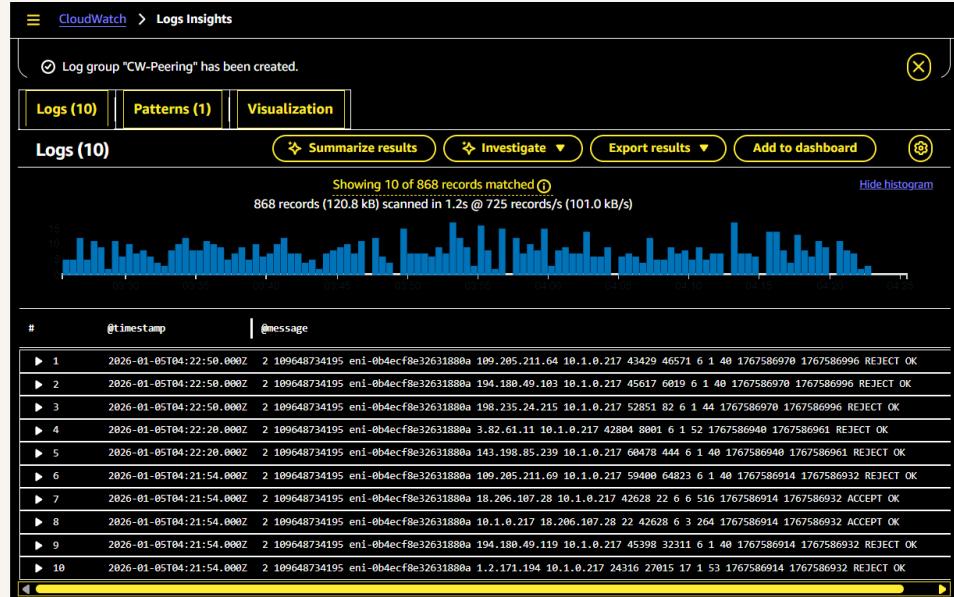




VPC Monitoring with Flow Logs



Michael K.





Michael K.

NextWork Student

nextwork.org

Introducing Today's Project!

What is Amazon VPC?

What is Amazon VPC? Amazon VPC (Virtual Private Cloud) is a logically isolated virtual network within AWS where I can launch and manage AWS resources (such as EC2 instances, RDS databases, and load balancers) using my own defined IP address range, subnets, route tables, and network gateways.

How I used Amazon VPC in this project

In today's project, I used Amazon VPC to create a peered network and monitor traffic between my peered VPCs (traffic in my network).

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

One thing I didn't expect in this project was its complexity and relationship between the different services used for monitoring.

This project took me...

This project took me 3 hours.

In the first part of my project...

Step 1 - Set up VPCs

In this step, I will Create two VPCs from scratch for VPC Peering.

Step 2 - Launch EC2 instances

In this step, I will launch an EC2 instance in each VPC, so i can use them to test my VPC peering connection.

Step 3 - Set up Logs

In this step, I will set up a way to track all inbound and outbound network traffic and set up a space that stores all of these records.

Step 4 - Set IAM permissions for Logs

In this step, I will give VPC Flow Logs the permission to write logs and send them to CloudWatch (log group), and finish setting up my subnet's flow log.

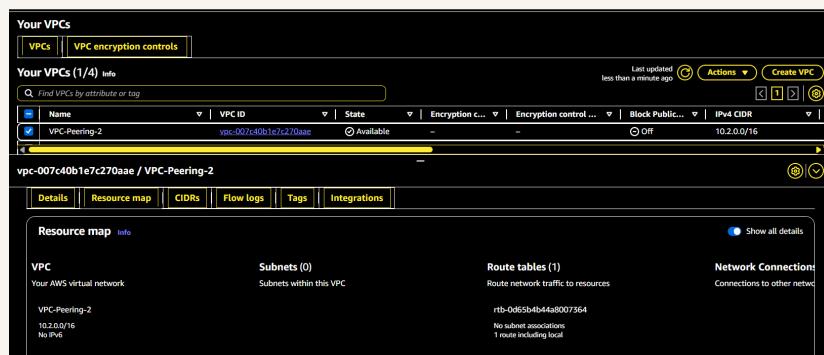
Multi-VPC Architecture

I started my project by launching 2 VPCs, 2 Subnets and their components.

