

ICT & Infra S3 S/NO week 7: AWS VPC Monitoring

Date: Nov 2022
Version 1.0

Class: I3 CB01

Student numbers: 4642295, 4216709, 4961854

Student names: Ryan Smith, Edris Rahimi, Heiko Morales

Introduction

S/NO: In this exercise you will learn how to monitor and analyse AWS VPC traffic with AWS CloudWatch and Athena tools.

How to deliver your assignments?

Fill in this document with required information. Answer questions and upload the document to Canvas at most one week after the assignment is given.

Assignment 1: Create log files for CloudWatch and S3 bucket

- Follow the demos from the lecture. Create necessary entities / configurations in AWS for a chosen VPC with respect to your case-study.

I started by creating an IAM role for this task and added a trusted relationship. I also added the policy I created for this task

[IAM](#) > [Roles](#) > [vpcflowlogsrole](#)

vpcflowlogsrole

[Delete](#)

Summary

[Edit](#)

Creation date

November 12, 2022, 12:18 (UTC+01:00)

ARN

[arn:aws:iam::807013657668:role/vpcflowlogsrole](#)

Last activity

None

Maximum session duration

1 hour

Permissions

Trust relationships

Tags

Access Advisor

Revoke sessions

Permissions policies (1) [Info](#)

You can attach up to 10 managed policies.

[Simulate](#)[Remove](#)[Add permissions](#) ▼

Filter policies by property or policy name and press enter.

< 1 >

<input type="checkbox"/>	Policy name	Type	Description
<input type="checkbox"/>	flowlogsRole	Customer managed	

Trusted entities

Entities that can assume this role under specified conditions.

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Sid": "",  
6       "Effect": "Allow",  
7       "Principal": {  
8         "Service": "vpc-flow-logs.amazonaws.com"  
9       },  
10      "Action": "sts:AssumeRole"  
11    }  
12  ]  
13 }
```

Permissions

Policy usage

Tags

Policy versions

Access Advisor

Policy summary

{ } JSON

Edit policy

?

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Action": [  
6         "logs:CreateLogGroup",  
7         "logs:CreateLogStream",  
8         "logs:DescribeLogGroups",  
9         "logs:DescribeLogStreams",  
10        "logs:PutLogEvents"  
11      ],  
12      "Effect": "Allow",  
13      "Resource": "*"   
14    }  
15  ]  
16 }
```

I then created 3 flow logs. The accept and reject flow log have the same destination which is a log group in cloudwatch, but the all flow log goes to the s3 bucket I created for this task

Flow logs (3) Info

Actions ▼

Create flow log

Q Filter flow logs

< 1 > ⚙

<input type="checkbox"/>	Name ▼	Flow log ID ▼	Filter
<input type="checkbox"/>	firepass-vpc-log-accept	fl-0ec9df7bfe4fb2e77	ACCEPT
<input type="checkbox"/>	firepass-vpc-log-reject	fl-003cd1c245cba0ea9	REJECT
<input type="checkbox"/>	firepass-vpc-log-all	fl-0ba5d7bbb6139b612	ALL

firepass-log-bucket Info

Objects

Properties

Permissions

Metrics

Management

Access Points

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant

Copy S3 URI

Copy URL

Download

Open

Delete

Actions ▼

Create folder

Upload

Q Find objects by prefix

<input type="checkbox"/>	Name ▲	Type ▼	Last modified ▼	Size ▼	Storage class
<input type="checkbox"/>	AWSLogs/	Folder	-	-	-

This is the log stream output when I created the 2 flow logs

sno7-vpc-log

▼ Log group details

ARN

arn:aws:logs:eu-central-1:807013657668:log-group:sno7-vpc-log:*

Creation time

1 hour ago

Retention

Never expire

Stored bytes

-

Metric filters

0

Subscription filters

0

Contributor Insights rules

-

Log streams

Metric filters

Subscription filters

Contributor Insights

Tags

Log streams (5)

🔍 Filter log streams or try prefix search

☐ Exact match

<input type="checkbox"/>	Log stream ▼	Last event time
<input type="checkbox"/>	eni-068164457f34cdfdf-reject	2022-11-12 13:13:30 (UTC+01:00)
<input type="checkbox"/>	eni-0144a1c8fc56970e6-accept	2022-11-12 12:54:35 (UTC+01:00)
<input type="checkbox"/>	eni-03187528e5ca8091c-accept	2022-11-12 12:49:45 (UTC+01:00)
<input type="checkbox"/>	eni-0447ae6220e715cf1-accept	2022-11-12 12:49:20 (UTC+01:00)
<input type="checkbox"/>	eni-068164457f34cdfdf-accept	2022-11-12 12:43:15 (UTC+01:00)

