Monitoring and Analyzing VPC Traffic Using AWS CloudWatch and Flow Logs

In this project, I set up two VPCs and established VPC Peering to enable communication between them. I configured VPC Flow Logs to collect network traffic data and used CloudWatch Log Insights to analyze the traffic. Additionally, I will be launching two EC2 instances to further test and validate the connectivity between the peered VPCs.

Let's Start the Project →

We are logged in as a **IAM user**.

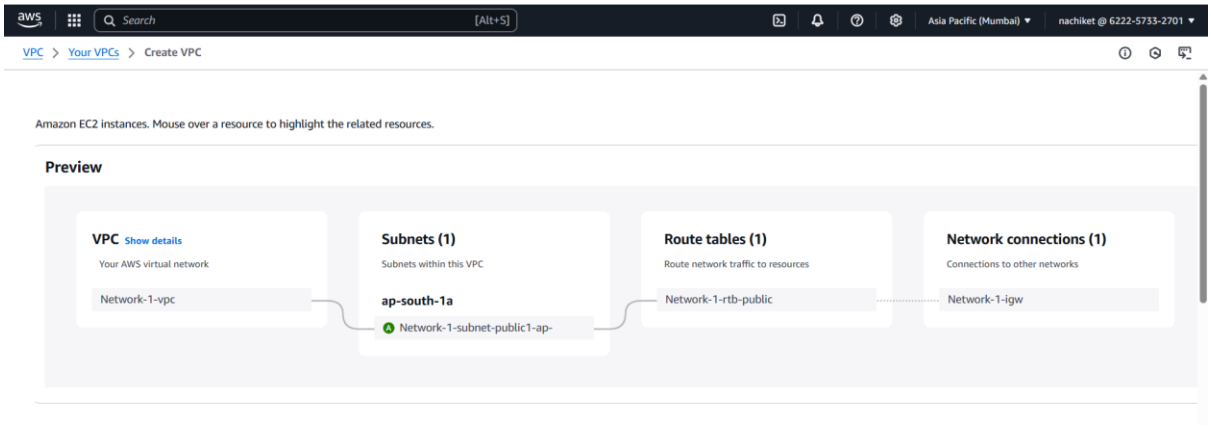
Step 1: Setup the 2 VPC's.

What is Amazon VPC?

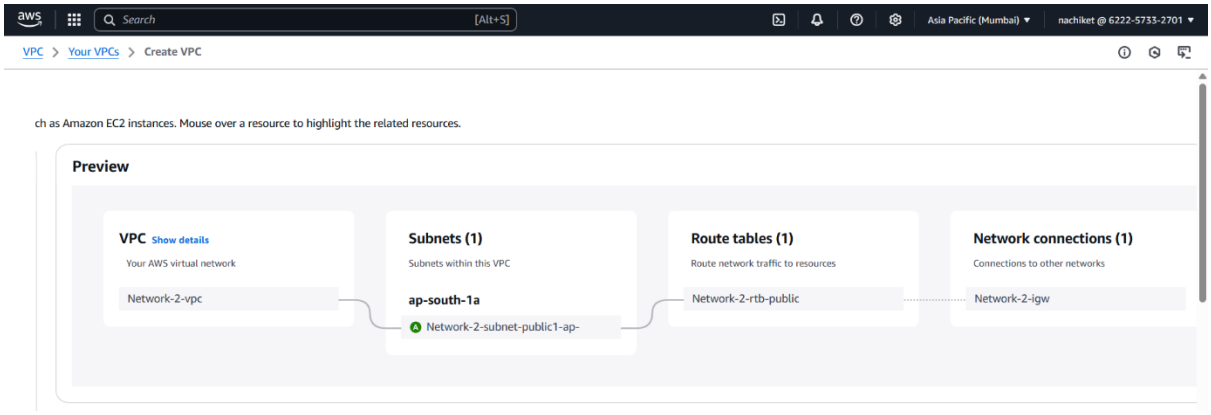
➔ Amazon VPC (Virtual Private Cloud) is a private network inside AWS where you can launch and manage resources securely. Think of it as your own isolated section of AWS, where you control networking, IP addresses, security, and how your resources communicate.

I navigated to the VPC section in AWS and selected the option to create a new VPC. I provided a name for the VPC and assigned the IP address range 10.1.0.0/16. I configured the following settings:

- Number of Availability Zones: 1
- Number of Public Subnets: 1
- Number of Private Subnets: 0
- VPC Endpoint: None
- NAT Gateway: None
- After configuring these settings, I successfully created the VPC.
- Create the subnet for respective VPC.