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 in order to avoid IP Address overlap.

I also launched EC2 instances in each subnet

My EC2 instances' security groups allow traffic in and out of my EC2 Instances. The Inbound rules for the first Instance security group were : SSH, and All ICMP IPv4 from the second VPC. The Outbound rules were : All Traffic to anywhere. The Inbound rules for the second Instance security group were : All ICMP IPv4 from the first VPC. The Outbound rules were : All Traffic to anywhere.



Logs

Logs are like a diary which record everything that happens in my region, from users logging in to errors popping up. It's the go-to place to understand what's going on with my systems, troubleshoot problems, and keep an eye on who's doing what.

Log groups are like big folders in AWS where i keep related logs together. Usually, logs from the same source or application will go into the same log group, BUT logs are also region-specific. This means log data gets created and saved in the region it was created, although you can use CloudWatch dashboards to bring together logs from different regions.

The screenshot shows the AWS VPC Peering interface. At the top, there is a navigation bar with tabs for 'VPCs' and 'VPC encryption controls'. Below this, a search bar says 'Find VPCs by attribute or tag'. A table lists four VPCs:

Name	VPC ID	State	Encryption c...	Encryption control ...	Block Public...	IPv4 CIDR
VPC-Peering-2	vpc-007e40b1c7270aae	Available	-	-	Off	10.2.0.0/16
VPC-Peering-1	vpc-008830ee07ac607a5	Available	-	-	Off	10.1.0.0/16
My Test VPC	vpc-0f1bb56fd256a6e2	Available	-	-	Off	10.0.0.0/16
-	vpc-098e81428b564094	Available	-	-	Off	172.31.0.0/16

Below the table, a specific VPC (vpc-008830ee07ac607a5 / VPC-Peering-1) is selected. The interface shows several tabs: 'Details', 'Resource map', 'CIDRs', 'Flow logs', 'Tags', and 'Integrations'. The 'Flow logs' tab is active, displaying one entry:

Name	Flow log ID	Traffic type	Destination type	Destination name
VPC-Peering-1-FL	fl-04d729eeafe7f5991	All	cloud-watch-logs	CW-Peering



Michael K.

NextWork Student

nextwork.org

IAM Policy and Roles

I created an IAM policy to give VPC Flow Logs the permission to record traffic and store them in my CloudWatch. This policy makes sure that my VPC Flow log can now send log data to my log group!

I also created an IAM role permission can't be directly attached to AWS services to perform some actions on AWS resources. Only roles containing these Policies (permissions) can.

A custom trust policy is a specific type of policy! While IAM policies help me define the actions a user/service can or cannot do, custom trust policies are used to very narrowly define who can use a role.

