AWS VPC Network Monitoring

By: Ritvik Indupuri

1. Project Overview

This project focuses on establishing a secure and observable network infrastructure within Amazon Web Services (AWS). It demonstrates the deployment of a multi-VPC environment, secure inter-VPC communication via peering, and the implementation of real-time network traffic monitoring using VPC Flow Logs integrated with Amazon CloudWatch. The objective was to gain deep insights into network behavior for security auditing, troubleshooting, and operational visibility.

---

2. Key Concepts

- **Amazon Virtual Private Cloud (VPC):** A logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define.
- **Amazon EC2 (Elastic Compute Cloud):** A web service that provides resizable compute capacity in the cloud. Used here to deploy virtual servers (instances) within the VPCs.
- **VPC Peering Connection:** A networking connection between two VPCs that enables you to route traffic directly between them using private IPv4 addresses.
- **VPC Flow Logs:** A feature that enables you to capture information about the IP traffic going to and from network interfaces in your VPC.
- **Amazon CloudWatch:** A monitoring and observability service that provides data and actionable insights for AWS, hybrid, and on-premises applications and infrastructure resources.

- **CloudWatch Logs Insights:** A fully integrated, interactive query service that enables you to quickly and efficiently explore, analyze, and visualize your log data.
- **AWS Identity and Access Management (IAM):** A web service that helps you securely control access to AWS resources. Used to manage permissions for services like VPC Flow Logs.

---

3. Architecture Overview

The project architecture consists of two distinct VPCs, each hosting an EC2 instance within a public subnet. A VPC peering connection facilitates private communication between these two VPCs. VPC Flow Logs are configured on one of the public subnets to capture all network traffic, which is then sent to a dedicated CloudWatch Log Group for analysis using CloudWatch Logs Insights.

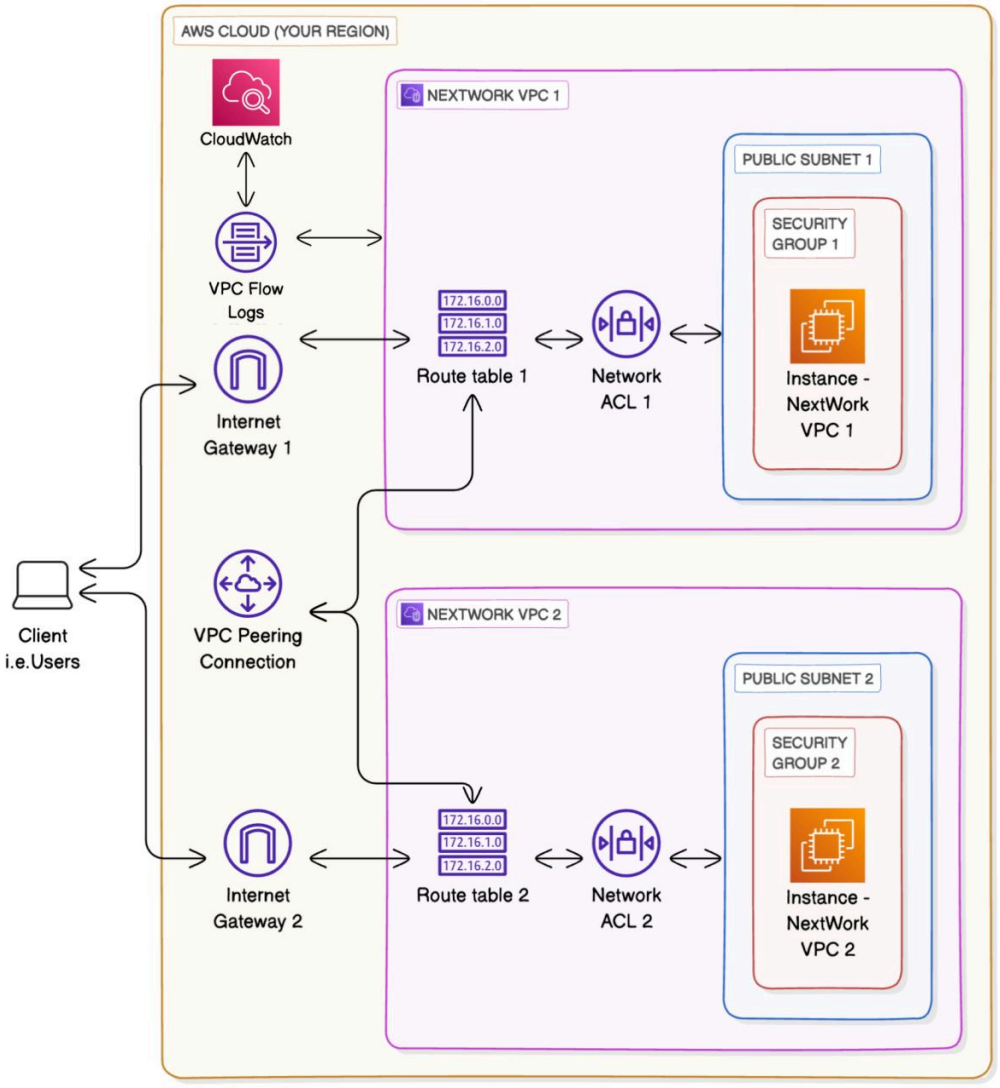

Figure 1: Architectural overview of the project

4. Implementation Steps

4.1. VPC Setup

Two VPCs were created to simulate a multi-network environment:

- **Homelab-1-vpc:** Configured with an IPv4 CIDR block of 10.1.0.0/16 and a single public subnet.
- **Homelab-2-vpc:** Configured with an IPv4 CIDR block of 10.2.0.0/16 and a single public

subnet.

