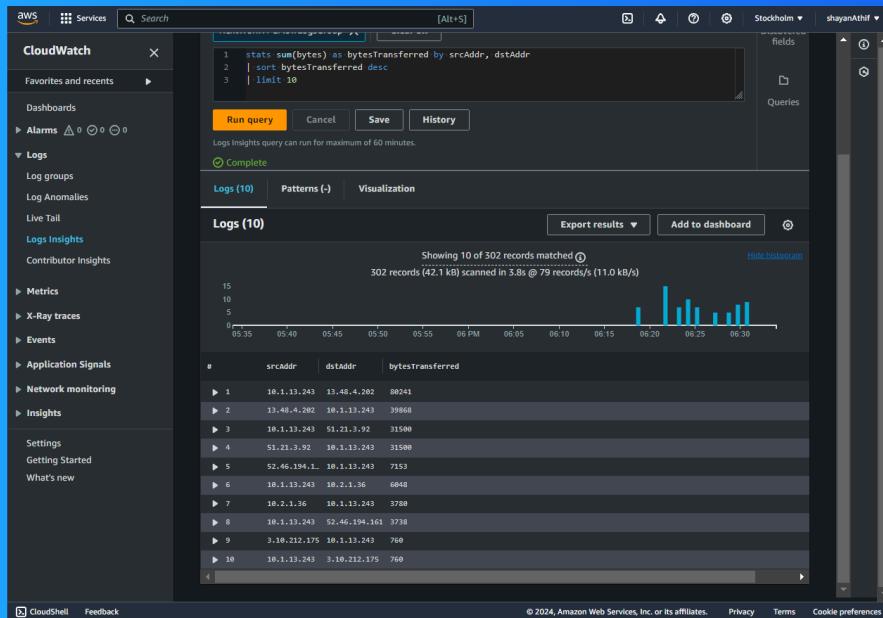




VPC Monitoring with Flow Logs



sirajudeen athif



S

sirajudeen athif
NextWork Student

NextWork.org

Introducing Today's Project!

What is Amazon VPC?

Amazon VPC is a service that allows users to create private networks within the AWS cloud. It provides enhanced security, customizable network configurations and easy scalability.

How I used Amazon VPC in this project

I used Amazon VPC to create two virtual private cloud (VPC), established a peering connection between them and monitored the network.

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

That it is very simple to use AWS, given you have some practice.

This project took me...

Around 2 hours.

In the first part of my project...

Step 1 - Set up VPCs

Create VPCs from scratch, establish VPC Peering Connections between them, and set up monitoring to track traffic and analyze network performance.

Step 2 - Launch EC2 instances

Launch EC2 instances in both VPCs to establish a peering connection and then, monitor network traffic and performance between them.

Step 3 - Set up Logs

We will enable AWS CloudWatch Logs to capture and record all network traffic, including requests and responses, for monitoring and analysis purposes.

Step 4 - Set IAM permissions for Logs

We're granting VPC Flow Logs permission to write logs and send them to CloudWatch, allowing us to capture and store network traffic data.

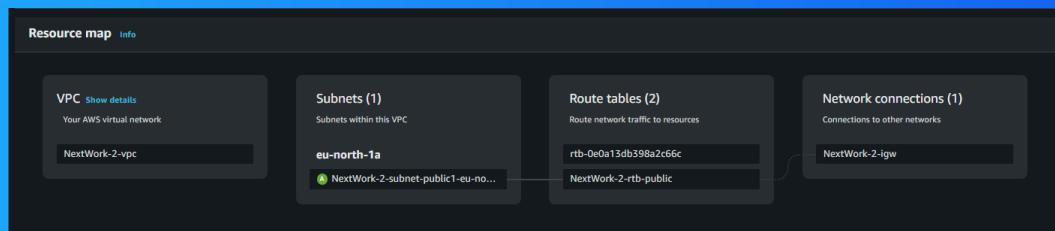
Multi-VPC Architecture

I launched 2 VPCs, each with a single Public Subnet, and configured them for use.

The CIDR blocks for VPCs 1 and 2 are 10.1.0.0/16 and 10.2.0.0/16, respectively. They must be unique because AWS reserves one IPv4 CIDR block per account, ensuring each network gets its own IP address range.

I also launched EC2 instances in each subnet

Initially, Security Group rules only allowed SSH inbound traffic. We added an explicit ICMP rule to enable pinging between instances, allowing for later network testing and verification.

