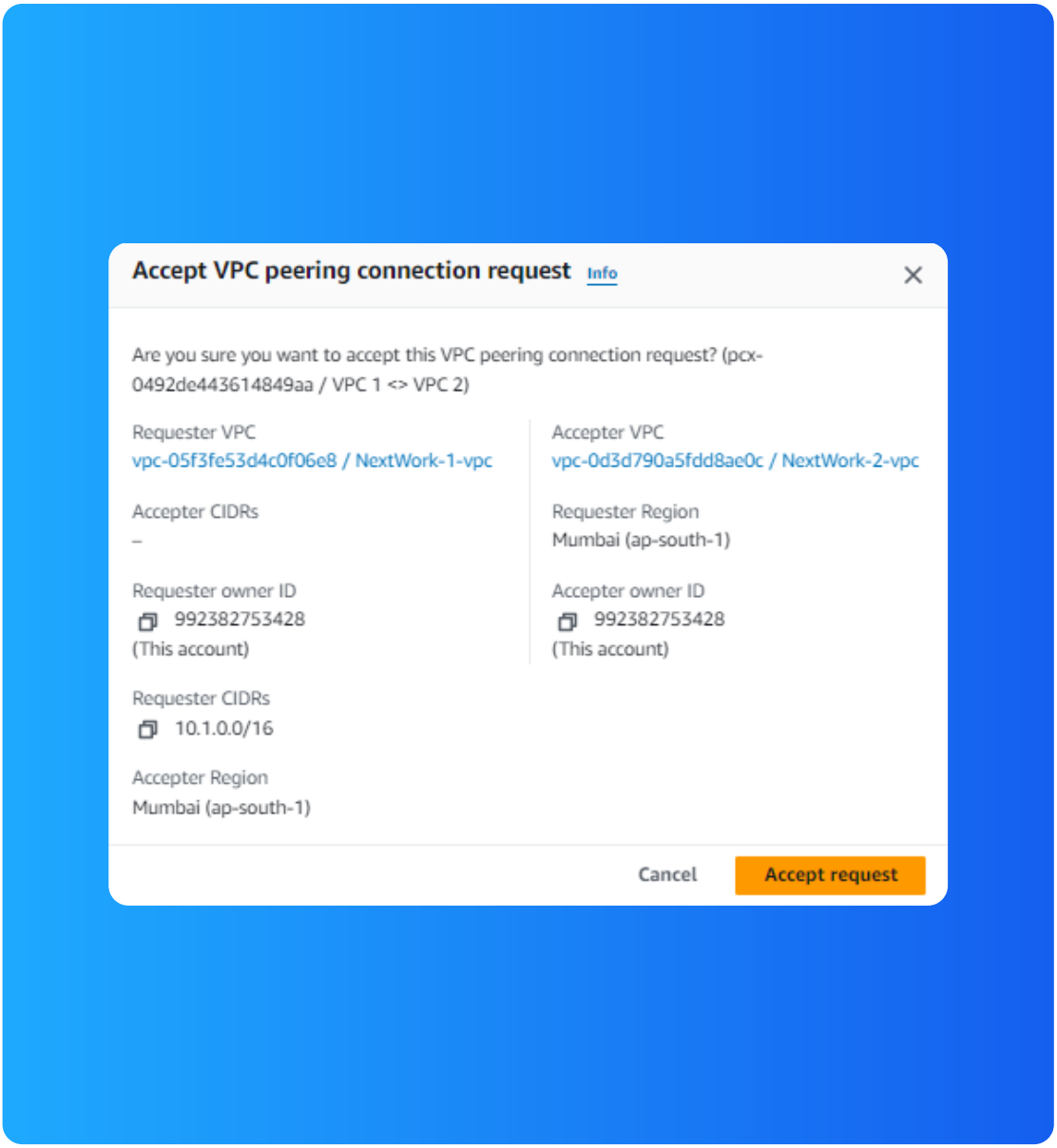
NextWork.org

VPC Peering

Roshan Thomas



Roshan Thomas
NextWork Student

Introducing Today's Project!

What is Amazon VPC?

Amazon VPC (Virtual Private Cloud) allows you to create isolated networks within AWS, giving control over IP ranges, subnets, route tables, and gateways. It's useful for securing and organizing cloud resources efficiently.