Each VPC was set up using the "VPC and more" wizard, including an Internet Gateway and a default public route table.

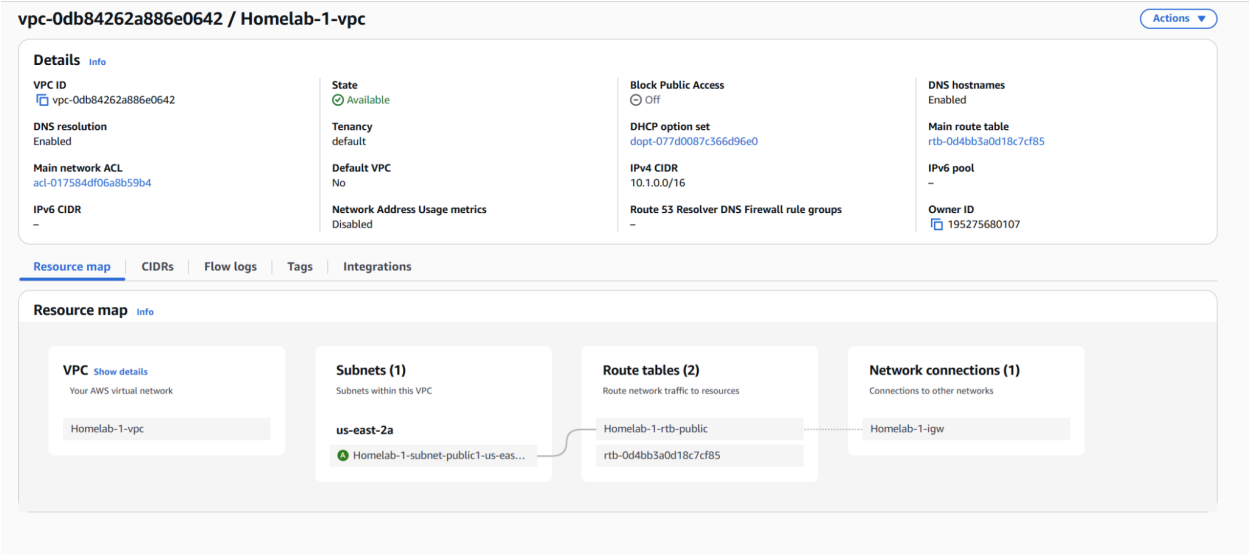

Figure 1: VPC Resource Map for Homelab-1-vpc.

4.2. EC2 Instance Deployment

One EC2 instance (t2.micro, Amazon Linux 2023 AMI) was launched into the public subnet of each VPC.

- **Instance - Homelab VPC 1:** Deployed in Homelab-1-vpc.
- **Instance - Homelab VPC 2:** Deployed in Homelab-2-vpc.

Security Groups were configured for each instance to allow inbound ICMP traffic from 0.0.0.0/0 (all IP addresses) to facilitate connectivity testing.

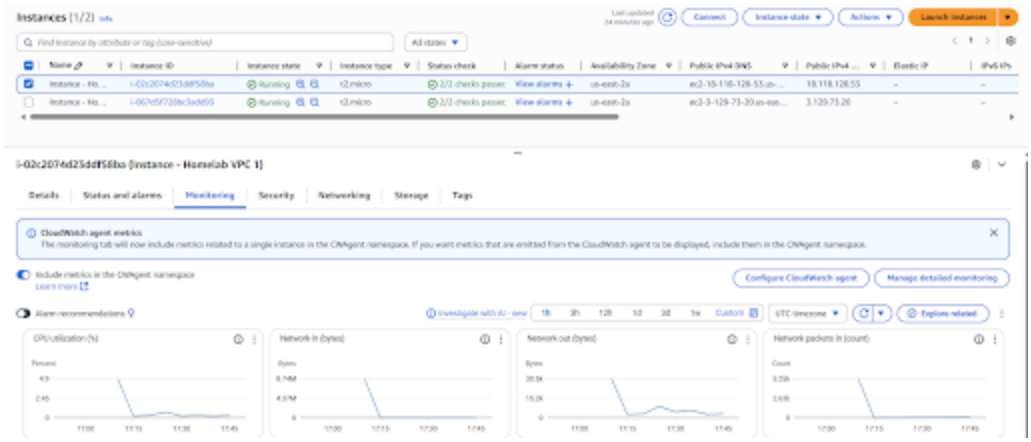

Figure 3: EC2 Instance - Homelab VPC 2 Monitoring Tab.

4.3. CloudWatch Log Group Creation

A dedicated CloudWatch Log Group, HomelabVPCFlowLog, was created to centralize all VPC Flow Log data. This log group was configured with a default retention policy.

