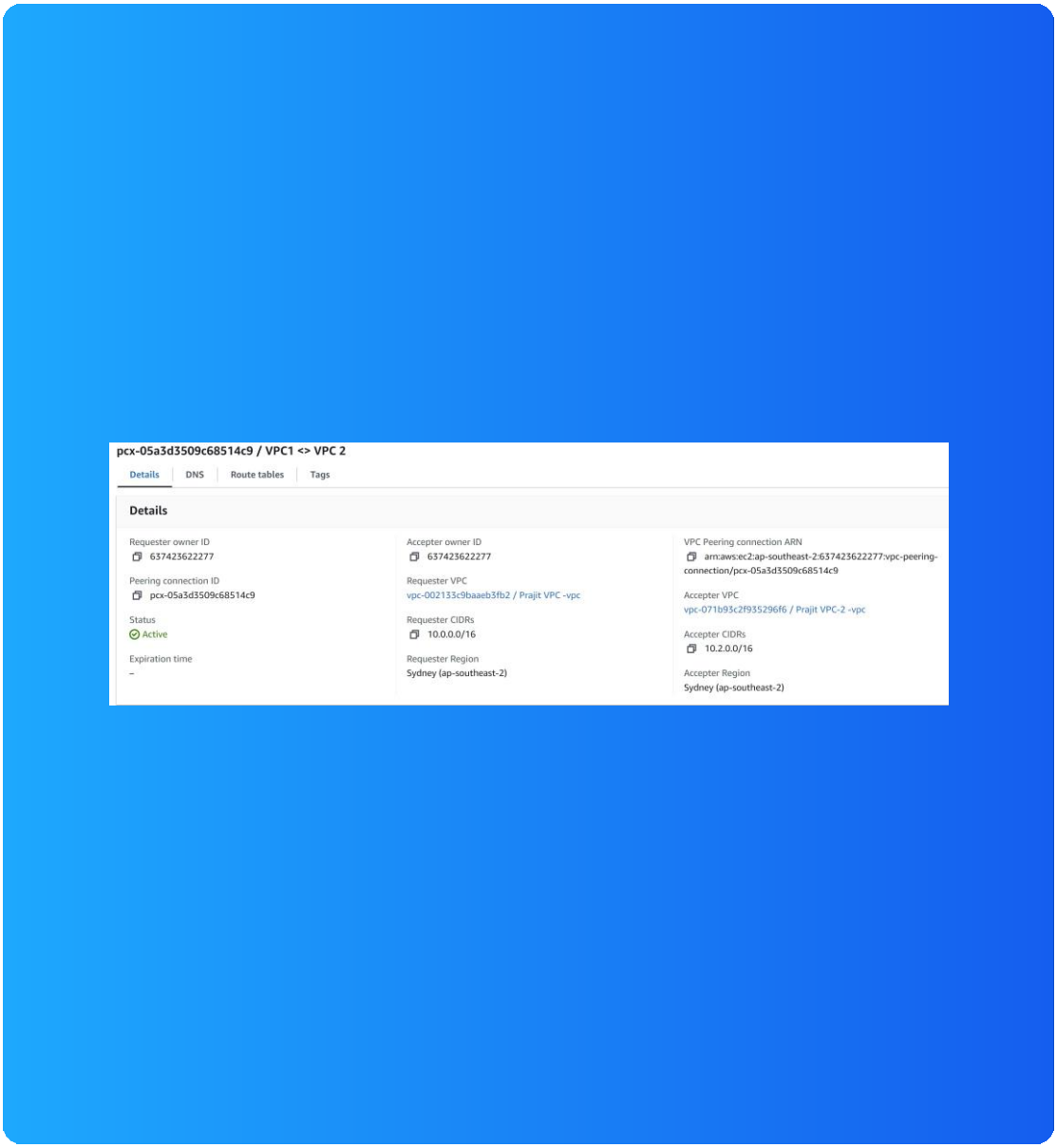
NextWork.org

VPC Peering

Prajit Venkatachalam

https://www.linkedin.com/in/prajit-venkatachalam-435b2a150/



V

Prajit Venkatachalam
NextWork Student

Introducing Today's Project!

What is Amazon VPC?

Amazon VPC (Virtual Private Cloud) exists within an AWS region and is used to build a private and secure connection for resources in the subnets. Through an internet gateway, these resources and users can access internet to communicate each other.

