



VPC Peering

J

Michael K.

⌚ A VPC peering connection **pcx-0f397645ce8fb08f0 / VPC -Peering-1 < VPC-Peering-2** has been requested. ⚗

pcx-0f397645ce8fb08f0 / VPC -Peering-1 < VPC-Peering-2

Actions ▾

ⓘ Pending acceptance You can accept or reject this peering connection request using the 'Actions' menu. You have until Wednesday, January 7, 2026 at 20:09:44 EST to accept or reject the request, otherwise it expires. ⚗

Details <small>Info</small>	
Requester owner ID 109648734195	Acceptor owner ID 109648734195
Peering connection ID pcx-0f397645ce8fb08f0	Requester VPC vpc-06de5ccb386c46b47 / My-Test-VPC-Peering-1
Status Pending Acceptance by 109648734195	Requester CIDs 10.1.0.0/16
Expiration time Wednesday, January 7, 2026 at 20:09:44 EST	Requester Region N. Virginia (us-east-1)
	VPC Peering connection ARN arn:aws:ec2:us-east-1:109648734195:vpc-peering-connection/pcx-0f397645ce8fb08f0
	Acceptor VPC vpc-0f929ff39f6055b60 / My-Test-VPC-Peering-2
	Acceptor CIDs -
	Acceptor Region N. Virginia (us-east-1)



Michael K.

NextWork Student

nextwork.org

Introducing Today's Project!

What is Amazon VPC?

Amazon VPC (Virtual Private Cloud) is a logically isolated virtual network within AWS where I can launch and manage AWS resources (such as EC2 instances, RDS databases, and load balancers) using my own defined IP address range, subnets, route tables, and network gateways.

How I used Amazon VPC in this project

In today's project, I used Amazon VPC to test VPC Peering between 2 VPCs.

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

One thing I didn't expect in this project was specific configuration of the Network ACL at the subnet Level for the VPC Peering to work.

This project took me...

This project took me 1 hour.

In the first part of my project...

Step 1 - Set up my VPC

In this step, I will create two VPCs with their resources.

Step 2 - Create a Peering Connection

In this step, I will set up a connection link between my VPCs.

Step 3 - Update Route Tables

In this step, I will set up a way for traffic coming from VPC 1 to get to VPC 2 and set up a way for traffic coming from VPC 2 to get to VPC 1.

Step 4 - Launch EC2 Instances

In this step, I will launch an EC2 instance in each VPC, so i can use them to test my VPC peering connection later.

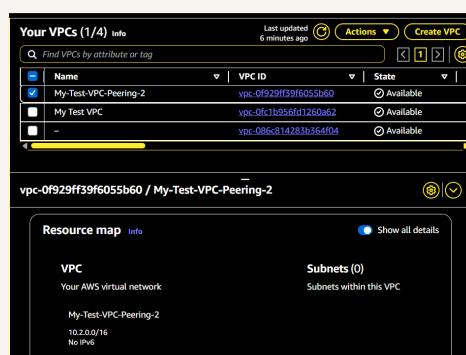
Multi-VPC Architecture

I started my project by launching 2 VPCs with one subnet in each.

The CIDR blocks for VPCs 1 and 2 are unique. They have to be unique in order to avoid resources in each VPC to overlap because other their identical IP addresses.

I also launched 2 EC2 instances

I didn't set up key pairs for these EC2 instances because the key pairs serve to give direct access to the Instances through the Internet using SSH. In this case I don't need to do that since I'll be connecting to my instance with Instance Connect which will generate it automatically and temporarily for me to be able to connect.

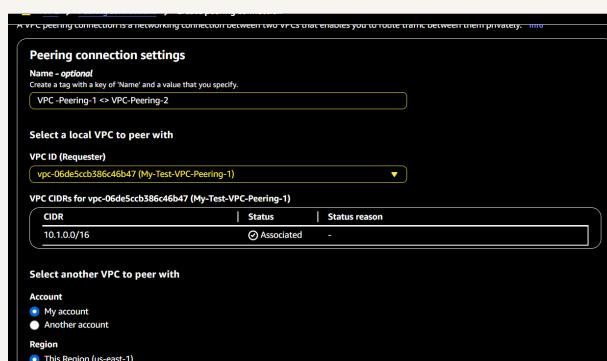


VPC Peering

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.

The difference between a Requester and an Acceptor in a peering connection is that the Requester is the VPC that initiates a peering connection by sending an invitation to the Acceptor.



Updating route tables

After accepting a peering connection, my VPCs' route tables need to be updated inorder to create a route that each VPC will use to connect to the other.

My VPCs' new routes have a destination of each other. The routes' target was the VPC Peering created earlier.

The screenshot shows the AWS Route Tables interface. At the top, a success message says "Updated routes for rtb-0c197c3fcb050472b / VPC-Peering-2-route successfully". Below it, the route table ID is rtb-0c197c3fcb050472b. The main configuration includes:

Route table ID rtb-0c197c3fcb050472b	Main Yes	Explicit subnet associations -	Edge associations -
VPC vpc-0f929ff39f6055b60 My-Test-VPC-Peering-2	Owner ID 109648734195		

Below the main configuration, there are tabs for Routes, Subnet associations, Edge associations, Route propagation, and Tags. The Routes tab is selected, showing two routes:

Destination	Target	Status	Propagated	Route Origin
10.1.0.0/16	peering-0f397645...	Active	No	Create Route
10.2.0.0/16	local	Active	No	Create Route Table

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 my first Instance again.