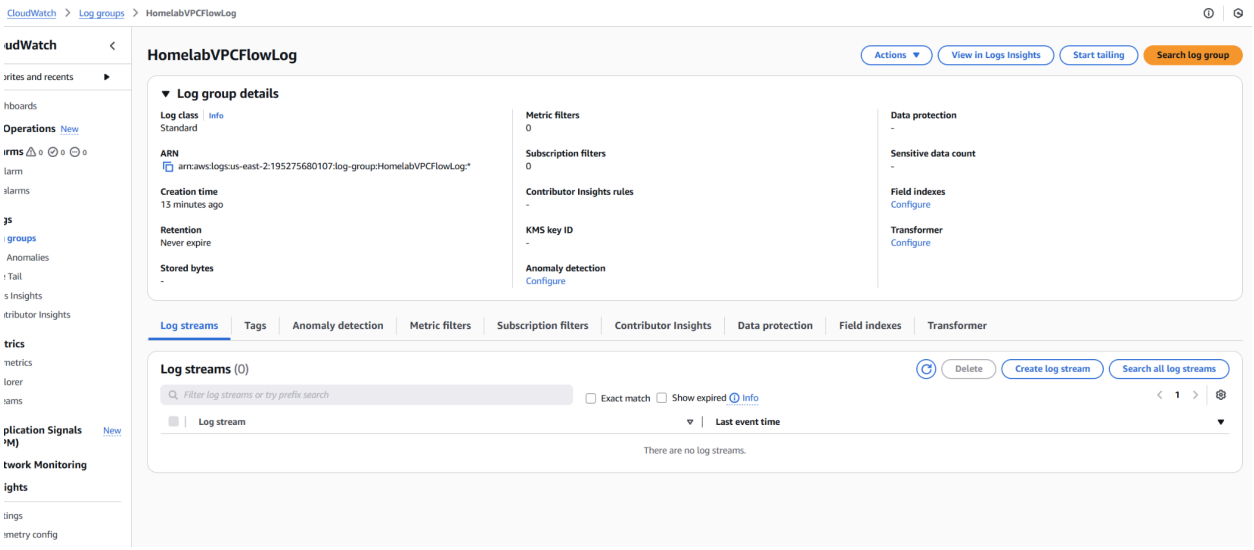

Figure 4: CloudWatch Log Group Details for HomelabVPCFlowLog.

4.4. IAM Policy and Role for Flow Logs

To enable VPC Flow Logs to publish data to CloudWatch Logs, a custom IAM policy and role

were created adhering to the principle of least privilege:

- **IAM Policy (NextWorkVPCFlowLogsPolicy):** Granted specific permissions for CloudWatch Logs actions, including logs:CreateLogGroup, logs:CreateLogStream, logs:PutLogEvents, logs:DescribeLogGroups, and logs:DescribeLogStreams.

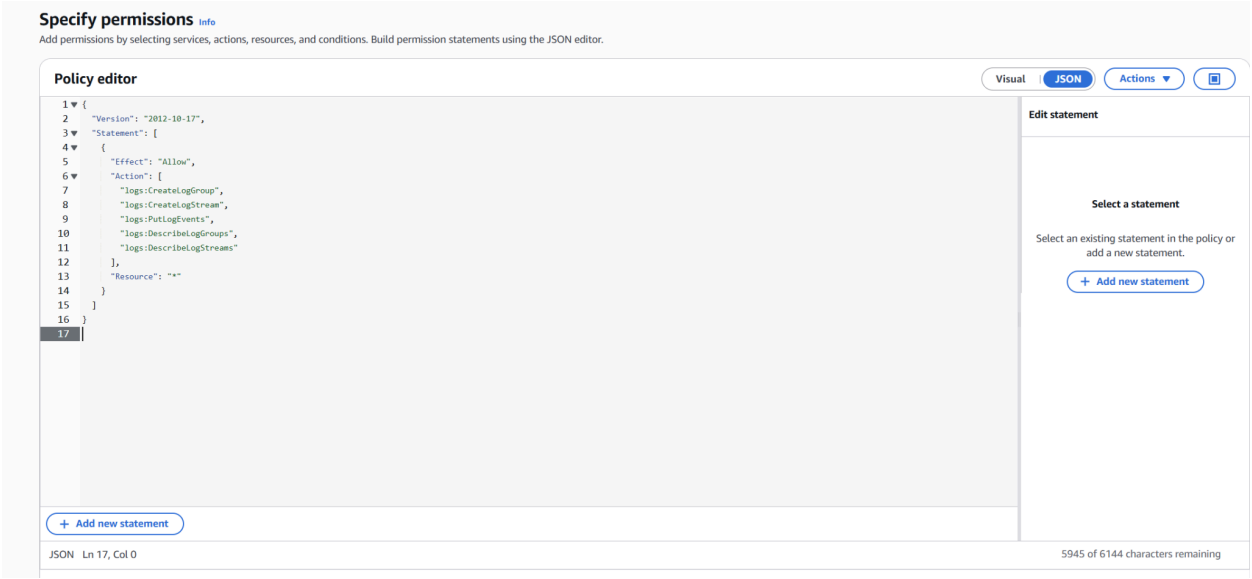

Figure 5: IAM Policy JSON for VPC Flow Logs.