How I used Amazon VPC in this project

In today's project, I used Amazon VPC to set up a multi-VPC architecture (VPC-1 & VPC-2), established a peering connection to connect the VPCs, and updated the security groups of two EC2 instances with SSH and ICMP traffic rules to test and validate.

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

One thing I didn't expect in this project was the need to assign public IPv4 addresses to my EC2 instances to make EC2 Instance Connect work. I fixed this by creating an Elastic IPv4 address and associating it with my VPC-1 EC2 instance.

This project took me...

This project took me 3 hours to complete, include writing the report.

V
Prajit Venkatachalam
NextWork Student

In the first part of my project...

Step 1 Set up my VPC

In this step, I am using the VPC resource map/launch wizard to create two VPCs and their components in few minutes.

Step 2 Create a Peering Connection

In this step, I set up a VPC peering connection, which is a VPC component designed to directly connect two VPCs together.

Step 3 Update Route Tables

In this step, I will update both the VPC's public route table for 'peering connection' with correct destination address to set up communication i.e. CIDR block of VPC-2 in VPC-1 public route table, and CIDR block of VPC-1 in VPC-2 public route table.

Step 4 Launch EC2 Instances

In this step, I am launching EC2 instances in VPC-1 and VPC-2, so that I can connect to my instances later and test the VPC connectivity.

Prajit Venkatachalam
NextWork Student

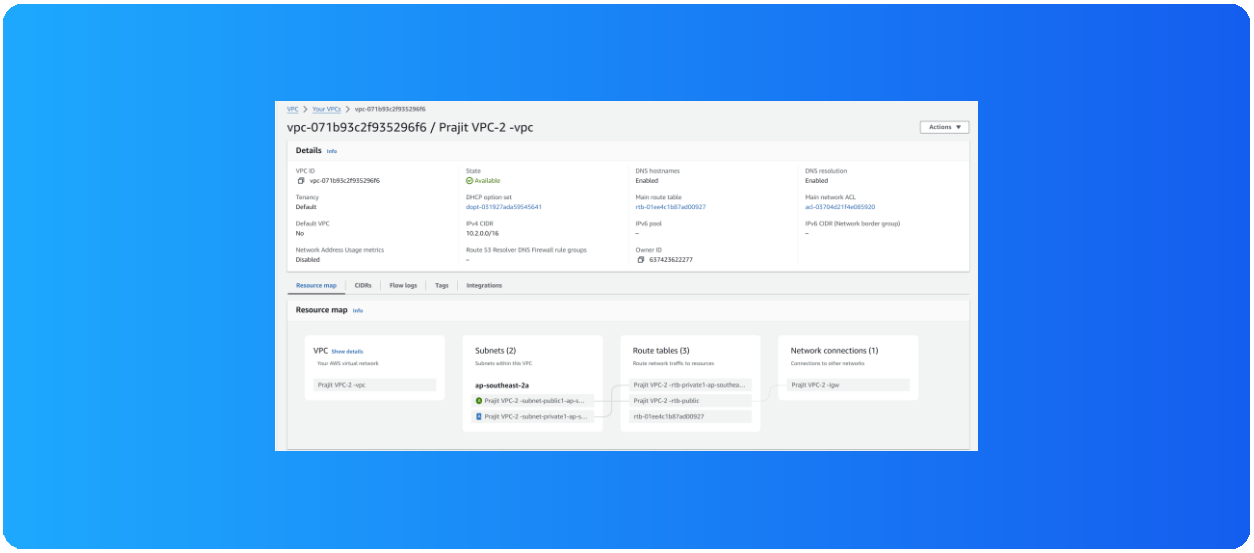
Multi-VPC Architecture

I started my project by launching two VPCs. They have unique CIDR blocks, and each of them has one public subnet.

The CIDR blocks for VPCs 1 and 2 are 10.1.0.0/16 and 10.2.0.0/16. They need to be unique because, once a VPC peering connection is set up, route tables require distinct addresses for proper routing between the VPCs.

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 directly connect with our EC2 instance later in this project, which handles key pair creation and management for us.