How I used Amazon VPC in this project

I used Amazon VPC to create two isolated virtual networks (VPCs) and establish a peering connection between them. This allowed me to securely connect resources in the two VPCs without exposing them to the public internet.

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

I didn't expect the complexity of configuring network ACLs and security groups to allow VPC communication. Understanding default rules and adjusting for ICMP traffic was crucial.

This project took me...

1 Hour and 45 mins.

Roshan Thomas
NextWork Student

NextWork.org

In the first part of my project...

Step 1 - Set up my VPC

In this step, we're creating two VPCs from scratch using the VPC wizard and the visual VPC resource map to speed up the process. This will help us quickly set up the foundation for VPC peering.

Step 2 - Create a Peering Connection

In this step, we're creating a peering connection to link the two VPCs. This will allow the VPCs to communicate with each other by enabling traffic flow between them.

Step 3 - Update Route Tables

We're updating the route tables of both VPCs to direct traffic between VPC 1 and VPC 2. This will enable proper routing so that traffic from one VPC can reach the other through the peering connection.

Step 4 - Launch EC2 Instances

We're going to launch an EC2 instance in each of the VPCs we created earlier. This will give us machines to interact with and test the VPC peering connection between them.

Roshan Thomas
NextWork Student

NextWork.org

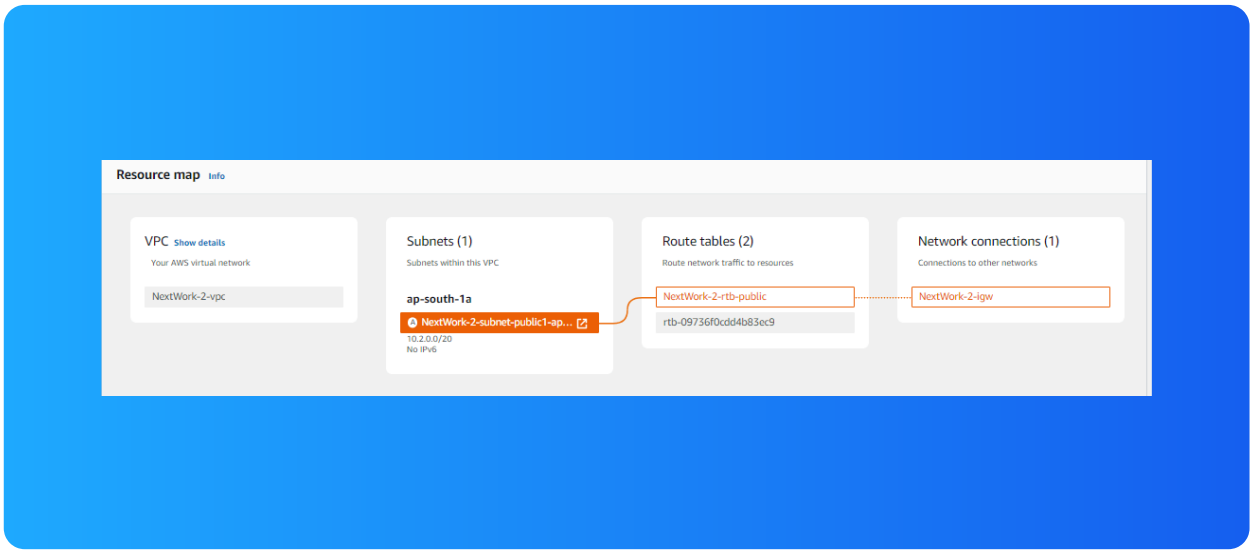
Multi-VPC Architecture

I started my project by launching two VPCs, each with one public subnet. The first VPC used CIDR block 10.1.0.0/16 and the second VPC used 10.2.0.0/16. Both had no private subnets, no NAT gateways, and default tenancy.

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 addresses could cause routing conflicts and prevent proper communication between the VPCs.

I also launched 2 EC2 instances

I didn't set up key pairs for these EC2 instances as I'm going to use the default key pair provided by AWS. This will allow me to access the instances using the AWS Management Console or the AWS CLI.



