

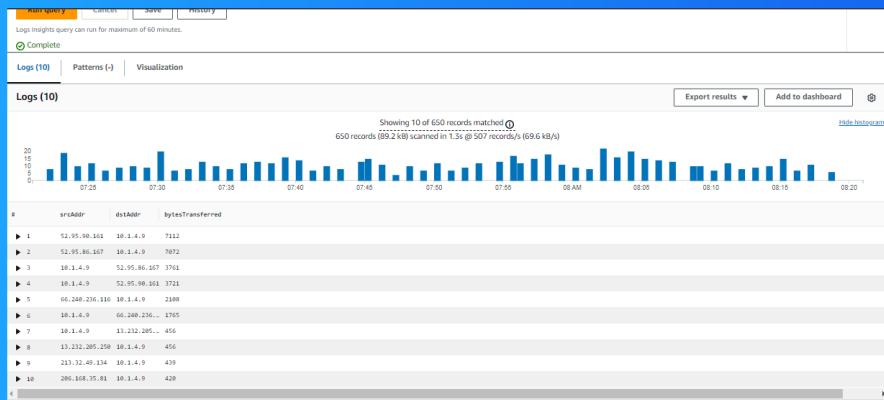


NextWork.org

# VPC Monitoring with Flow Logs



Roshan Thomas





Roshan Thomas  
NextWork Student

[NextWork.org](http://NextWork.org)

# Introducing Today's Project!

## What is Amazon VPC?

Amazon VPC (Virtual Private Cloud) allows you to create an isolated network within AWS, providing control over network settings, security, and access. It's useful for securely hosting applications, managing traffic, and integrating on-premise network.

## How I used Amazon VPC in this project

I used VPC to create a secure network for my EC2 instances. I set up subnets, configured routing, and used security groups to control traffic. VPC Flow Logs helped me monitor network activity and analyze data flow.

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

One thing I didn't expect was the complexity of configuring network ACLs and security groups to allow ICMP traffic for ping tests. This highlighted the importance of precise security settings for network connectivity.

## This project took me...

2 Hours and 30 Mins



# In the first part of my project...

## Step 1 - Set up VPCs

I'm going to create two VPCs to practice setting up and configuring virtual private networks in AWS. This will also help me review concepts from previous projects and prepare for more complex network architectures.

## Step 2 - Launch EC2 instances

I launched one EC2 instance in each VPC to test the VPC peering connection and prepare for future network monitoring.

## Step 3 - Set up Logs

I'll set up VPC Flow Logs to track network traffic and store the data. This will help me analyze data flow, identify threats, and optimize network performance.

## Step 4 - Set IAM permissions for Logs

I'm going to create an IAM role and policy to grant VPC Flow Logs the necessary permissions to write logs and send them to CloudWatch. This will enable me to effectively monitor and analyze network traffic within my VPCs.

