



VPC Monitoring with Flow Logs



[Prajit Venkatachalam](#)

Logs (10)

Export results ▼

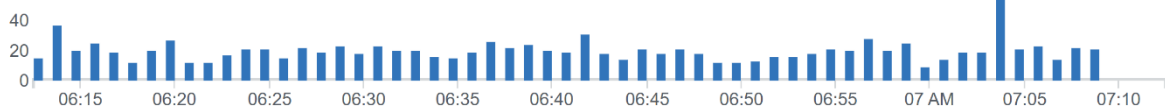
Add to dashboard



Showing 10 of 1,142 records matched ⓘ

[Hide histogram](#)

1,142 records (158.3 kB) scanned in 0.7s @ 1,629 records/s (225.8 kB/s)



#	srcAddr	dstAddr	bytesTransferred
▶ 1	10.1.13.23	13.239.158.4	137472
▶ 2	10.1.13.23	10.2.5.25	70728
▶ 3	13.239.158.4	10.1.13.23	65144
▶ 4	3.27.199.179	10.1.13.23	57456
▶ 5	10.1.13.23	3.27.199.179	57456
▶ 6	208.87.240.43	10.1.13.23	17691
▶ 7	99.83.81.77	10.1.13.23	14293
▶ 8	10.2.5.25	10.1.13.23	10500
▶ 9	10.1.13.23	99.83.81.77	7538
▶ 10	83.222.191.166	10.1.13.23	3400



Introducing Today's Project!

What is Amazon VPC?

Amazon VPC is a fundamental AWS resource that lets us control the underlying network for our resources, allowing us to manage traffic flow, monitor security, and organize resources efficiently.

How I used Amazon VPC in this project

In today's project, I used Amazon VPCs primarily to capture logs of network activity and traffic flow between two created VPCs, storing these logs in CloudWatch.

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

Logs Insights has numerous built-in queries that provide various options for analyzing network traffic.

This project took me...

This project took me 3.5 hours to complete include documentation.



In the first part of my project...

Step 1 Set up VPCs

In this step, I will set up two VPCs from scratch. Network monitoring can still be done with just a single VPC, but it's great to have the extra challenge and tackle VPC peering in this project.

Step 2 Launch EC2 instances

In this step, I will be launching two EC2 instances, one in each VPC. This is important to set up the remainder of our project. My EC2 instances will generate traffic so that the VPC flow logs can monitor it.

Step 3 Set up Logs

In this step, I will set up VPC flow logs to start monitoring network traffic. I will also set up a storage space for my flow logs.

Step 4 Set IAM permissions for Logs

In this step, I will grant VPC flow logs the permissions to create logs and upload them to my log group in CloudWatch.



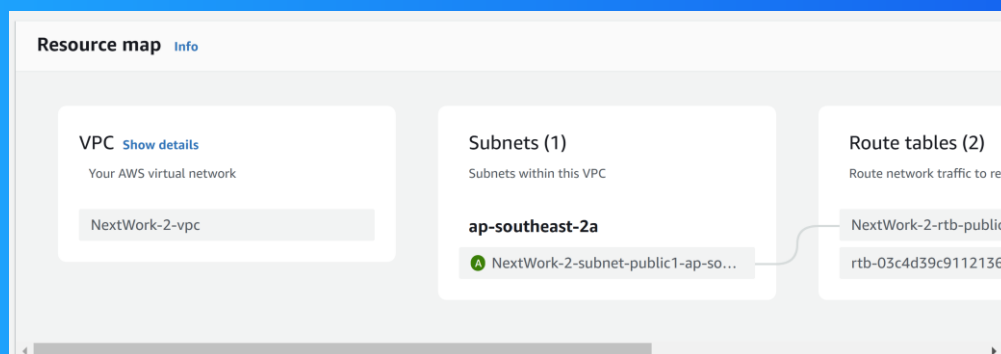
Multi-VPC Architecture

I started my project by launching two VPCs. I created two public subnets (one public subnet in each VPC) with no private subnets.

The CIDR blocks for VPCs 1 and 2 are 10.1.0.0/16 and 10.2.0.0/16. They must be unique because having overlapping CIDR blocks will cause network routing and traffic issues down the line when traffic needs to go from one VPC to another.

I also launched EC2 instances in each subnet

My EC2 instances' security groups allow SSH and ICMP traffic. This is because EC2 Instance Connect will need to access my EC2 instances using SSH, and I also need to allow ICMP traffic for connectivity tests later.



