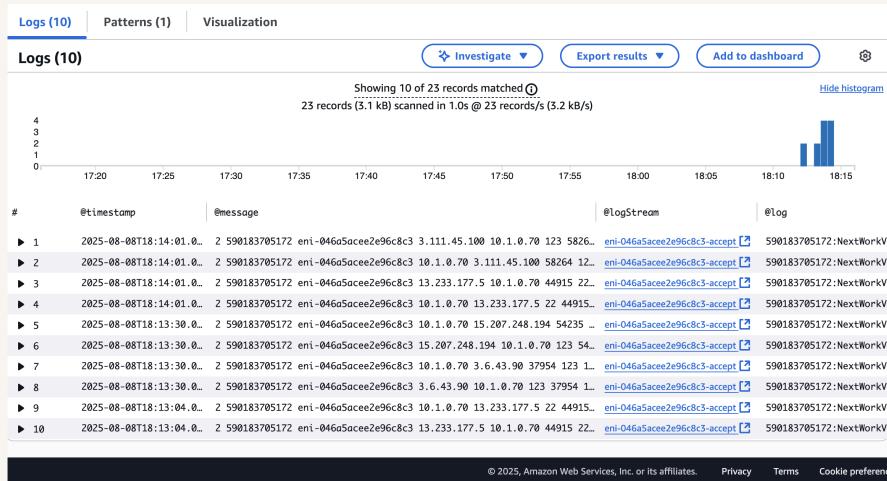




# VPC Monitoring with Flow Logs

G

Gaurav Balpande





# Introducing Today's Project!

## What is Amazon VPC?

Amazon VPC is the fundamental networking tool which is used for creation of private space within the AWS region. It is needed for security and to easily manage the resources.

## How I used Amazon VPC in this project

in today's project i had perform various monitoring tool which include cloudwatch and flow logs.

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

One thing I never expected is that we can easily create diagram for queries of logs with insight feature..

## This project took me...

it took near to 2.30 hr and it slightly hard.

# In the first part of my project...

## Step 1 - Set up VPCs

In this step, we are going to build the infrastructure by building two VPC with one public subnet in each and also some security tool we would apply.

## Step 2 - Launch EC2 instances

In this step we are going to launch Ec2 instance in each VPC so that we will be able to send data from one VPC to other.

## Step 3 - Set up Logs

In this step, we are going to add the monitoring tool AWS FLOW LOGS. It is a place where all the logs will be stored and we can make decision based on it.

## Step 4 - Set IAM permissions for Logs

In this step, we are going to build IAM role as it would not be possible for flow logs to send information to Amazon cloudwatch and also for storing and logging we would need IAM role.

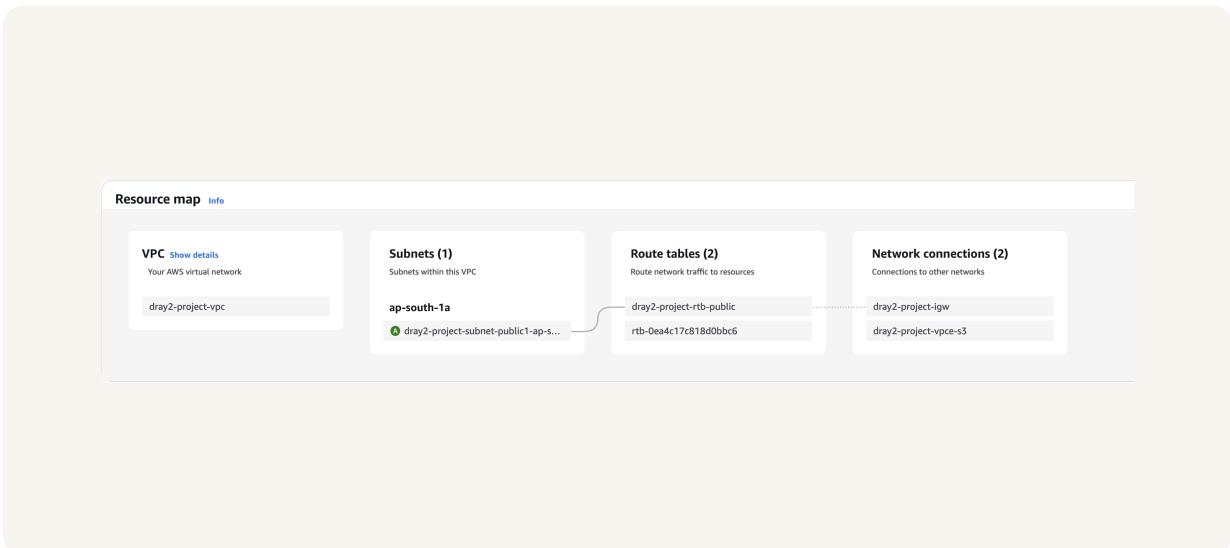
# Multi-VPC Architecture

I started my project by launching two VPC with one public subnet in each.

