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.

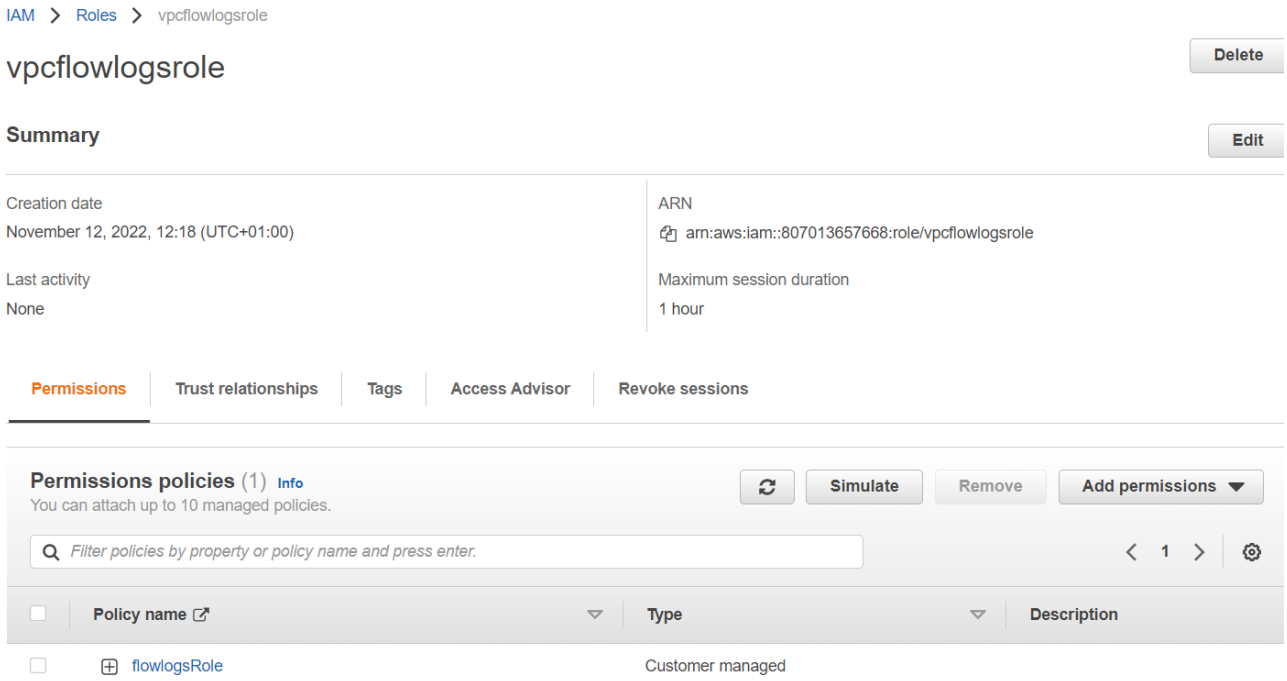
How to deliver your assignments?

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



Trusted entities

Entities that can assume this role under specified conditions.

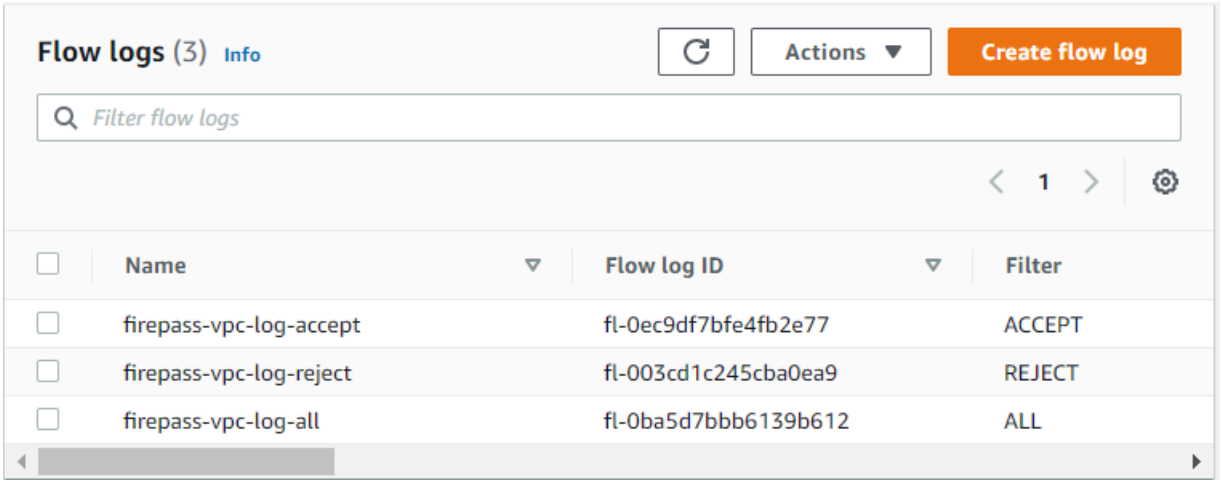
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "vpc-flow-logs.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