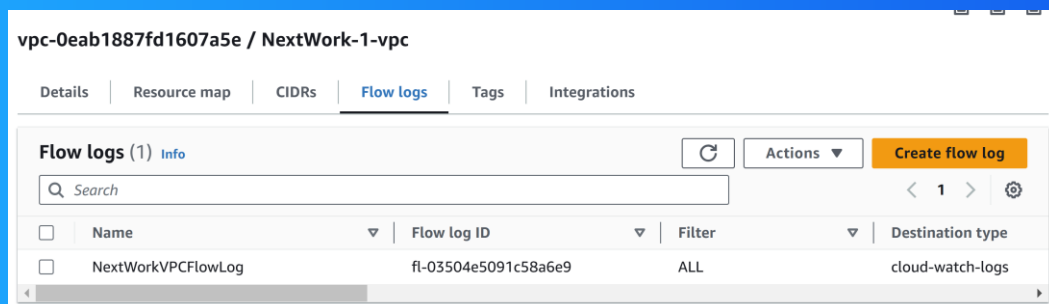


Logs

Logs are similar to diary entries for my computer systems; they provide detailed records of activities related to the traffic, resources, or AWS services being monitored.

Log groups are collections of logs that belong to a specific project, application, or source, often organized together within the same log group.

I also set up a flow log for VPC 1





IAM Policy and Roles

I created an IAM policy to define rules that grant policy holders, such as my VPC flow logs service, the ability to create log streams and upload them to CloudWatch.

I also created an IAM role because services like VPC flow logs need to be associated with a role rather than a JSON policy. Creating an IAM role is necessary to provide the access required for recording and uploading logs.

A custom trust policy is a specific type of policy that defines who or what is allowed to assume an IAM role.

Custom trust policy

Create a custom trust policy to enable others to perform actions in this account.

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Sid": "Statement1",  
6       "Effect": "Allow",  
7       "Principal": {  
8         "Service": "vpc-flow-logs.amazonaws.com"  
9       },  
10      "Action": "sts:AssumeRole"  
11    }  
12  ]  
13 }
```



In the second part of my project...

Step 5 Ping testing and troubleshooting

In this step, I am generating network traffic. This is important for discussions related to cloud networking and engineering.

Step 6 Set up a peering connection

In this step, I will set up a peering connection so that both VPCs can communicate with each other.

Step 7 Update VPC route tables

In this step, I will update the route tables for my two VPCs so that traffic destined for the other VPC can be directed through the peering connection instead of the public internet

Step 8 Analyze flow logs

In this step, I am tracking the network data collected from my VPCs and analyzing it to extract insights.

[illegible]

I was able to receive ping replies when I ran the ping test using the other second EC2 instance's public IP address, which indicates that my second instance is allowing ICMP traffic.

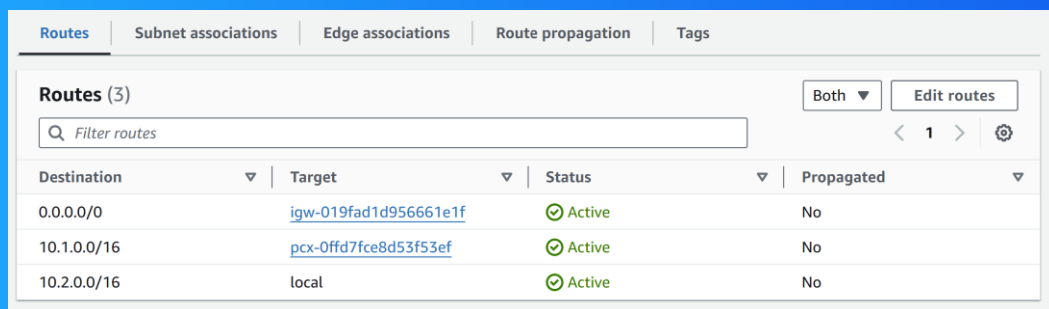


Connectivity troubleshooting

Upon reviewing VPC 1's route table, I found that the ping test using Instance 2's private address failed because I had not configured the route table to direct traffic between the two VPCs.

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

I also updated the route tables for both VPCs so that traffic from one VPC to the other's private address is directed through a peering connection rather than the public internet.



Routes	Subnet associations	Edge associations	Route propagation	Tags
Routes (3)				
<input type="text" value="Filter routes"/>				
Both Edit routes				
< 1 > ⚙				
Destination	Target	Status	Propagated	
0.0.0.0/0	igw-019fad1d956661e1f	Active	No	
10.1.0.0/16	pcx-0ffd7fce8d53f53ef	Active	No	
10.2.0.0/16	local	Active	No	



Connectivity troubleshooting

I received ping replies from Instance 2's private IP address! This means that setting up the peering connection and configuring the route table resolved the connectivity issue, allowing traffic to navigate between the two VPCs.