The CIDR blocks for VPCs 1 and 2 are 10.1.0.0/16 and 10.2.0.0/16 They have to be unique because we are going to peer and peering can create a conflict.

## I also launched EC2 instances in each subnet

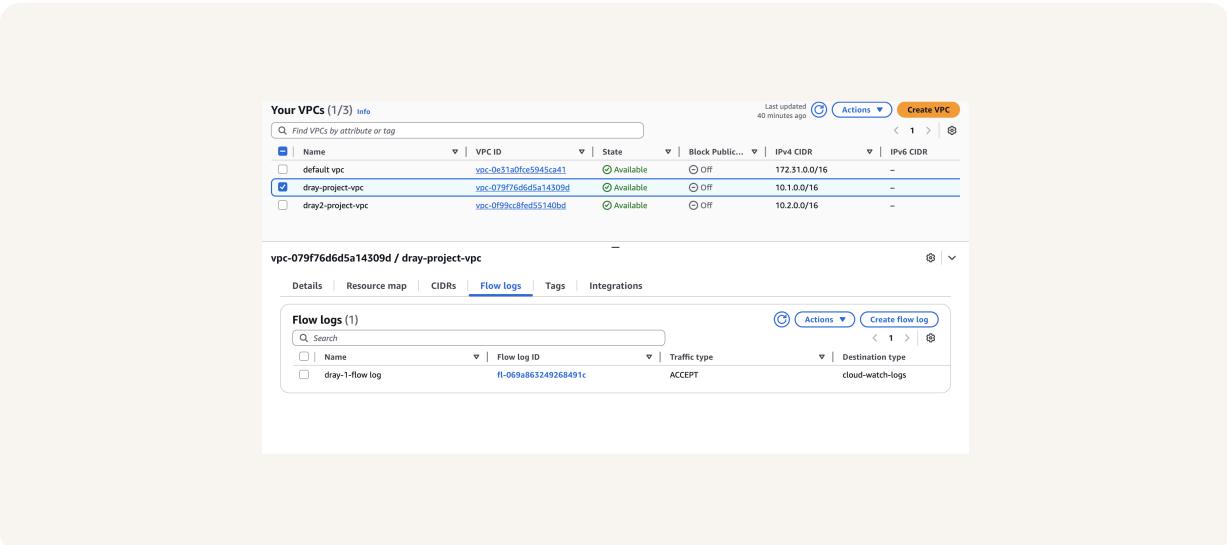
My EC2 instances' security groups allow all ICMP traffic This is because for sending data from one VPC to other we need to have ICMP messages be able to send.



# Logs

Logs are like a diary which store all the information related to a topic.

Log groups are like a folder which contain information related to a specific type and can be used for debugging and finding solution.



# IAM Policy and Roles

I created an IAM policy because for various task like creation and description of log group,log stream in cloudwatch we would need IAM role.

I also created an IAM role because for various task like creation and description of log group and log stream required IAM roles.

A custom trust policy is to very narrowly assign policy to specific resource.

The screenshot shows the 'Custom trust policy' configuration page in the AWS IAM console. The left panel displays the JSON code for the policy: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 }|The right panel contains a sidebar with the following sections:

- Edit statement**: A button to edit the existing statement.
- Select a statement**: A link to select an existing statement or add a new one.
- Add new statement**: A button to add a new statement.

# In the second part of my project...

## Step 5 - Ping testing and troubleshooting

In this step for checking the flow log we will send message from one instance to other instance and that too in different VPC. So we will create instance in two VPC and send data from one to other.

## Step 6 - Set up a peering connection

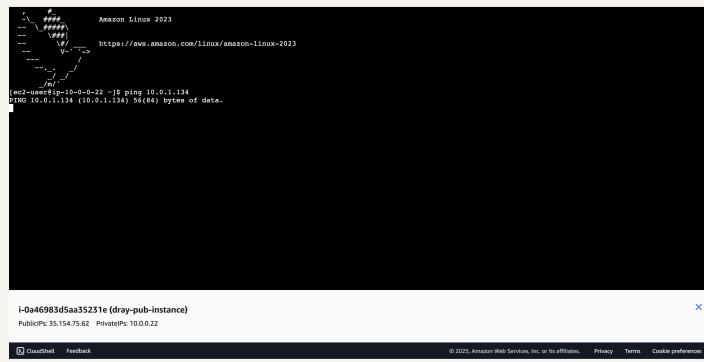
In this step,we are going to create a VPC peering between the two VPC so that there is a private connection between the two.

