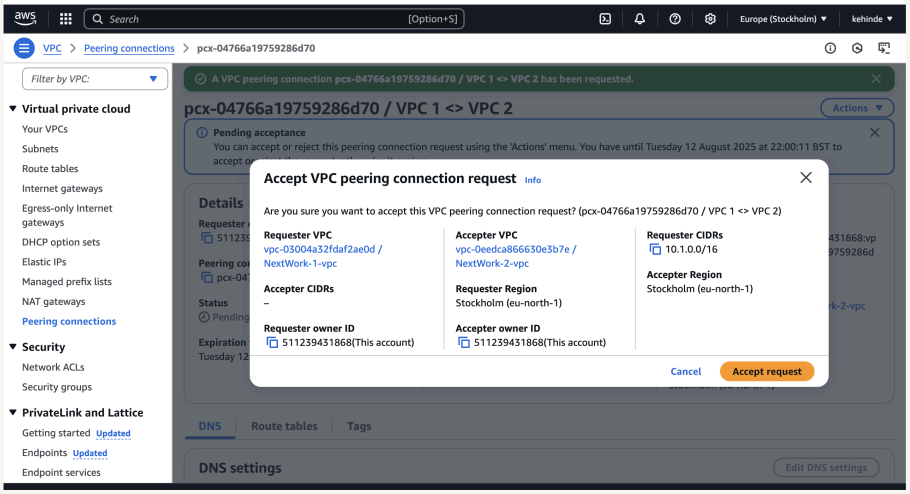
nextwork.org

VPC Peering

Kehinde Abiuwa



Introducing Today's Project!

What is Amazon VPC?

Amazon VPC is a Virtual Private Cloud service that lets you create your own isolated virtual network within the AWS cloud. It allows you to launch AWS resources—like EC2 instances—inside a customizable network that you control, including IP address ranges, subnets, route tables, and security settings. It is useful because it gives you full control over your network environment, enhances security by isolating resources, enables you to configure network traffic rules, and supports building scalable and secure cloud architectures tailored to your specific needs

How I used Amazon VPC in this project

I used amazon in todays project to test connection between 2 vpc's

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

nothing really

Kehinde Abiuwa
NextWork Student

This project took me...

It took me 40minutes to complete this project

In the first part of my project...

Step 1 - Set up my VPC

In this step, I am going to Create two VPCs from scratch and then Use the visual VPC resource map to create my VPCs quick.

Step 2 - Create a Peering Connection

In this step, I am going to bridge my two vpc's together by creating a vpc peering connection

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 as well as Set up a way for traffic coming from VPC 2 to get to VPC 1.

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 your VPC peering connection later.

Multi-VPC Architecture

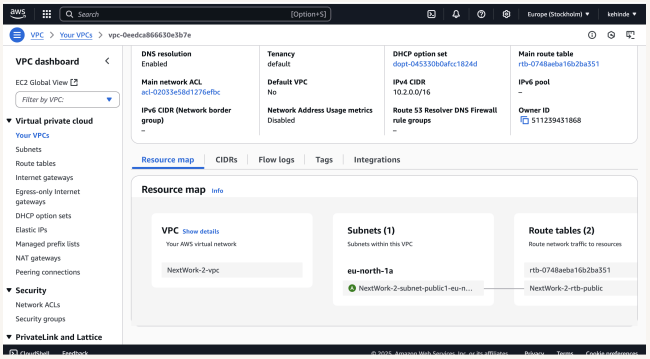
I started my project by launching a vpc i called nextwork-1 with one public subnet, and then i launched my second vpc called nextwork-2 with one public subnet as well

Each VPC must have a 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 as EC2 Instance Connect allows me to securely access the instances without needing to manage my own SSH key pairs. Instead, AWS automatically generates and manages temporary keys behind the scenes during each connection, making the process simpler and more secure. Since I’ve already practiced creating and using key pairs in previous projects, and this project focuses on using EC2 Instance Connect, there was no need to repeat that step