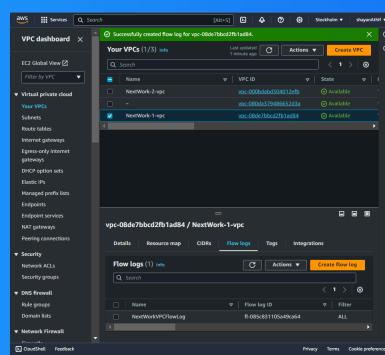


Logs

Logs act as a detailed account of all system activities, recording every event, interaction, and transaction that occurs within the system.

Log Groups are folders where AWS keeps related logs together, making it easy to organize and view the logs for my system.

I also set up a flow log for VPC 1

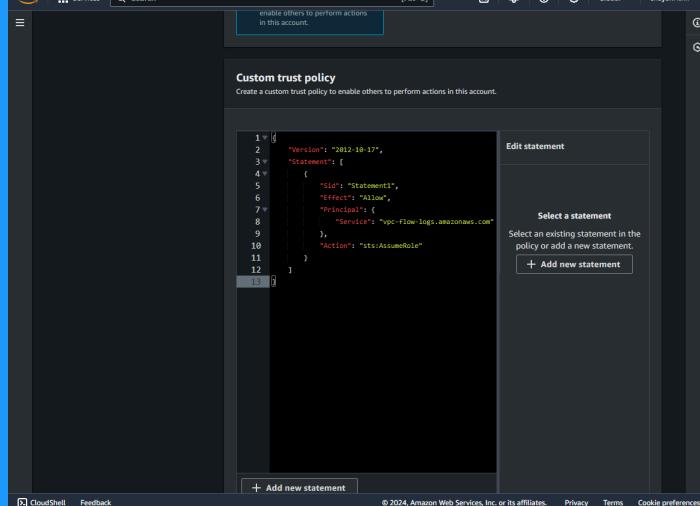


IAM Policy and Roles

By default, VPC Flow Logs can't write logs to CloudWatch. We need to give it permissions to do so, ensuring our VPC can now send its log data to the correct CloudWatch log group.

I created an IAM role because you can't assign an IAM policy directly to a service, you have to assign ROLES to a service.

Custom Trust Policies are a specific type of policy that allows us to precisely control who can use an IAM Role.



In the second part of my project...

Step 5 - Ping testing and troubleshooting

We're setting up a VPC Peering Connection to create a secure, private link between the two VPCs, bypassing the internet and allowing them to communicate directly with each other.

Step 6 - Set up a peering connection

we will be establishing the peering connection between the two VPCs.

Step 7 - Update VPC route tables

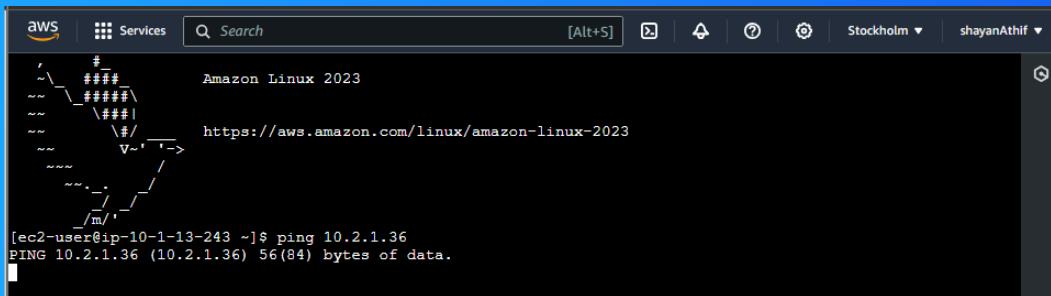
Now that the peering connection is all set up, we now need to set up a way for the traffic coming from both the VPCs.

Step 8 - Analyze flow logs

we will be reviewing the flow logs recorded by the VPCs public subnet.

Connectivity troubleshooting

The first ping test didn't work, meaning I sent the ICMP request but got no response from the other EC2 instance, indicating a connectivity issue.



A screenshot of a terminal window on an Amazon Linux 2023 instance. The window title bar shows 'Amazon Linux 2023' and the URL 'https://aws.amazon.com/linux/amazon-linux-2023'. The terminal prompt is '[ec2-user@ip-10-1-13-243 ~]\$'. The user runs the command 'ping 10.2.1.36', which returns the output 'PING 10.2.1.36 (10.2.1.36) 56(84) bytes of data.'.