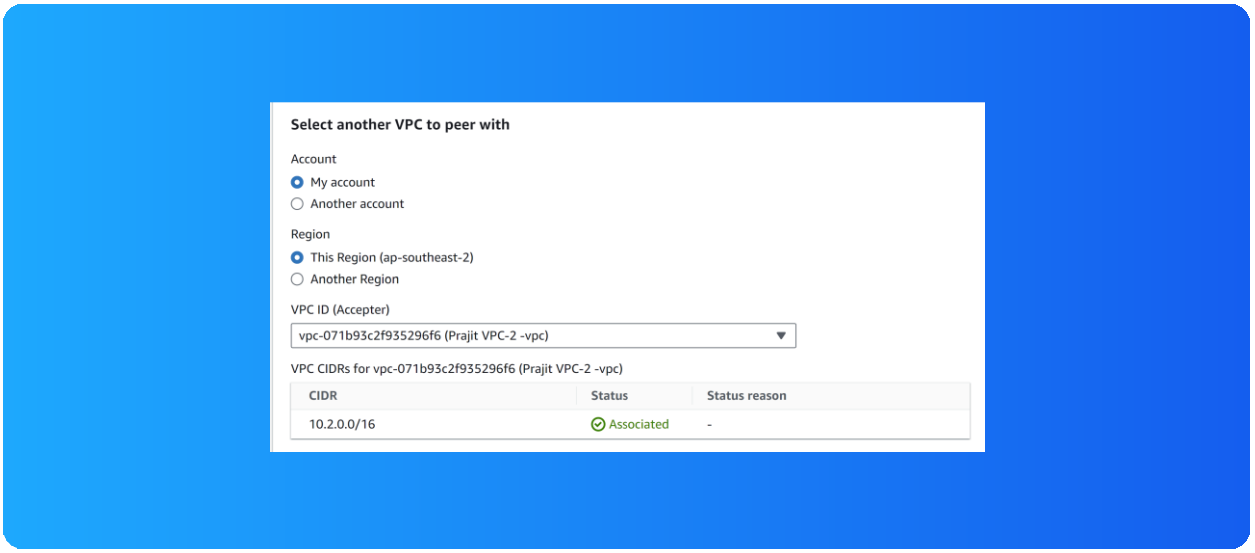
Prajit Venkatachalam
NextWork Student

VPC Peering

A VPC peering connection is a setup between two VPCs to connect and achieve data transfer securely over private IPs. The route tables in both VPCs should be updated with the correct destination and target to enable communication between the VPCs.

VPCs would use peering connections to establish connection between VPCs in separate AWS accounts. This allows businesses to securely collaborate and share resources across accounts without exposing them to the public internet.

The difference between a Requester and an Accepter in a peering connection is that a VPC that initiates a peering connection and sends invitation to another VPC is called requester, and VPC that receives invitation and accepts it is called accepter.