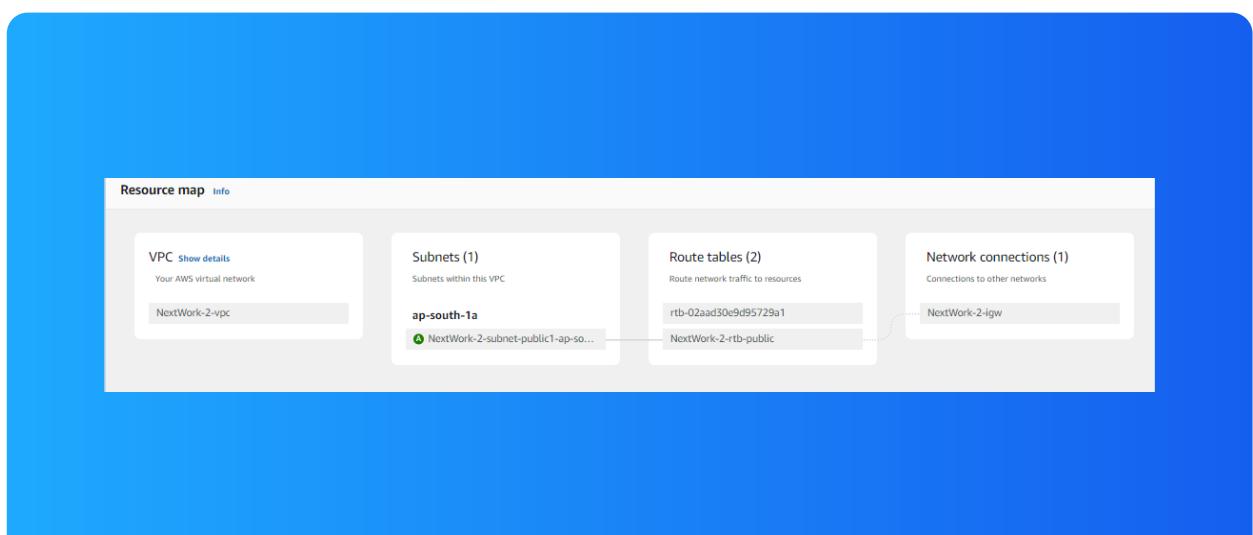
# Multi-VPC Architecture

I started my project by launching two VPCs. Each VPC has two subnets: a public subnet and a private subnet. This setup allows for both internet connectivity and isolation of resources within the private subnet.

The CIDR blocks for VPCs 1 and 2 are 10.1.0.0/16 and 10.2.0.0/16, respectively. They have to be unique to ensure that the IP addresses within each VPC do not overlap and avoid conflicts when the VPCs are peered or connected to other networks.

## I also launched EC2 instances in each subnet

My EC2 instances' security groups allow inbound ICMP traffic from all IP addresses (0.0.0.0/0). This is because ICMP messages are used for ping tests, which are essential for verifying network connectivity between the instances and other resources.



**Roshan Thomas**  
NextWork Student

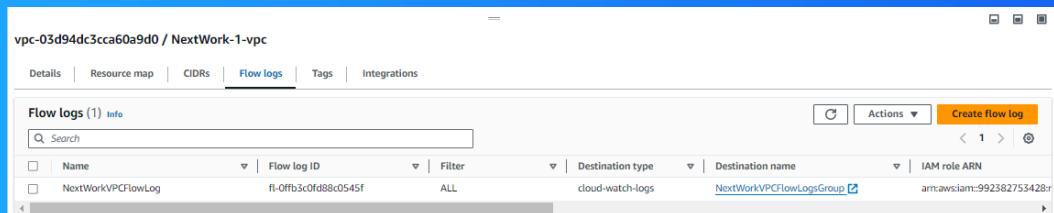
[NextWork.org](http://NextWork.org)

# Logs

Logs are records of events or activities that occur within a system. They provide valuable information about a system's behavior, including errors, warnings, and performance metrics.

Log groups are containers that organize and store log streams. They allow you to group related logs together for easier management and analysis.

## I also set up a flow log for VPC 1



# IAM Policy and Roles

I created an IAM policy to grant VPC Flow Logs the necessary permissions to write logs and send them to CloudWatch. This allows me to effectively monitor and analyze network traffic within my VPCs.

I also created an IAM role because it provides a secure way to grant VPC Flow Logs the necessary permissions to write logs and send them to CloudWatch. This helps prevent unauthorized access to sensitive log data.

A custom trust policy is a JSON document that specifies which entities are allowed to assume an IAM role. It grants permissions to specific AWS services or other principals to use the role's associated permissions.

```
1▼ {
2    "Version": "2012-10-17",
3    "Statement": [
4        {
5            "Sid": "Statement1",
6            "Effect": "Allow",
7            "Principal": {"Service": "vpc-flow-logs.amazonaws.com"},
8            "Action": "sts:AssumeRole"
9        }
10    ]
11}
12}
```

A circular profile picture of a man with dark hair and a beard, wearing a light-colored striped shirt.

