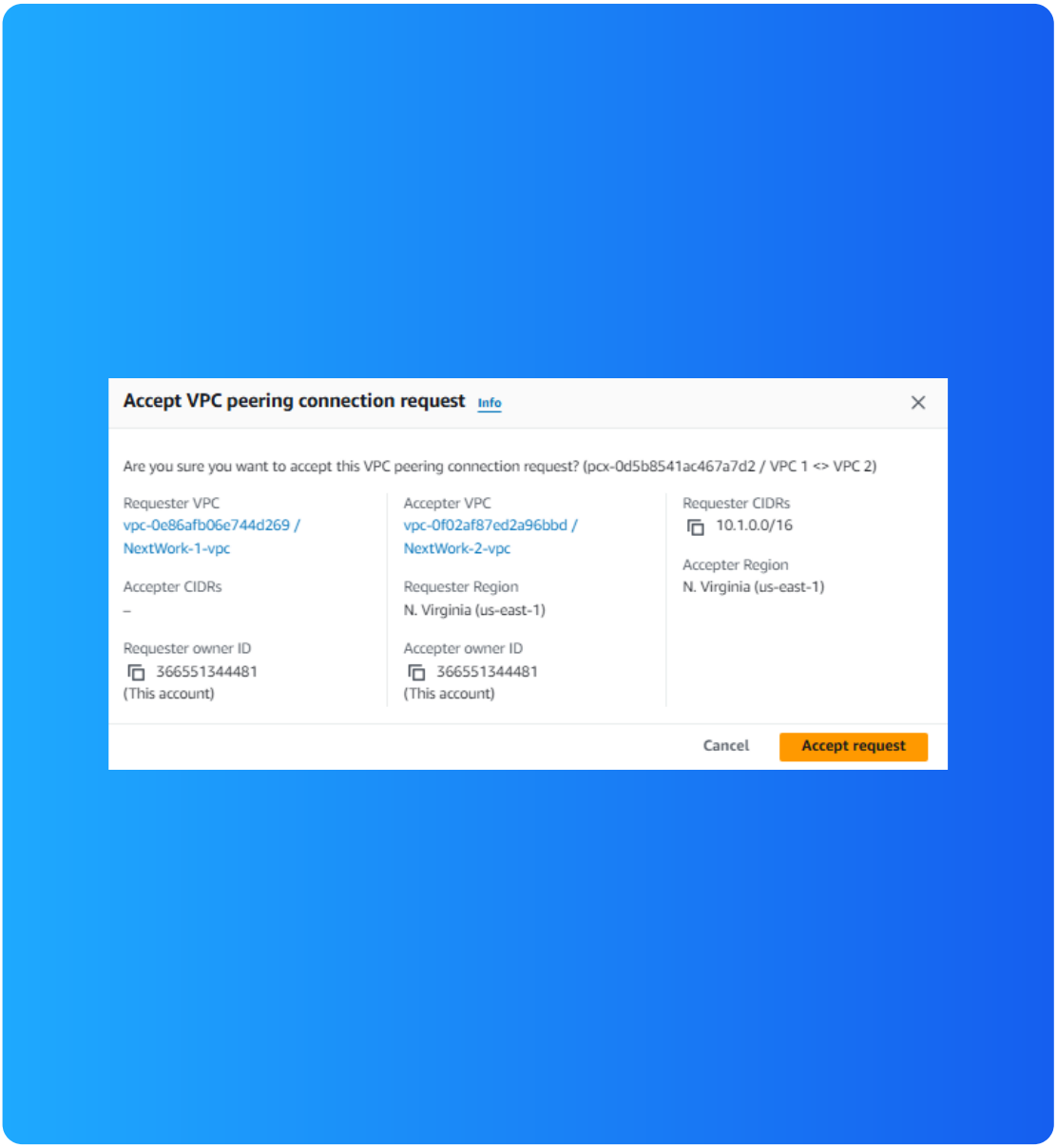
NextWork.org

VPC Peering

Gloria



G
Gloria
NextWork Student

Introducing Today's Project!

What is Amazon VPC?

Amazon VPC (Virtual Private Cloud) is a secure, isolated network in AWS where you can launch resources like EC2 instances. It’s useful because it gives control over networking, such as IP ranges, subnets and security.