These are the log event outputs

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" value="Filter events"/>			
	Timestamp	Message	Log stream name
▶	2022-11-12T12:33:08.000+01:00	2 807013657668 eni-0144a1c8fc56970e6 61.177.173.46 19...	eni-0144a1c8fc56970e6-accept
▶	2022-11-12T12:33:08.000+01:00	2 807013657668 eni-0144a1c8fc56970e6 61.177.173.46 19...	eni-0144a1c8fc56970e6-accept
▶	2022-11-12T12:33:08.000+01:00	2 807013657668 eni-0144a1c8fc56970e6 192.168.16.169 6...	eni-0144a1c8fc56970e6-accept
▶	2022-11-12T12:33:08.000+01:00	2 807013657668 eni-0144a1c8fc56970e6 192.168.16.169 6...	eni-0144a1c8fc56970e6-accept
▶	2022-11-12T12:33:17.000+01:00	2 807013657668 eni-0447ae6220e715cf1 192.168.16.235 5...	eni-0447ae6220e715cf1-accept
▶	2022-11-12T12:33:43.000+01:00	2 807013657668 eni-03187528e5ca8091c 192.168.0.83 20...	eni-03187528e5ca8091c-accept
▶	2022-11-12T12:33:43.000+01:00	2 807013657668 eni-03187528e5ca8091c 20.40.73.192 192...	eni-03187528e5ca8091c-accept
▶	2022-11-12T12:34:07.000+01:00	2 807013657668 eni-0144a1c8fc56970e6 61.177.173.46 19...	eni-0144a1c8fc56970e6-accept
▶	2022-11-12T12:34:07.000+01:00	2 807013657668 eni-0144a1c8fc56970e6 192.168.16.169 6...	eni-0144a1c8fc56970e6-accept
▶	2022-11-12T12:34:14.000+01:00	2 807013657668 eni-0447ae6220e715cf1 192.168.16.235 1...	eni-0447ae6220e715cf1-accept
▶	2022-11-12T12:34:14.000+01:00	2 807013657668 eni-0447ae6220e715cf1 192.168.16.235 5...	eni-0447ae6220e715cf1-accept
▶	2022-11-12T12:34:14.000+01:00	2 807013657668 eni-0447ae6220e715cf1 183.136.225.32 1...	eni-0447ae6220e715cf1-accept
▶	2022-11-12T12:35:17.000+01:00	2 807013657668 eni-0447ae6220e715cf1 192.168.16.235 5...	eni-0447ae6220e715cf1-accept
▶	2022-11-12T12:35:17.000+01:00	2 807013657668 eni-0447ae6220e715cf1 192.168.16.235 5...	eni-0447ae6220e715cf1-accept
▶	2022-11-12T12:36:18.000+01:00	2 807013657668 eni-0447ae6220e715cf1 192.168.16.235 5...	eni-0447ae6220e715cf1-accept
▶	2022-11-12T12:36:21.000+01:00	2 807013657668 eni-068164457f34cdfdf 192.168.0.40 212...	eni-068164457f34cdfdf-accept
▶	2022-11-12T12:36:21.000+01:00	2 807013657668 eni-068164457f34cdfdf 212.102.58.164 1...	eni-068164457f34cdfdf-accept
▶	2022-11-12T12:36:39.000+01:00	2 807013657668 eni-03187528e5ca8091c 192.168.0.83 118...	eni-03187528e5ca8091c-accept
▶	2022-11-12T12:36:39.000+01:00	2 807013657668 eni-03187528e5ca8091c 192.168.0.83 118...	eni-03187528e5ca8091c-accept
▶	2022-11-12T12:36:39.000+01:00	2 807013657668 eni-03187528e5ca8091c 118.34.123.43 19...	eni-03187528e5ca8091c-accept
▶	2022-11-12T12:36:39.000+01:00	2 807013657668 eni-03187528e5ca8091c 118.34.123.43 19...	eni-03187528e5ca8091c-accept
▶	2022-11-12T12:37:15.000+01:00	2 807013657668 eni-0447ae6220e715cf1 192.168.16.235 2...	eni-0447ae6220e715cf1-accept
▶	2022-11-12T12:37:15.000+01:00	2 807013657668 eni-0447ae6220e715cf1 212.102.58.164 1...	eni-0447ae6220e715cf1-accept
▶	2022-11-12T12:37:15.000+01:00	2 807013657668 eni-0447ae6220e715cf1 192.168.16.235 5...	eni-0447ae6220e715cf1-accept

