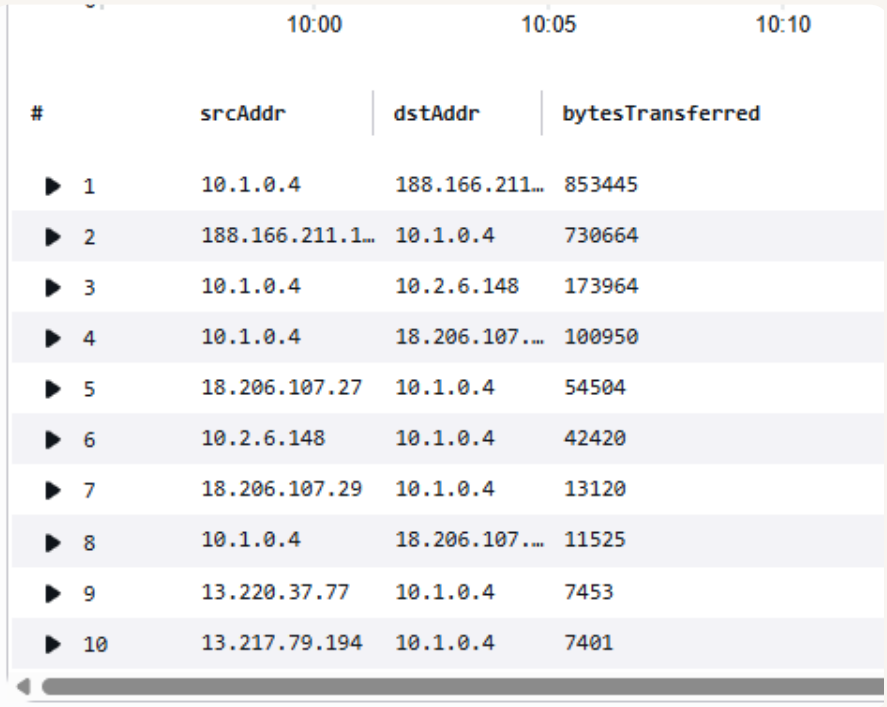
nextwork.org

VPC Monitoring with Flow Logs

Louis Moyo



10:00 10:05 10:10

#	srcAddr	dstAddr	bytesTransferred
1	10.1.0.4	188.166.211…	853445
2	188.166.211.1…	10.1.0.4	730664
3	10.1.0.4	10.2.6.148	173964
4	10.1.0.4	18.206.107.…	100950
5	18.206.107.27	10.1.0.4	54504
6	10.2.6.148	10.1.0.4	42420
7	18.206.107.29	10.1.0.4	13120
8	10.1.0.4	18.206.107.…	11525
9	13.220.37.77	10.1.0.4	7453
10	13.217.79.194	10.1.0.4	7401

L
Louis Moyo
NextWork Student

Introducing Today's Project!

What is Amazon VPC?

Amazon VPC is a secure, isolated section of the AWS cloud where you can launch resources inside a virtual network that you control. It's useful because it lets you customise IP ranges, subnets, route tables, and security settings, enabling secure and efficient communication between resources.

How I used Amazon VPC in this project

I used Amazon VPC to create two separate networks, set up a VPC peering connection, and monitor their traffic using VPC Flow Logs and CloudWatch Log Insights.

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

I didn't expect to see source and destination IPs swapped compared to the example logs - it highlighted how traffic direction depends on which instance initiates the connection.

This project took me...

About 2 hours from VPC setup to running Flow Logs queries and analysing the results.

In the first part of my project...

Step 1 - Set up VPCs

I'm going to use the VPC wizard to quickly create two new VPCs (unique CIDRs, one public subnet each). This sets a clean foundation to revisit peering and then enable Flow Logs + CloudWatch so I can monitor and analyse traffic between them.

Step 2 - Launch EC2 instances

I'm going to launch one EC2 instance in each VPC so they can communicate over the VPC peering connection later, generating network traffic that we can capture and analyse with VPC Flow Logs in CloudWatch.

Step 3 - Set up Logs

I'm going to enable VPC Flow Logs to capture all inbound and outbound traffic for my VPCs, and configure them to store the log records in CloudWatch so I can later search, filter, and analyse the network activity between my instances.

Step 4 - Set IAM permissions for Logs

I'm going to create an IAM policy and role that grants VPC Flow Logs permission to write log data to CloudWatch, then link that role to my subnet's flow log so network traffic records can be stored and analysed.