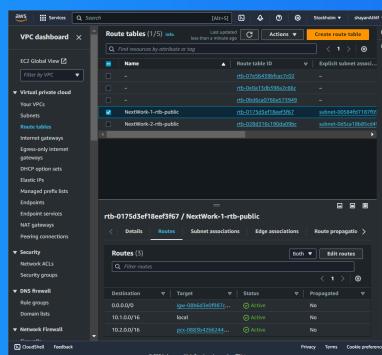
Running the ping test with the other instance's public IP allowed replies, confirming Instance 2 responds. This enables communication between both the instances using public IPs.

Connectivity troubleshooting

In VPC 1's route table, a route to 0.0.0.0/0 was found that directs traffic from anywhere, including Instance 2, through a public gateway. This prevents a direct connection and causes the ping test to fail.

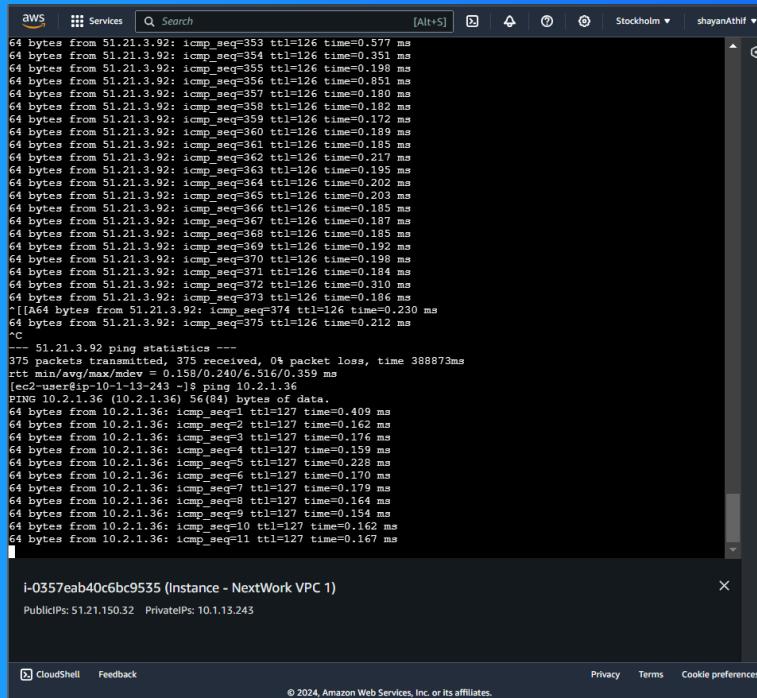
To solve this, I set up a peering connection between my VPCs

I also updated both VPCs' route tables because even with a Peering Connection established, there's no direct route from VPC 1 to VPC 2 unless a route in the route table of VPC 1 was added.



Connectivity troubleshooting

I received ping replies from Instance 2's private IP address! This means we have successfully created a connection between the two VPCs.