## Step 7 - Analyze flow logs

In this step,we are finally going to observe and analysis the logs on cloudwatch and will analyse the insights.

# Connectivity troubleshooting

My first ping test between my EC2 instances had no replies, which means there is no VPC peering.



The screenshot shows a terminal window titled "Amazon Linux 2023" with a cat logo icon. The command entered is "ping 10.0.1.134". The output shows the ping command being run and a reply from the target instance.

```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-10-0-0-22 ~]$ ping 10.0.1.134
PING 10.0.1.134 (10.0.1.134) 56(84) bytes of data.

I-0a46983d5aa35231e (dray-pub-instance)
PublicIP: 35.154.75.62 PrivateIP: 10.0.0.22

CloudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

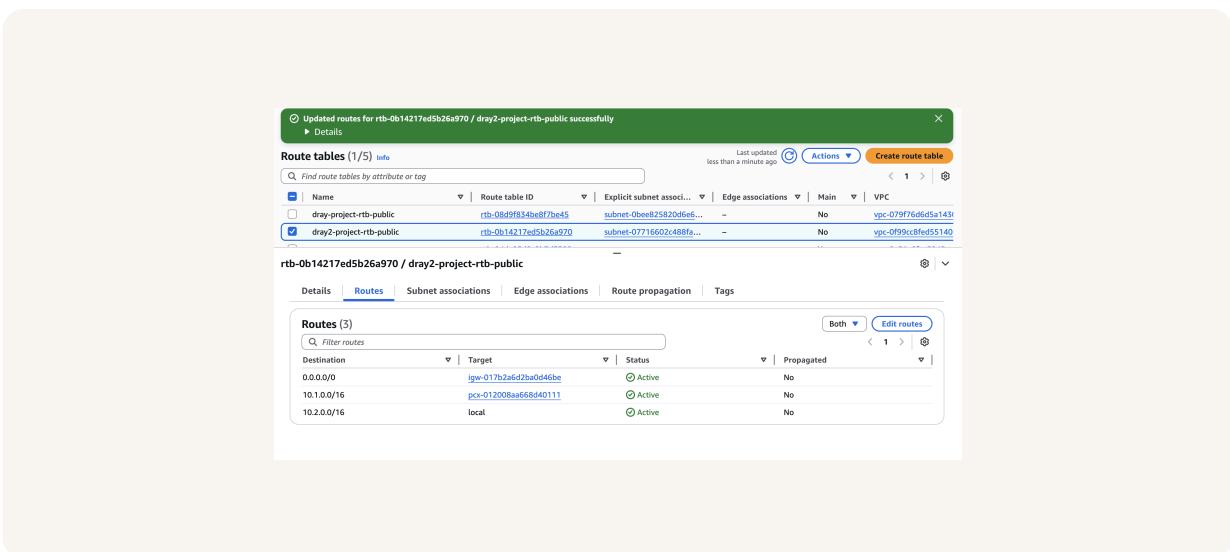
I could receive ping replies if I ran the ping test using the other instance's public IP address, which means that the public ip can be ping from any instance.

# Connectivity troubleshooting

Looking at VPC 1's route table, I identified that the ping test with Instance 2's private address failed because there is no route table for VPC peering.

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

I also updated both VPCs' route tables so that there is peering connection between the two instances.





# Connectivity troubleshooting

I received ping replies from Instance 2's private IP address! This means that the peering connection is set successfully.

The screenshot shows a terminal window in AWS CloudShell. The terminal output displays a ping session between two instances. The destination IP is 10.0.1.134. The ping command used was "ping 10.0.1.134". The output shows multiple ICMP echo replies from the target instance, indicating a successful connection. The terminal also shows the user's last login information and some system logs at the bottom.

