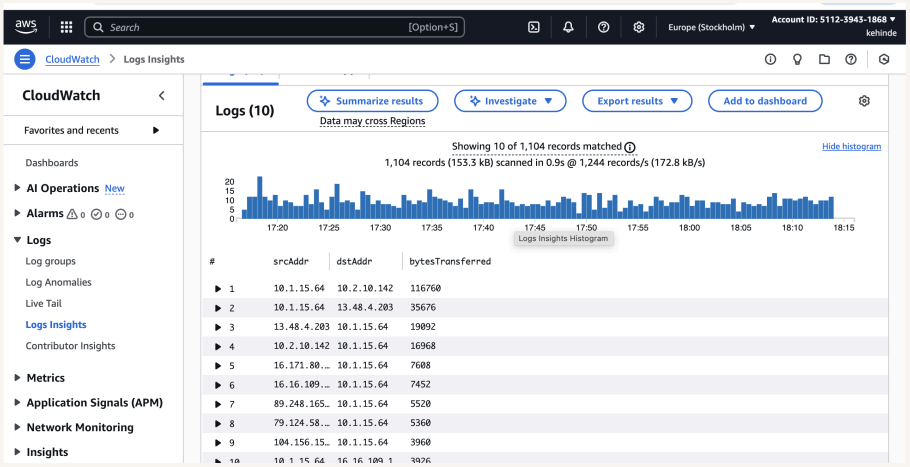
nextwork.org

VPC Monitoring with Flow Logs

Kehinde Abiuwa



Kehinde Abiuwa
NextWork Student

Introducing Today's Project!

What is Amazon VPC?

Amazon VPC is a service that lets you create a secure, isolated virtual network in the AWS cloud, giving you full control over your networking environment for deploying and managing AWS resources. It is useful because it allows you to securely control network traffic, isolate resources, and customize your cloud network to meet specific architectural and security requirements.

How I used Amazon VPC in this project

I used Amazon VPC in today's project to test the peering connection and communication between instances in two different VPCs. I also used the VPC to monitor traffic between the instances and to check and analyze flow logs.

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

Nothing really

This project took me...

I used 60MINUTES on this project

In the first part of my project...

Step 1 - Set up VPCs

I am about to setup two VPCs because I want to test their peering connection

Step 2 - Launch EC2 instances

In this step I am going to launch an EC2 instance in each VPC, so I can use them to test your VPC peering connection later

Step 3 - Set up Logs

In this step I am going to use a tool called VPC Flow logs to monitor the traffic going in and out of my vpc. I will do this by Setting up a way to track all inbound and outbound network traffic & also Setting up a space that stores all of these records.

Step 4 - Set IAM permissions for Logs

In this step, I am going to give VPC Flow Logs the necessary permissions to write logs to CloudWatch Logs and finish setting up my subnet's flow log. I will accomplish both tasks using an IAM role with the appropriate policy attached.

Multi-VPC Architecture

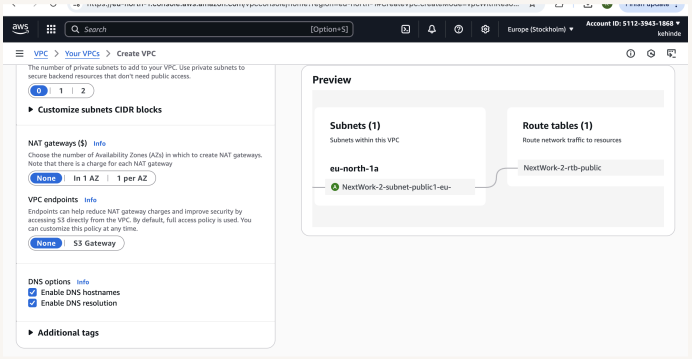
I started my project by launching my first VPC with only 1 public subnet & 1 availability zone, I also set the ipv4 cidr block to 10.1.0.0/16, making it unique.