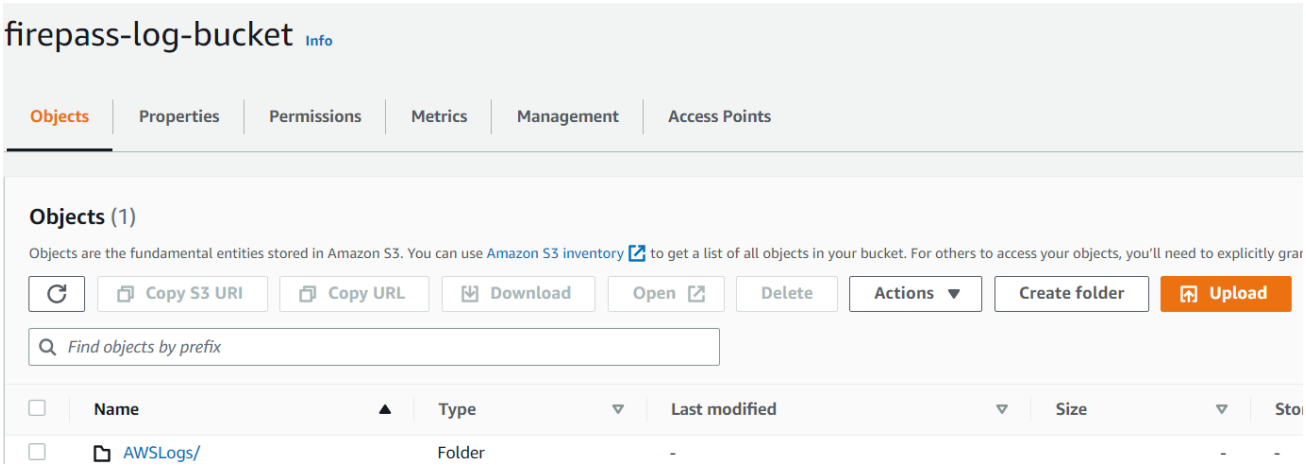
Permissions | Policy usage | Tags | Policy versions | Access Advisor

Policy summary | { } JSON | Edit policy

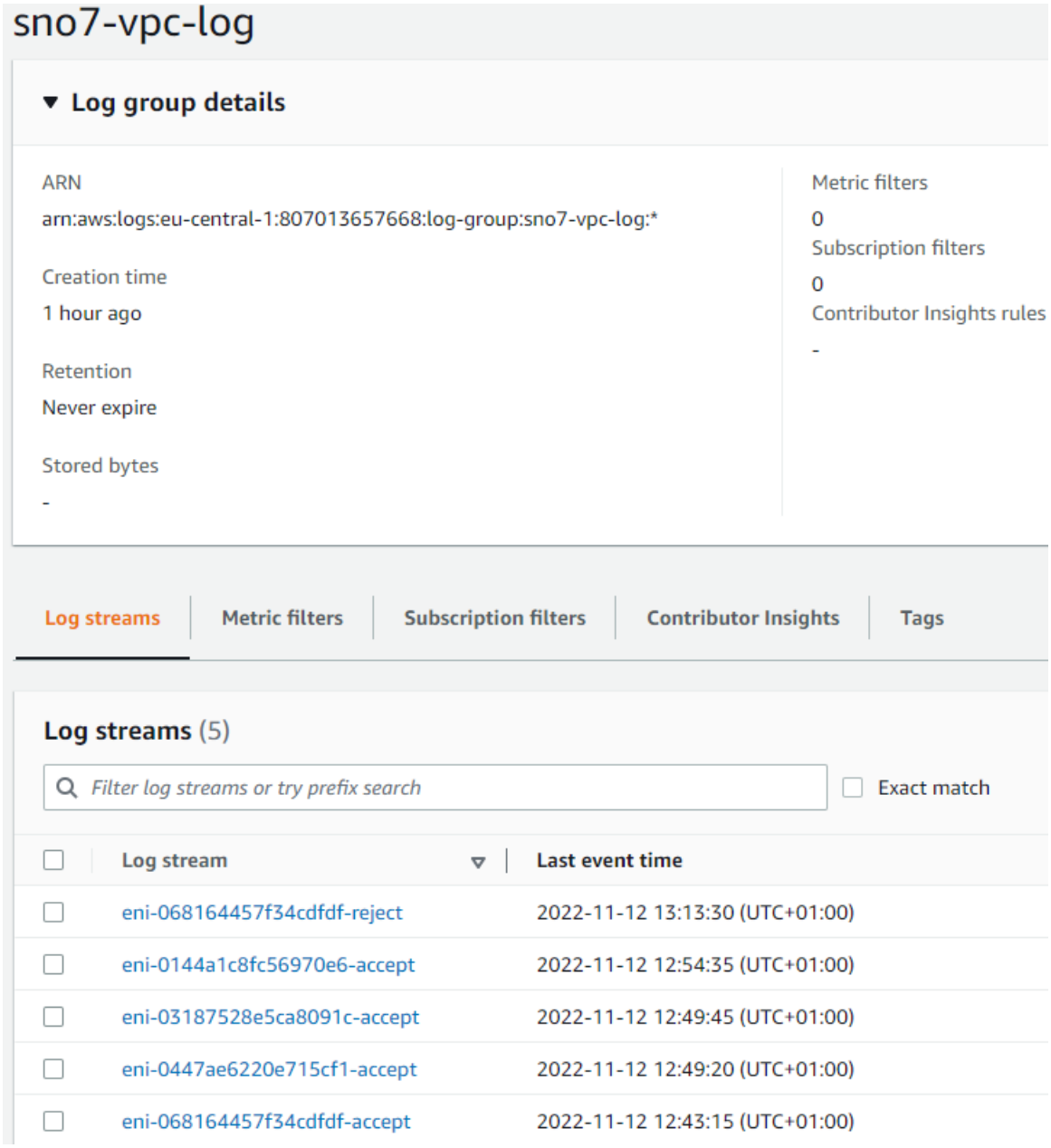
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