Kehinde Abiuwa
NextWork Student



Kehinde Abiuwa
NextWork Student

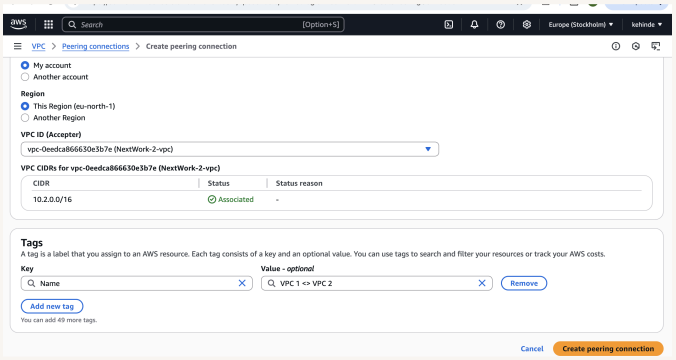
VPC Peering

A VPC peering connection is a network connection between two Virtual Private Clouds (VPCs) that allows them to communicate with each other privately using IP addresses as if they were within the same network. It enables instances in different VPCs—either within the same AWS account or across different accounts—to send traffic to each other without using the public internet, VPNs, or gateways. VPC peering is useful for sharing resources securely across environments while maintaining low latency and high throughput.

VPCs would use peering connections to securely and privately communicate with each other across different virtual networks without needing to route traffic over the public internet. This is especially useful for connecting microservices, environments (like dev and prod), or VPCs across AWS accounts while maintaining low latency, high bandwidth, and enhanced security. Peering connections help simplify architecture by allowing direct routing between VPCs using private IP addresses, which is ideal for sharing data or services across isolated networks

The difference between a Requester and an Accepter in a peering connection is that the Requester is the VPC that initiates the peering connection, while the Accepter is the VPC that must approve or accept the request for the connection to become active. Both sides can be in the same AWS account or in different accounts, but no traffic will flow between the VPCs until the Accepter explicitly accepts the connection request. Once accepted, the peering connection is established, and both VPCs can configure route tables and security groups to allow communication.