Roshan Thomas
NextWork Student

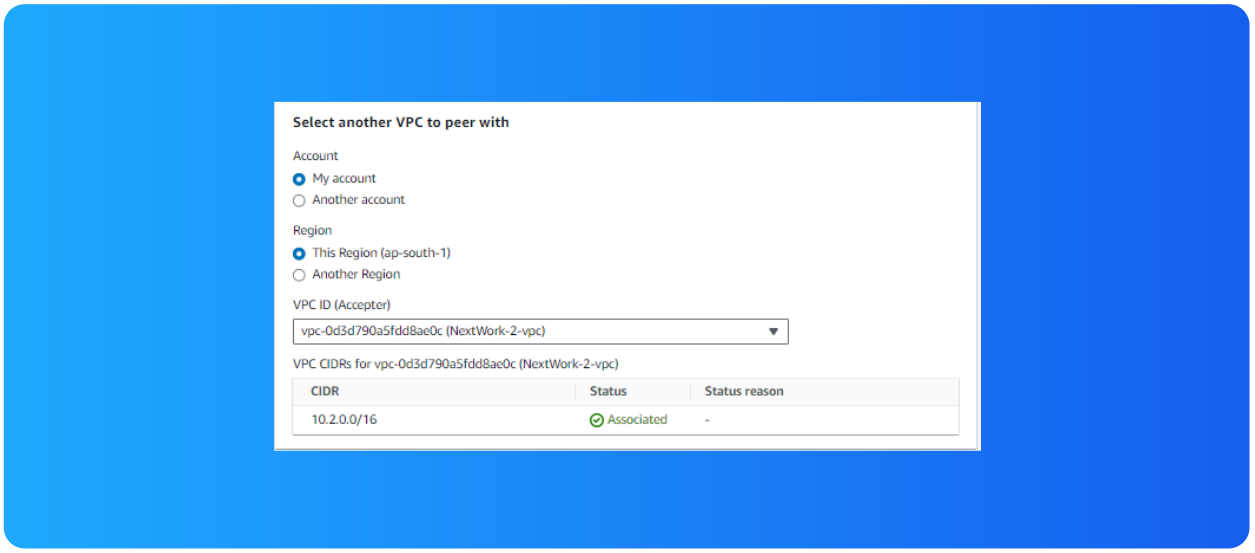
NextWork.org

VPC Peering

A VPC peering connection is a networking connection that allows two VPCs to communicate directly with each other using private IP addresses, enabling traffic between them without going over the public internet.

VPCs would use peering connections to enable secure, direct communication between different VPCs without using the public internet, improving performance and security across isolated cloud networks.

The difference between a Requester and an Accepter in a peering connection is that the Requester initiates the connection request, while the Accepter is the VPC owner who accepts or denies the peering request.