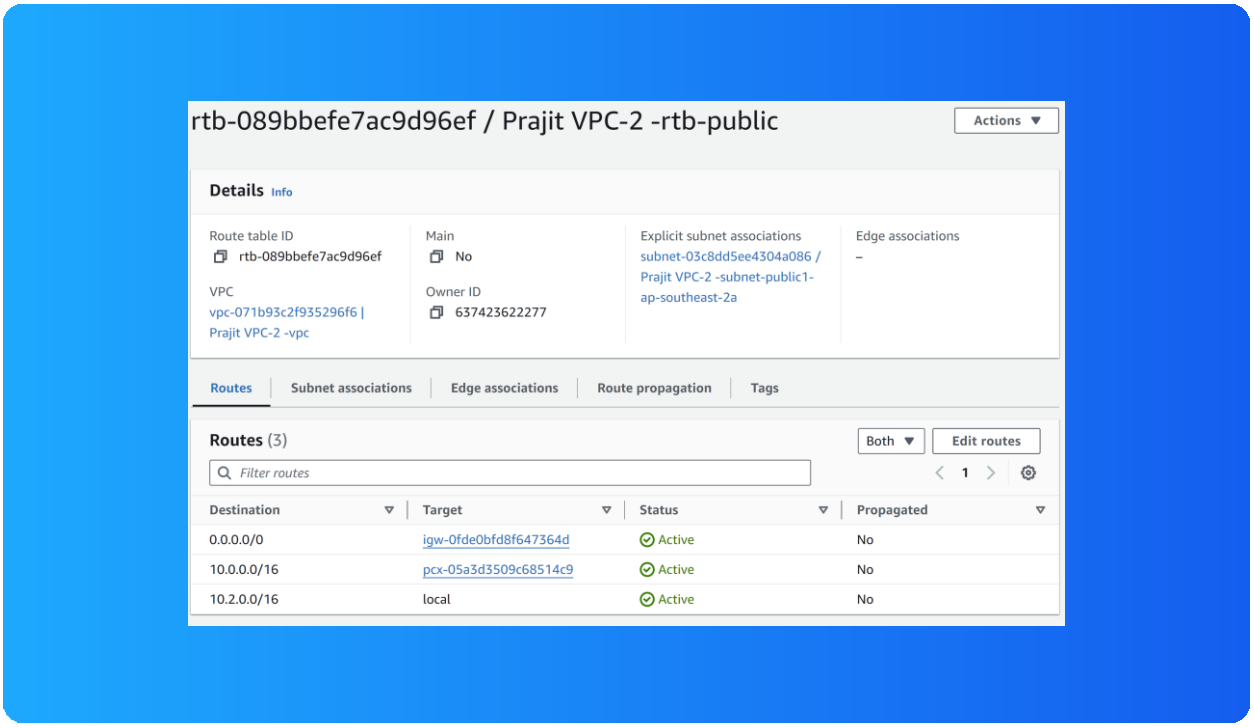


Prajit Venkatachalam
NextWork Student

Updating route tables

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 the other VPC.

My VPCs' new routes have the CIDR block of VPC-2 as the destination in the public route table of VPC-1, and the CIDR block of VPC-1 in the public route table of VPC-2. The routes' target is the peering connection that I updated.



In the second part of my project...

Step 5 Use EC2 Instance Connect

I am using EC2 Instance Connect to directly connect to my first EC2 instance. I need to do this in order to use the instance for connectivity tests later in this project.

Step 6 Connect to EC2 Instance 1

I reconnected the EC2 Instance Connect to connect directly to EC2 instance (VPC-1 instance), but I still encountered an error due to the missing SSH protocol. This was fixed by adding the SSH protocol to the security group of the VPC-1 EC2 instance.

Step 7 Test VPC Peering

In this step, I will attempt a direct connection from the VPC-1 EC2 instance to the VPC-2 EC2 instance. This will help me validate the peering connection setup that I established.

