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

**Class:**

**Student numbers:**

**Student names:**

Date: Sep 2020  
Version 1.0

# 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 the 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 an S3 bucket*

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

Provide screenshots and descriptions of the steps above

Created flow logs by following the steps in the demo and filling in the required spots, also created an S3 bucket, and connected a flow log to the bucket. The VPC flow logs for CloudWatch catch all traffic accepted and rejected with a 1-minute aggregation interval for faster analysis during testing.

Afbeelding met tekst, schermopname, nummer, Lettertype

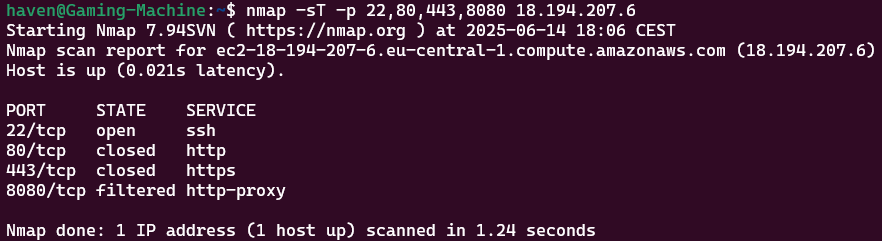
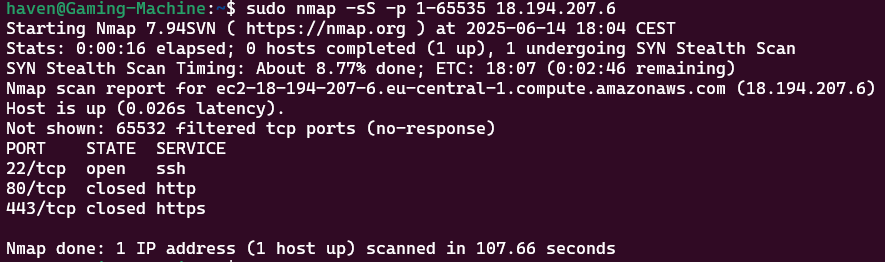
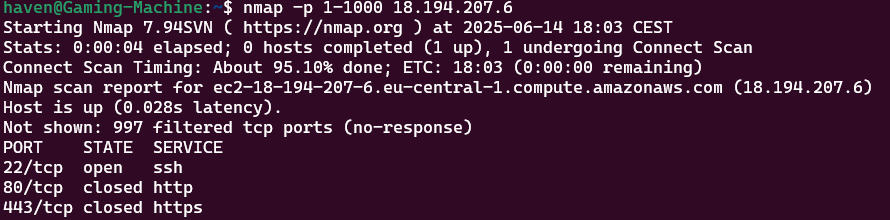
Door AI gegenereerde inhoud is mogelijk onjuist.

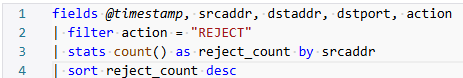
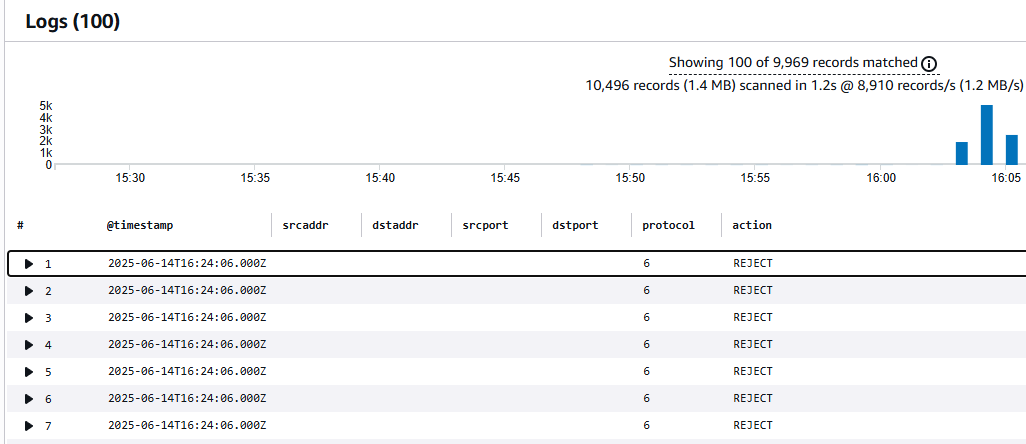
Here you can see my bucket with a log file in it. You would create flow logs in S3 for long-term storage.