And these are the log event outputs using this filter:

[version, accountid, interfaceid, srcaddr, dstaddr, srcport, dstport, protocol !=6 && protocol > 1, packets, bytes, start, end, action, logstatus]

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" value="[version, accountid, interfaceid, srcaddr, dstaddr, srcport, dstport, protocol !=6 && protocol > 1, packets, bytes, start, end, action, logstatus]"/>			
▼	Timestamp	Message	Log stream name
▼	2022-11-12T13:13:15.000+01:00	2 807013657668 eni-0144a1c8fc56970e6 45.142.192.10 19...	eni-0144a1c8fc56970e6-reject
		2 807013657668 eni-0144a1c8fc56970e6 45.142.192.10 192.168.16.169 39073 1900 17 1 127 1668255195 1668255249 REJECT OK	
▼	2022-11-12T13:15:15.000+01:00	2 807013657668 eni-0144a1c8fc56970e6 222.147.109.13 1...	eni-0144a1c8fc56970e6-reject
		2 807013657668 eni-0144a1c8fc56970e6 222.147.109.13 192.168.16.169 0 0 47 1 564 1668255315 1668255369 REJECT OK	

These are the results of the sample queries. I have added the query used to get each result

stats count(*) as records by srcAddr | sort records desc

sno7-vpc-log X

```
1 stats count(*) as records by srcAddr | sort records desc
2
```

Run query

Cancel

Save

History

Queries are allowed to run for up to 15 minutes.

Logs

Visualization



#	srcAddr	records
▶ 1	192.168.0.83	361
▶ 2	3.1.221.199	315
▶ 3	192.168.16.2...	172
▶ 4	192.168.16.1...	153
▶ 5	89.248.163.2...	111
▶ 6	46.17.96.38	99
▶ 7	52.59.235.149	78
▶ 8	5.188.87.3	69
▶ 9	46.161.27.85	57
▶ 10	92.63.197.133	51
▶ 11	45.143.200.1...	50
▶ 12	185.156.73.1...	49
▶ 13	45.227.253.99	47

stats sum(bytes) as bytesTransferred by srcAddr, srcPort, dstAddr

| sort bytesTransferred desc

| limit 10

sno7-vpc-log X

```
1 stats sum(bytes) as bytesTransferred by srcAddr, dstAddr
2 | sort bytesTransferred desc
3 | limit 10
```

Run query

Cancel

Save

History

Queries are allowed to run for up to 15 minutes.

Logs

Visualization

Export results ▼

Showing 10 of 3,916 records matched ⓘ

3,916 records (554.0 kB) scanned in 4.5s @ 862 records/s (1

100
80
60
40
20
0

2022

Apr

Jul

#	srcAddr	dstAddr	bytesTransferred
▶ 1	52.59.235.149	192.168.16.235	17220
▶ 2	192.168.16.235	171.212.102.210	10273
▶ 3	192.168.16.169	61.177.173.49	7200
▶ 4	192.168.0.40	212.102.58.164	6906
▶ 5	192.168.16.169	45.33.80.243	6767
▶ 6	192.168.0.40	45.79.181.94	6767
▶ 7	34.243.160.129	192.168.16.169	5780
▶ 8	192.168.16.169	61.177.173.35	5760
▶ 9	192.168.0.40	113.200.105.23	4949
▶ 10	192.168.0.83	45.79.172.21	4911

sno7-vpc-log X

```
1 filter (srcPort > 1023 and srcAddr != "10.1.254.10") |
2   |> stats count(*) as records by srcAddr, srcPort, dstAddr |
3   |> sort records desc | limit 5
```

Run query

Cancel

Save

History

Queries are allowed to run for up to 15 minutes.

