Amazon VPC in this project

I used Amazon VPC to set up a multi-VPC Architecture (I set up two VPCs), I also used it to create a peering connection and to update security groups rules to run a succesfull connectivity test to validate my peering set up

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

I didn't expect to need a public IPv4 Address for EC2 instance to work and also didn't expect that Elastic IPs can assign static IPv4 to resources .

This project took me...

This project took me about 2hours + including studiing time



In the first part of my project...

Step 1 - Set up my VPC

In this step i will be using VPC resources/map wizard to create two VPCs and their components in just minutes

Step 2 - Create a Peering Connection

In this step , I will be setting up a VPC Peering connections, which is a VPC Component designed to directly connect two VPCs together

Step 3 - Update Route Tables

In this project , i will be setting up a way for traffic coming from VPC 1 to get to VPC 2 and for traffic coming from VPC 2 to get to VPC 1

Step 4 - Launch EC2 Instances

In this step - I will be Launching an EC2 instance in each VPC (VPC 1 and VPC 2), so that i can connect them directly to my instance and use them to test my VPC peering connection later.

Joba Amunigun
NextWork Student

nextwork.org

Multi-VPC Architecture

I started my project by launching two VPCs - They have unique CIDR Blocks and 1 Subnet

The CIDR blocks for VPCs 1 and 2 are "10.1.0.0/16" and " 10.2.0.0/16" respectively - They have unique IPv4 CIDR block 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 because i am using EC2 instance connect to connect directly with the EC2 Instance in this project (AWS handles Key Pair Creation and mangement for me)





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. This means data can now be transferred between VPCs without going through the public internet. Without a peering connection, data transfers between VPCs would use resources' public address - meaning VPCs have to communicate over the public internet.

VPCs would use peering connections to let 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.

The difference between a Requester and an Acceptor in a peering connection is that the Requester initiates a peering connection - The requester will send the other VPC an invitation to connect! While, the Acceptor is the VPC that receives a peering connection request! The Acceptor can either accept or decline the invitation. For this project - The Requester VPC was "NextWork-1-vpc" and the Acceptor VPC was Acceptor VPC "NextWork-2-vpc" Because peering connection isn't actually made until the other VPC also agrees to it i also accepted the connection to establishing the VPC peering



Joba Amunigun
NextWork Student

nextwork.org

Select another VPC to peer with

Account

My account
 Another account

Region

This Region (us-east-2)
 Another Region

VPC ID (Acceptor)

vpc-0d404db945d06a242 (NextWork-2-vpc) ▾

VPC CIDRs for vpc-0d404db945d06a242 (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 need to be updated because the default route table doesn't have a route using the peering connection yet

- This needs to be set up so that resources can be directed to the peering connection when trying to reach other VPC.

My VPCs' new routes have a destination of the other VPCs CIDR block . The routes' target was the pairng connection i set up

Routes (3)					
Filter routes				Both	
Destination	Target	Status	Propagated		
0.0.0.0/0	igw-0d64fcc8f771a4887	Active	No		
10.1.0.0/16	local	Active	No		
10.2.0.0/16	pxc-0c5e7627c1697ae4f	Active	No		



In the second part of my project...

Step 5 - Use EC2 Instance Connect

In this step ,I will be testing my VPC peering connection - I will need one of my EC2 instances to try talk to the other. Hence, I will use EC2 Instance Connect to connect to my EC2 instance.

Step 6 - Connect to EC2 Instance 1

In this step , I will be re-attempting my connecting to my EC2 instance and reolving another error preventing me from using EC2 instance connect to connect directly to the instance

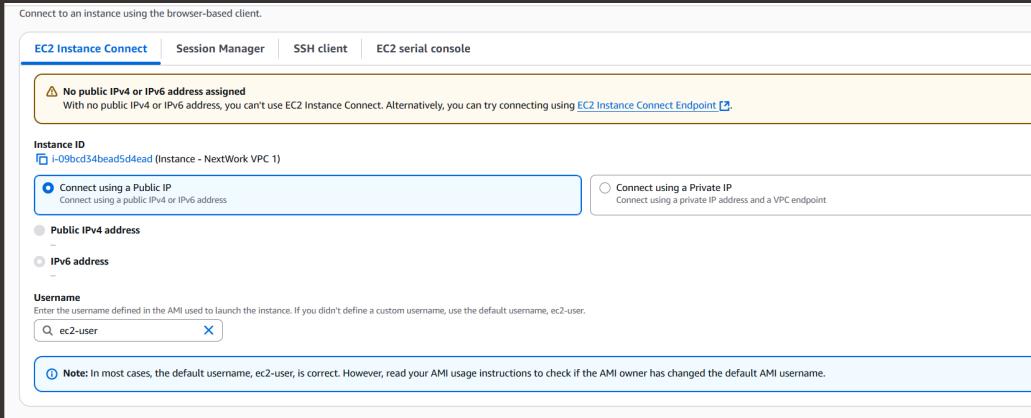
Step 7 - Test VPC Peering

In this step, I will be using the instance "NextWork VPC-1" to attempt a direct connection with Instance "NextWork VPC-2" so that i can validate that my peering connection is set up properly

Troubleshooting Instance Connect

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

'I was stopped from using EC2 Instance Connect as my instance did not have a IPv4 address. In order for EC2 Instance 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 IPv4 addresses that get allocated to your AWS account, and is yours to delegate to an EC2 instance - "Static" would mean that because EC2 instances by default have dynamic IPs, their Public IPv4 addresses change every time they're restarted. Having an Elastic IP is like having a permanent address in a city, instead of having to move from location to location every time your instance restarts.