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

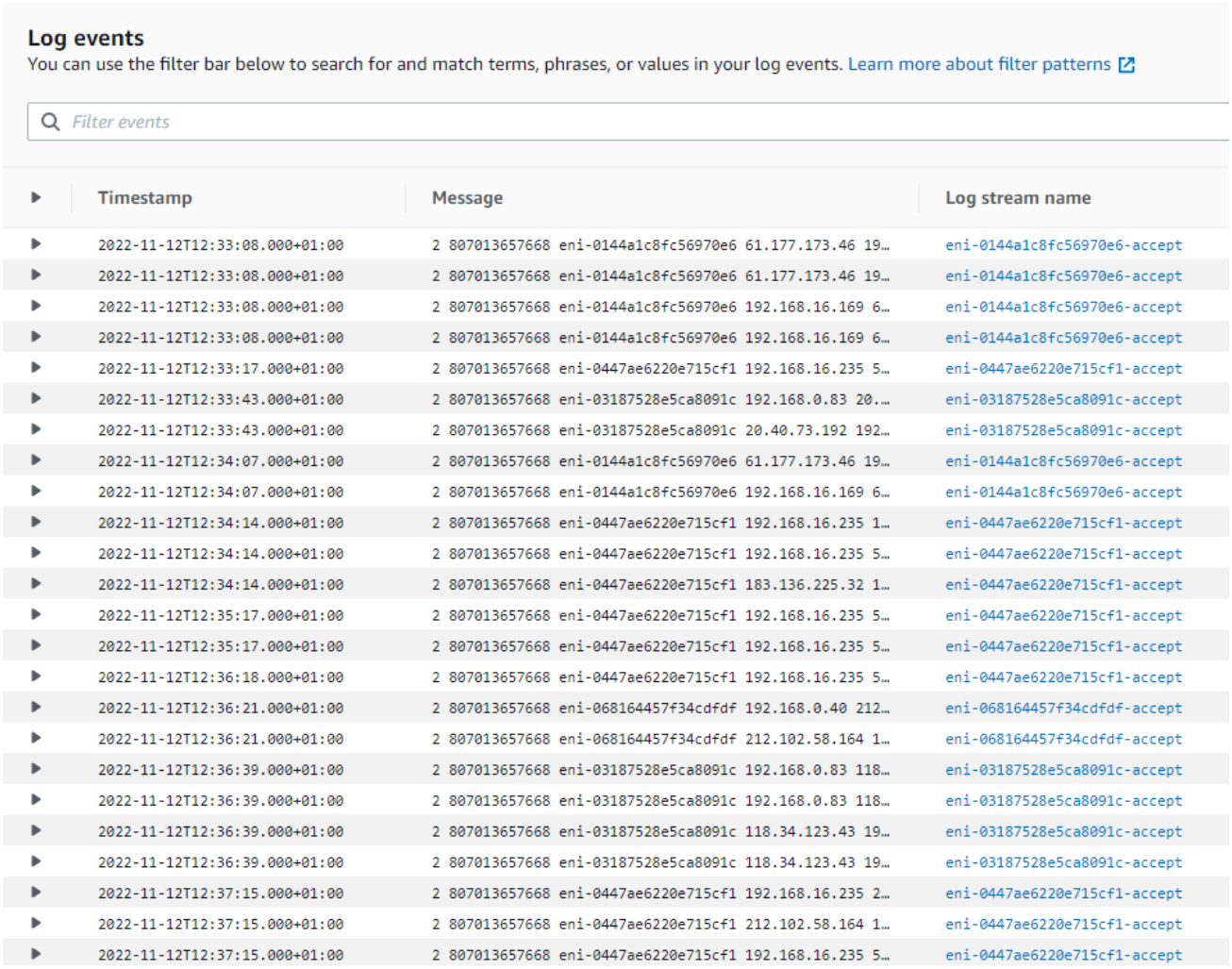




This is the log stream output when I created the 2 flow logs



These are the log event outputs



Log events

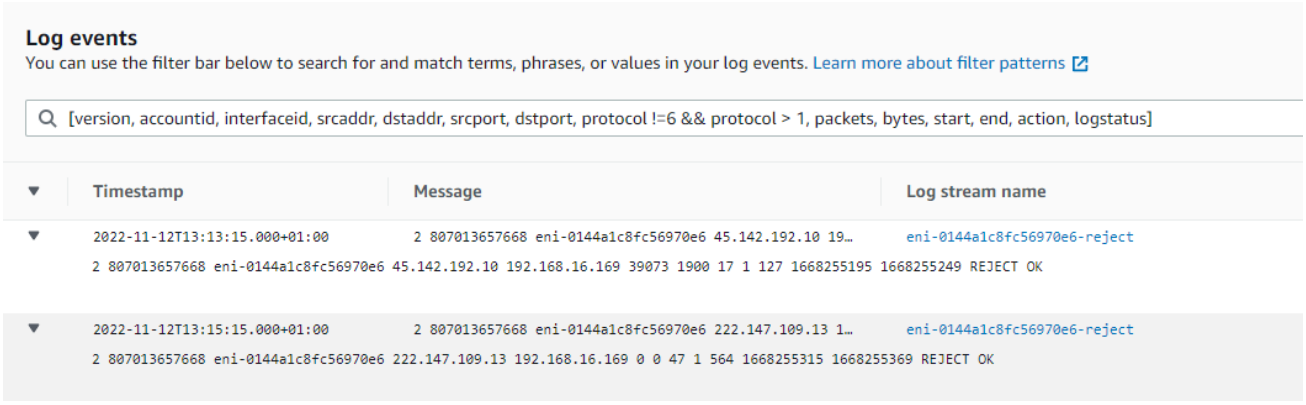
You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