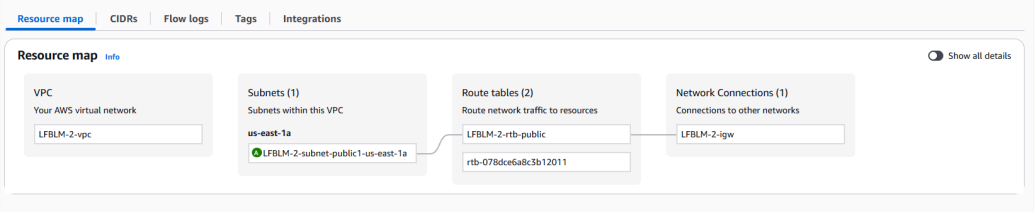
Multi-VPC Architecture

I started my project by launching two new VPCs from scratch using the VPC wizard, each with one public subnet, no private subnets, default tenancy, and no NAT gateways or 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 ranges would cause routing conflicts and prevent proper communication when connected via VPC peering

I also launched EC2 instances in each subnet

My EC2 instances' security groups allow ICMP from all IPs for Instance 1, but only from 10.1.0.0/16 for Instance 2. This is because Instance 1 needs to be reachable from anywhere, while Instance 2 only needs pings from VPC 1.

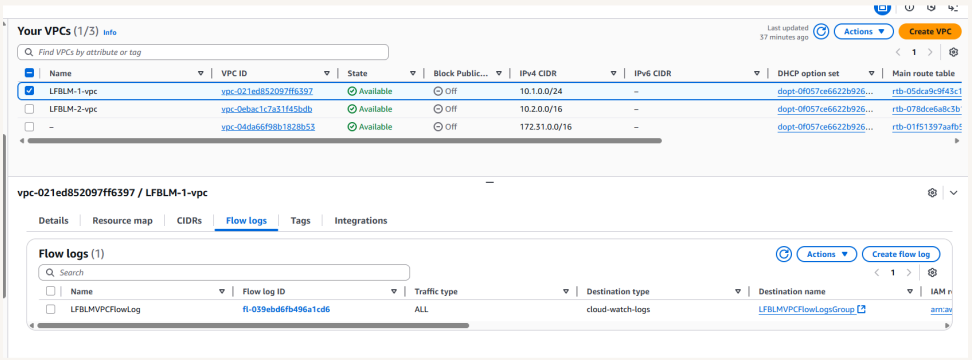


Louis Moyo
NextWork Student

Logs

Logs are detailed records of events and activities in a system, such as network traffic, user actions, and errors, used to monitor performance, troubleshoot issues, and track security-related events.

Log groups are containers in CloudWatch that store and organise related logs from the same source or application, making it easier to manage, search, and analyse logs for monitoring and troubleshooting.

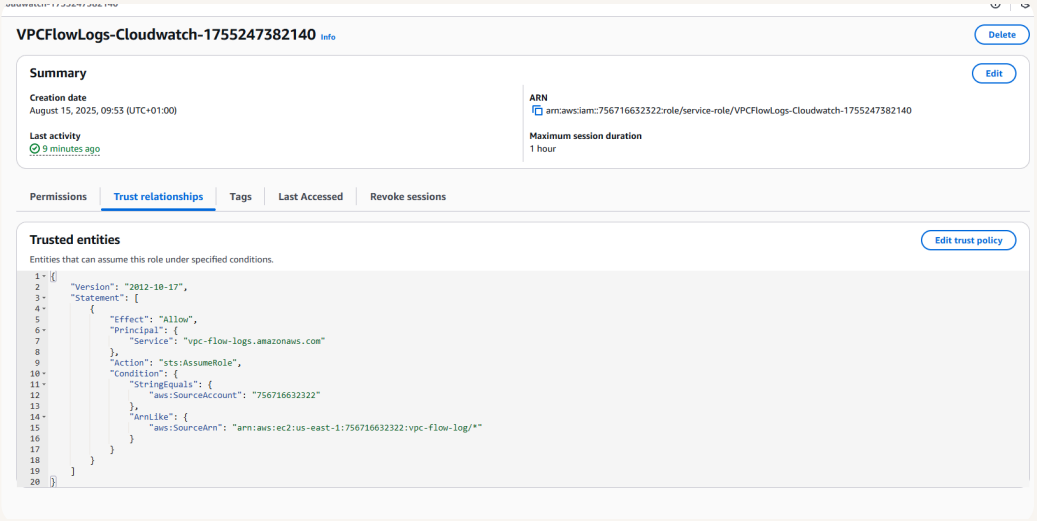


IAM Policy and Roles

I created an IAM policy because VPC Flow Logs needs explicit permission to create log groups, create log streams, and send log events to CloudWatch, and this policy grants those permissions for any log group or stream.