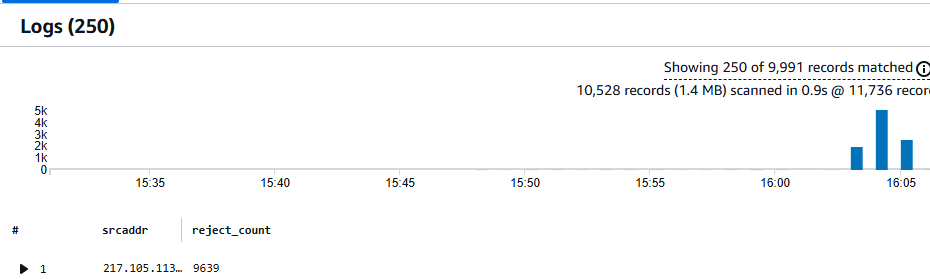
## *Assignment 2: Analyse CloudWatch or S3 bucket log files*

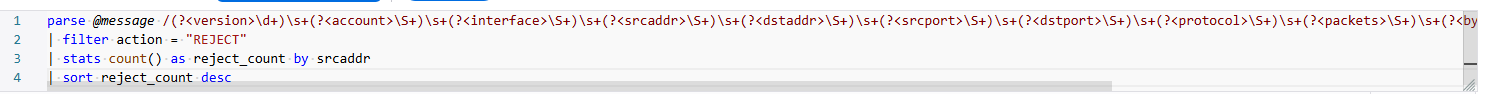
* Think about some useful scenarios (DDoS, port scanning) for your VPC's malicious network activities forensics.
* Run this malicious scenario on your VPC, subnet, or network interface.
* Demonstrate the analysis of the network flow logs and indicate these malicious activities in the search results or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

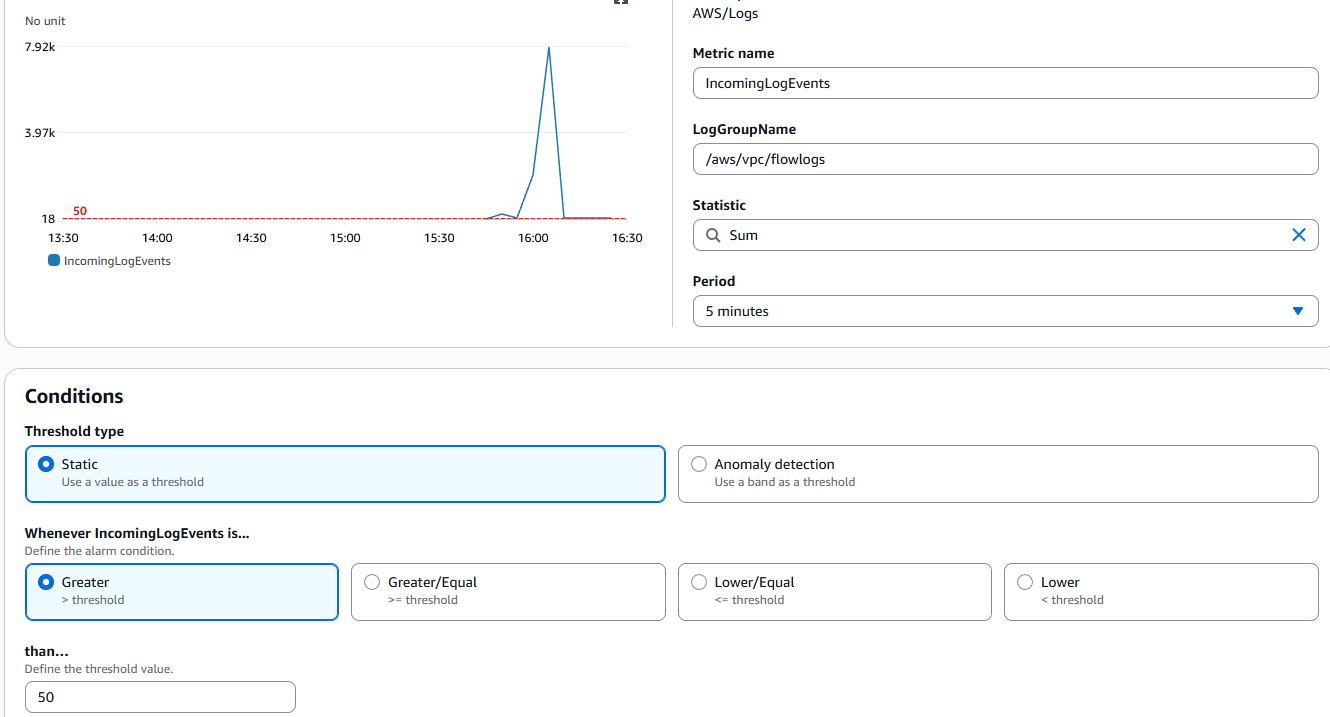
Provide screenshots and descriptions of the steps above.

  
I performed three nmap scans: a normal one, an aggressive one, and a silent one. I did this to simulate malicious activities.