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

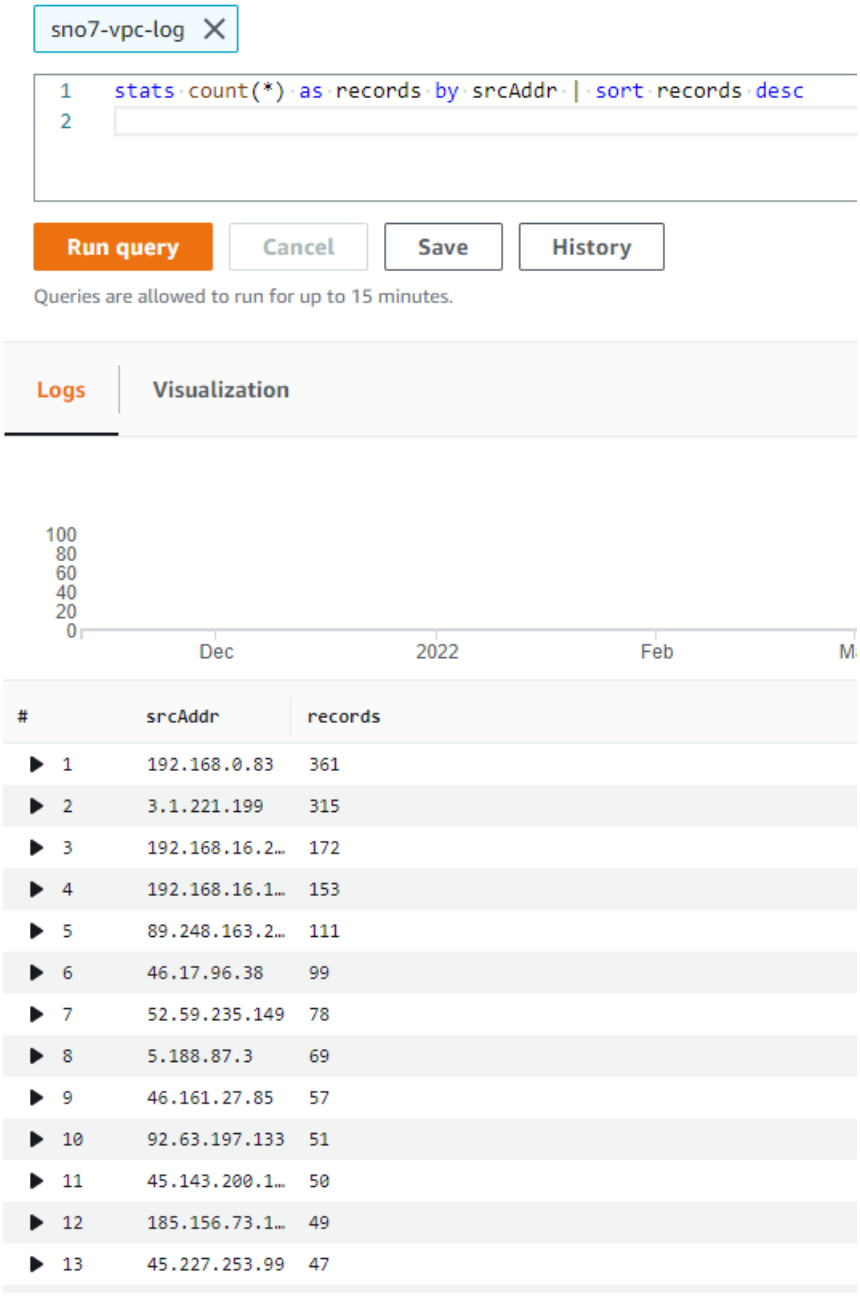
You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