I also created an IAM role because VPC Flow Logs needs explicit permission to publish log data into CloudWatch, and creating a role with the right trust policy ensures only this service can assume it and perform that action.

A custom trust policy is a special IAM policy that specifies who or what can assume a role, rather than what actions they can take. In this case, it restricts access so only the vpc-flow-logs.amazonaws.com service can use the role.


VPCFlowLogs-Cloudwatch-1755247382140 Info
Delete
Summary
Edit
Creation date
August 15, 2025, 09:53 (UTC+01:00)
ARN
arn:aws:iam::756716632322:role/service-role/VPCFlowLogs-Cloudwatch-1755247382140
Last activity
9 minutes ago
Maximum session duration
1 hour
Permissions
Trust relationships
Tags
Last Accessed
Revoke sessions
Trusted entities
Edit trust policy
Entities that can assume this role under specified conditions.
{
"Version": "2012-10-17",
"Statement": [
{
"Effect": "Allow",
"Principal": {
"Service": "vpc-flow-logs.amazonaws.com"
},
"Action": "sts:AssumeRole",
"Condition": {
"StringEquals": {
"aws:SourceAccount": "756716632322"
},
"ArnLike": {
"aws:SourceArn": "arn:aws:ec2:us-east-1:756716632322:vpc-flow-log/*"
}
}
}
]
}

In the second part of my project...

Step 5 - Ping testing and troubleshooting

I'm going to connect to Instance 1 and try sending test messages to Instance 2's private IP address. This will both confirm that the VPC peering connection works and generate traffic that should appear in the flow logs for monitoring.

Step 6 - Set up a peering connection

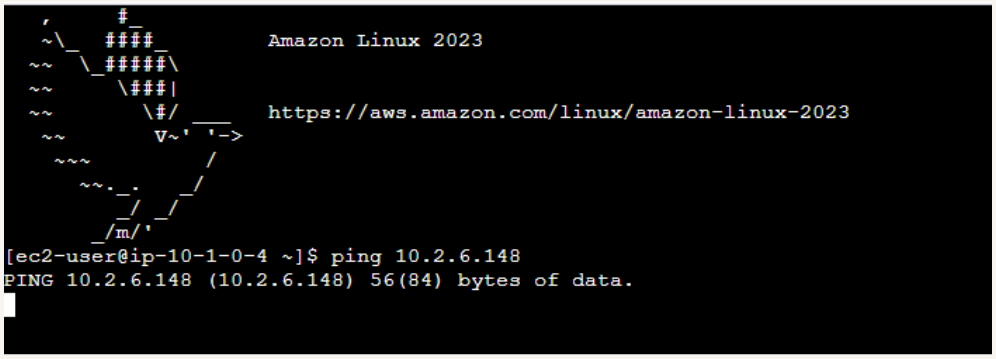
I'm going to create a VPC peering connection between my two VPCs and update their route tables so they can communicate directly using private IP addresses, ensuring faster, more secure traffic without going over the public internet.

Step 7 - Analyze flow logs

I'm going to review VPC Flow Logs for VPC 1's public subnet to see what network activity was recorded and analyse the traffic for useful insights.

Connectivity troubleshooting

My first ping test between my EC2 instances had no replies, which means the connection over the VPC peering link isn't working yet, likely due to missing or incorrect network permissions, such as NACL rules or route table entries.



I could receive ping replies if I ran the ping test using the other instance's public IP address, which means Instance 2 is set up to respond to ICMP requests and Instance 1 can reach it over the public internet, though this doesn't confirm VPC peering works.

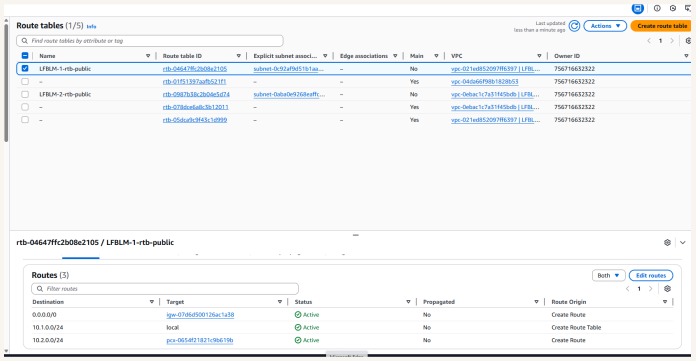
Louis Moyo
NextWork Student

Connectivity troubleshooting

he issue is VPC 1's route table lacks a route to 10.2.0.0/16 via the peering connection, so private IP traffic goes over the internet. Adding this route ensures traffic stays on the private AWS network, improving speed and security.