Now, I created a second VPC with the IP address range 10.2.0.0/16.



Why does each VPC need a unique IP address range?

➔ If both VPCs had the same CIDR block, it would cause an IP address overlap, leading to routing and traffic issues when communication occurs between them. By assigning unique IP ranges, we ensure proper routing and avoid conflicts.

Step 2: Launch 2 EC2 Instances. (EC2 Instance will generate the traffic)

- Navigate to the EC2 Instances section and click on Launch Instance.
- Provide a name for the instance and select a Linux AMI. In this project, I used Amazon Linux.
- Choose an instance type – I selected t2.micro.
- Use the default key pair for SSH access.
- In network settings, select VPC1.
- Enable Auto-assign Public IP to allow internet access.
- Create a new security group, provide a name, and configure the following security rules:
    - SSH (port 22) – Allow from 0.0.0.0/0
    - All ICMP - IPv4 – Allow from 0.0.0.0/0 (for pinging between instances)
- Launch the instance.

Similarly Launch the 2nd Instance.

Step 3: Setup Flow Logs

What is Amazon VPC Flow Logs?

➔ Amazon **VPC Flow Logs** is a feature that **captures and records network traffic** going **to and from** your **VPC, subnets, or network interfaces**. It helps you **monitor, troubleshoot, and analyze** network activity for security and performance insights.

- Search for CloudWatch in the AWS console and open it.
- Navigate to the Logs section and select Log Groups.
- Click on Create Log Group.
- Provide a name for the log group.
- Set Retention Settings to Never Expire.
- Choose Log Class as Standard.
- Click on Create to finalize the log group.

What is Log?

➔ Log keeps records of everything that happens. For example – User Logging, Error, Accepted Traffic, Denied Traffic.

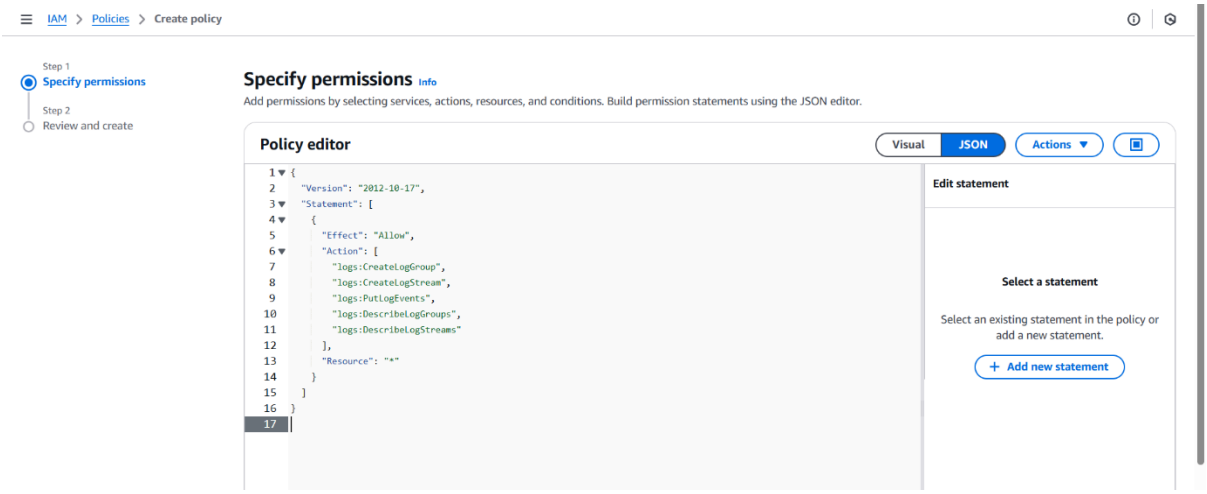
- Go to the VPC page in the AWS console.
- Select the VPC1 that you created.
- Inside the VPC settings, navigate to the Flow Logs section.
- Click on Create Flow Log.
- Provide a name for the Flow Log.
- Set Filter to All (to capture all traffic: accepted, rejected, and both).
- Set Maximum Aggregation Interval to 1 minute.
- Choose Destination as Send to CloudWatch Logs.
- Select the Log Group you created earlier in CloudWatch.
- Click Create to finalize the Flow Log setup.

Why doesn’t it have permission to create and upload logs to CloudWatch?