```
Last login: Fri Aug  8 05:45:16 2025 from 13.233.177.3
[ec2-user@ip-10-0-0-22 ~]$ ping 10.0.1.134
PING 10.0.1.134 (10.0.1.134) 56(84) bytes of data.
64 bytes from 10.0.1.134: icmp_seq=0 ttl=127 time=0.192 ms
64 bytes from 10.0.1.134: icmp_seq=1 ttl=127 time=0.190 ms
64 bytes from 10.0.1.134: icmp_seq=100 ttl=127 time=0.199 ms
64 bytes from 10.0.1.134: icmp_seq=101 ttl=127 time=0.199 ms
64 bytes from 10.0.1.134: icmp_seq=102 ttl=127 time=0.189 ms
64 bytes from 10.0.1.134: icmp_seq=103 ttl=127 time=0.197 ms
64 bytes from 10.0.1.134: icmp_seq=104 ttl=127 time=0.189 ms
64 bytes from 10.0.1.134: icmp_seq=105 ttl=127 time=0.197 ms
64 bytes from 10.0.1.134: icmp_seq=106 ttl=127 time=0.176 ms
64 bytes from 10.0.1.134: icmp_seq=107 ttl=127 time=0.179 ms
64 bytes from 10.0.1.134: icmp_seq=108 ttl=127 time=0.177 ms
64 bytes from 10.0.1.134: icmp_seq=109 ttl=127 time=0.186 ms
64 bytes from 10.0.1.134: icmp_seq=110 ttl=127 time=0.186 ms
64 bytes from 10.0.1.134: icmp_seq=111 ttl=127 time=0.183 ms
64 bytes from 10.0.1.134: icmp_seq=112 ttl=127 time=0.185 ms
64 bytes from 10.0.1.134: icmp_seq=113 ttl=127 time=0.185 ms
64 bytes from 10.0.1.134: icmp_seq=114 ttl=127 time=0.176 ms
64 bytes from 10.0.1.134: icmp_seq=115 ttl=127 time=0.187 ms
64 bytes from 10.0.1.134: icmp_seq=116 ttl=127 time=0.172 ms
64 bytes from 10.0.1.134: icmp_seq=117 ttl=127 time=0.176 ms
64 bytes from 10.0.1.134: icmp_seq=118 ttl=127 time=0.241 ms
64 bytes from 10.0.1.134: icmp_seq=119 ttl=127 time=0.183 ms
64 bytes from 10.0.1.134: icmp_seq=120 ttl=127 time=0.197 ms
64 bytes from 10.0.1.134: icmp_seq=121 ttl=127 time=0.189 ms
64 bytes from 10.0.1.134: icmp_seq=122 ttl=127 time=0.192 ms
64 bytes from 10.0.1.134: icmp_seq=123 ttl=127 time=0.192 ms
64 bytes from 10.0.1.134: icmp_seq=124 ttl=127 time=0.178 ms
64 bytes from 10.0.1.134: icmp_seq=125 ttl=127 time=0.202 ms
64 bytes from 10.0.1.134: icmp_seq=126 ttl=127 time=0.194 ms
64 bytes from 10.0.1.134: icmp_seq=127 ttl=127 time=0.178 ms
64 bytes from 10.0.1.134: icmp_seq=128 ttl=127 time=0.203 ms
```

i-0a46983d5aa35231e (dray-pub-instance)  
PublicIPs: 35.154.75.62 PrivateIPs: 10.0.0.22

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

# Analyzing flow logs

Flow logs tell us about all the necessary information like source port, destination port, entry time, completion time, id, ENI id, etc

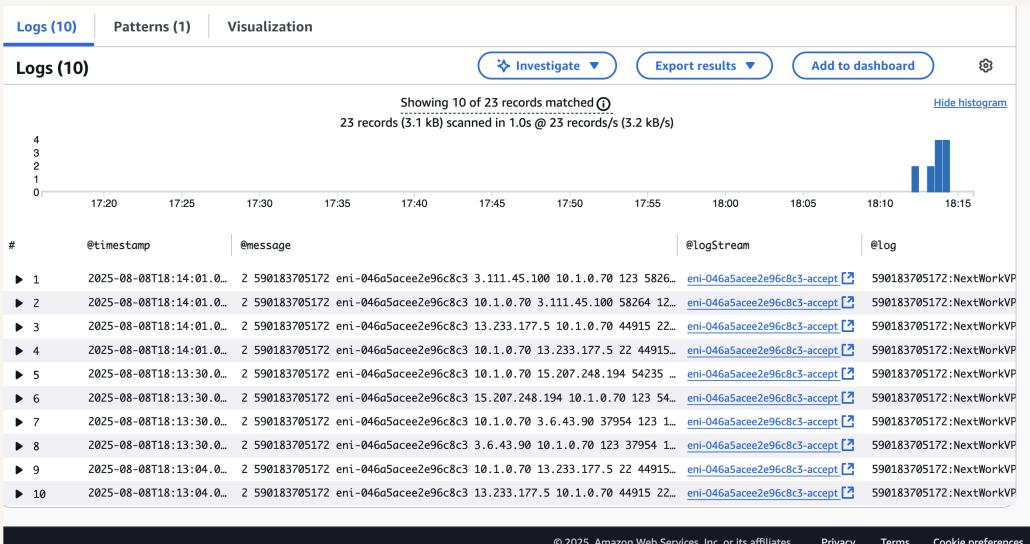
For example, the flow log I've captured tells us  
2025-08-08T18:08:59.000Z 2  
590183705172 eni-046a5acee2e96c8c3 10.1.0.70 45.79.191.178 22 36174 6 1 44 1754676539 1754676565 ACCEPT OK  