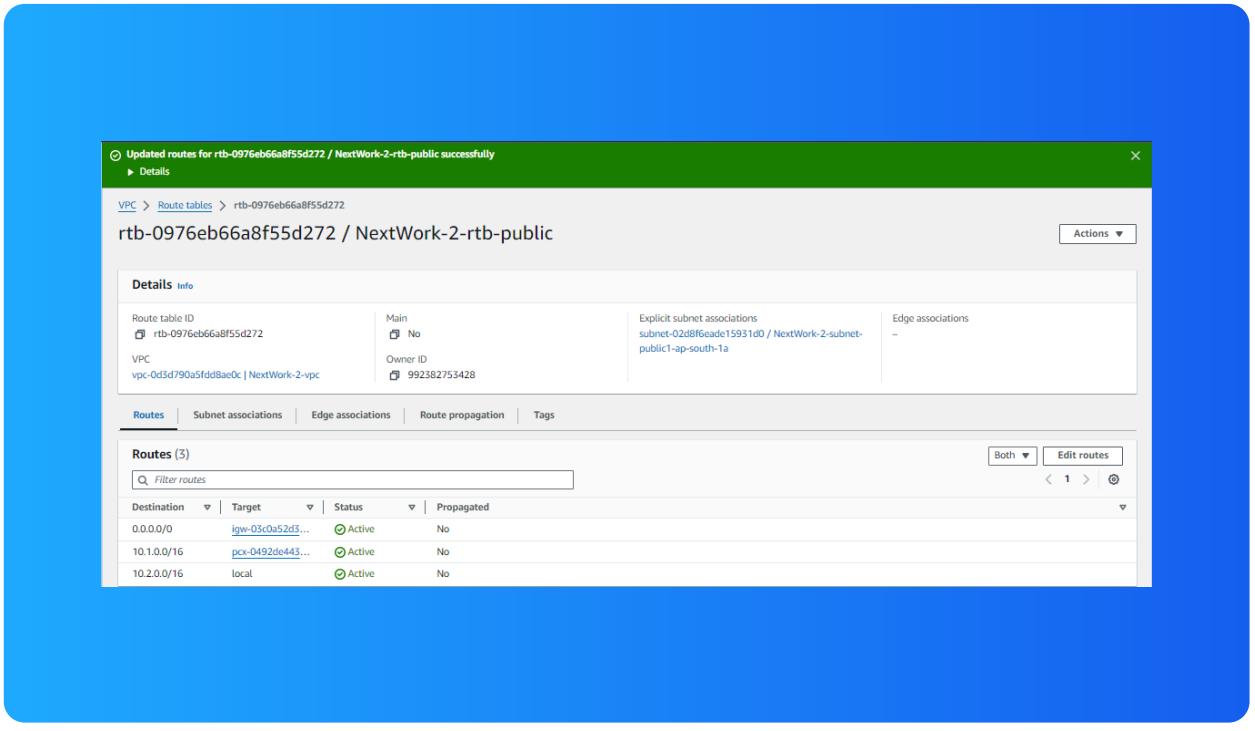


Roshan Thomas
NextWork Student

Updating route tables

After accepting a peering connection, my VPCs' route tables need to be updated because they must know how to direct traffic to the other VPC. Without new routes, the VPCs can't communicate through the peering connection.

My VPCs' new routes have a destination of the CIDR block of the other VPC (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.



Roshan Thomas
NextWork Student

In the second part of my project...

Step 5 - Use EC2 Instance Connect

We're going to use EC2 Instance Connect to connect to our first EC2 instance. This will allow us to access the instance's command line and run commands to test the VPC peering connection.

Step 6 - Connect to EC2 Instance 1

We're going to try connecting to our first EC2 instance again using EC2 Instance Connect. This time, we should be able to connect successfully since we've resolved the IP address issue.

Step 7 - Test VPC Peering

We're going to test the VPC peering connection by having Instance 1 send messages to Instance 2. This will help us verify that the peering connection is working correctly and that the instances can communicate with each other across VPC boundaries.