Kehinde Abiuwa
NextWork Student

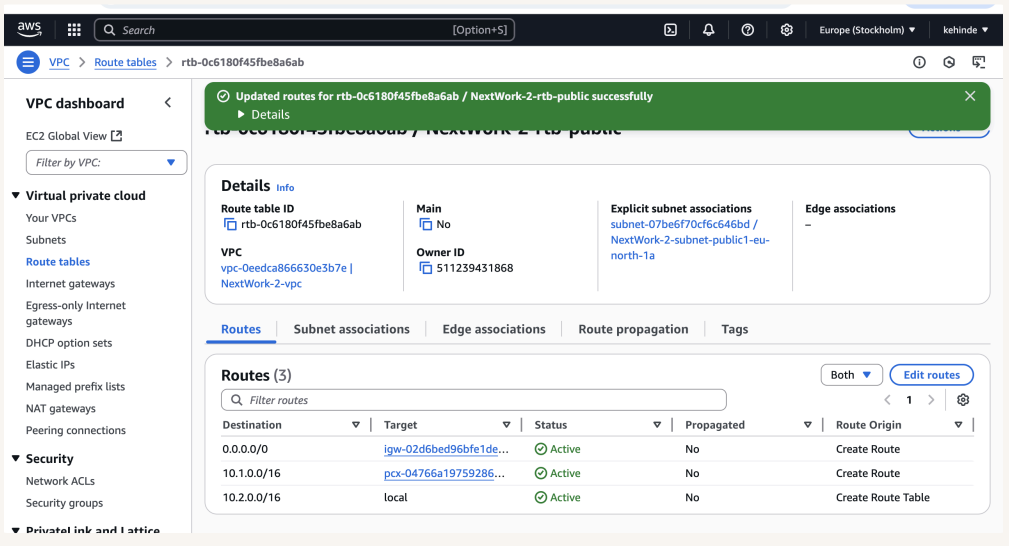


Kehinde Abiuwa
NextWork Student

Updating route tables

After accepting a peering connection, my VPCs' route tables need to be updated because they don't automatically know how to reach each other's IP address ranges. By adding a new route, I'm explicitly telling each VPC how to send traffic to the other VPC through the peering connection. Without updating the route tables, even though the peering connection exists, the instances in each VPC wouldn't be able to communicate because the network has no defined path to the remote VPC's CIDR block.

My VPCs' new routes have a destination of the cidr block of the receiving vpc . The routes' target was the peering connection i created



In the second part of my project...

Step 5 - Use EC2 Instance Connect

In this step, I am going to Use EC2 Instance Connect to connect to your first EC2 instance & Fix a connection error!

Step 6 - Connect to EC2 Instance 1

In this step, I am going to Use EC2 Instance Connect to connect to Instance 1 once again

Step 7 - Test VPC Peering

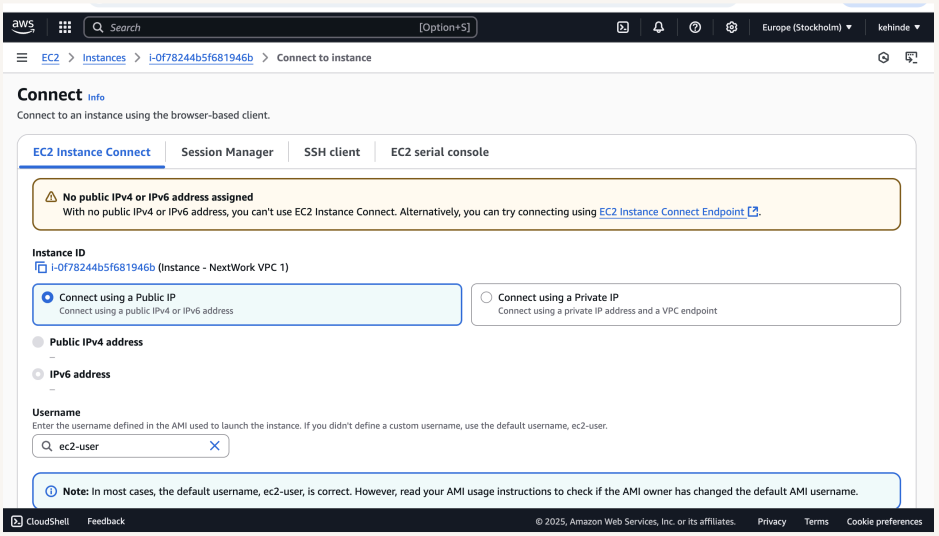
In this step I am going to Get Instance 1 to send test messages to Instance 2 & Solve connection errors until Instance 2 is able to send messages back.