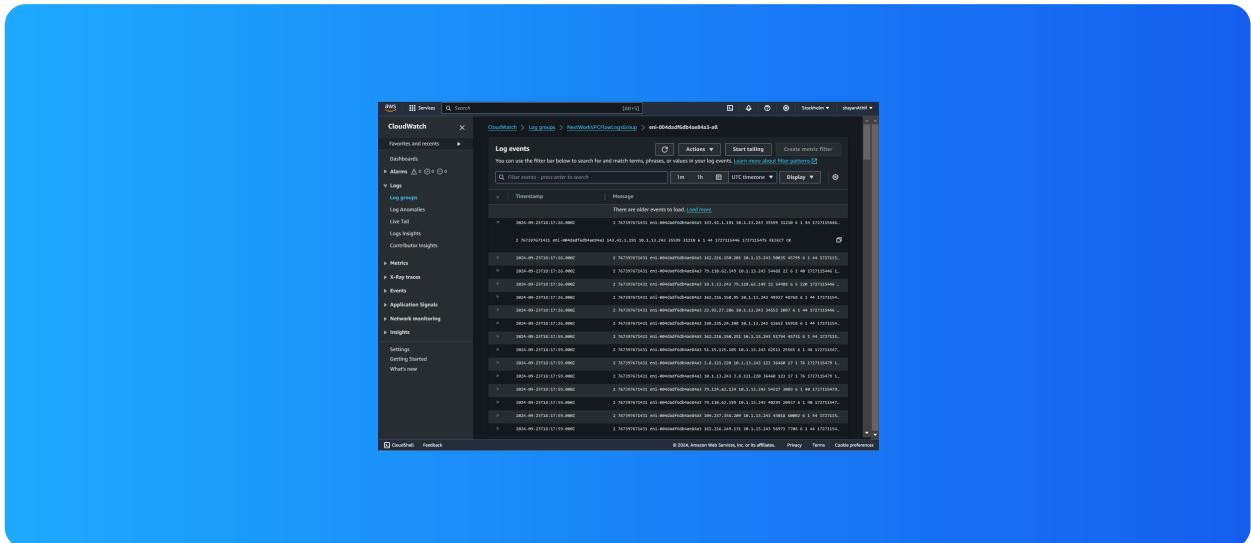
The screenshot shows a terminal window in AWS CloudShell with the following output:

```
aws Services Search [Alt+S] Stockholm shayanAthif
64 bytes from 51.21.3.92: icmp_seq=353 ttl=126 time=0.577 ms
64 bytes from 51.21.3.92: icmp_seq=354 ttl=126 time=0.361 ms
64 bytes from 51.21.3.92: icmp_seq=355 ttl=126 time=0.198 ms
64 bytes from 51.21.3.92: icmp_seq=356 ttl=126 time=0.851 ms
64 bytes from 51.21.3.92: icmp_seq=357 ttl=126 time=0.180 ms
64 bytes from 51.21.3.92: icmp_seq=358 ttl=126 time=0.182 ms
64 bytes from 51.21.3.92: icmp_seq=359 ttl=126 time=0.172 ms
64 bytes from 51.21.3.92: icmp_seq=360 ttl=126 time=0.189 ms
64 bytes from 51.21.3.92: icmp_seq=361 ttl=126 time=0.185 ms
64 bytes from 51.21.3.92: icmp_seq=362 ttl=126 time=0.217 ms
64 bytes from 51.21.3.92: icmp_seq=363 ttl=126 time=0.195 ms
64 bytes from 51.21.3.92: icmp_seq=364 ttl=126 time=0.209 ms
64 bytes from 51.21.3.92: icmp_seq=365 ttl=126 time=0.193 ms
64 bytes from 51.21.3.92: icmp_seq=366 ttl=126 time=0.195 ms
64 bytes from 51.21.3.92: icmp_seq=367 ttl=126 time=0.187 ms
64 bytes from 51.21.3.92: icmp_seq=368 ttl=126 time=0.185 ms
64 bytes from 51.21.3.92: icmp_seq=369 ttl=126 time=0.192 ms
64 bytes from 51.21.3.92: icmp_seq=370 ttl=126 time=0.198 ms
64 bytes from 51.21.3.92: icmp_seq=371 ttl=126 time=0.184 ms
64 bytes from 51.21.3.92: icmp_seq=372 ttl=126 time=0.310 ms
64 bytes from 51.21.3.92: icmp_seq=373 ttl=126 time=0.186 ms
^[[A64 bytes from 51.21.3.92: icmp_seq=374 ttl=126 time=0.230 ms
64 bytes from 51.21.3.92: icmp_seq=375 ttl=126 time=0.212 ms
^[[A
-- 51.21.3.92 ping statistics --
375 packets transmitted, 0% packet loss, time 388873ms
rtt min/avg/max/mdev = 0.158/0.240/6.516/0.359 ms
[ec2-user@ip-10-1-13-243 ~]$ ping 10.2.1.36
PING 10.2.1.36 (10.2.1.36) 56(84) bytes of data.
64 bytes from 10.2.1.36: icmp_seq=1 ttl=127 time=0.409 ms
64 bytes from 10.2.1.36: icmp_seq=2 ttl=127 time=0.162 ms
64 bytes from 10.2.1.36: icmp_seq=3 ttl=127 time=0.176 ms
64 bytes from 10.2.1.36: icmp_seq=4 ttl=127 time=0.159 ms
64 bytes from 10.2.1.36: icmp_seq=5 ttl=127 time=0.151 ms
64 bytes from 10.2.1.36: icmp_seq=6 ttl=127 time=0.170 ms
64 bytes from 10.2.1.36: icmp_seq=7 ttl=127 time=0.179 ms
64 bytes from 10.2.1.36: icmp_seq=8 ttl=127 time=0.164 ms
64 bytes from 10.2.1.36: icmp_seq=9 ttl=127 time=0.154 ms
64 bytes from 10.2.1.36: icmp_seq=10 ttl=127 time=0.162 ms
64 bytes from 10.2.1.36: icmp_seq=11 ttl=127 time=0.167 ms
^[[A
i-0357eab40c6bc9535 (Instance - NextWork VPC 1)
PublicIPs: 51.21.150.32 PrivateIPs: 10.1.13.243
CloudShell Feedback Privacy Terms Cookie preferences
© 2024, Amazon Web Services, Inc. or its affiliates.
```

Analyzing flow logs

Flow logs provide detailed network metrics, including source/destination IP addresses, protocols, packet lengths, and more.