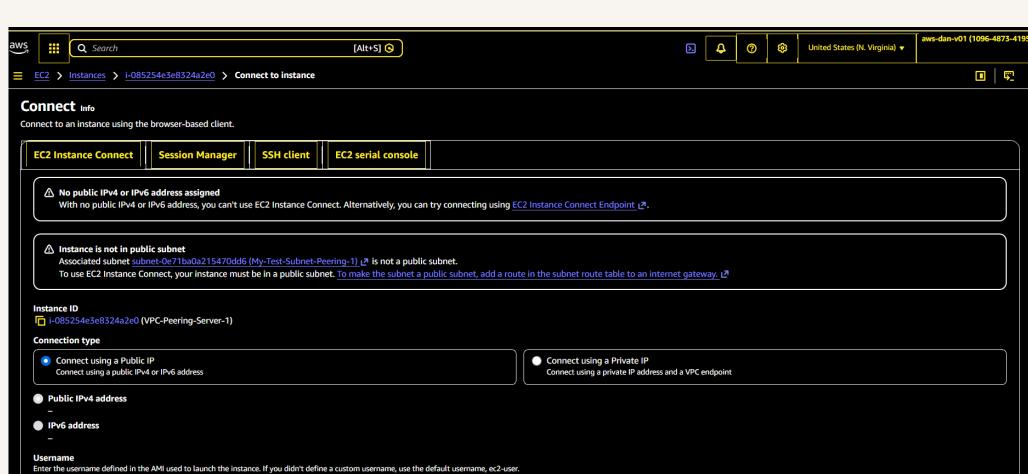
Step 7 - Test VPC Peering

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

Troubleshooting Instance Connect

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

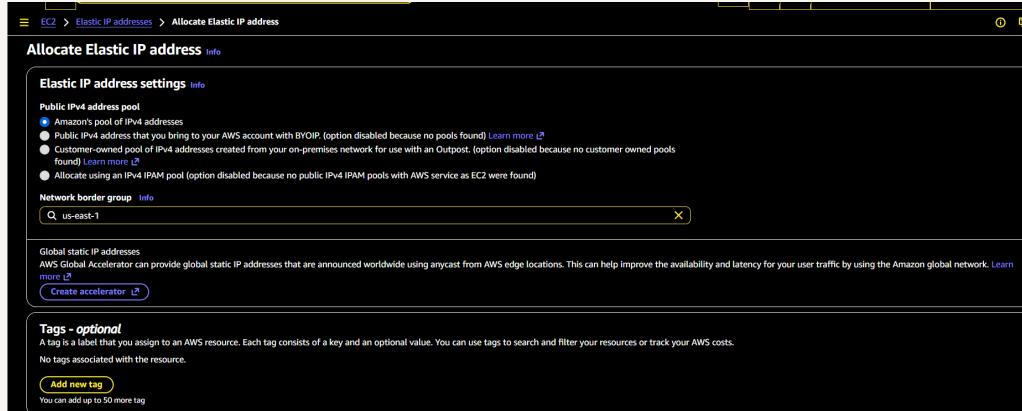
I was stopped from using EC2 Instance Connect because of the error Indicating the absence a public IP Address in my server.



Elastic IP addresses

To resolve this error, I set up Elastic IP addresses. Elastic IP addresses are static IP Addresses that can be allocated to AWS resources (like servers). These Addresses don't change even after the Instance has stopped or restarted.

Associating an Elastic IP address resolved the error because the Instance connect could now have a public IP Address to connect to the server.



Troubleshooting ping issues

To test VPC peering, I ran the command <<ping "Private IP of the second server">>

A successful ping test would validate my VPC peering connection because it will confirm the communication between both servers. Since the communication comes from resources from 2 different VPCs then the VPC Peering is validated.

I had to update my second EC2 instance's security group by adding an ICMP Inbound rule which will allow the Ping to reach the Instance or server.

```
ec2-user@ip-10-1-0-200 ~]$ ping 10.2.0.214 10.2.0.214
PING 10.2.0.214 (10.2.0.214) 56(84) bytes of data. bytes of data.
64 bytes from 10.2.0.214: icmp_seq=1 ttl=127 time=0.472 ms time=0.472 ms
64 bytes from 10.2.0.214: icmp_seq=2 ttl=127 time=0.476 ms time=0.476 ms
64 bytes from 10.2.0.214: icmp_seq=3 ttl=127 time=0.466 ms time=0.466 ms
^C
--- 10.2.0.214 ping statistics ---atistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2076msloss, time 2076ms
rtt min/avg/max/mdev = 0.466/0.471/0.476/0.004 ms6/0.004 ms
[ec2-user@ip-10-1-0-200 ~]$ 0 ~] $
```



nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