- **IAM Role (NextWorkVPCFlowLogsRole):** Configured with a trust policy allowing vpc-flow-logs.amazonaws.com to assume the role, ensuring only the Flow Logs service could utilize these permissions. The NextWorkVPCFlowLogsPolicy was attached to this role.

4.5. VPC Flow Log Configuration

A VPC Flow Log was configured for the public subnet of Homelab-1-vpc.

- **Filter:** Set to All to capture all accepted and rejected traffic.
- **Maximum Aggregation Interval:** Set to 1 minute for granular data collection.
- **Destination:** Set to Send to CloudWatch Logs, pointing to the HomelabVPCFlowLog log group.
- **IAM Role:** The newly created NextWorkVPCFlowLogsRole was assigned.

4.6. VPC Peering Connection

A VPC peering connection (pcx-06bb3966b7cf8679) was initiated from Homelab-1-vpc (Requester) to Homelab-2-vpc (Accepter) and subsequently accepted.

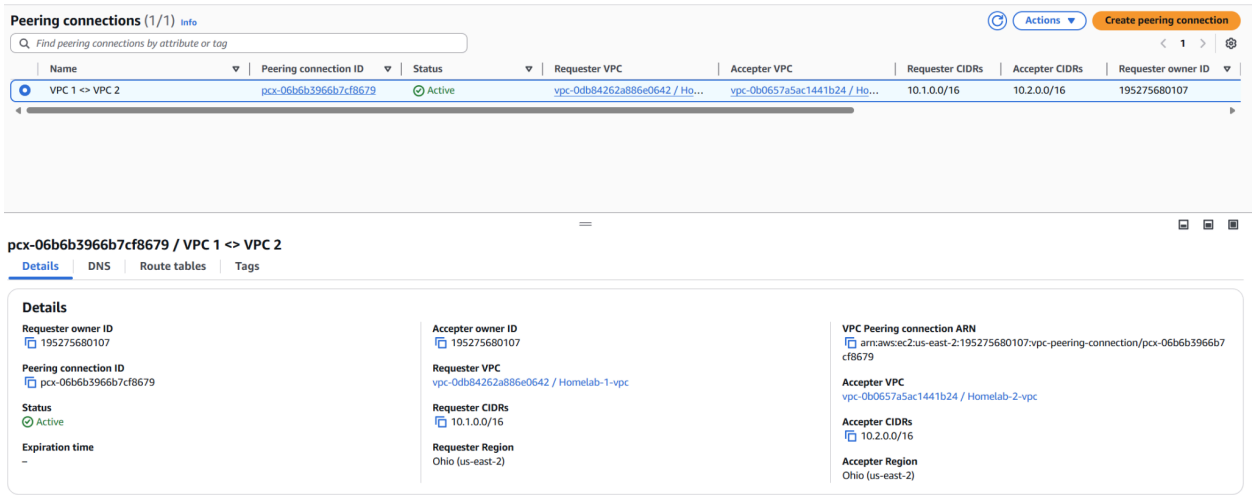

Figure 6: Active VPC Peering Connection Details.

4.7. Route Table Configuration

To enable communication over the peering connection, route tables for both VPCs were updated:

- **Homelab-1-vpc's Public Route Table:** A route was added with a Destination of

10.2.0.0/16 (Homelab-2-vpc's CIDR) and a Target of the peering connection (pcx-06bb3966b7cf8679).

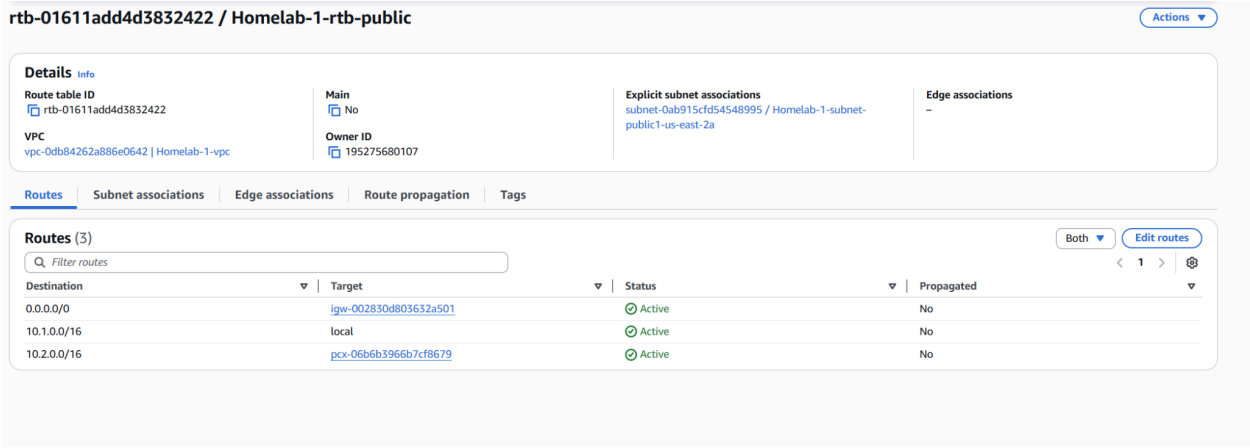

Figure 7: Homelab-1-vpc Public Route Table with Peering Route.