Kehinde Abiuwa
NextWork Student

Troubleshooting Instance Connect

Next, I used EC2 Instance Connect to connect to my vpc 1

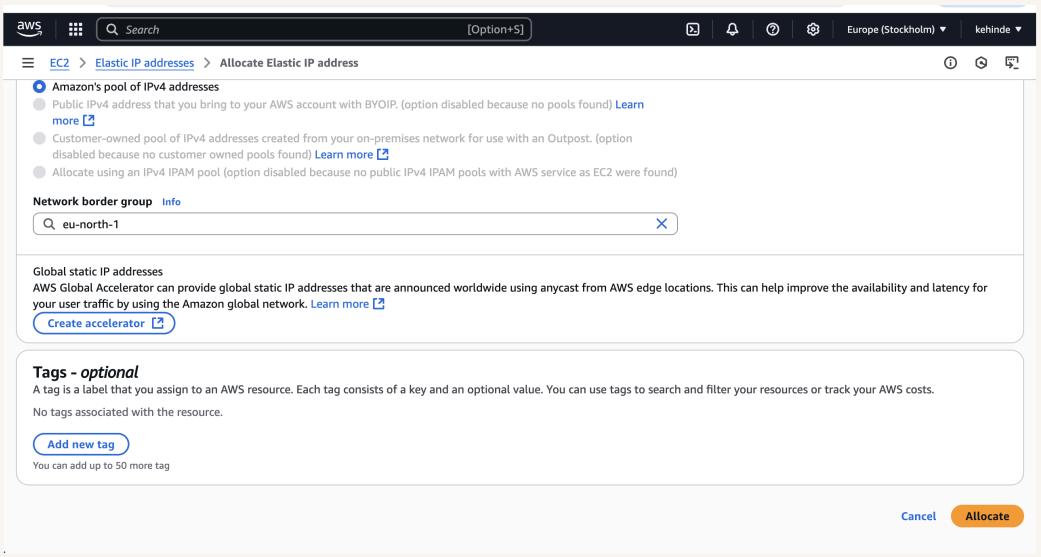
I was stopped from using EC2 Instance Connect as No public IPv4 address was assigned to my vpc 1 instance



Elastic IP addresses

To resolve this error, I set up Elastic IP addresses. Elastic IP addresses are static, public IPv4 addresses provided by AWS that you can allocate to your AWS account and associate with your EC2 instances. Unlike regular public IP addresses, which can change when an instance is stopped and restarted, Elastic IPs remain the same, ensuring consistent access to your instance. This is especially useful for services or configurations that require a stable IP address for external communication or firewall rules.

Associating an Elastic IP address resolved the error because it provided a stable and persistent public IP address for my EC2 instance, ensuring that I could reliably connect to it even after stopping and restarting the instance. Without an Elastic IP, the public IP assigned to an EC2 instance can change, which may lead to connection issues or misconfigured network settings. By using an Elastic IP, I ensured consistent access and avoided errors related to dynamic IP changes.

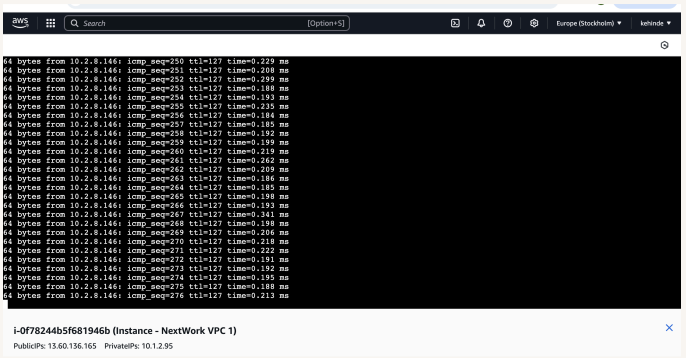

aws
Search
[Option+S]
Europe (Stockholm)
kehinde
EC2
Elastic IP addresses
Allocate Elastic IP address
Amazon's pool of IPv4 addresses
Public IPv4 address that you bring to your AWS account with BYOIP. (option disabled because no pools found) Learn more
Customer-owned pool of IPv4 addresses created from 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)
Network border group Info
eu-north-1
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 Ping command

A successful ping test would validate my VPC peering connection because it confirms that instances in the two VPCs can reach each other over the private network using their internal IP addresses. This means the peering connection is active, the route tables are correctly configured to direct traffic through the peering link, and security group or network ACL rules allow ICMP (ping) traffic. If the ping fails, it could indicate issues with routing, firewall rules, or that the peering connection isn't properly established.

I had to update my second EC2 instance's security group because I noticed there was no inbound rule allowing ICMP traffic. I added a new rule that allows inbound ICMP traffic so that the instance could respond to ping requests, which is essential for validating the VPC peering connection



The place to learn & showcase your skills

Check out nextwork.org for more projects