The CIDR blocks for VPCs 1 and 2 are different, non-overlapping IP address ranges. They have to be unique because VPC peering requires that the IP ranges of the connected VPCs do not overlap. If the CIDR blocks were the same or overlapped, AWS wouldn’t know how to correctly route traffic between them, which would cause conflicts and connectivity issues. Having unique CIDR blocks ensures that traffic can be accurately directed between the two VPCs without ambiguity

I also launched EC2 instances in each subnet

My EC2 instances' security groups allow icmp traffic from anywhere. This is because I'll be doing a new ping test in just a second and it's going to require my security group to allow in ICMP traffic from all IP addresses, not just from the other VPC

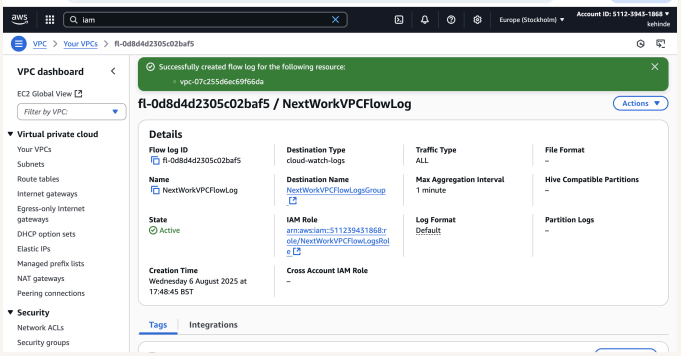
Kehinde Abiuwa
NextWork Student



Logs

Logs are like a diary for your computer systems. They record everything that happens, from users logging in to errors popping up. It's the go-to place to understand what's going on with your systems, troubleshoot problems, and keep an eye on who's doing what.

Log groups are containers in Amazon CloudWatch Logs that organize and store log streams from the same source, application, or resource. Think of them as big folders where related logs are grouped together for easier management and analysis. Each log group can contain multiple log streams, and logs are stored in the region where they are generated. Although log data is region-specific, CloudWatch dashboards can be used to visualize and monitor logs across multiple regions

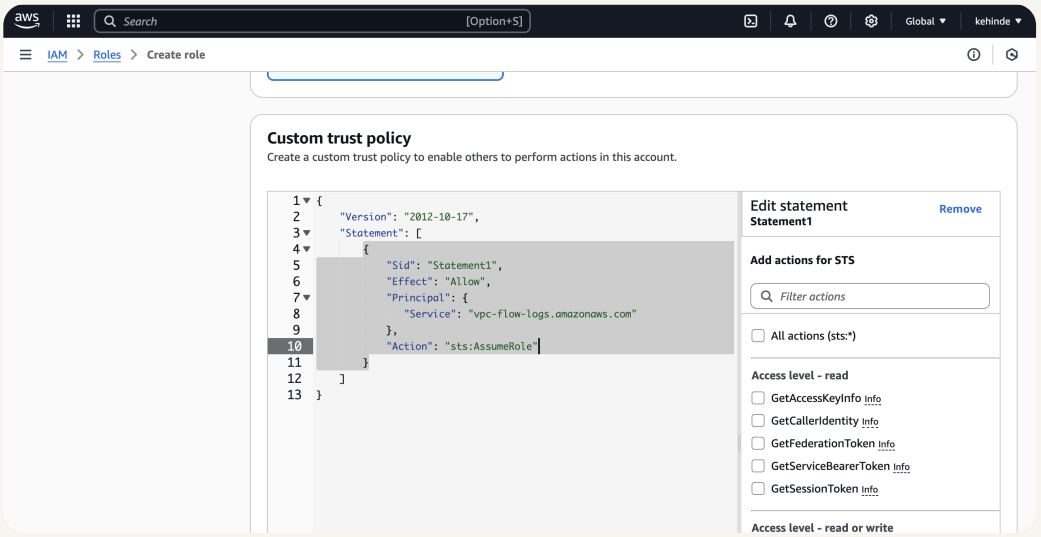


IAM Policy and Roles

I created an IAM policy because VPC Flow Logs by default don't have the permission to record logs and store them in the CloudWatch log group. This policy makes sure that my VPC can now send log data to my log group.

I also created an IAM role because you can’t assign an IAM policy directly to an AWS service. Instead, you need to create a role, attach the policy to that role, and then assign the role to the service—in this case, VPC Flow Logs. This role allows the service to assume the permissions defined in the policy, such as writing logs to the CloudWatch log group I created earlier.