Logs

Visualization

Export results ▼

Add to

Showing 5 of 3,372 records matched ⓘ

4,006 records (566.9 kB) scanned in 3.5s @ 1,141 records/s (161.5 kB



#	srcAddr	srcPort	dstAddr	records
▶ 1	46.17.96.38	47428	192.168.16.2...	117
▶ 2	5.188.87.3	51533	192.168.0.83	48
▶ 3	46.161.27.85	50357	192.168.0.83	39
▶ 4	89.248.165.51	41866	192.168.16.1...	35
▶ 5	89.248.165.83	42046	192.168.0.40	33

filter (action="REJECT") | stats count_distinct(dstPort) as portcount by srcAddr | sort portcount desc | limit 5

sno7-vpc-log X

```
1 filter (action="REJECT") | stats count_distinct(dstPort) as portcount by srcAddr | sort
```

Run query

Cancel

Save

History

Queries are allowed to run for up to 15 minutes.

Logs

Visualization

Export results ▼

Add to dashboard



Showing 5 of 2,672 records matched ⓘ

[Hide histogram](#)

4,017 records (568.4 kB) scanned in 3.9s @ 1,038 records/s (147.0 kB/s)



	srcAddr	portcount
▶ 1	46.17.96.38	117
▶ 2	5.188.87.3	80
▶ 3	45.227.253.99	61
▶ 4	45.143.200.1...	56
▶ 5	185.156.73.1...	56

filter (action="REJECT") | stats count_distinct(dstPort) as portcount, count_distinct(srcAddr) as sourcecount by bin(15m)

sno7-vpc-log X

```
1 _distinct(dstPort) as portcount, count_distinct(srcAddr) as sourcecount by bin(15m)
```

Run query

Cancel

Save

History

Queries are allowed to run for up to 15 minutes.

Logs

Visualization

Export results ▼

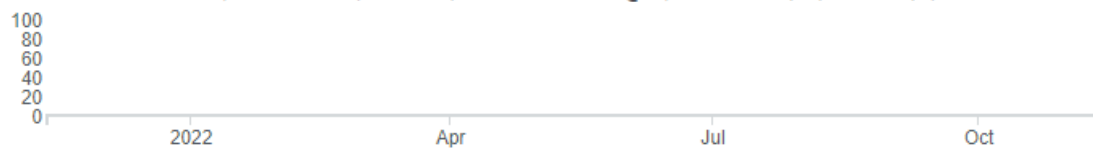
Add to dashboard



Showing 6 of 2,694 records matched ⓘ

[Hide histogram](#)

4,052 records (573.4 kB) scanned in 3.9s @ 1,040 records/s (147.2 kB/s)



#	bin(15m)	portcount	sourcecount
▶ 1	2022-11-12T14:15:00.00...	7	6
▶ 2	2022-11-12T14:00:00.00...	447	217
▶ 3	2022-11-12T13:45:00.00...	506	244
▶ 4	2022-11-12T13:30:00.00...	593	232
▶ 5	2022-11-12T13:15:00.00...	533	241
▶ 6	2022-11-12T13:00:00.00...	97	73

After that, I switched to Athena and ran this query

```
CREATE EXTERNAL TABLE IF NOT EXISTS vpc_flow_logs (  
    version int,  
    account string,  
    interfaceid string,  
    sourceaddress string,  
    destinationaddress string,  
    sourceport int,  
    destinationport int,  
    protocol int,  
    numpackets int,  
    numbytes bigint,  
    starttime int,  
    endtime int,  
    action string,  
    logstatus string  
)  
  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY ' '  
  
LOCATION s3://firepass-log-bucket/AWSLogs/807013657668/vpcflowlogs/  
  
TBLPROPERTIES ("skip.header.line.count"="1");
```

After that ran, I ran this query to check the output of the last one

```
SELECT sourceaddress,  
       approx_distinct(destinationport) AS destPortCount  
FROM "default"."vpc_flow_logs"  
WHERE action = 'REJECT'  
       AND from_unixtime(starttime) > date_add('day', -365, current_timestamp)  
GROUP BY sourceaddress  
ORDER BY destPortCount DESC limit 10
```

Results (10)

Search rows

#	sourceaddress	destPortCount
1	46.17.96.38	149
2	89.248.163.217	111
3	5.188.87.3	98
4	46.161.27.85	77
5	45.227.253.99	77
6	45.143.200.102	66
7	185.156.73.153	66
8	92.63.197.133	63
9	185.156.73.57	60
10	92.63.197.154	58

Assignment 2: Analyse CloudWatch or S3 bucket log files

- Think about some useful scenario (DDoS, port scanning) for your VPC malicious network activities forensics.
- Run this malicious scenario to your VPC, subnet or network interface.
- Demonstrate the analysis of the network flow logs and indicate this malicious activities in search results or CloudWatch Insights graph.
- Create an alarm and notification by email/sms if this malicious activity is detected (CloudWatch).