[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

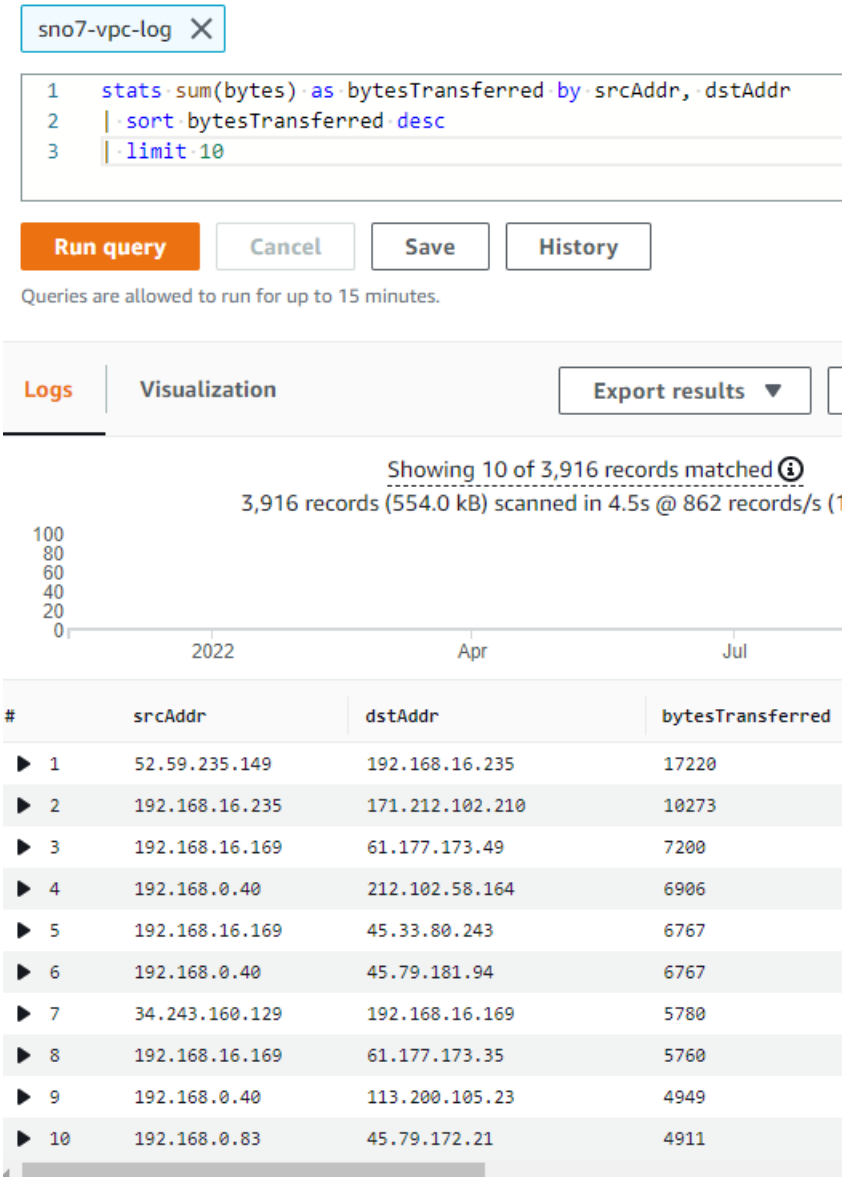


#	srcAddr	records
1	192.168.0.83	361
2	3.1.221.199	315
3	192.168.16.2…	172
4	192.168.16.1…	153
5	89.248.163.2…	111
6	46.17.96.38	99
7	52.59.235.149	78
8	5.188.87.3	69
9	46.161.27.85	57
10	92.63.197.133	51
11	45.143.200.1…	50
12	185.156.73.1…	49
13	45.227.253.99	47

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

| sort bytesTransferred desc

| limit 10



sno7-vpc-log

```
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
```

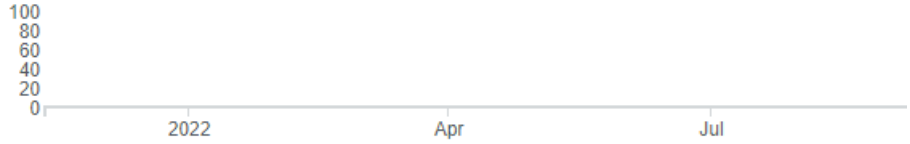
Run query | Cancel | Save | History

Queries are allowed to run for up to 15 minutes.