A custom trust policy is a JSON document that defines which entities (such as AWS services, users, or other roles) are allowed to assume a specific IAM role. It’s part of the role’s configuration and controls who or what can use that role.


aws
Search
[Option+S]
Global
kehinde
IAM
Roles
Create role
Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.
{
"Version": "2012-10-17",
"Statement": [
{
"Sid": "Statement1",
"Effect": "Allow",
"Principal": {
"Service": "vpc-flow-logs.amazonaws.com"
},
"Action": "sts:AssumeRole"
}
]
}
Edit statement
Statement1
Remove
Add actions for STS
Filter actions
All actions (sts:*)
Access level - read
GetAccessKeyInfo Info
GetCallerIdentity Info
GetFederationToken Info
GetServiceBearerToken Info
GetSessionToken Info
Access level - read or write

In the second part of my project...

Step 5 - Ping testing and troubleshooting

In this step I am going to generate network traffic to see if my flow logs can pick up on them, I am generating the network traffic by getting my instance in vpc 1 to send a message to my instance in vpc 2, If I am able to get the instances to talk to each other, this means I am also testing my vpc peering setup at the same time.

Step 6 - Set up a peering connection

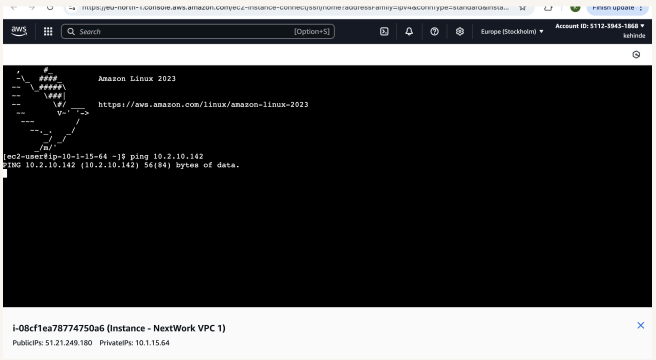
In this step I am going to setup a connecting between my instance in vpc 1 and my instance in vpc 2 by creating a peering connection and setting it up for them to be able to communicate with each other.

Step 7 - Analyze flow logs

In this step I am going to check out what VPC Flow Logs has recorded about my network's activity. I will do this by reviewing the flow logs recorded aboout VPC 1's public subnet and analyzing the flow logs to get some insights.

Connectivity troubleshooting

My first ping test between my EC2 instances had no replies, which means the instances were unable to communicate with each other over the network



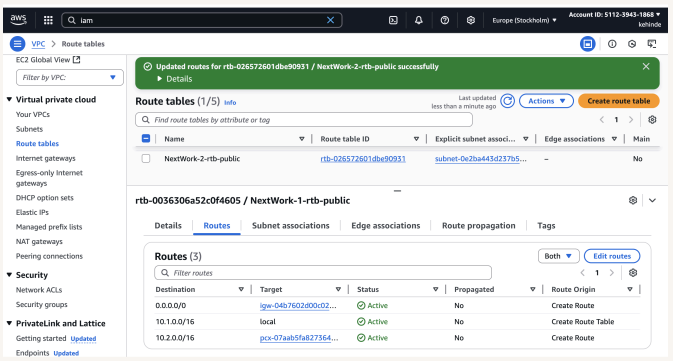
I could receive ping replies if I ran the ping test using the other instance's public IP address, which means the instances were able to communicate over the internet, but not over the private VPC network.

Connectivity troubleshooting

Looking at VPC 1's route table, I identified that the ping test using Instance 2's private IP address failed because there is no route that directs traffic to VPC 2's private CIDR block through the VPC peering connection. Instead, I see a route with a destination of 0.0.0.0/0, which sends traffic through the internet gateway, exposing it to the public internet. This means that while the instances can reach each other using their public IPs, they cannot communicate privately because the necessary private route via the peering connection is missing