Roshan Thomas  
NextWork Student

[NextWork.org](http://NextWork.org)

# In the second part of my project...

## Step 5 - Ping testing and troubleshooting

I'll use EC2 Instance Connect to access Instance 1 and send test messages to Instance 2 to generate network traffic and test the VPC peering connection.

## Step 6 - Set up a peering connection

I'm going to set up a peering connection between my two VPCs. This will create a link between the networks, allowing resources in each VPC to communicate with each other.

## Step 7 - Update VPC route tables

## Step 8 - Analyze flow logs

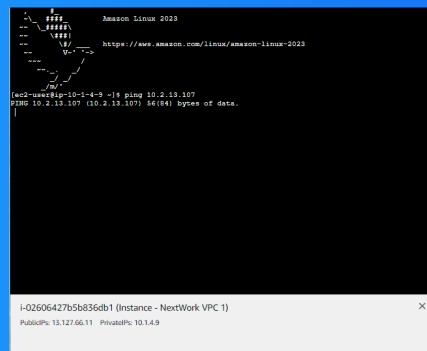
I'm going to review the flow logs recorded for VPC 1's public subnet to analyze network traffic activity within that subnet. This will help me understand data flow patterns, identify security threats, and optimize network performance.

Roshan Thomas  
NextWork Student

[NextWork.org](https://NextWork.org)

# Connectivity troubleshooting

My first ping test between my EC2 instances had no replies, which means there might be a problem with the connection between them. This could be due to blocked ICMP traffic or misconfigured network settings.



I could receive ping replies using the public IP address, indicating a successful private network peering connection. However, there might be an issue with security settings restricting ICMP traffic within the private subnet.

Roshan Thomas  
NextWork Student

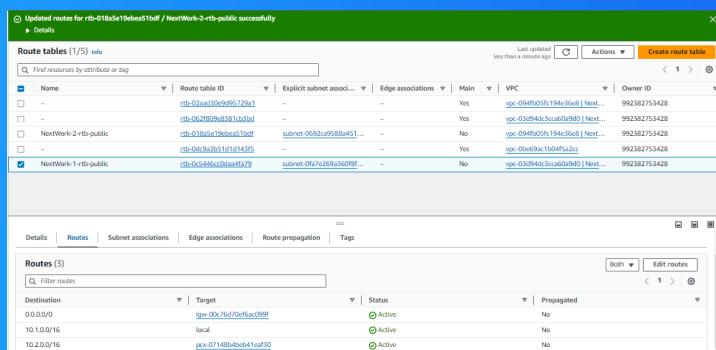
[NextWork.org](http://NextWork.org)

# Connectivity troubleshooting

Looking at VPC 1's route table, I identified that the ping test failed due to a missing route to VPC 2's private subnet. This indicates a misconfiguration in the routing settings.

## To solve this, I set up a peering connection between my VPCs

I also updated both VPCs' route tables so that resources in each VPC could reach the other VPC through the newly created peering connection. This ensured that network traffic could flow between the instances in the two VPCs.