- **Homelab-2-vpc's Public Route Table:** A corresponding route was added with a Destination of 10.1.0.0/16 (Homelab-1-vpc's CIDR) and a Target of the peering connection.

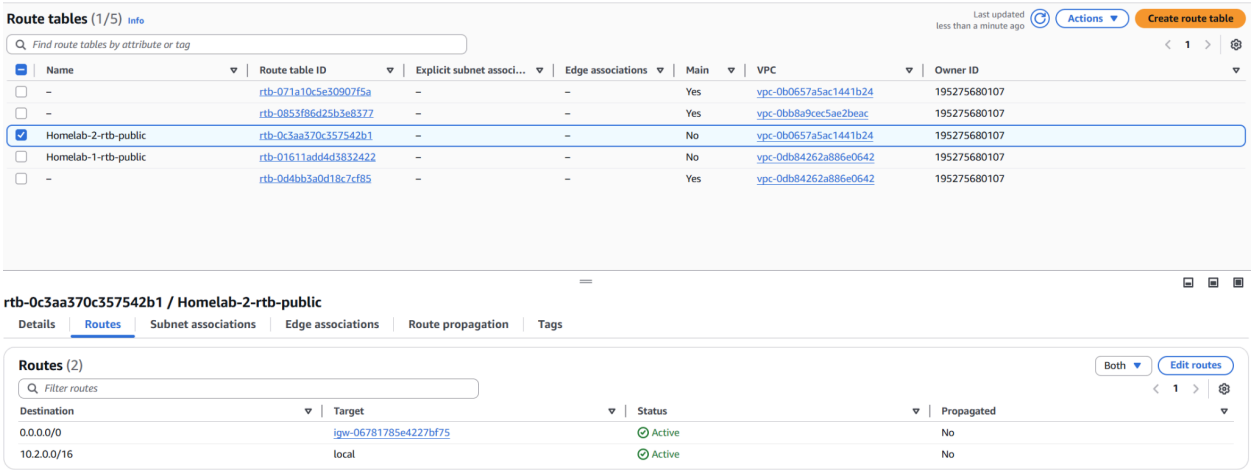

Figure 8: Homelab-2-vpc Public Route Table with Peering Route.

4.8. Connectivity Validation

Initial ping tests from Instance - Homelab VPC 1 to Instance - Homelab VPC 2's private IP address (10.2.11.243) initially failed, indicating a missing route.

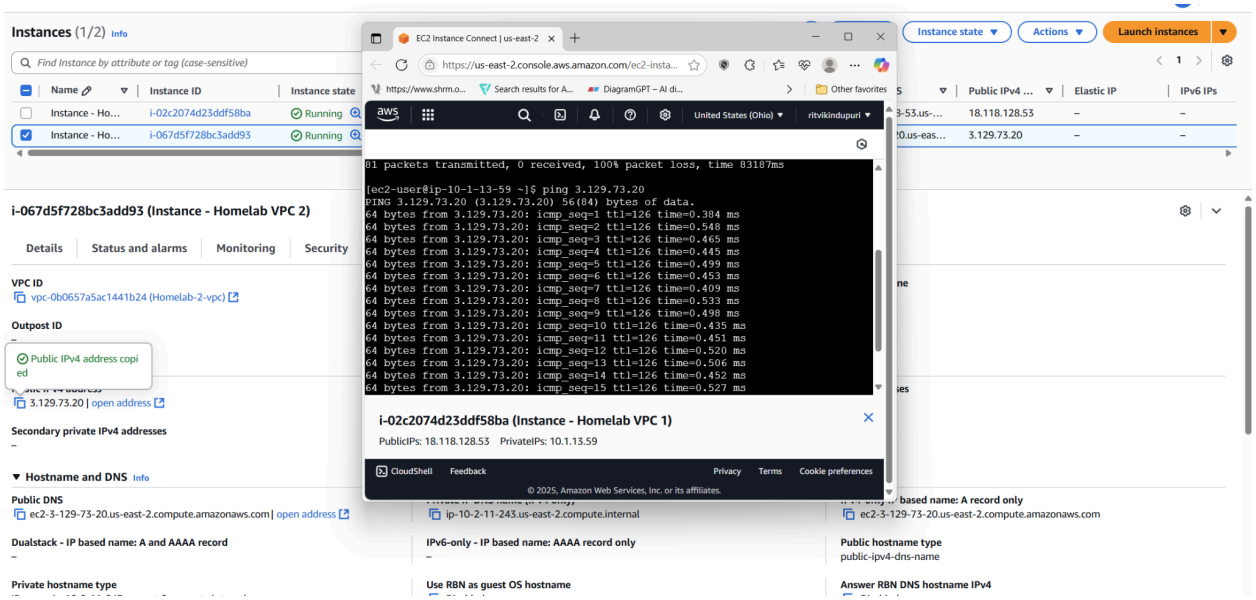

Figure 9: Initial Failed Ping Test to Private IP.

After configuring the route tables with the peering connection, subsequent ping tests to the private IP were successful, confirming direct private communication.