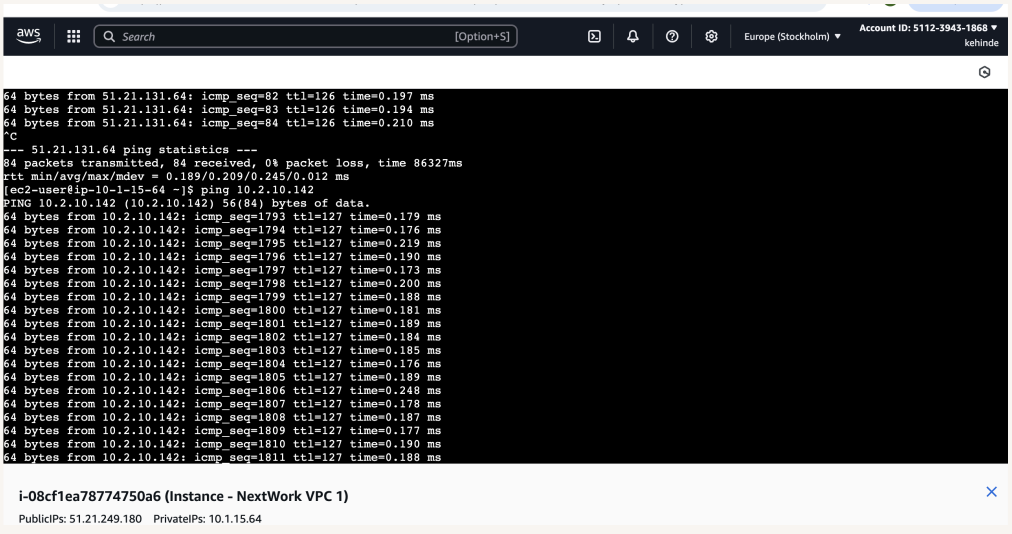
To solve this, I set up a peering connection between my VPCs

I also updated both VPCs' route tables so that the peering connection could fully work. It's one thing to set up the peering connection, but it's another to tell the VPCs how to locate each other using their private IP ranges. Without proper routes in the route tables, the VPCs won't know how to send traffic to one another—even if the peering connection itself is active



Connectivity troubleshooting

I received ping replies from Instance 2's private IP address! This means the two instances can now successfully communicate over the private network, which confirms that the VPC peering connection is working, and that the route tables and security group rules have been correctly configured to allow ICMP (ping) traffic between them. It also verifies that both VPCs can locate each other using their private IP ranges, as intended in a secure, internal network setup

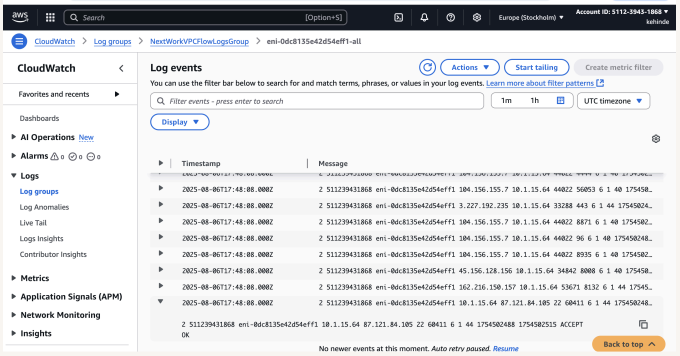


Analyzing flow logs

Flow logs tell us about the network traffic going to and from network interfaces in a VPC by capturing key information in a series of fields. The different parts of a flow log typically include the version, which is the format version of the flow log record; the account ID of the flow log owner; and the interface ID, which identifies the network interface (ENI) associated with the traffic. They also include the source and destination IP addresses and port numbers involved in the communication, as well as the protocol number (such as 6 for TCP or 17 for UDP). Additionally, flow logs record the number of packets and bytes transferred, the start and end time of each traffic flow, and the action indicating whether the traffic was accepted or rejected. Finally, the log status shows the result of the logging attempt (e.g., OK, NODATA, or SKIPDATA). These parts help you analyze, troubleshoot, and monitor network behavior within your VP

For example, the flow log I've captured tells us that network traffic flowed successfully from a private IP address 10.1.15.64 to a public IP address 87.121.84.105 on port 22 (which is used for SSH). The log entry shows: Version: 2 Account ID: 511239431868 Network Interface (ENI): eni-0dc8135e42d54eff1 Source IP: 10.1.15.64 Destination IP: 87.121.84.105 Source Port: 60411 Destination Port: 22 Protocol: 6 (which represents TCP) Packets: 1 Bytes: 44 Start Time: 1754502488 End Time: 1754502515 Action: ACCEPT Log Status: OK This means a TCP connection attempt (likely SSH) was successfully accepted by the security settings on the instance, and the data was logged without any issue (OK). It confirms that the instance at 10.1.15.64 was able to reach the external IP over SSH, and the VPC Flow Logs captured that event correctly.