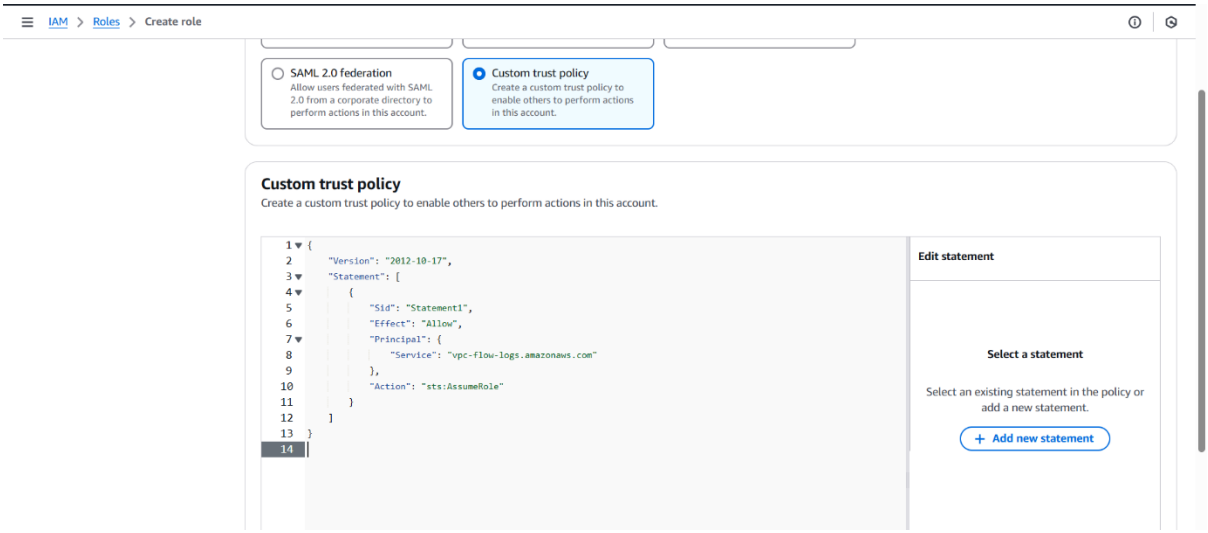
➔ The issue is due to IAM policies and roles. Without the proper IAM role and permissions, VPC Flow Logs cannot send data to CloudWatch Logs. To resolve this, an IAM role with the required policy must be attached to grant the necessary permissions.

Step 4: Create IAM role and Policy.

- Go to the IAM page in the AWS console.
- Navigate to the Policies section and click on Create Policy.
- Select the JSON tab to define the policy using JSON format.
- Enter the required permissions in JSON format.
- Click Next, review the policy, and provide a name for it.
- Click Create Policy to finalize.



- Navigate to the Roles section and click on Create Role.
- Under Trusted Entity Type, select Custom Trust Policy.
- Enter the custom trust policy code (specific to VPC Flow Logs).
- Click Next and search for VPC Flow Logs in the permissions section.
- Select the appropriate permissions for VPC Flow Logs.
- Provide a name for the role.
- Click Create Role to complete the process.



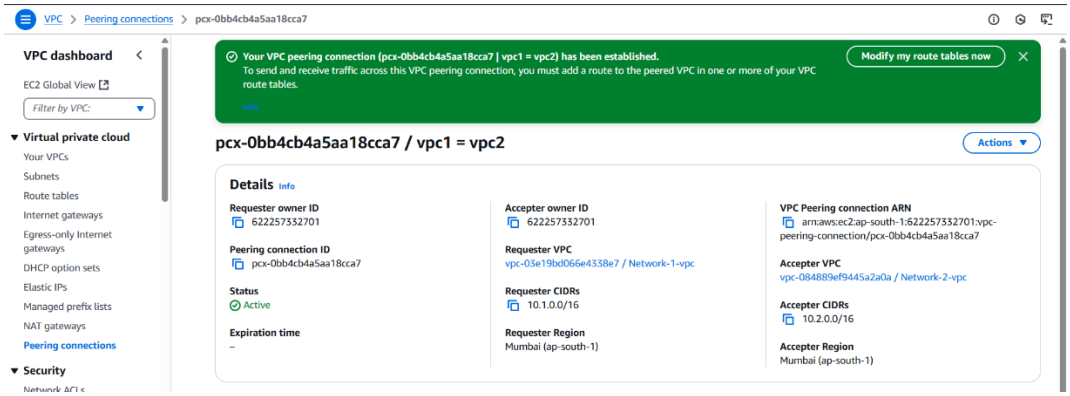
Now, go to the VPC section. Search for the IAM role we just created in the IAM section of the VPC. Then, create a Flow Log using this role.