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

I also updated both VPCs' route tables so that traffic destined for the other VPC's private IP range is routed through the VPC peering connection, allowing direct private communication without going over the public internet.



Connectivity troubleshooting

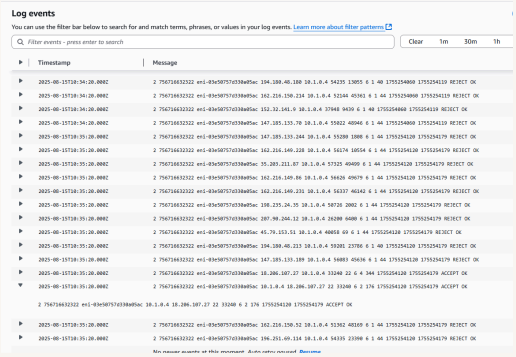
I received ping replies from Instance 2's private IP address! This means the VPC peering connection and route tables are now correctly configured, allowing direct communication between the two instances over their private IPs without going through the public internet.

Louis Moyo
NextWork Student

Analyzing flow logs

Flow logs tell us about network traffic in and out of an interface — version, AWS account ID, network interface ID, source/destination IPs, ports, protocol, packets, bytes, timestamps, action (ACCEPT/REJECT), and log status (OK).

For example, the flow log I've captured tells us that 176 bytes of TCP traffic (port 22) were successfully sent from 18.206.107.27 to 10.1.0.4, and the connection was allowed by both the NACL and SG.

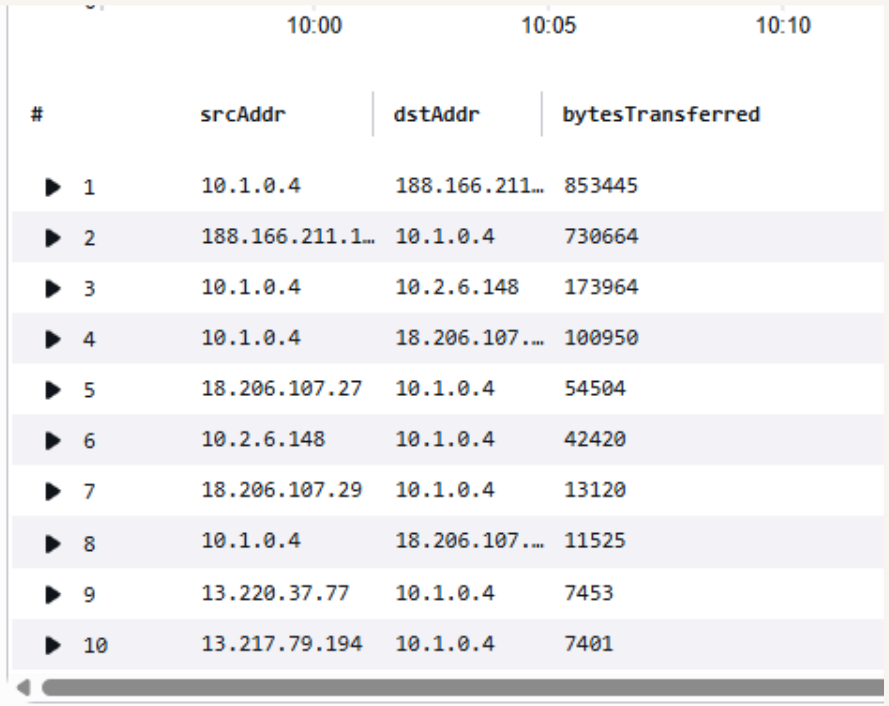


Louis Moyo
NextWork Student

Logs Insights

Logs Insights is an interactive tool in CloudWatch that lets you search, analyse, and visualise log data using a query language, helping you quickly identify patterns, troubleshoot issues, and gain insights from your logs in near real-time.

I ran the query to return the top 10 byte transfers by source and destination IP addresses. This query analyses the flow logs to identify the largest data transfers, helping pinpoint high-traffic connections for monitoring or troubleshooting.



10:00 10:05 10:10

#	srcAddr	dstAddr	bytesTransferred
1	10.1.0.4	188.166.211…	853445
2	188.166.211.1…	10.1.0.4	730664
3	10.1.0.4	10.2.6.148	173964
4	10.1.0.4	18.206.107.…	100950
5	18.206.107.27	10.1.0.4	54504
6	10.2.6.148	10.1.0.4	42420
7	18.206.107.29	10.1.0.4	13120
8	10.1.0.4	18.206.107.…	11525
9	13.220.37.77	10.1.0.4	7453
10	13.217.79.194	10.1.0.4	7401

The place to learn & showcase your skills

Check out nextwork.org for more projects