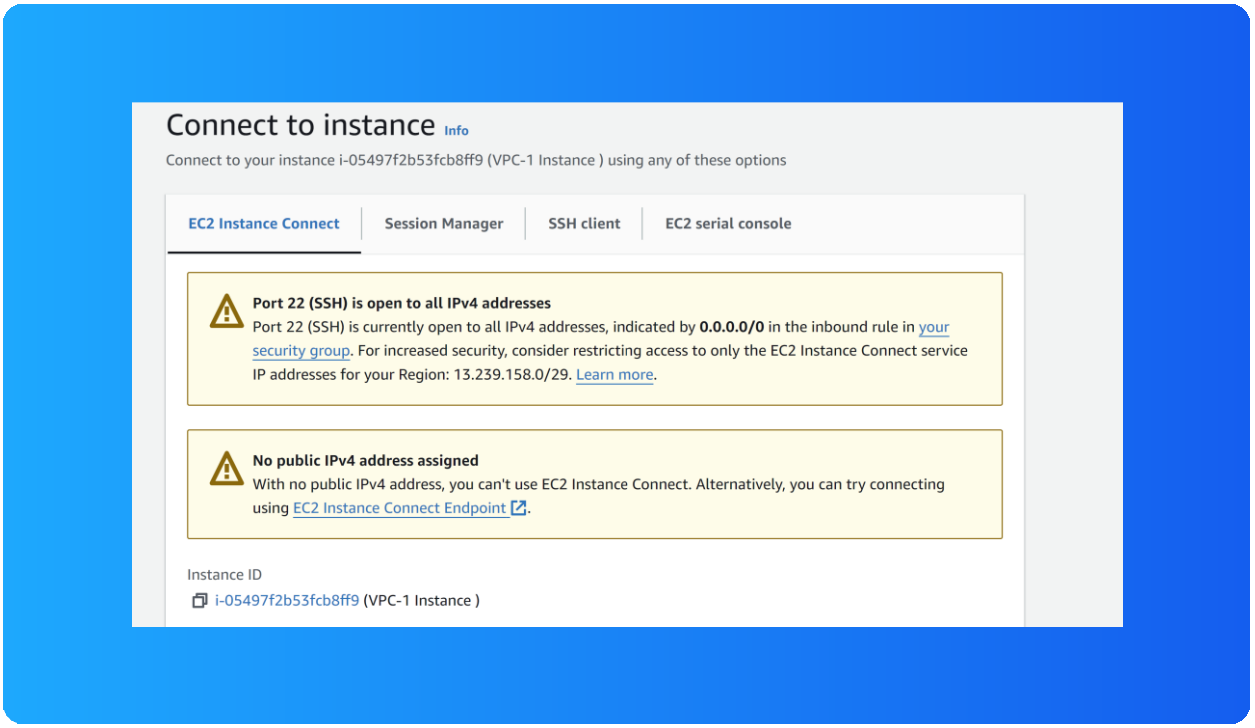
V

Prajit Venkatachalam
NextWork Student

Troubleshooting Instance Connect

Next, I used EC2 Instance Connect to directly connect to my EC2 instance (VPC-1 instance) from the AWS Management Console.

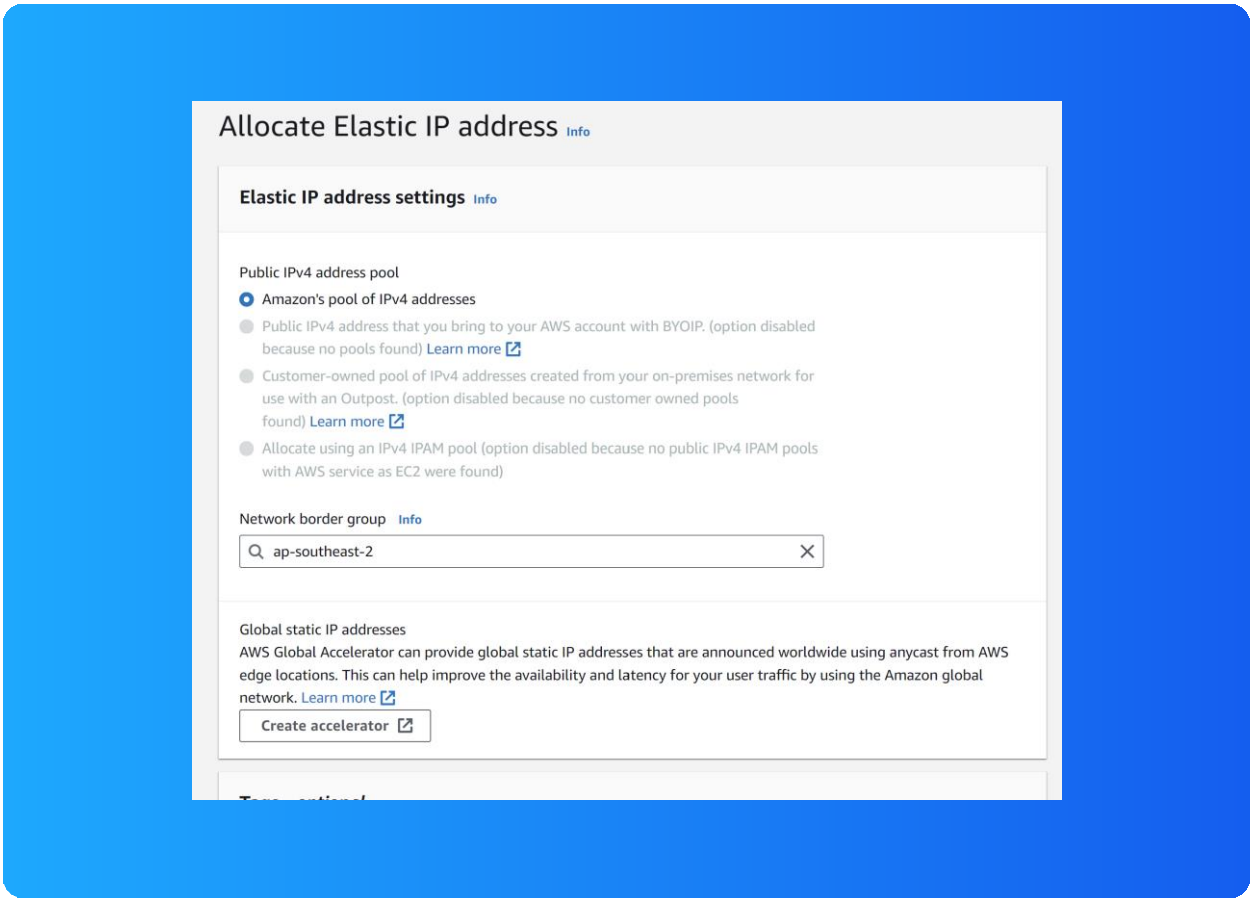
I was unable to use EC2 Instance Connect because a public IPv4 address was not assigned to the EC2 instance (VPC-1 instance). The public IPv4 address is an essential element for an EC2 instance in a public subnet to enable EC2 Instance Connect.