Flow Log is all setup! This means network traffic going into and out of your VPC is now getting tracked.

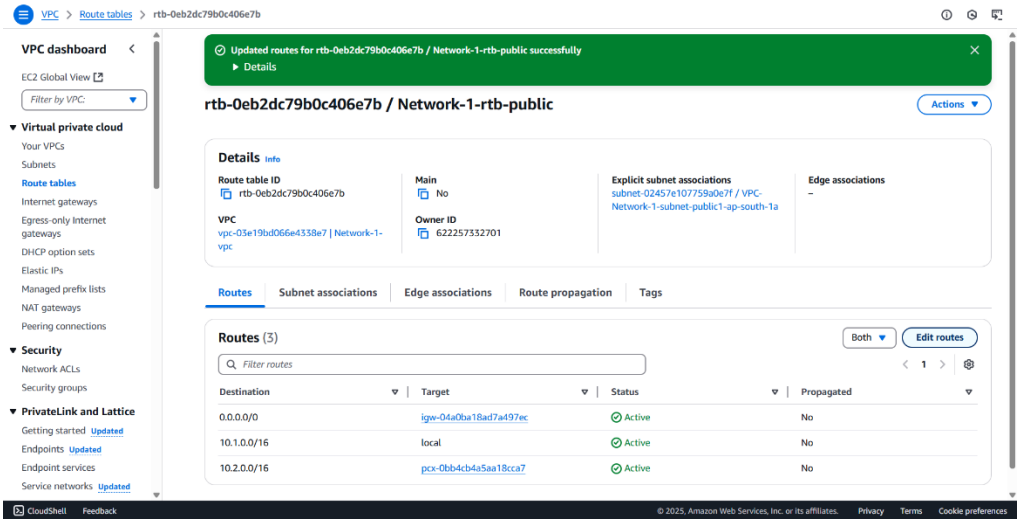
Step 5: VPC Peering

- Go to the VPC tab in the AWS console.
- Navigate to Peering Connections and create a new peering connection.
- Select the Sender VPC and Receiver VPC, then initiate the request.
- Accept the Peering Request from the receiving VPC.
- Update the Route Tables for both VPCs:
    - Add the IP address range of the second VPC in the first VPC's route table.
    - Add the IP address range of the first VPC in the second VPC's route table.
- Now, the private IPs of instances in both VPCs can communicate with each other.