Logs | Visualization | Export results | Add to

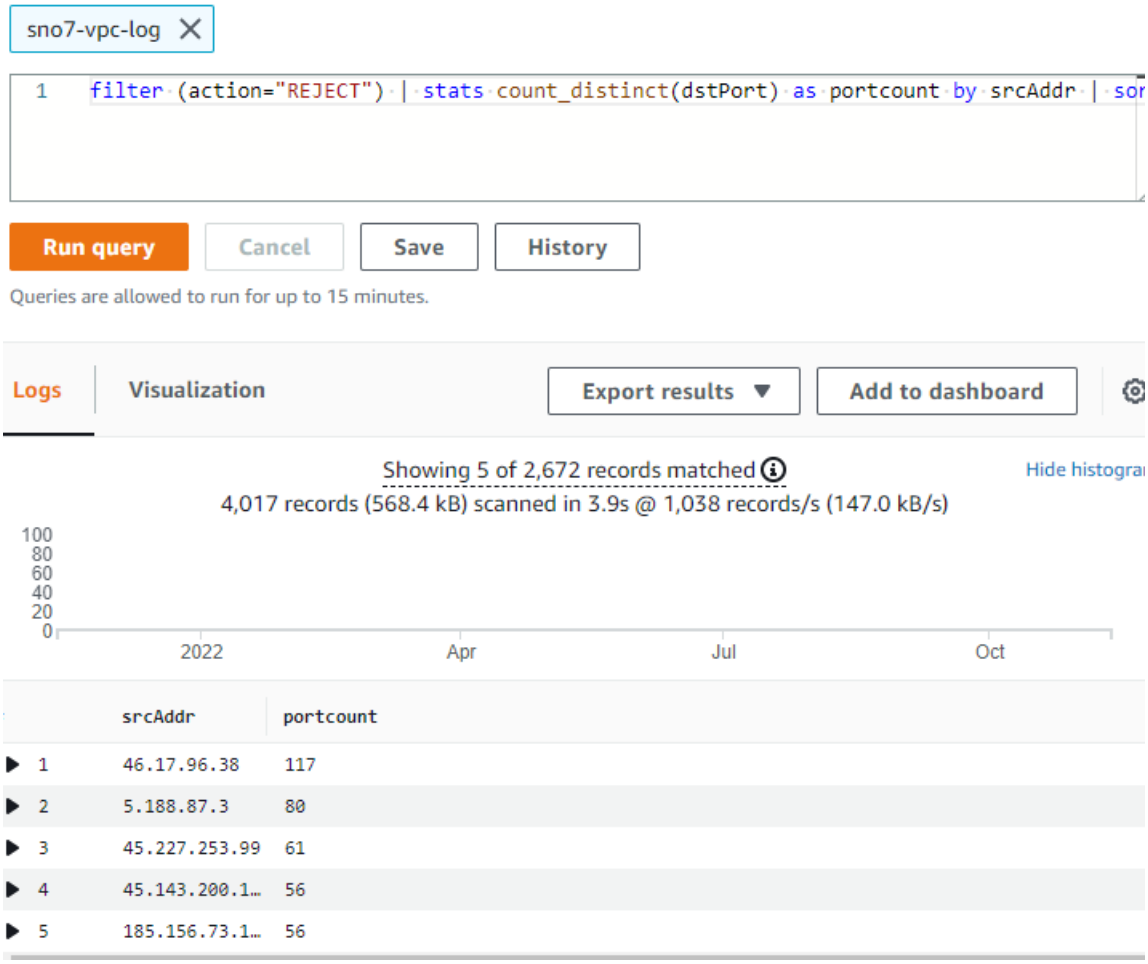
Showing 5 of 3,372 records matched

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

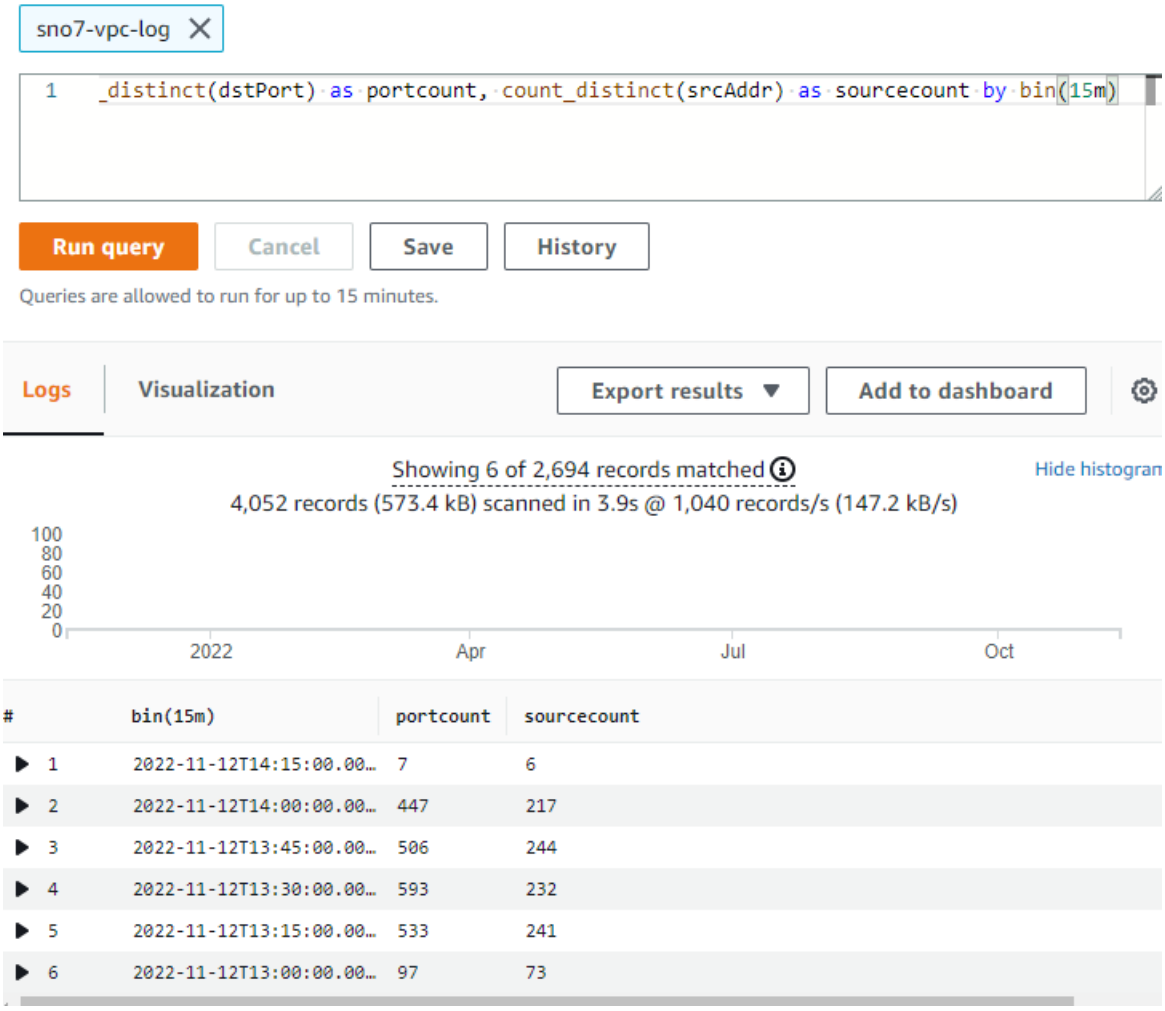


#	srcAddr	srcPort	dstAddr	records
1	46.17.96.38	47428	192.168.16.2…	117
2	5.188.87.3	51533	192.168.0.83	48
3	46.161.27.85	50357	192.168.0.83	39
4	89.248.165.51	41866	192.168.16.1…	35