How I used Amazon VPC in this project

I used Amazon VPC in today's project to create two isolated networking environments, set up a peering connection between them, configure route tables for communication, and launch EC2 instances to test the connectivity between the VPCs.

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

One thing I didn't expect in this project was the need to handle multiple setup configurations, such as resolving connection errors and updating security group rules, which added unexpected complexity to testing the VPC peering connection.

This project took me...

This project took me about 40 minutes, primarily due to the time spent on setup, resolving connection errors, and configuring the VPCs and security group rules.

G
Gloria
NextWork Student

In the first part of my project...

Step 1 - Set up my VPC

I am setting up two Virtual Private Clouds (VPCs) from scratch using the VPC wizard. By using the visual VPC resource map, I will create and configure the VPCs quickly to establish isolated networking environments.

Step 2 - Create a Peering Connection

In this step, I am creating a peering connection to link my two VPCs. This will allow resources in each VPC to communicate securely with each other by establishing a direct network connection between them.

Step 3 - Update Route Tables

In this step, I am configuring the route tables for both VPCs to enable traffic to flow between them. This ensures that resources in VPC 1 can communicate with resources in VPC 2 and vice versa.

Step 4 - Launch EC2 Instances

In this step, I am launching one EC2 instance in each VPC. These instances will be used later to test the connectivity and functionality of the VPC peering connection.

Gloria
NextWork Student

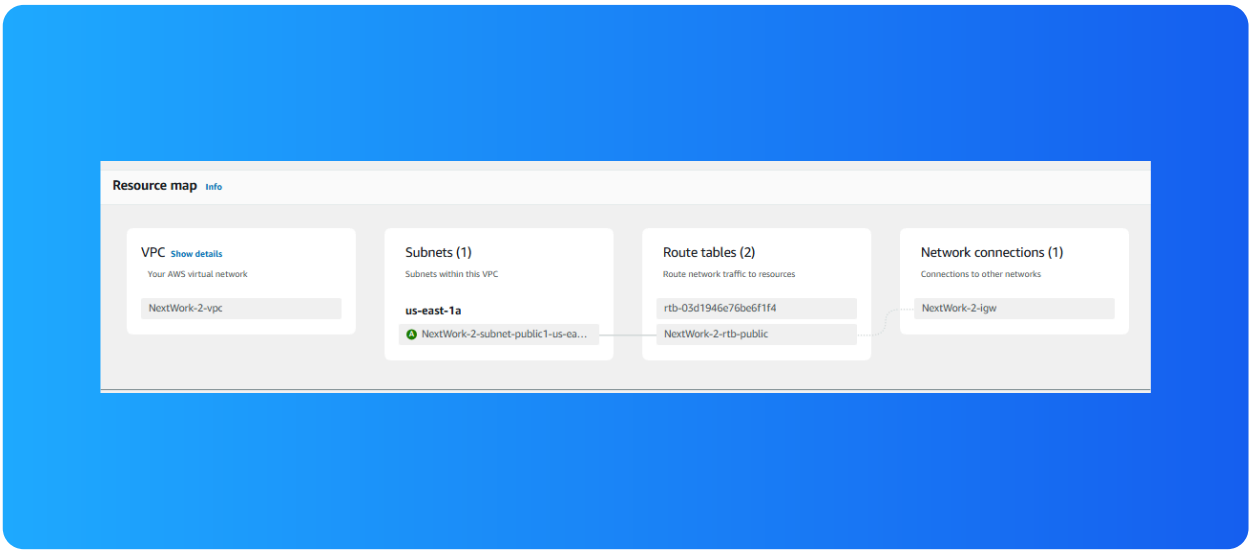
Multi-VPC Architecture

I launched a second VPC named "NextWork-2" with a unique IPv4 CIDR block of 10.2.0.0/16. It includes 1 public subnet in a single Availability Zone with no private subnets, NAT gateways, or VPC endpoints.

The CIDR blocks for VPCs 1 and 2 are 10.1.0.0/16 and 10.2.0.0/16. They have to be unique because overlapping IP ranges can lead to routing conflicts. I did this to ensure smooth communication between resources in each VPC.

I also launched 2 EC2 instances

I didn't set up key pairs for these EC2 instances as AWS EC2 Instance Connect manages key pairs automatically. Since I've already practiced setting up key pairs in previous projects, it wasn't necessary to repeat the process here.