# Connectivity troubleshooting

I received ping replies from Instance 2's private IP address! This means that the network peering connection is now working correctly, and resources in the two VPCs can communicate with each other.

```
64 bytes from 65.2.33.89: icmp_seq=292 ttl=126 time=0.454 ms
64 bytes from 65.2.33.89: icmp_seq=293 ttl=126 time=0.510 ms
64 bytes from 65.2.33.89: icmp_seq=294 ttl=126 time=0.449 ms
64 bytes from 65.2.33.89: icmp_seq=295 ttl=126 time=0.500 ms
64 bytes from 65.2.33.89: icmp_seq=296 ttl=126 time=0.634 ms
64 bytes from 65.2.33.89: icmp_seq=297 ttl=126 time=0.858 ms
64 bytes from 65.2.33.89: icmp_seq=298 ttl=126 time=0.396 ms
64 bytes from 65.2.33.89: icmp_seq=299 ttl=126 time=0.580 ms
64 bytes from 65.2.33.89: icmp_seq=300 ttl=126 time=0.427 ms
64 bytes from 65.2.33.89: icmp_seq=301 ttl=126 time=0.480 ms
64 bytes from 65.2.33.89: icmp_seq=302 ttl=126 time=0.512 ms
64 bytes from 65.2.33.89: icmp_seq=303 ttl=126 time=0.472 ms
64 bytes from 65.2.33.89: icmp_seq=304 ttl=126 time=0.490 ms
64 bytes from 65.2.33.89: icmp_seq=305 ttl=126 time=0.443 ms
64 bytes from 65.2.33.89: icmp_seq=306 ttl=126 time=0.467 ms
64 bytes from 65.2.33.89: icmp_seq=307 ttl=126 time=0.495 ms
64 bytes from 65.2.33.89: icmp_seq=308 ttl=126 time=0.467 ms
64 bytes from 65.2.33.89: icmp_seq=309 ttl=126 time=0.424 ms
64 bytes from 65.2.33.89: icmp_seq=310 ttl=126 time=0.576 ms
64 bytes from 65.2.33.89: icmp_seq=311 ttl=126 time=0.485 ms
64 bytes from 65.2.33.89: icmp_seq=312 ttl=126 time=0.577 ms
64 bytes from 65.2.33.89: icmp_seq=313 ttl=126 time=0.517 ms
64 bytes from 65.2.33.89: icmp_seq=314 ttl=126 time=0.479 ms
64 bytes from 65.2.33.89: icmp_seq=315 ttl=126 time=0.425 ms
64 bytes from 65.2.33.89: icmp_seq=316 ttl=126 time=0.454 ms
64 bytes from 65.2.33.89: icmp_seq=317 ttl=126 time=0.442 ms
```



Roshan Thomas  
NextWork Student

[NextWork.org](http://NextWork.org)

# Analyzing flow logs

Flow logs provide insights into network traffic, including source and destination IP addresses, protocols, packet counts, bytes transferred, and timestamps.

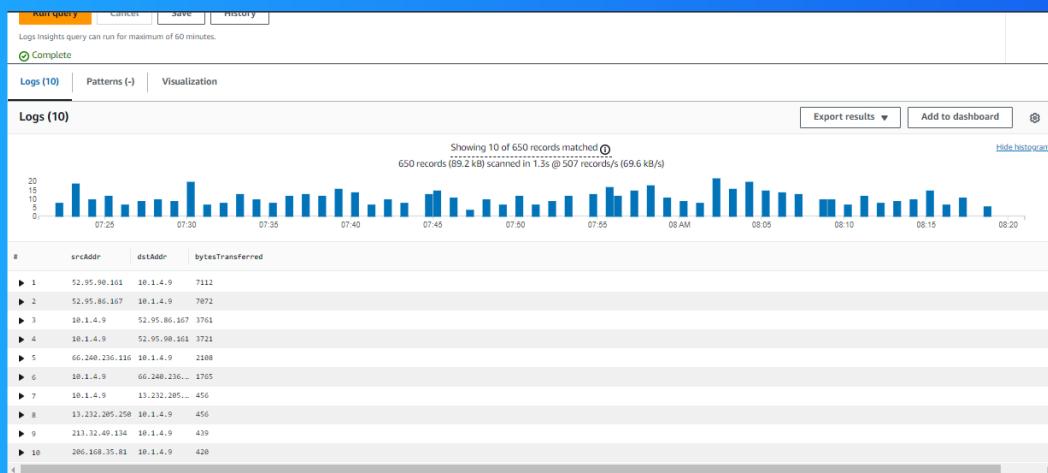
For example, the flow log shows a successful connection from a source IP address to my EC2 instance. It includes details like protocol, ports, and status.