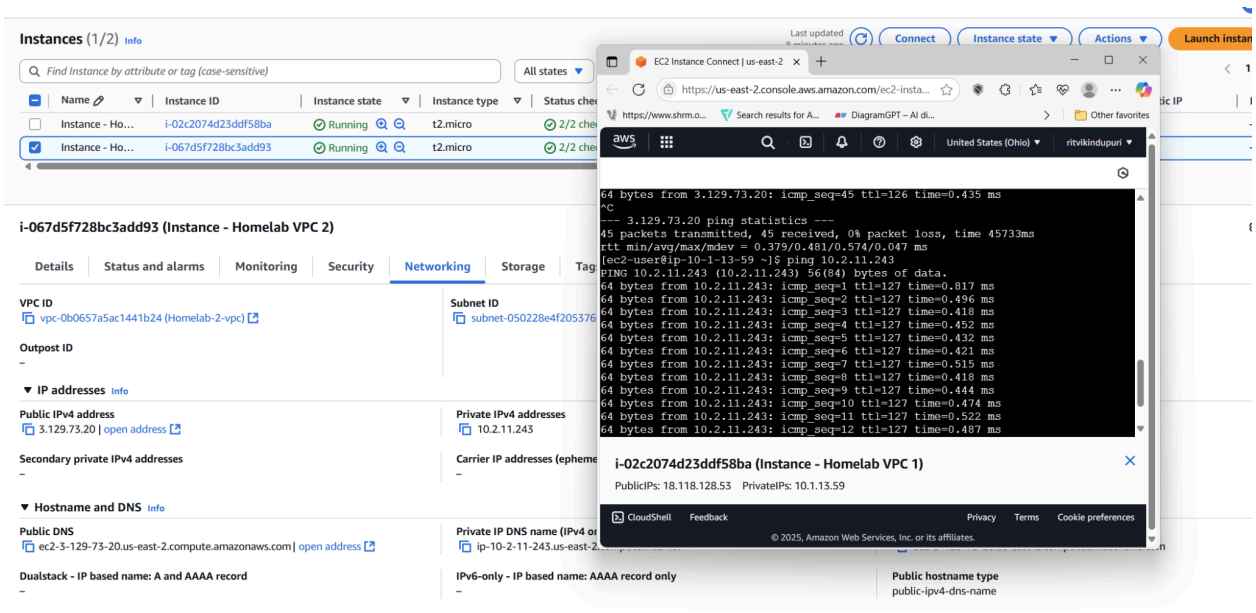

Figure 10: Successful Ping Test to Private IP after Route Configuration.

4.9. Flow Log Analysis with CloudWatch Logs Insights

VPC Flow Logs captured detailed network traffic information, including accepted and rejected connections.

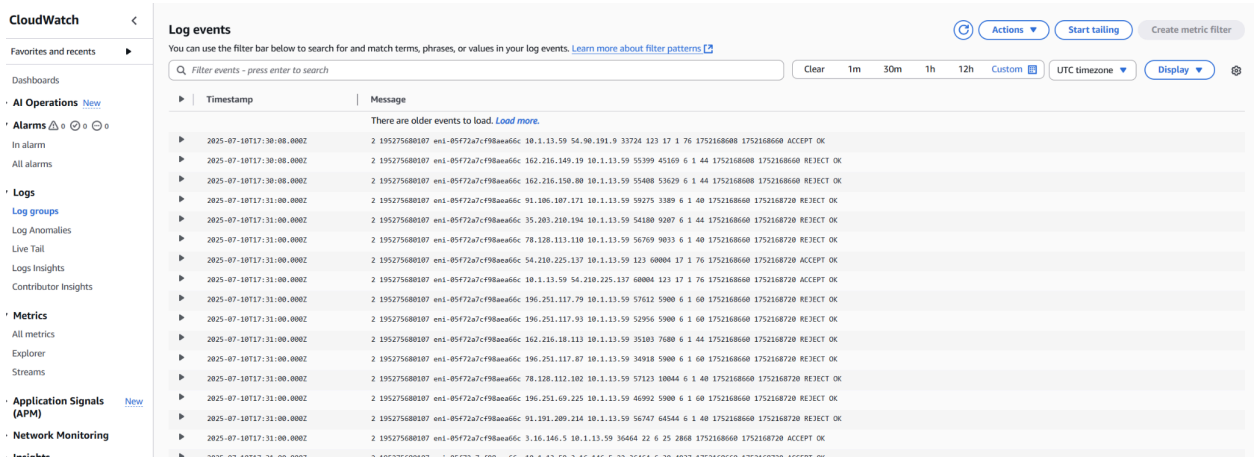

Figure 11: CloudWatch Log Events for VPC Flow Logs.

CloudWatch Logs Insights was used to query and analyze this data:

- **Top Byte Transfers:** A query was run to identify the top 10 byte transfers by source and destination IP addresses. This query processed 434 records (60.2 KB) in 0.8 seconds, revealing a top transfer of 25,569 bytes.