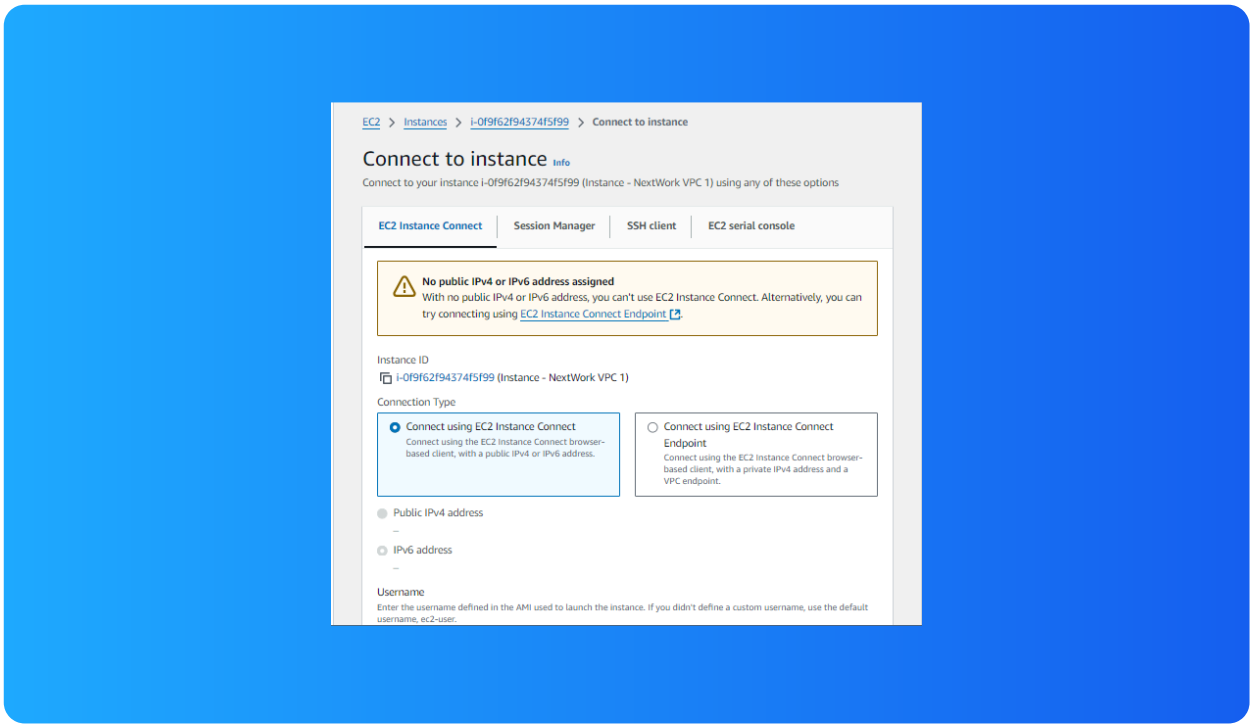
Roshan Thomas
NextWork Student

NextWork.org

Troubleshooting Instance Connect

Next, I used EC2 Instance Connect to connect to my first EC2 instance because it provides a secure and reliable way to access the instance's command line without having to manage SSH keys.

I was stopped from using EC2 Instance Connect as my EC2 instance did not have a public IPv4 address assigned. I had to assign a public IP address to the instance before I could connect using EC2 Instance Connect.