J

Michael K.
NextWork Student

nextwork.org

Custom trust policy

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

```
1 v {  
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 will get Instance 1 to send test messages to Instance 2 to test my VPC Peering.

Step 6 - Set up a peering connection

In this step, I will set up a connection link between your VPCs.

Step 7 - Analyze flow logs

In this step, I will review the flow logs recorded about VPC 1's public subnet and analyse the flow logs to get some insights.



Michael K.

NextWork Student

nextwork.org

Connectivity troubleshooting

My first ping test between my EC2 instances had no replies, which means the connection to the second server was not reachable. but i solved the problem and it worked.,

I could receive ping replies if I ran the ping test using the other instance's public IP address, which means it is publicly reachable.

Connectivity troubleshooting

Looking at VPC 1's route table, I identified that the ping test with Instance 2's private address failed because the ICMP rule was not setup at the security group level. Also, the peering rule was not setup at the Network ACL level for both Instances.

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

I also updated both VPCs' route tables so that it can have the peering route for the connection with each other.

The screenshot shows the AWS Route Tables and Routes interface. At the top, there is a table titled "Route tables (1/6) Info". The table lists several route tables, including "My-test-private-route-table", "My-test-public-route-table", "Route-Peering-2", and "Route-Peering-1". Below this, a specific route table is selected: "rtb-0a7604d5d502110f5 / Route-Peering-1". Under this table, there is another table titled "Routes (3)".

Route tables (1/6) Info					
Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
<input checked="" type="checkbox"/> My-test-private-route-table	rtb-08d655399e226093f	subnet-01ab18999834c7...	-	No	vpc-0fc1b895bf1260a62 My ...
<input type="checkbox"/> -	rtb-0b0961b1baed72052	-	-	Yes	vpc-066c814233b364f04
<input type="checkbox"/> -	rtb-054e81248811e6047	subnet-036c315d0f0f494f...	-	Yes	vpc-0f10f95b012350a62 My T...
<input type="checkbox"/> -	rtb-0d65b4d444b8007364	subnet-01c450d4965a75...	-	Yes	vpc-00744061e7c270aae VPC...
<input type="checkbox"/> -	rtb-03a85185ae5024597	-	-	Yes	vpc-008830ee07ae607a5 VPC...
<input checked="" type="checkbox"/> Route-Peering-1	rtb-0a7604d5d502110f5	subnet-08bd094bd14273...	-	No	vpc-008830ee07ae607a5 VPC...

rtb-0a7604d5d502110f5 / Route-Peering-1																									
Details	Routes	Subnet associations	Edge associations	Route propagation	Tags																				
Routes (3) <input style="float: right; margin-right: 10px;" type="button" value="Both"/> <input style="float: right; margin-right: 10px;" type="button" value="Edit routes"/> <input style="margin-bottom: 5px;" type="button" value="Filter routes"/> <table border="1"> <thead> <tr> <th>Destination</th> <th>Target</th> <th>Status</th> <th>Propagated</th> <th>Route Origin</th> </tr> </thead> <tbody> <tr> <td>0.0.0.0/0</td> <td>ipw-Oct19d279119eb5a9</td> <td><input checked="" type="radio"/> Active</td> <td>No</td> <td>Create Route</td> </tr> <tr> <td>10.1.0.0/16</td> <td>local</td> <td><input checked="" type="radio"/> Active</td> <td>No</td> <td>Create Route Table</td> </tr> <tr> <td>10.2.0.0/16</td> <td>pcc-090c5ed4f00423ca80</td> <td><input checked="" type="radio"/> Active</td> <td>No</td> <td>Create Route</td> </tr> </tbody> </table>						Destination	Target	Status	Propagated	Route Origin	0.0.0.0/0	ipw-Oct19d279119eb5a9	<input checked="" type="radio"/> Active	No	Create Route	10.1.0.0/16	local	<input checked="" type="radio"/> Active	No	Create Route Table	10.2.0.0/16	pcc-090c5ed4f00423ca80	<input checked="" type="radio"/> Active	No	Create Route
Destination	Target	Status	Propagated	Route Origin																					
0.0.0.0/0	ipw-Oct19d279119eb5a9	<input checked="" type="radio"/> Active	No	Create Route																					
10.1.0.0/16	local	<input checked="" type="radio"/> Active	No	Create Route Table																					
10.2.0.0/16	pcc-090c5ed4f00423ca80	<input checked="" type="radio"/> Active	No	Create Route																					

Connectivity troubleshooting

I received ping replies from Instance 2's private IP address! This means that the Peering was successful.