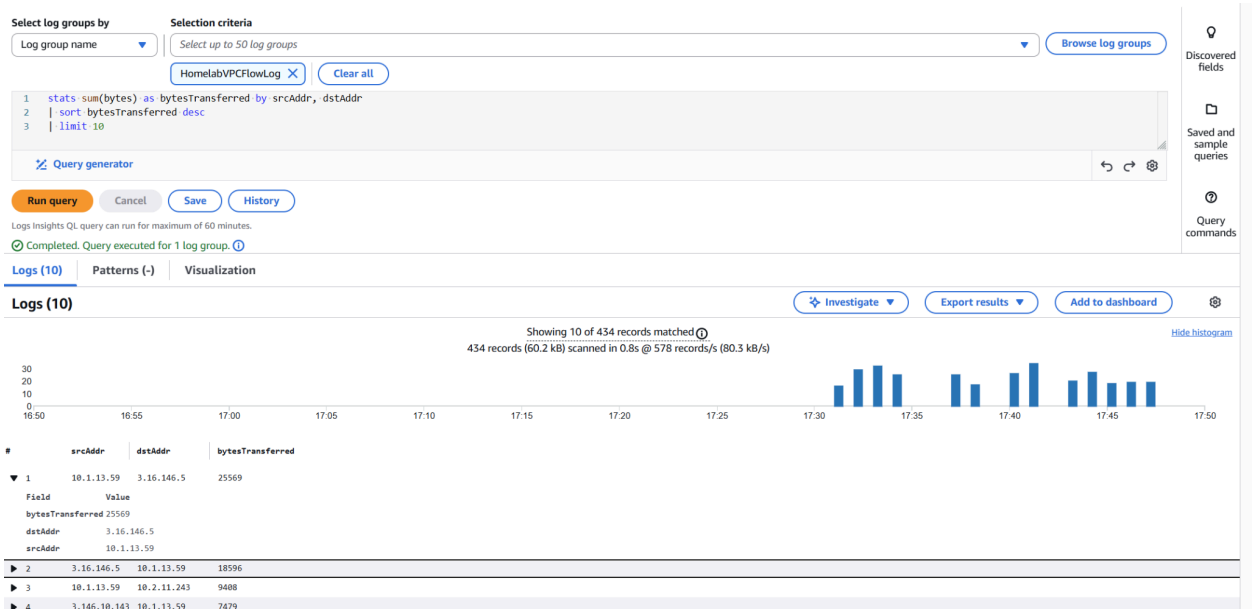

Figure 12: CloudWatch Logs Insights Query for Top Byte Transfers.

- **Top Packet Transfers:** A similar query identified top packet transfers, processing 469 records in 0.8 seconds.

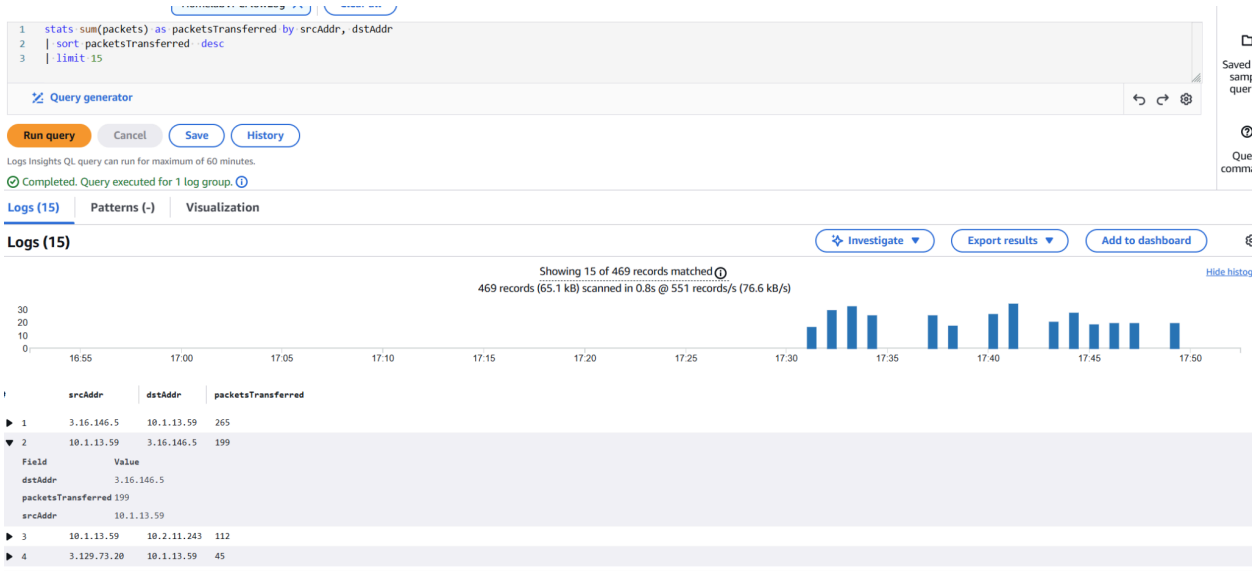

Figure 13: CloudWatch Logs Insights Query for Top Packet Transfers.

Rejected Connections: Queries were executed to filter for and count rejected connections, identifying instances like 17 rejections from a single source IP (89.248.163.112), crucial for security auditing.