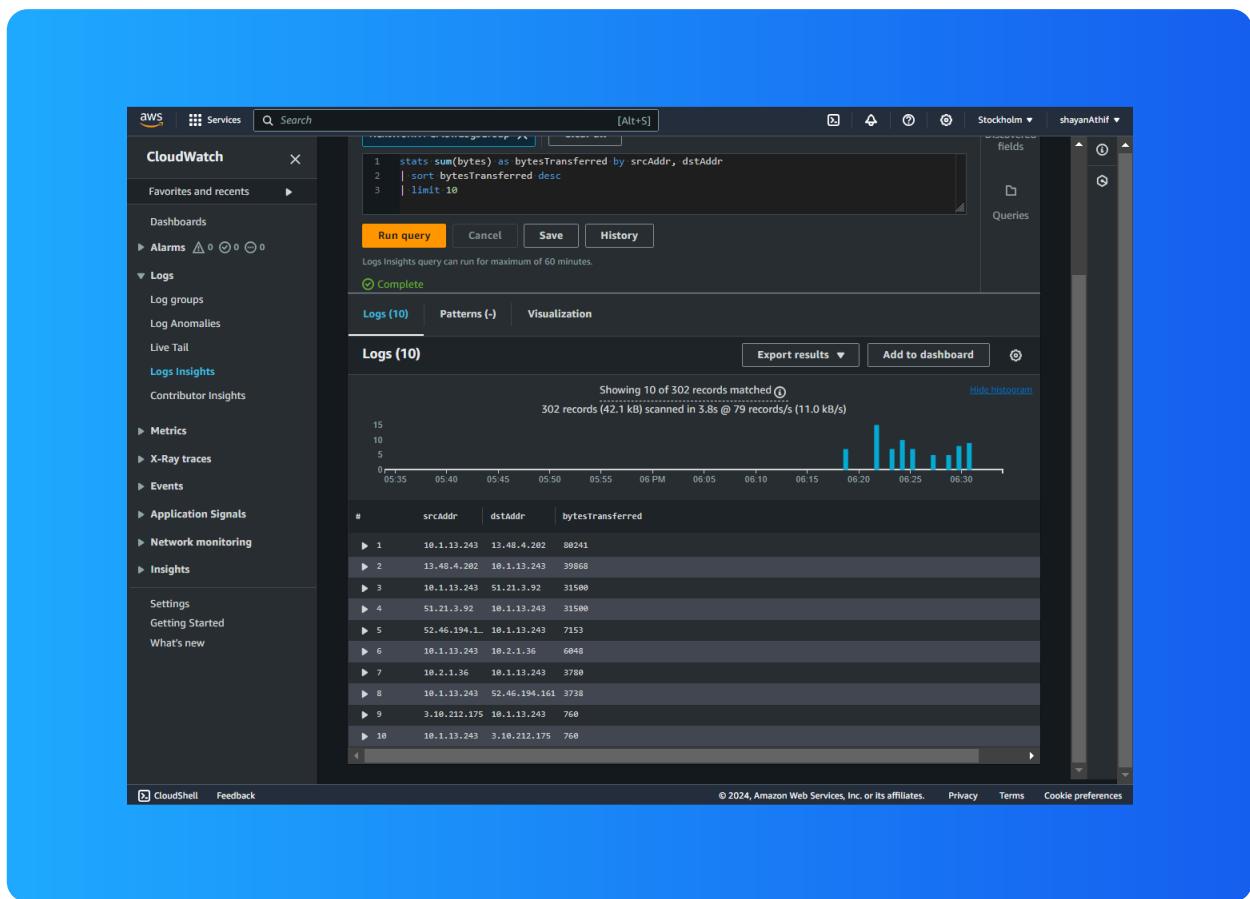
For example, the flow log I've captured tells us that it was an unsuccessful data transfer: 44 bytes via TCP (port 22) from 143.42.1.191 to 10.1.13.243, with 1 packet, in "REJECT" traffic category.



Logs Insights

Log Insights uses CloudWatch queries to filter, analyze, and combine log data, helping with troubleshooting and understanding network traffic patterns.

I ran the query to select the top 10 byte transfers by source and destination IP addresses. This query analyzes the top 10 biggest data transfers between IP addresses in this network.





NextWork.org

Everyone should be in a job they love.

Check out nextwork.org for
more projects