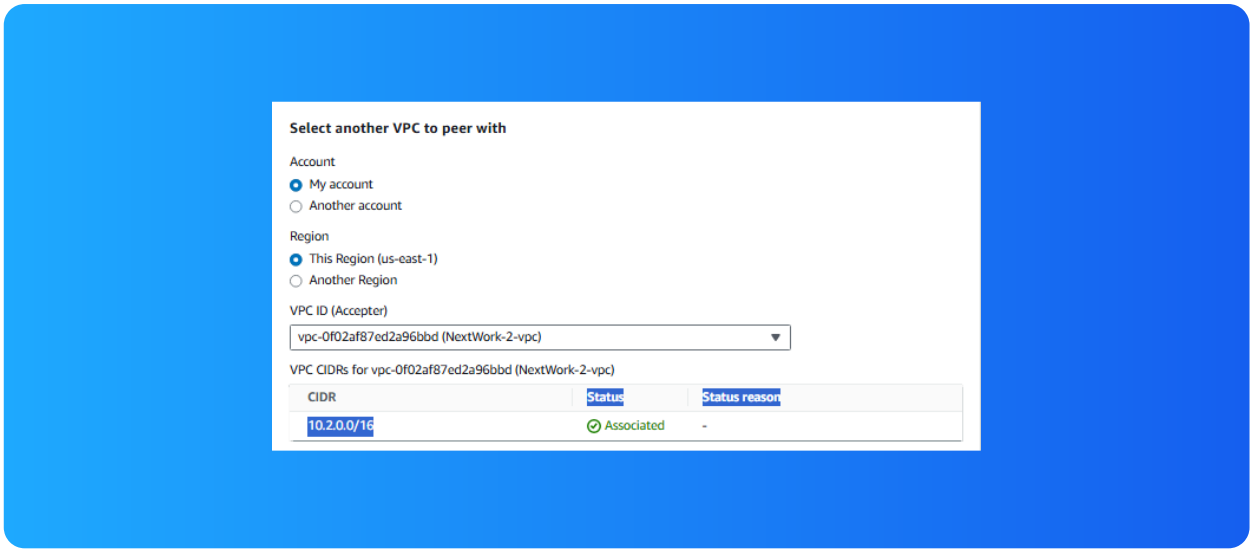
Gloria
NextWork Student

VPC Peering

A VPC peering connection is a networking link between two VPCs that enables them to communicate directly as if they were part of the same network, without using the public internet or gateways.

VPCs would use peering connections to enable direct communication between them, ensuring secure, low-latency networking without relying on the public internet, and allowing seamless sharing of resources across VPCs.

The difference between a Requester and an Accepter in a peering connection is that the Requester starts the connection request, while the Accepter approves it. Both roles work together to create a link allowing communication between their networks.