▶ 2024-10-20T08:04:14.000Z	2 9923827513420 eni-decoder#4421088889 13.132.143.177 10.1.4.9 52160 8132 6 1 40 1729411664 1729411577 REJECT OK
▶ 2024-10-20T08:04:14.000Z	2 9923827513420 eni-decoder#4421088889 47.254.215.64 10.1.4.9 28017 5983 17 1 29 1729411664 1729411577 ACCEPT OK
▶ 2024-10-20T08:04:14.000Z	2 9923827513420 eni-decoder#4421088889 80.75.232.9 10.1.4.9 32879 2668 6 1 40 1729411664 1729411577 REJECT OK
▶ 2024-10-20T08:05:18.000Z	2 9923827513420 eni-decoder#4421088889 64.62.197.83 10.1.4.9 20748 37 4 29 1729411558 1729411577 REJECT OK
▶ 2024-10-20T08:05:18.000Z	2 9923827513420 eni-decoder#4421088889 76.143.195.2 10.1.4.9 51617 23 6 1 40 1729411558 1729411577 REJECT OK
▶ 2024-10-20T08:05:18.000Z	2 9923827513420 eni-decoder#4421088889 59.120.129.245 10.1.4.9 15555 79 6 1 40 1729411558 1729411577 REJECT OK
▶ 2024-10-20T08:05:18.000Z	2 9923827513420 eni-decoder#4421088889 281.169.35.16 10.1.4.9 44465 4596 17 1 304 1729411558 1729411577 REJECT OK
▶ 2024-10-20T08:05:18.000Z	2 9923827513420 eni-decoder#4421088889 183.77.172.11 10.1.4.9 40012 8080 6 1 40 1729411558 1729411577 REJECT OK
▶ 2024-10-20T08:05:18.000Z	2 9923827513420 eni-decoder#4421088889 65.49.1.39 10.1.4.9 54840 9643 6 1 40 1729411558 1729411577 REJECT OK
▶ 2024-10-20T08:05:18.000Z	2 9923827513420 eni-decoder#4421088889 74.207.153.22 10.1.4.9 69485 3456 6 1 44 1729411558 1729411577 REJECT OK
▶ 2024-10-20T08:05:18.000Z	2 9923827513420 eni-decoder#4421088889 125.17.181.218 10.1.4.9 24465 3419 6 1 52 1729411558 1729411577 REJECT OK
▶ 2024-10-20T08:05:18.000Z	2 9923827513420 eni-decoder#4421088889 13.233.186.146 10.1.4.9 123 56745 37 1 76 1729411558 1729411577 ACCEPT OK
▶ 2024-10-20T08:05:18.000Z	2 9923827513420 eni-decoder#4421088889 10.1.4.9 31.233.186.146 56746 123 37 1 76 1729411558 1729411577 ACCEPT OK
▶ 2024-10-20T08:05:18.000Z	2 9923827513420 eni-decoder#4421088889 51.140.6.36 10.1.4.9 0 1 1 36 1729411558 1729411577 ACCEPT OK
▶ 2024-10-20T08:05:18.000Z	2 9923827513420 eni-decoder#4421088889 10.1.4.9 51.140.6.36 0 1 1 36 1729411558 1729411577 ACCEPT OK
▶ 2024-10-20T08:05:18.000Z	2 9923827513420 eni-decoder#4421088889 131.41.206.142 10.1.4.9 59918 8728 6 1 40 1729411558 1729411577 REJECT OK
▶ 2024-10-20T08:05:18.000Z	2 9923827513420 eni-decoder#4421088889 162.216.156.75 10.1.4.9 54094 48537 6 1 44 1729411558 1729411577 REJECT OK
▶ 2024-10-20T08:05:18.000Z	2 9923827513420 eni-decoder#4421088889 45.227.252.228 10.1.4.9 51428 41389 6 1 40 1729411558 1729411577 REJECT OK
▶ 2024-10-20T08:05:18.000Z	2 9923827513420 eni-decoder#4421088889 1729411558 1729411577 REJECT OK

# Logs Insights

Logs Insights is a powerful tool that allows you to interactively search, analyze, and visualize log data in CloudWatch Logs. It provides a user-friendly interface for querying log data, discovering log fields, and performing various analysis tasks.

I ran the query fields: source IP, destination IP, protocol, status | stats count() to analyze flow log data and understand traffic patterns.





NextWork.org

# Everyone should be in a job they love.

Check out nextwork.org for  
more projects