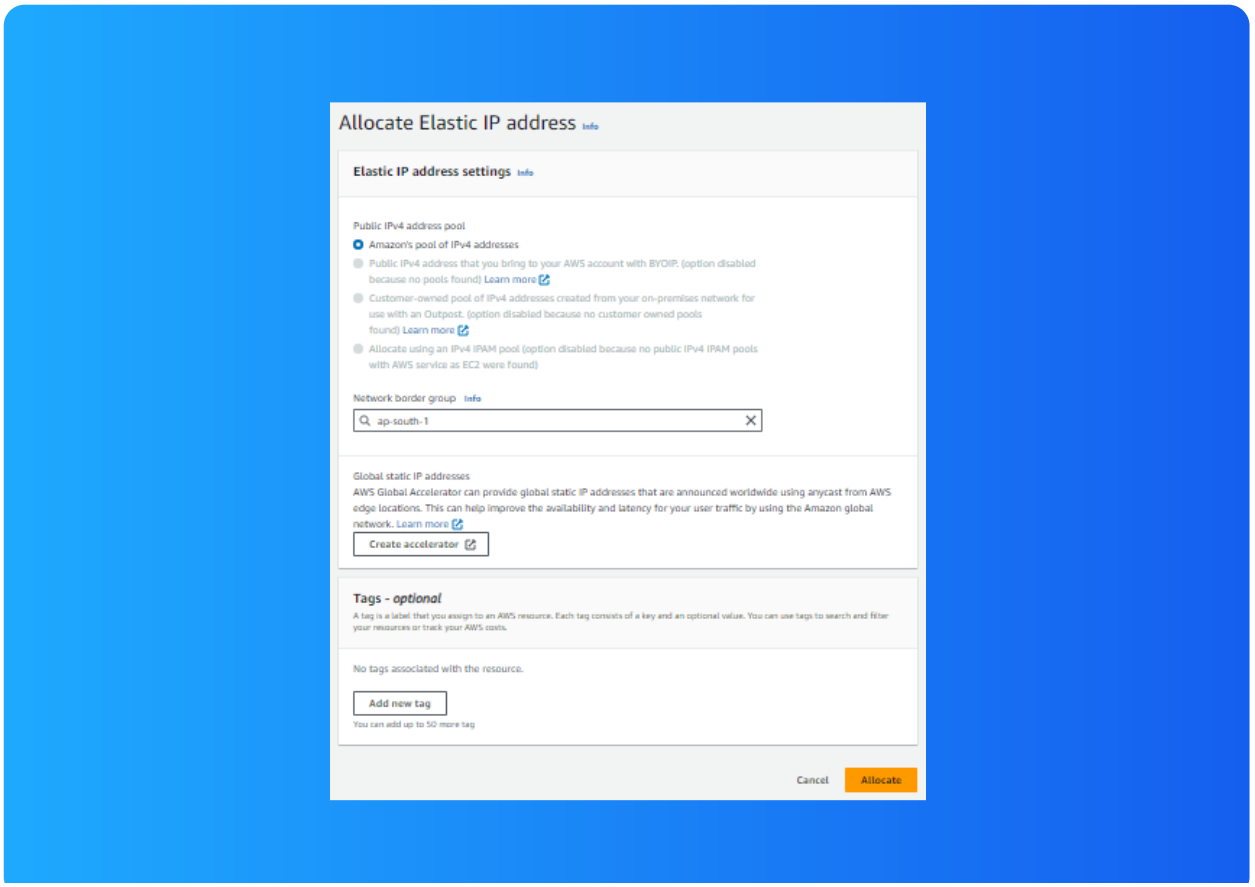
Roshan Thomas
NextWork Student

NextWork.org

Elastic IP addresses

To resolve this error, I set up Elastic IP addresses. Elastic IP addresses are static IP addresses that can be associated with an EC2 instance or used independently. They provide a consistent IP address even if the underlying EC2 instance is replaced

Associating an Elastic IP address resolved the error because it provided a consistent public IP address for my EC2 instance. This allowed me to connect to the instance using EC2 Instance Connect even if the underlying instance was rebooted or replace



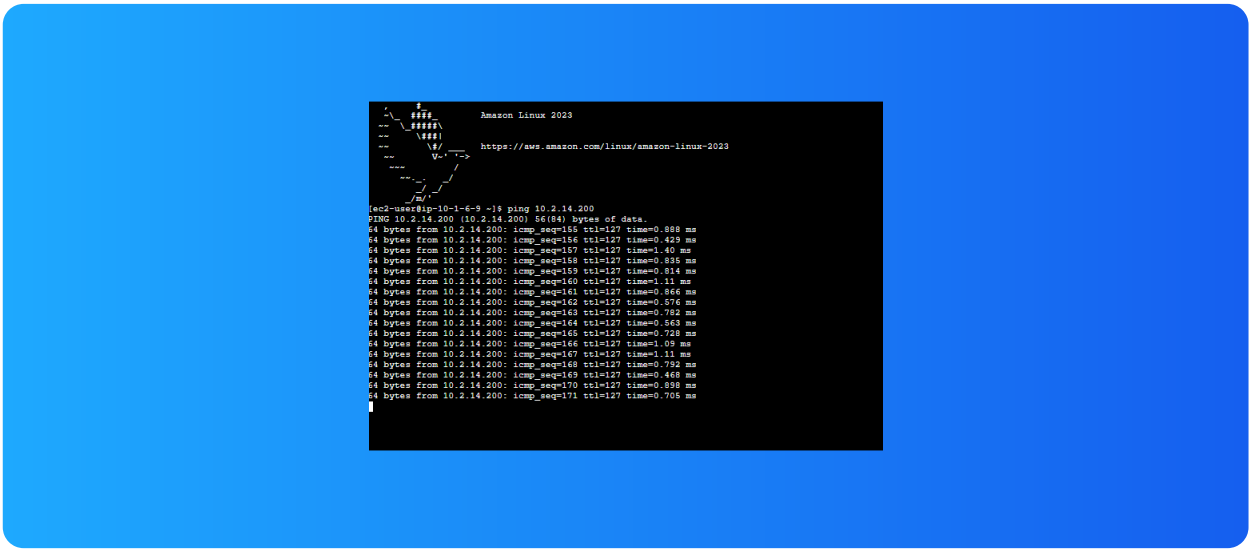
Roshan Thomas
NextWork Student

Troubleshooting ping issues

To test VPC peering, I ran the command ping <Instance 2's private IP address> from Instance 1's terminal. This command sends ICMP packets to Instance 2's private IP address, which should be successful if the VPC peering connection is working correct.

A successful ping test would validate my VPC peering connection because it would indicate that ICMP packets are able to traverse the VPC peering connection and reach the other instance, which means the connection is working as expected.

I had to update my second EC2 instance's security group because the default rules did not allow ICMP traffic from the first instance's VPC. I added a new rule that allowed ICMP traffic from the CIDR block of the first instance's VPC.



Everyone should be in a job they love.

Check out nextwork.org for more projects