Peering Connection Successful



Route Table Update

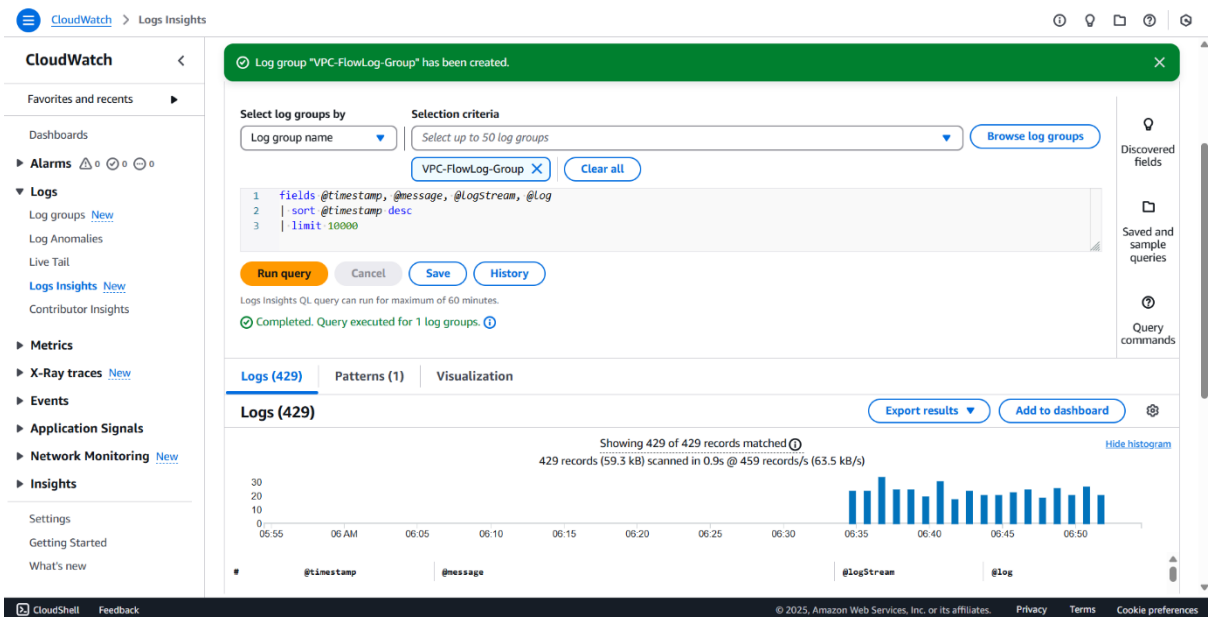


Accessing Private IP Address

```
[ec2-user@ip-10-1-2-57 ~]$ ping 3.109.186.45
PING 3.109.186.45 (3.109.186.45) 56(84) bytes of data.
64 bytes from 3.109.186.45: icmp_seq=1 ttl=126 time=2.85 ms
64 bytes from 3.109.186.45: icmp_seq=2 ttl=126 time=0.470 ms
64 bytes from 3.109.186.45: icmp_seq=3 ttl=126 time=0.422 ms
64 bytes from 3.109.186.45: icmp_seq=4 ttl=126 time=0.516 ms
64 bytes from 3.109.186.45: icmp_seq=5 ttl=126 time=0.462 ms
64 bytes from 3.109.186.45: icmp_seq=6 ttl=126 time=0.471 ms
64 bytes from 3.109.186.45: icmp_seq=7 ttl=126 time=0.476 ms
64 bytes from 3.109.186.45: icmp_seq=8 ttl=126 time=2.42 ms
64 bytes from 3.109.186.45: icmp_seq=9 ttl=126 time=0.521 ms
64 bytes from 3.109.186.45: icmp_seq=10 ttl=126 time=0.554 ms
64 bytes from 3.109.186.45: icmp_seq=11 ttl=126 time=0.542 ms
64 bytes from 3.109.186.45: icmp_seq=12 ttl=126 time=0.514 ms
64 bytes from 3.109.186.45: icmp_seq=13 ttl=126 time=0.466 ms
64 bytes from 3.109.186.45: icmp_seq=14 ttl=126 time=0.520 ms
64 bytes from 3.109.186.45: icmp_seq=15 ttl=126 time=0.519 ms
^C
--- 3.109.186.45 ping statistics ---
15 packets transmitted, 15 received, 0% packet loss, time 14550ms
rtt min/avg/max/mdev = 0.422/0.782/2.853/0.733 ms
[ec2-user@ip-10-1-2-57 ~]$
```

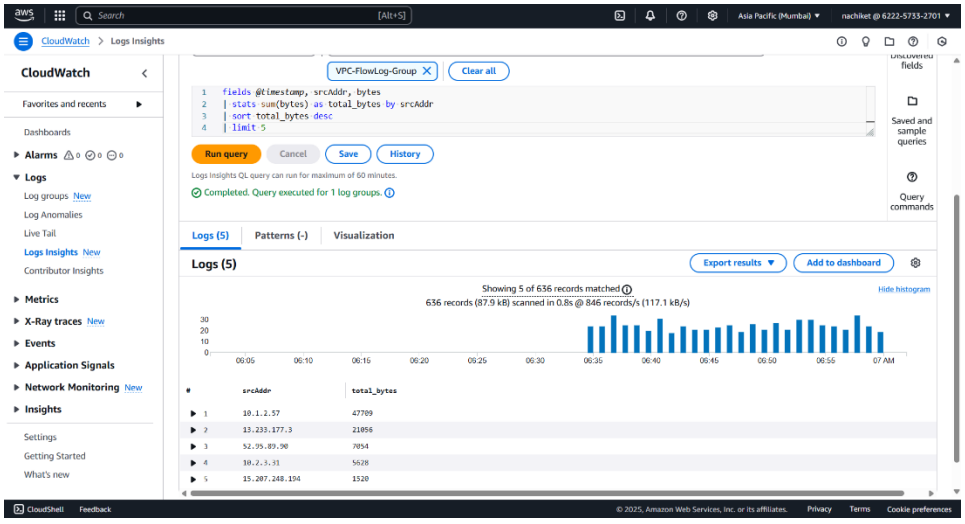
Step 8: Analyse the Flow Logs.

- Go to the VPC Flow Logs section.
- Open CloudWatch Logs Insights and select the log group where Flow Logs are stored.
- Run queries based on your requirements.



List of queries which I run :

1. Top 5 IP’s sending request.



2. Accepted IP’s

3.