```
--- 10.2.5.25 ping statistics ---
1998 packets transmitted, 125 received, 93.7437% packet loss, time 2076568ms
rtt min/avg/max/mdev = 0.415/0.718/1.575/0.237 ms
[ec2-user@ip-10-1-13-23 ~]$ ping 3.27.199.179
PING 3.27.199.179 (3.27.199.179) 56(84) bytes of data.
64 bytes from 3.27.199.179: icmp_seq=1 ttl=126 time=1.02 ms
64 bytes from 3.27.199.179: icmp_seq=2 ttl=126 time=0.917 ms
64 bytes from 3.27.199.179: icmp_seq=3 ttl=126 time=0.512 ms
64 bytes from 3.27.199.179: icmp_seq=4 ttl=126 time=0.755 ms
64 bytes from 3.27.199.179: icmp_seq=5 ttl=126 time=0.880 ms
64 bytes from 3.27.199.179: icmp_seq=6 ttl=126 time=0.546 ms
64 bytes from 3.27.199.179: icmp_seq=7 ttl=126 time=0.601 ms
64 bytes from 3.27.199.179: icmp_seq=8 ttl=126 time=0.777 ms
64 bytes from 3.27.199.179: icmp_seq=9 ttl=126 time=0.491 ms
64 bytes from 3.27.199.179: icmp_seq=10 ttl=126 time=1.11 ms
64 bytes from 3.27.199.179: icmp_seq=11 ttl=126 time=0.925 ms
64 bytes from 3.27.199.179: icmp_seq=12 ttl=126 time=0.549 ms
64 bytes from 3.27.199.179: icmp_seq=13 ttl=126 time=0.769 ms
64 bytes from 3.27.199.179: icmp_seq=14 ttl=126 time=0.456 ms
```



Analyzing flow logs

Flow logs tell us about the source and destination of the network traffic, the amount of data being transferred, and whether the traffic was accepted or rejected.

For example, the flow log I've captured shows that traffic went from 220.132.14.214 to 10.1.13.23. It also indicates the traffic accepted by the security groups and network ACLs of my VPC.

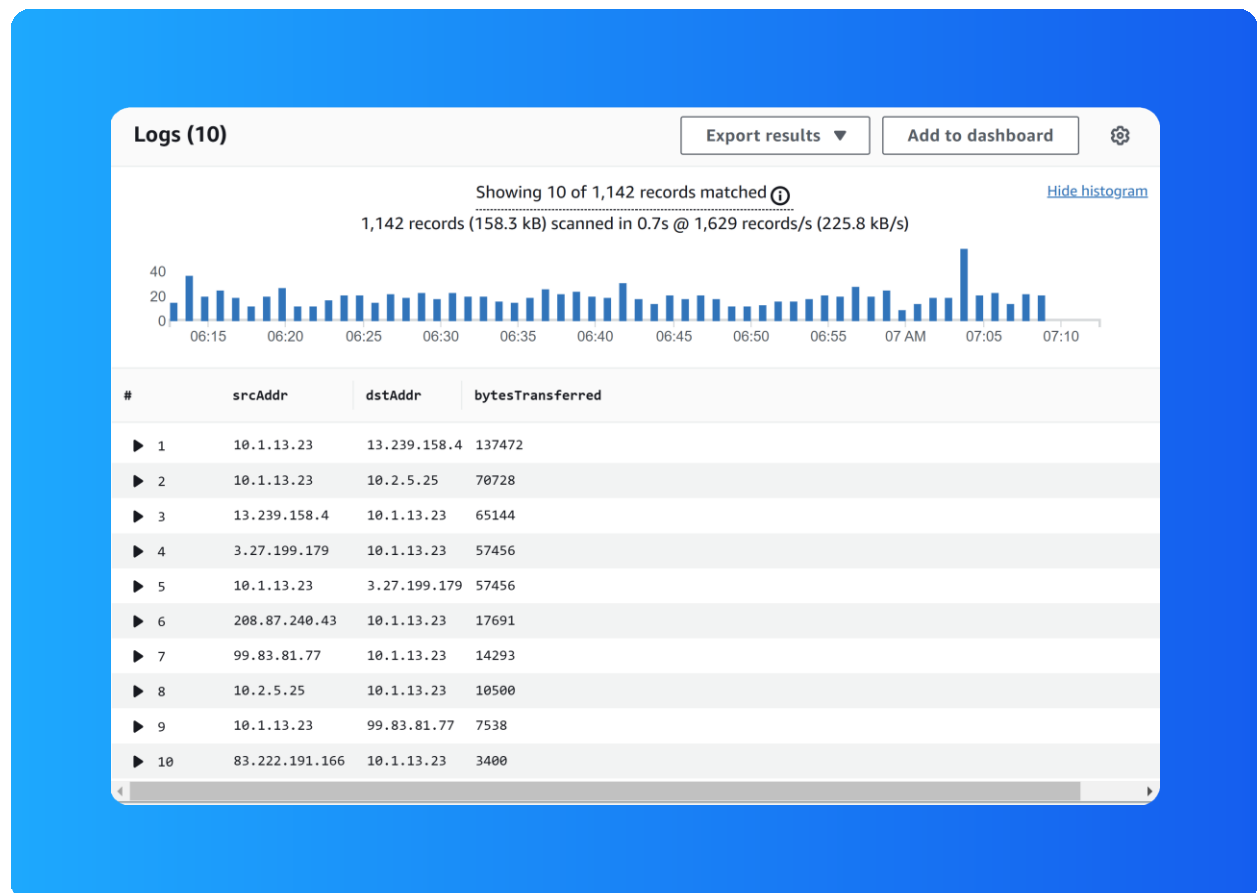
Timestamp	Message
	There are older events to load. Load more
2024-10-28T06:34:49.000Z	2 637423622277 eni-ef5598706acc783 205.230.31.3 10.1.13.23 54034 8140 6 1 40 1730097289 1730097345 REJECT OK