```
~~ \#####\###\
~~ \###| \###|
~~ \|/ __ https://aws.amazon.com/linux/amazon-linux-2023x/amazon-linux-2023
~~ V~'-'> V~' '-'>
~~~ /   /
~~~_ _/_/_/_/
/_/_/_/_/_/_/
/m/'/_/m/'

Last login: Mon Jan  5 00:39:56 2026 from 18.206.107.27 18.206.107.27
[ec2-user@ip-10-1-0-217 ~]$ ping 10.2.0.148 10.2.0.148
PING 10.2.0.148 (10.2.0.148) 56(84) bytes of data. bytes of data.
64 bytes from 10.2.0.148: icmp_seq=1 ttl=127 time=0.435 ms time=0.435 ms
64 bytes from 10.2.0.148: icmp_seq=2 ttl=127 time=0.466 ms time=0.466 ms
64 bytes from 10.2.0.148: icmp_seq=3 ttl=127 time=0.457 ms time=0.457 ms
64 bytes from 10.2.0.148: icmp_seq=4 ttl=127 time=0.448 ms time=0.448 ms
64 bytes from 10.2.0.148: icmp_seq=5 ttl=127 time=0.439 ms time=0.439 ms
64 bytes from 10.2.0.148: icmp_seq=6 ttl=127 time=0.457 ms time=0.457 ms
64 bytes from 10.2.0.148: icmp_seq=7 ttl=127 time=0.440 ms time=0.440 ms
64 bytes from 10.2.0.148: icmp_seq=8 ttl=127 time=0.439 ms time=0.439 ms
64 bytes from 10.2.0.148: icmp_seq=9 ttl=127 time=0.445 ms time=0.445 ms
64 bytes from 10.2.0.148: icmp_seq=10 ttl=127 time=0.451 ms time=0.451 ms
64 bytes from 10.2.0.148: icmp_seq=11 ttl=127 time=0.452 ms time=0.452 ms
64 bytes from 10.2.0.148: icmp_seq=12 ttl=127 time=0.449 ms time=0.449 ms
64 bytes from 10.2.0.148: icmp_seq=13 ttl=127 time=0.447 ms time=0.447 ms
64 bytes from 10.2.0.148: icmp_seq=14 ttl=127 time=0.441 ms time=0.441 ms
64 bytes from 10.2.0.148: icmp_seq=15 ttl=127 time=0.451 ms time=0.451 ms
```

Analyzing flow logs

Flow logs tell me about the number of data that were sent successfully from one IP address (e.g 18.237.140.165) to another, the protocol and port used, the number of packets transferred and if the traffic was allowed ("ACCEPT") or not ("REJECT").

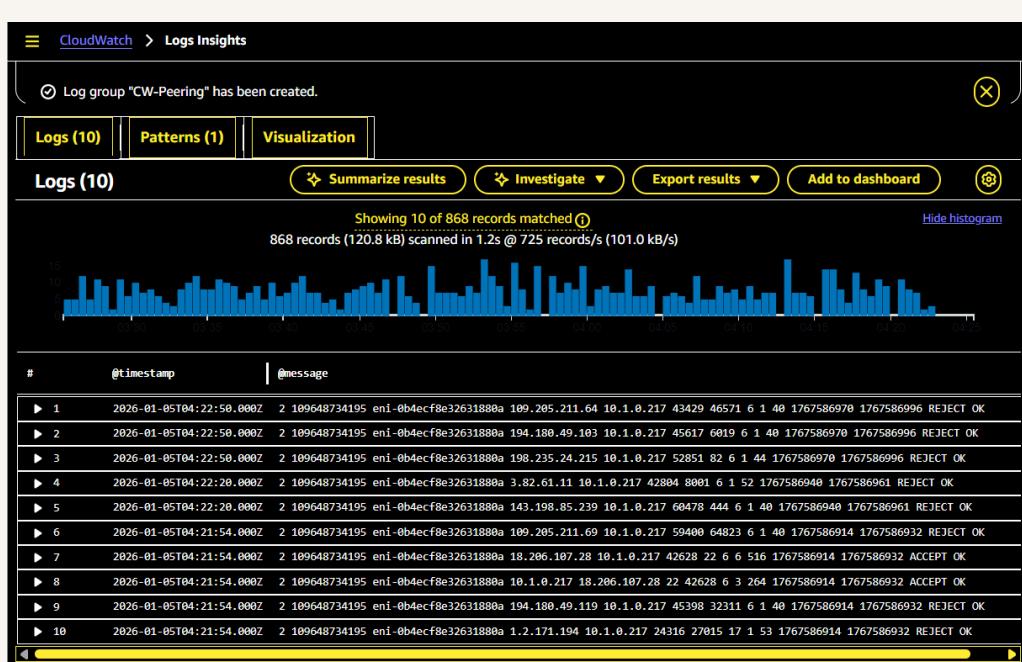
For example, the flow log I've captured tells me that 344 bytes of data were sent successfully from the IP address 18.237.140.165 to 10.1.5.112 using TCP protocol on port 22, with 4 packets transferred and the traffic was allowed ("ACCEPT").

Log events	
Actions ▾ Start tailing Create metric filter	
Filter events - press enter to search	
Timestamp	Message
	There are older events to load. Load more.