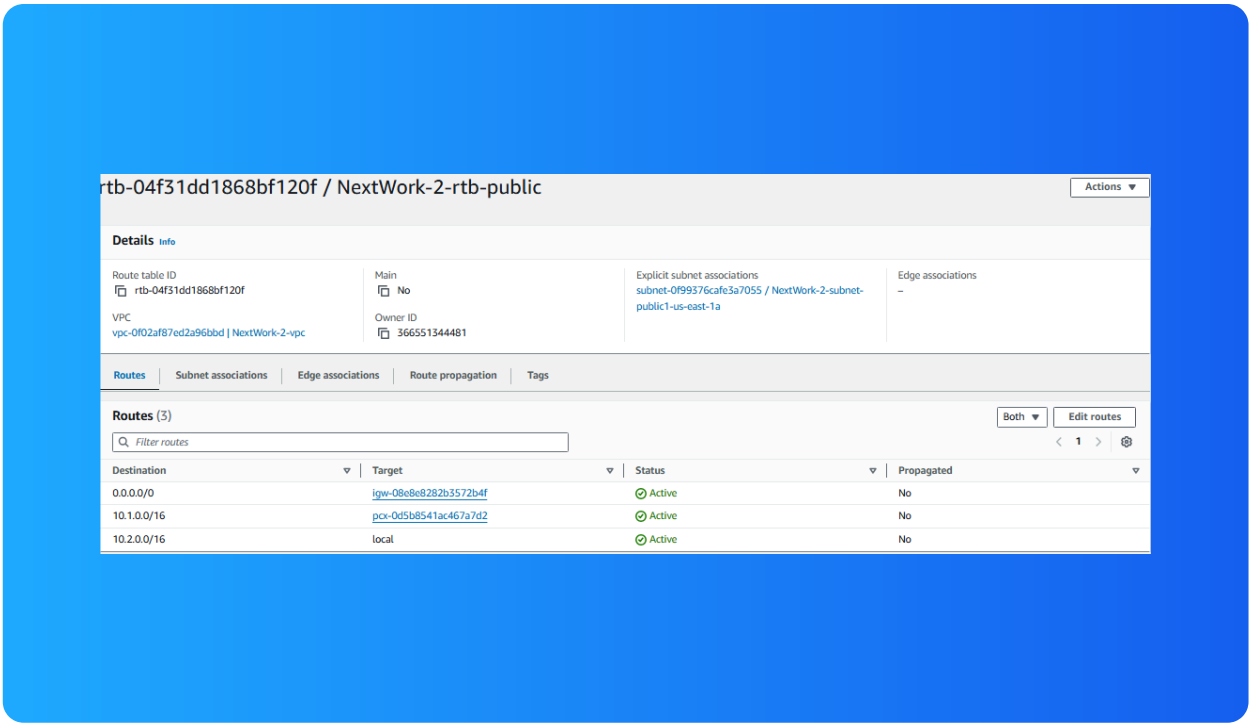


Gloria
NextWork Student

Updating route tables

After accepting a peering connection, my VPCs' route tables need to be updated because this allows traffic to route correctly between the two VPCs, enabling communication between their resources over the peering connection.

My VPCs' new routes have a destination of 10.2.0.0/16 for VPC 1 and 10.1.0.0/16 for VPC 2. The routes' target was the peering connection established between the two VPCs.



G
Gloria
NextWork Student

In the second part of my project...

Step 5 - Use EC2 Instance Connect

In this step, I am connecting to my first EC2 instance using EC2 Instance Connect. Once connected, I will test the communication between this instance and the second EC2 instance to verify the VPC peering connection, troubleshooting any connection

Step 6 - Connect to EC2 Instance 1

✍ In this step, I am using EC2 Instance Connect to reconnect to Instance 1 and troubleshoot another connection error that might arise, ensuring the instance is properly configured for communication.

Step 7 - Test VPC Peering

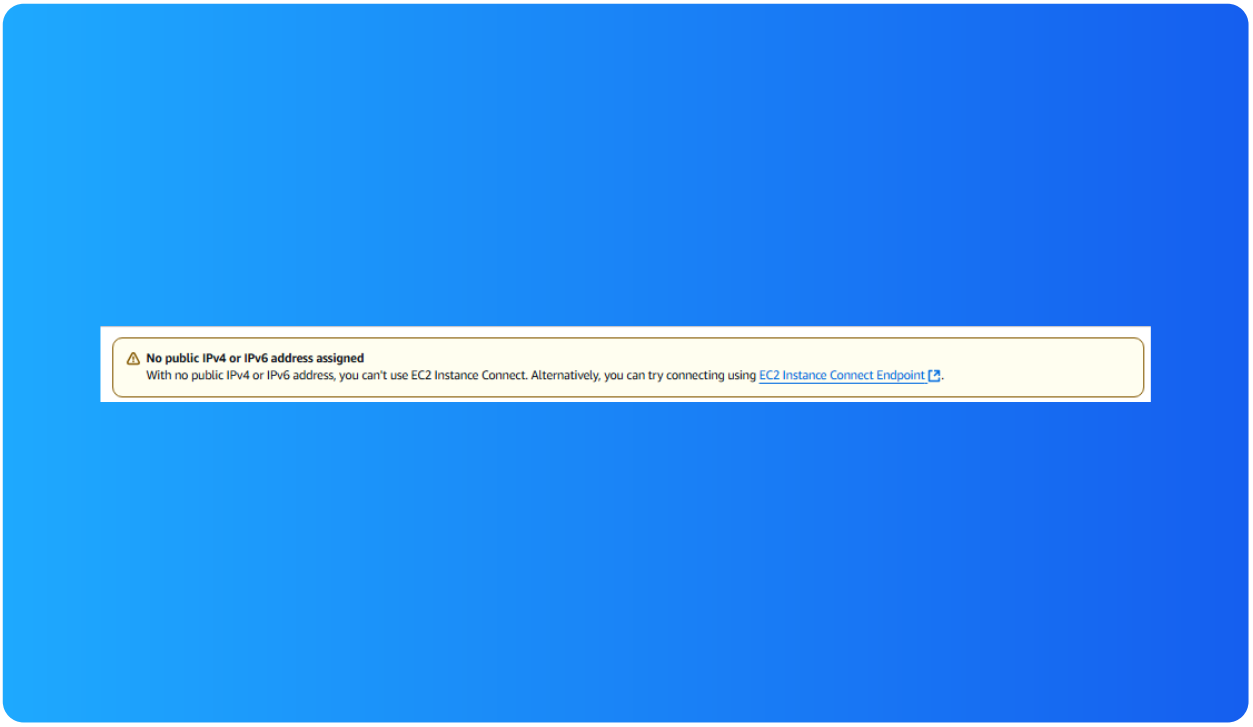
✍ In this step, I am testing the VPC peering connection by sending test messages from Instance 1 in VPC 1 to Instance 2 in VPC 2. I will troubleshoot any connection errors to ensure successful two-way communication between the instances.