Associating an Elastic IP address resolved the error because it gives my EC2 Instance a Public IP address fulfilling all the requirements for my EC2 instance to work



Joba Amunigun
NextWork Student

nextwork.org

EC2 > Elastic IP addresses > Allocate Elastic IP address

Allocate Elastic IP address Info

Elastic IP address settings Info

Public IPv4 address pool

Amazon's pool of IPv4 addresses

Customer-owned pool of IPv4 addresses created from your on-premises network for use with an Outpost. (option disabled because no customer owned pools found) [Learn more](#)

Allocate using an IPv4 IPAM pool (option disabled because no public IPv4 IPAM pools with AWS service as EC2 were found)

Global static IP addresses

AWS Global Accelerator can provide global static IP addresses that are announced worldwide using anycast from AWS edge locations. This can help improve the availability and latency for your user traffic by using the Amazon global network. [Learn more](#)

[Create accelerator](#)

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

[Add new tag](#)

You can add up to 50 more tag

[Cancel](#) [Allocate](#)



Troubleshooting ping issues

To test VPC peering, I ran the command "ping 10.x.xx.xxx" (Which is the "NextWork VPC 2's Private IPv4 address ")

A successful ping test would validate my VPC peering connection - this ping test will not get ANY replies from other EC2 Instance if the connection did not connect the two VPCs. Hence, getting ping replies = connection was set up properly .

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

```
64 bytes from 10.2.11.194: icmp_seq=463 ttl=127 time=0.634 ms
64 bytes from 10.2.11.194: icmp_seq=464 ttl=127 time=0.512 ms
64 bytes from 10.2.11.194: icmp_seq=465 ttl=127 time=0.513 ms
64 bytes from 10.2.11.194: icmp_seq=466 ttl=127 time=0.661 ms
64 bytes from 10.2.11.194: icmp_seq=467 ttl=127 time=0.538 ms
64 bytes from 10.2.11.194: icmp_seq=468 ttl=127 time=0.514 ms
64 bytes from 10.2.11.194: icmp_seq=469 ttl=127 time=0.514 ms
64 bytes from 10.2.11.194: icmp_seq=470 ttl=127 time=0.547 ms
64 bytes from 10.2.11.194: icmp_seq=471 ttl=127 time=0.542 ms
64 bytes from 10.2.11.194: icmp_seq=472 ttl=127 time=0.513 ms
64 bytes from 10.2.11.194: icmp_seq=473 ttl=127 time=0.521 ms
64 bytes from 10.2.11.194: icmp_seq=474 ttl=127 time=0.546 ms
64 bytes from 10.2.11.194: icmp_seq=475 ttl=127 time=0.546 ms
64 bytes from 10.2.11.194: icmp_seq=476 ttl=127 time=0.589 ms
64 bytes from 10.2.11.194: icmp_seq=477 ttl=127 time=0.650 ms
64 bytes from 10.2.11.194: icmp_seq=478 ttl=127 time=0.584 ms
64 bytes from 10.2.11.194: icmp_seq=479 ttl=127 time=0.593 ms
64 bytes from 10.2.11.194: icmp_seq=480 ttl=127 time=0.569 ms
64 bytes from 10.2.11.194: icmp_seq=481 ttl=127 time=0.598 ms
64 bytes from 10.2.11.194: icmp_seq=482 ttl=127 time=0.571 ms
64 bytes from 10.2.11.194: icmp_seq=483 ttl=127 time=0.564 ms
64 bytes from 10.2.11.194: icmp_seq=484 ttl=127 time=0.494 ms
64 bytes from 10.2.11.194: icmp_seq=485 ttl=127 time=0.538 ms
64 bytes from 10.2.11.194: icmp_seq=486 ttl=127 time=0.545 ms
64 bytes from 10.2.11.194: icmp_seq=487 ttl=127 time=0.491 ms
64 bytes from 10.2.11.194: icmp_seq=488 ttl=127 time=0.733 ms
64 bytes from 10.2.11.194: icmp_seq=489 ttl=127 time=0.586 ms
*686 bytes from 10.2.11.194: icmp_seq=490 ttl=127 time=0.498 ms
64 bytes from 10.2.11.194: icmp_seq=491 ttl=127 time=0.505 ms
64 bytes from 10.2.11.194: icmp_seq=492 ttl=127 time=0.492 ms
```

i-09bcd34head5d4ead (Instance - NextWork VPC 1)



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