1754676539 1754676565 ACCEPT OK 2025-08-08T18:09:29.000Z 2 590183705172  
eni-046a5acee2e96c8c3 3.111

Log events	
You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns.	
<input type="text"/> Filter events - press enter to search	
Clear	1m
50m	1h
12h	Custom
UTC timezone	Display
▶   Timestamp	Message
	No older events at this moment. <a href="#">Resume</a>
▶ 2025-08-08T18:08:59.000Z	2 590183705172 eni-046a5acee2e96c8c3 10.1.0.70 45.79.191.178 22 36174 6 1 44 1754676539 1754676565 ACCEPT OK
▶ 2025-08-08T18:09:29.000Z	2 590183705172 eni-046a5acee2e96c8c3 3.111.45.108 10.1.0.70 123 37348 17 1 76 1754676539 1754676565 ACCEPT OK
▶ 2025-08-08T18:09:29.000Z	2 590183705172 eni-046a5acee2e96c8c3 10.1.0.70 3.111.45.108 37348 123 17 1 76 1754676539 1754676565 ACCEPT OK
▶ 2025-08-08T18:09:29.000Z	2 590183705172 eni-046a5acee2e96c8c3 3.6.43.98 10.1.0.70 123 47014 17 1 76 1754676569 1754676593 ACCEPT OK
▶ 2025-08-08T18:09:29.000Z	2 590183705172 eni-046a5acee2e96c8c3 10.1.0.70 3.6.43.98 47014 123 17 1 76 1754676569 1754676593 ACCEPT OK
▶ 2025-08-08T18:09:29.000Z	2 590183705172 eni-046a5acee2e96c8c3 15.207.248.194 10.1.0.70 123 44515 17 1 76 1754676569 1754676593 ACCEPT OK
▶ 2025-08-08T18:09:29.000Z	2 590183705172 eni-046a5acee2e96c8c3 10.1.0.70 15.207.248.194 44515 123 17 1 76 1754676569 1754676593 ACCEPT OK
▶ 2025-08-08T18:11:36.000Z	2 590183705172 eni-046a5acee2e96c8c3 3.6.43.98 10.1.0.70 123 39773 17 1 76 1754676569 1754676718 ACCEPT OK
▶ 2025-08-08T18:11:36.000Z	2 590183705172 eni-046a5acee2e96c8c3 10.1.0.70 3.6.43.98 39773 123 17 1 76 1754676569 1754676718 ACCEPT OK
▶ 2025-08-08T18:11:36.000Z	2 590183705172 eni-046a5acee2e96c8c3 15.209.20.166 10.1.0.70 123 54874 17 1 76 1754676569 1754676718 ACCEPT OK
▶ 2025-08-08T18:11:36.000Z	2 590183705172 eni-046a5acee2e96c8c3 10.1.0.70 15.209.20.166 54874 123 17 1 76 1754676569 1754676718 ACCEPT OK
▶ 2025-08-08T18:12:00.000Z	2 590183705172 eni-046a5acee2e96c8c3 15.233.177.3 10.1.0.70 123 44915 22 6 47 4296 1754676720 1754676745 ACCEPT OK
▶ 2025-08-08T18:12:00.000Z	2 590183705172 eni-046a5acee2e96c8c3 10.1.0.70 15.233.177.3 44915 6 43 6789 1754676720 1754676745 ACCEPT OK

# Logs Insights

Logs Insights is a tool in aws which helps us in understanding the logs by providing the feature of queries and diagram.

I ran the query Top 10 byte transfers by source and destination IP addresses This query analyzes the top 10 address with highest traffic.





[nextwork.org](https://nextwork.org)

# The place to learn & showcase your skills

Check out [nextwork.org](https://nextwork.org) for more projects