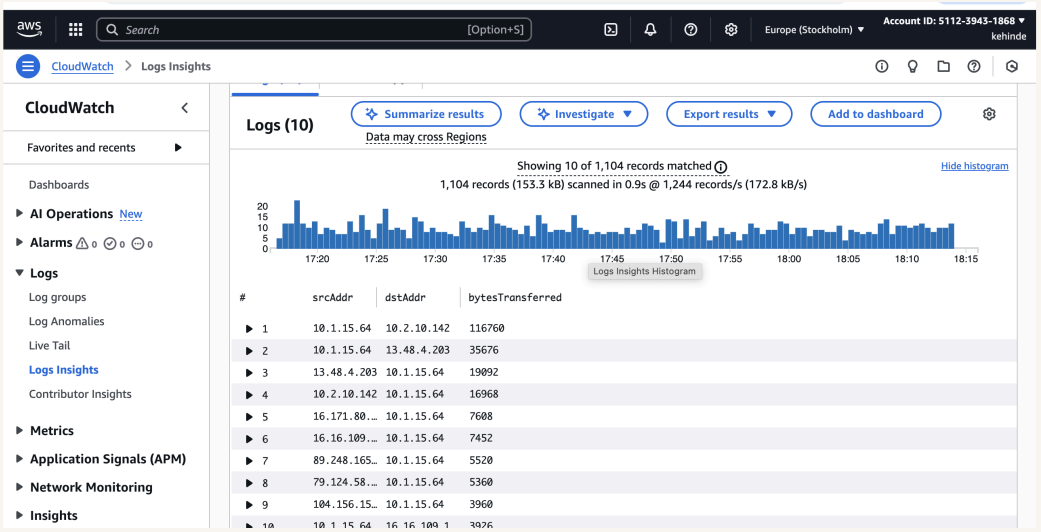
Kehinde Abiuwa
NextWork Student



Logs Insights

Logs Insights is a powerful query tool within Amazon CloudWatch that lets you search, analyze, and visualize log data stored in your log groups. It allows you to run custom queries to filter and extract specific information from large volumes of logs quickly. Logs Insights is useful for debugging applications, identifying performance issues, and tracking specific events in near real-time. It supports a SQL-like query language and can display results in both table and graph formats, making it easier to gain insights from raw log data.

I ran the query 'Top 10 byte transfers by source and destination IP addresses'. This query analyzes the flow log data to identify which source and destination IP address pairs transferred the most data over the selected time period. It helps reveal the heaviest traffic flows in the network by summing up the total bytes transferred between each pair and then listing the top 10 highest consumers of bandwidth.


aws
Search
[Option+S]
Europe (Stockholm)
Account ID: 5112-3943-1868
kehinde
CloudWatch
Logs Insights
CloudWatch
Favorites and recents
Dashboards
AI Operations New
Alarms
Logs
Log groups
Log Anomalies
Live Tail
Logs Insights
Contributor Insights
Metrics
Application Signals (APM)
Network Monitoring
Insights
Logs (10)
Summarize results
Investigate
Export results
Add to dashboard
Data may cross Regions
Showing 10 of 1,104 records matched
1,104 records (153.3 kB) scanned in 0.9s @ 1,244 records/s (172.8 kB/s)
Hide histogram
20
15
10
5
0
17:20
17:25
17:30
17:35
17:40
17:45
17:50
17:55
18:00
18:05
18:10
18:15
Logs Insights Histogram
srcAddr dstAddr bytesTransferred
1 10.1.15.64 10.2.10.142 116760
2 10.1.15.64 13.48.4.203 35676
3 13.48.4.203 10.1.15.64 19092
4 10.2.10.142 10.1.15.64 16968
5 16.171.80.… 10.1.15.64 7608
6 16.16.109.… 10.1.15.64 7452
7 89.248.165… 10.1.15.64 5520
8 79.124.58.… 10.1.15.64 5360
9 104.156.15… 10.1.15.64 3960

The place to learn & showcase your skills

Check out nextwork.org for more projects