2020-01-05T03:55:49,000Z	2 189648734195 en1:0&4cf0#02031880# 18.206.187.28 10.1.0.217 42628 22 6 4 344 1767585349 1767585369 ACCEPT OK
2020-01-05T03:55:49,000Z	2 189648734195 en1:0&4cf0#02031880# 18.1.0.217 31.206.187.28 22 42628 6 2 176 1767585349 1767585369 ACCEPT OK
2020-01-05T03:55:49,000Z	2 189648734195 en1:0&4cf0#02031880# 159.223.218.177 10.1.0.217 53024 22 6 11 1884 1767585349 1767585369 ACCEPT OK
2020-01-05T03:55:49,000Z	2 189648734195 en1:0&4cf0#02031880# 18.1.0.217 159.223.218.177 22 53024 6 11 2272 1767585349 1767585369 ACCEPT OK
2020-01-05T03:55:49,000Z	2 189648734195 en1:0&4cf0#02031880# 171.255.223.70 10.1.0.217 48796 3337 6 1 52 1767585349 1767585369 REJECT OK
2020-01-05T03:55:49,000Z	2 189648734195 en1:0&4cf0#02031880# 159.223.218.177 10.1.0.217 53046 22 6 11 1884 1767585349 1767585369 ACCEPT OK
2020-01-05T03:55:49,000Z	2 189648734195 en1:0&4cf0#02031880# 16.1.0.217 159.223.218.177 22 53046 6 11 2272 1767585349 1767585369 ACCEPT OK
2020-01-05T03:55:49,000Z	2 189648734195 en1:0&4cf0#02031880# 200.168.95.10 10.1.0.217 6382 222 6 1 68 1767585349 1767585369 REJECT OK
2020-01-05T03:55:49,000Z	2 189648734195 en1:0&4cf0#02031880# 3.269.239.246 10.1.0.217 53274 30877 6 1 44 1767585386 1767585394 REJECT OK
2020-01-05T03:55:49,000Z	2 189648734195 en1:0&4cf0#02031880# 162.216.159.163 10.1.0.217 53842 9899 6 1 44 1767585386 1767585394 REJECT OK
2020-01-05T03:55:49,000Z	2 189648734195 en1:0&4cf0#02031880# 194.188.40.195 10.1.0.217 45057 7265 6 1 48 1767585407 1767585416 REJECT OK
2020-01-05T03:55:49,000Z	2 189648734195 en1:0&4cf0#02031880# 79.124.63.154 10.1.0.217 42507 3389 6 1 46 1767585407 1767585416 REJECT OK

Logs Insights

Logs Insights is a CloudWatch feature that analyzes logs. In Log Insights, i use queries to filter, process and combine data to troubleshoot problems or better understand my network traffic!

I ran the query : fields @timestamp, @message | sort @timestamp desc | limit 10 This query analyzes "Top 10 byte transfers by source and destination IP addresses" and is all about discovering the top 10 biggest data transfers between IP addresses in my network!





nextwork.org

The place to learn & showcase your skills

Check out nextwork.org for more projects