Elastic IP addresses

To resolve this error, I set up Elastic IP addresses. An Elastic IP address is a static public IPv4 address that I requested for my AWS account and then associated with the EC2 instance (VPC-1 instance), which will use it as its public IPv4 address.

Associating an Elastic IP address resolved the error because it provides my EC2 instance with a public IPv4 address, which enables EC2 Instance Connect to work.



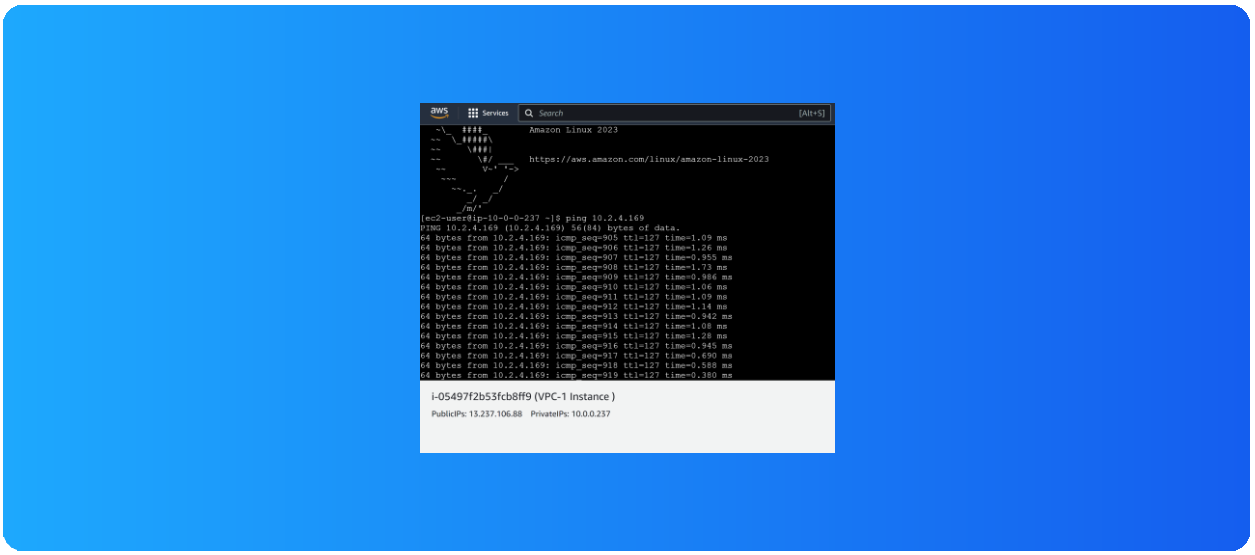
Prajit Venkatachalam
NextWork Student

Troubleshooting ping issues

To test VPC peering, I ran the command 'ping 10.2.4.169', which the private ipv4 address of the EC2 instance (VPC-2 instance).

A successful ping test would validate my VPC peering connection by receiving successful replies once I ping the private IP address of the VPC-2 instance from the VPC-1 Instance Connect terminal, which also connects the two VPCs successfully.

I had to update my second EC2 instance's (VPC-2 instance) security group because it was not allowing ICMP traffic from the VPC-1 instance. So, I added a new inbound rule to allow all ICMP traffic, which is the traffic type for ping messages.



Everyone should be in a job they love.

Check out nextwork.org for more projects