Gloria
NextWork Student

Troubleshooting Instance Connect

Next, I used EC2 Instance Connect to easily access my EC2 instance without the need to manage SSH keys, allowing me to quickly test connectivity between the two instances set up in my VPCs.

I was stopped from using EC2 Instance Connect because I disabled the auto-assign public IP address option when creating the instance, leaving it without an allocated IP address required for the connection.

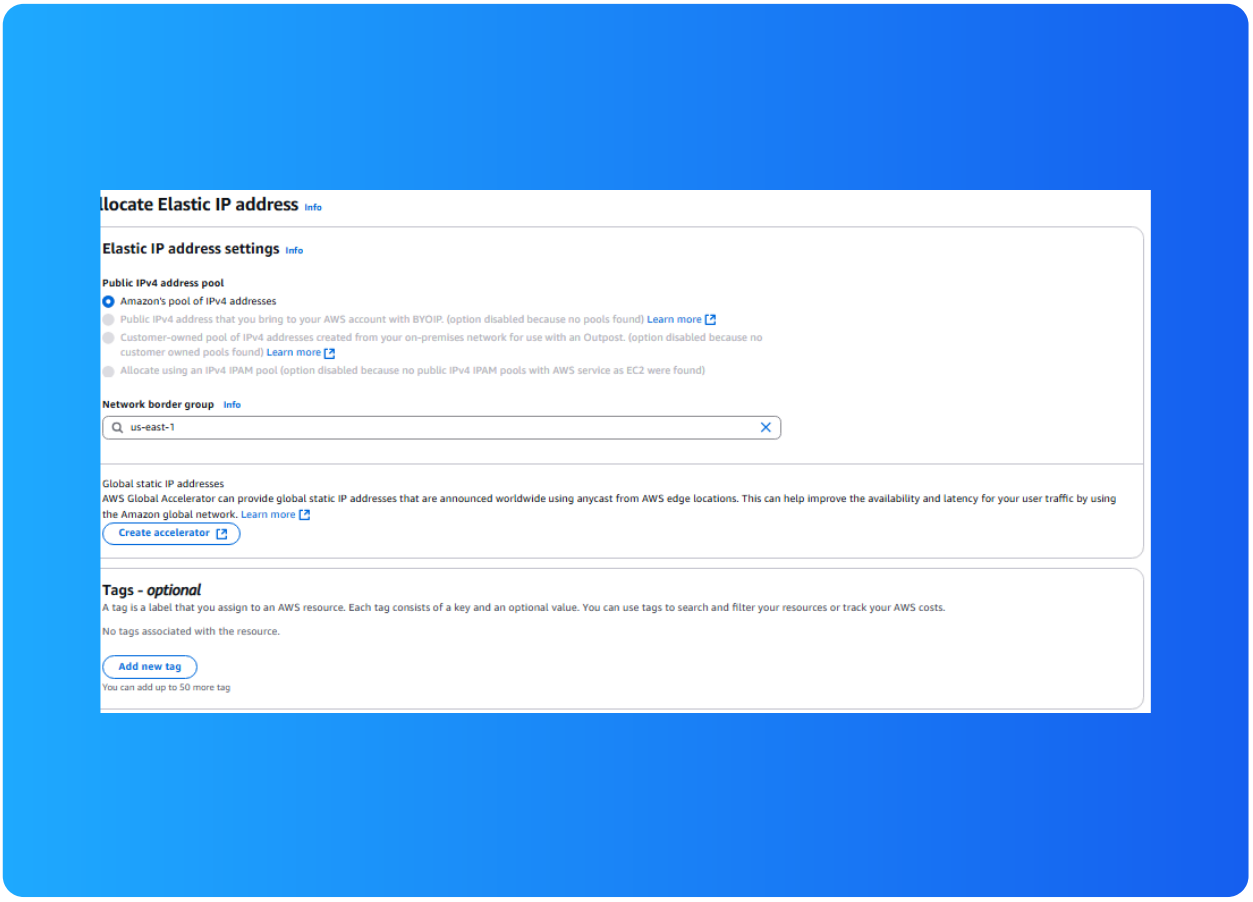


Gloria
NextWork Student

Elastic IP addresses

To resolve this error, I set up Elastic IP addresses. Elastic IP addresses are static, public IPv4 addresses that can be associated with an instance, allowing it to maintain the same IP address even if the instance is stopped, restarted, or replaced.

Associating an Elastic IP address resolved the error because it provided the instance with a static, public IP address, enabling EC2 Instance Connect to establish a connection to the instance successfully.