2024-10-28T06:34:49.000Z	2 637423622277 eni-ef5598706acc783 137.184.227.70 10.1.13.23 57440 2222 6 1 40 1730097289 1730097345 REJECT OK
2024-10-28T06:34:49.000Z	2 637423622277 eni-ef5598706acc783 13.239.158.4 10.1.13.23 37352 22 6 62 1360 1730097289 1730097345 ACCEPT OK
2024-10-28T06:34:49.000Z	2 637423622277 eni-ef5598706acc783 10.1.13.23 13.239.158.4 22 37352 6 60 8992 1730097289 1730097345 ACCEPT OK
2024-10-28T06:34:49.000Z	2 637423622277 eni-ef5598706acc783 220.132.14.214 10.1.13.23 57830 34567 0 1 40 1730097289 1730097345 REJECT OK
2024-10-28T06:34:49.000Z	2 637423622277 eni-ef5598706acc783 89.82.77.144 10.1.13.23 49088 5058 6 1 40 1730097289 1730097345 REJECT OK
2024-10-28T06:34:49.000Z	2 637423622277 eni-ef5598706acc783 83.222.191.166 10.1.13.23 54288 12388 6 1 40 1730097289 1730097345 REJECT OK
2024-10-28T06:35:45.000Z	2 637423622277 eni-ef5598706acc783 3.27.199.179 10.1.13.23 0 1 58 4872 1730097345 1730097405 ACCEPT OK
2024-10-28T06:35:45.000Z	2 637423622277 eni-ef5598706acc783 10.1.13.23 3.27.199.179 0 1 58 4872 1730097345 1730097405 ACCEPT OK
2024-10-28T06:35:45.000Z	2 637423622277 eni-ef5598706acc783 115.231.78.14 10.1.13.23 7669 9808 6 1 44 1730097345 1730097405 REJECT OK
2024-10-28T06:35:45.000Z	2 637423622277 eni-ef5598706acc783 162.142.125.89 10.1.13.23 11954 5351 17 1 12 1730097345 1730097405 Back to top



Logs Insights

Logs Insights is a specialized tool within Amazon CloudWatch that helps with analyzing logs and creating visual graphs and charts through queries.

I ran the query Top 10 byte transfers by source and destination IP addresses. This query analyzes the flow logs collected on EC2 instance 1 and returns the top 10 pairs of IP addresses based on the amount of data transferred between them.





NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