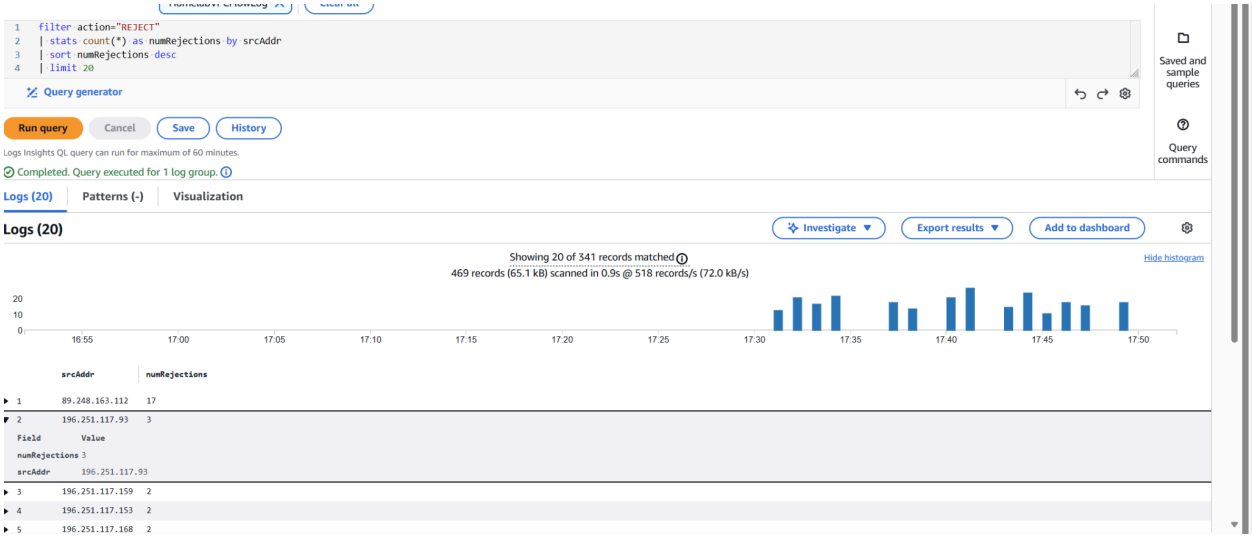

Figure 14: CloudWatch Logs Insights Query for Rejected Connections.

- **Average/Min/Max Bytes:** Further analysis included calculating average, minimum, and maximum byte transfers by source and destination.

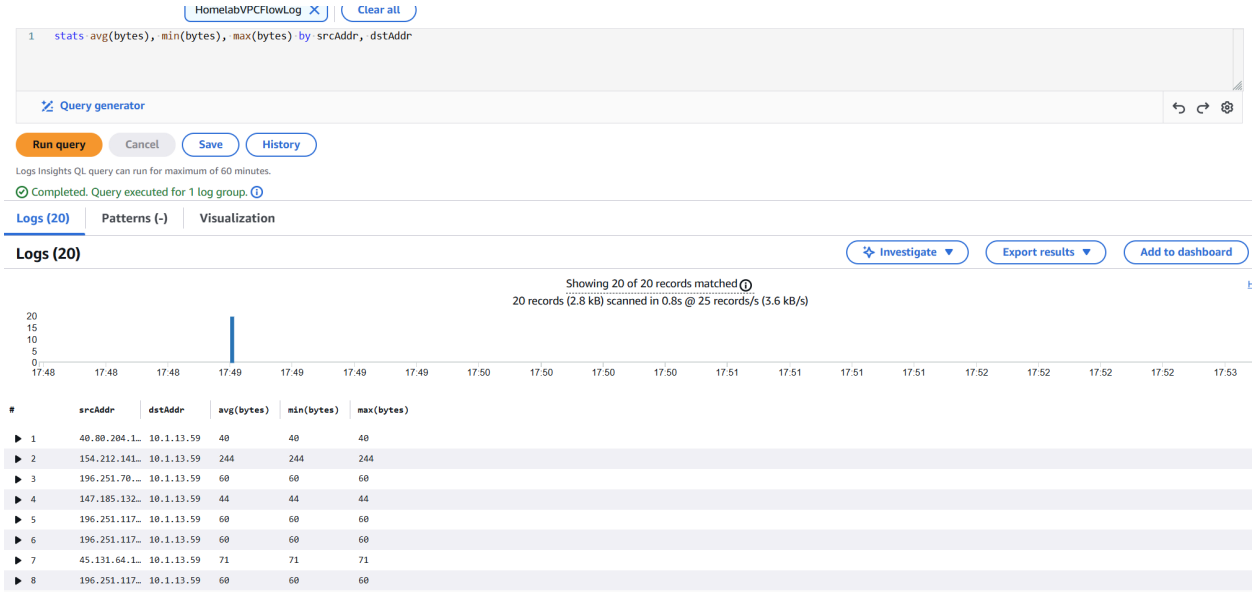

Figure 15: CloudWatch Logs Insights Query for Byte Transfer Statistics.

---

5. Conclusion

This project successfully demonstrated the end-to-end process of setting up a secure, peered VPC environment and implementing robust network monitoring. By configuring VPC Flow Logs and leveraging CloudWatch Logs Insights, comprehensive visibility into network traffic was achieved, enabling the identification of communication patterns, troubleshooting connectivity issues, and enhancing the overall security posture through analysis of accepted and rejected traffic.

---

6. Resource Cleanup

All AWS resources created during this project (EC2 instances, VPCs, peering connections, CloudWatch Log Groups, IAM roles/policies) were systematically terminated to prevent incurring unnecessary costs.