I started this task by using an nmap command on the Zabbix server to port scan

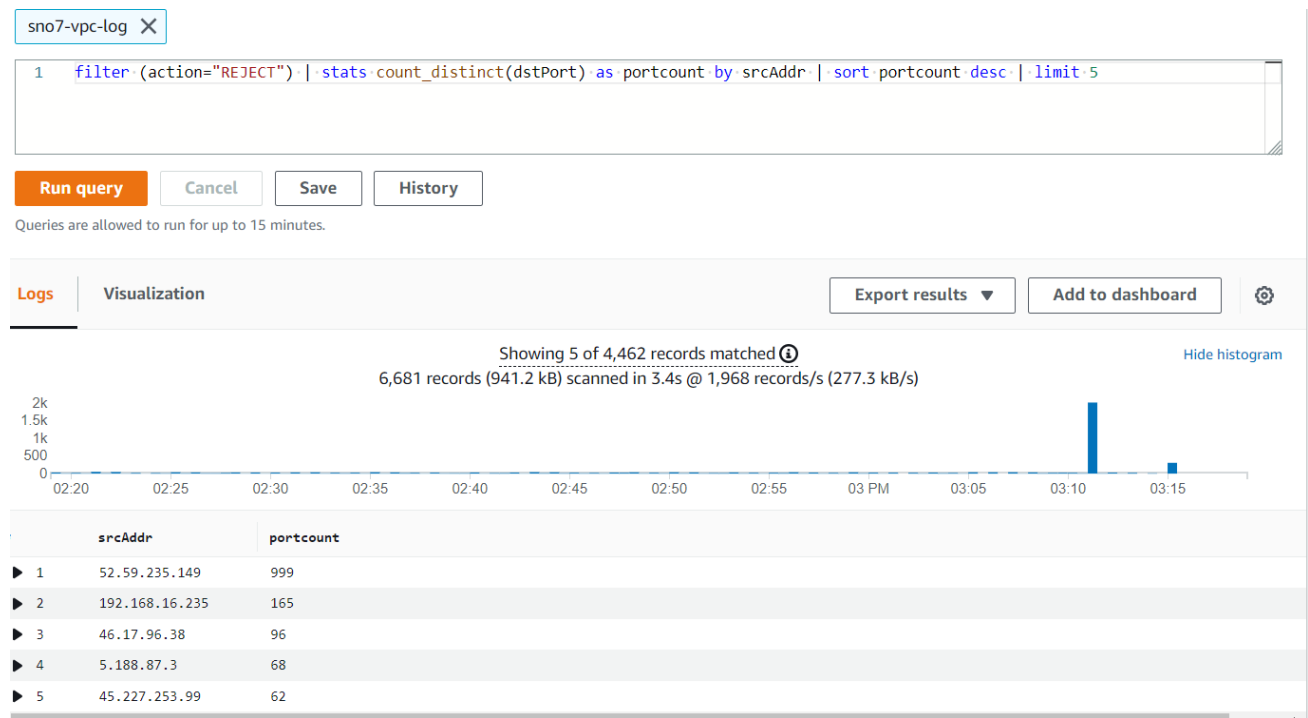
Instance summary for i-0a7fbf7fa758dd06d (Zabbix)
[Info](#)

Connect
Instance state ▼
Actions ▼

Updated less than a minute ago

Instance ID i-0a7fbf7fa758dd06d (Zabbix)	Public IPv4 address 52.59.235.149 open address	Private IPv4 addresses 192.168.16.235
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-52-59-235-149.eu-central-1.compute.amazonaws.com open address
Hostname type IP name: ip-192-168-16-235.eu-central-1.compute.internal	Private IP DNS name (IPv4 only) ip-192-168-16-235.eu-central-1.compute.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Auto-assigned IP address 52.59.235.149 [Public IP]	VPC ID vpc-08ed2bb7fdca49b9f (firepass-vpc)	Auto Scaling Group name -
IAM Role -	Subnet ID subnet-0ae357abaf727b3b9 (firepass-subnet-public2-eu-central-1b)	

After that, I checked the log and the port scan from the Zabbix server was confirmed



After that, I set a cloudwatch alarm to email me if malicious activity is detected

Alarms (1) ☐ Hide Auto Scaling alarms Clear select

Q Search

<input type="checkbox"/>	Name	State	Last state update
<input type="checkbox"/>	sno7-alarm	Insufficient data	2022-11-12 15:25:55