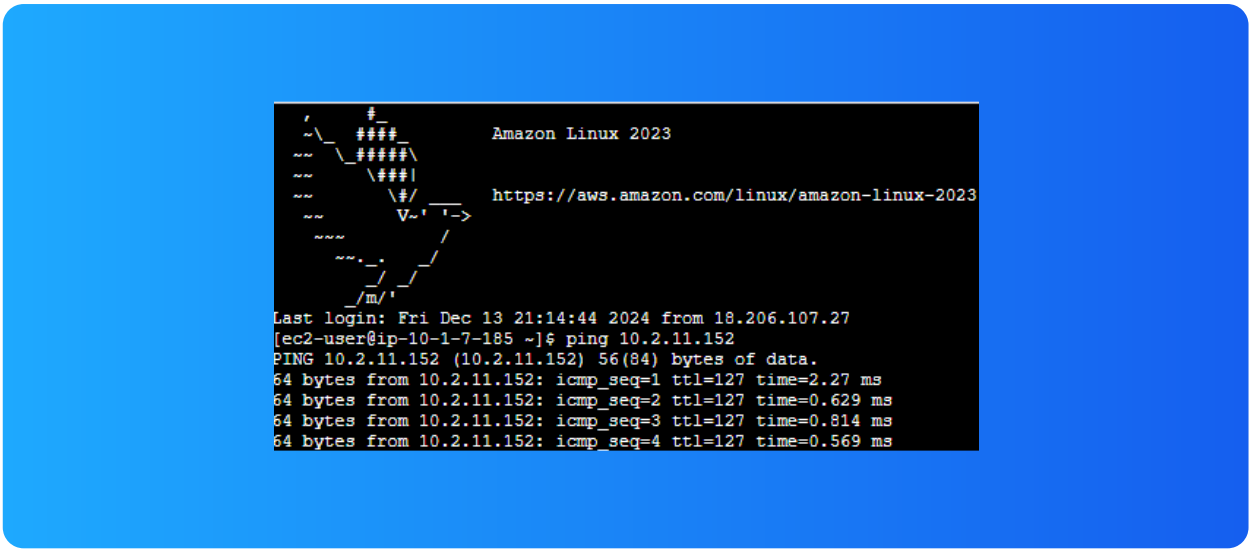
Gloria
NextWork Student

Troubleshooting ping issues

To test VPC peering, I ran the command `ping 10.2.11.152`(which is the private instance of my instance 2) from Instance 1 to check if it could successfully communicate with Instance 2 using its private IP address.

A successful ping test would validate my VPC peering connection because it confirms that the two instances in separate VPCs can communicate using their private IP addresses, proving the peering connection and routing tables are configured correctly.

I had to update my second EC2 instance's security group because it wasn't allowing ICMP traffic needed for the ping test. I added a new rule that allowed inbound ICMP traffic from the private IP range of the first VPC.



Everyone should be in a job they love.

Check out nextwork.org for more projects