5	89.248.165.83	42046	192.168.0.40	33

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



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



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

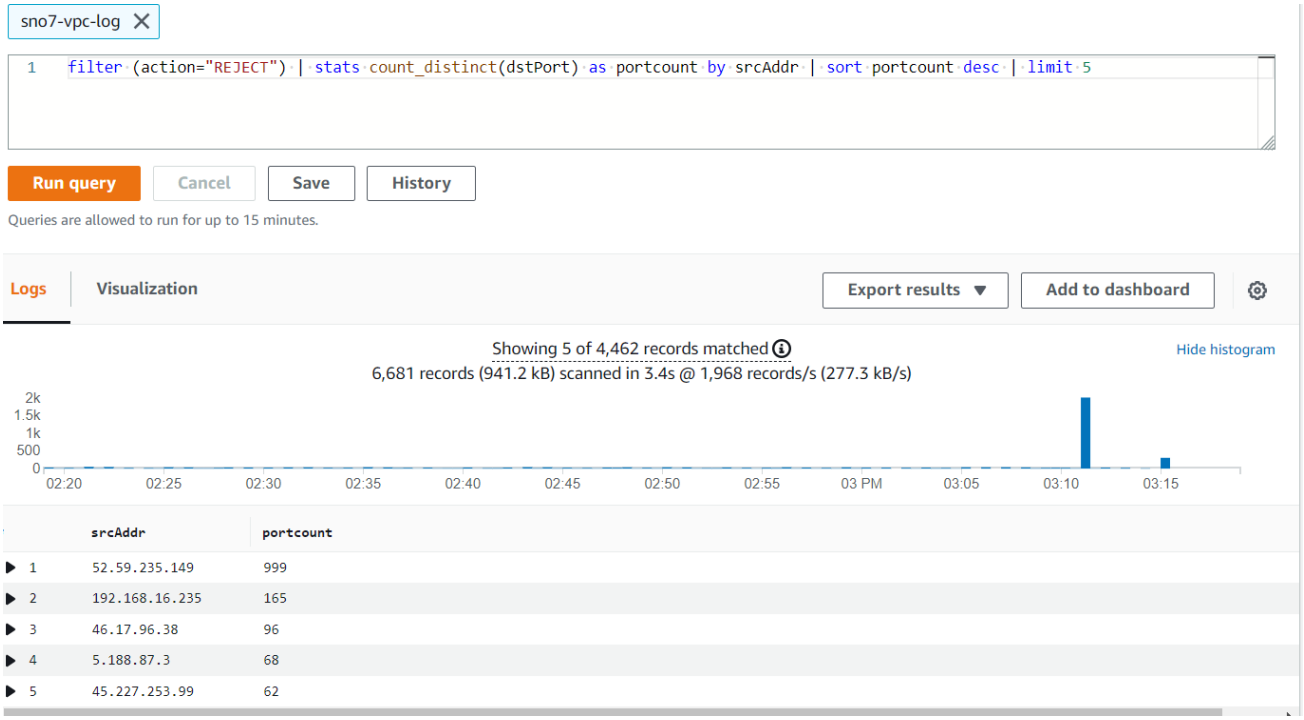
Instance summary for i-0a7fbf7fa758dd06d (Zabbix) Info

Updated less than a minute ago

Connect | Instance state | Actions

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	
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	Elastic IP addresses –
Auto-assigned IP address 52.59.235.149 [Public IP]	VPC ID vpc-08ed2bb7fdca49b9f (firepass-vpc)	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role –	Subnet ID subnet-0ae357abaf727b3b9 (firepass-subnet-public2-eu-central-1b)	Auto Scaling Group name –

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