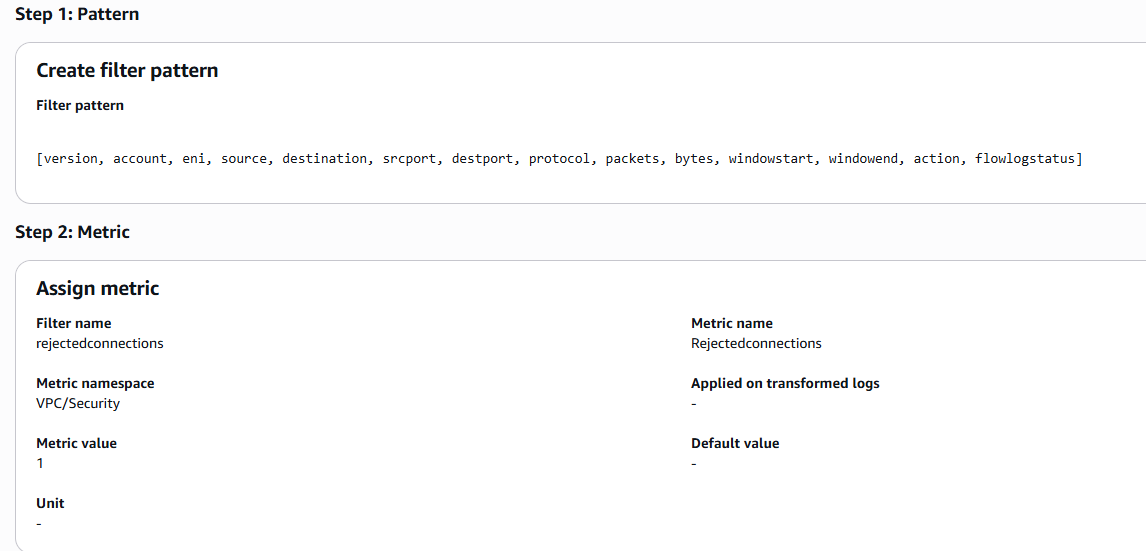


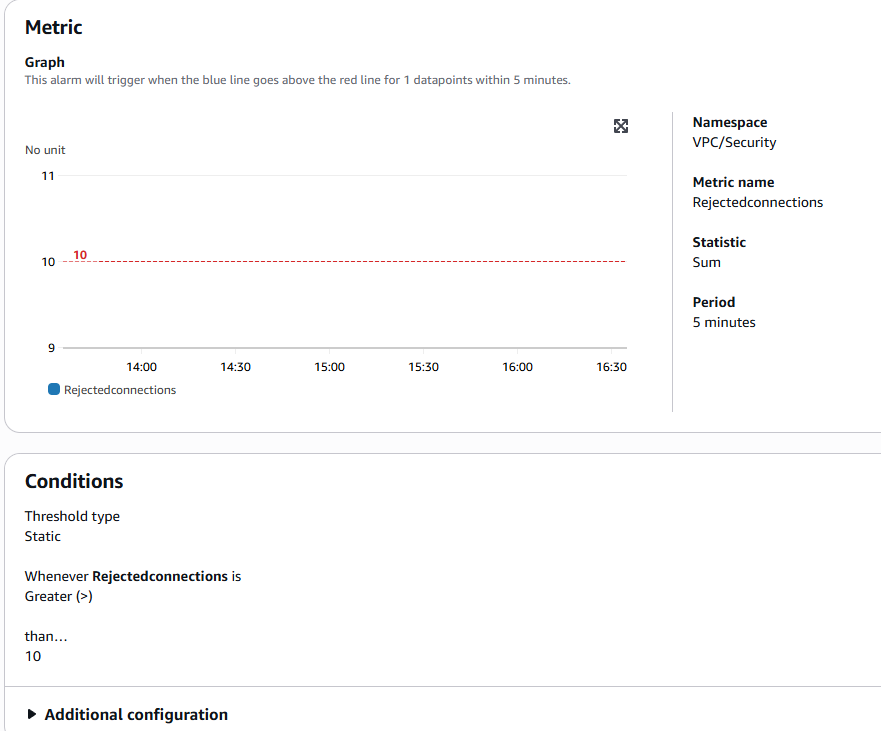


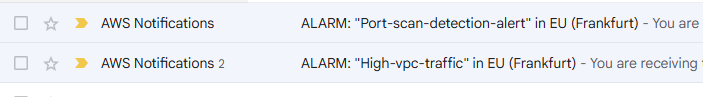
  
AI helped me create two queries, one showing all the rejected logs and the second one showing all the rejects per source IP. You can see my public IP with a reject count of 9639



Cloudwatch alarm, this alarm will go off if the average traffic is higher than 50. This can, of course, be changed to the normal traffic value.

  
I also wanted to have an alarm to detect a port scan, so I showed AI my logs, and I gave it this filter pattern. This pattern detects and gives every single thing in the logs a name, and when rejected connections get detected, the metric value goes up by 1

  
I then put this in an alarm again that goes off when something is greater than 10 in 5 minutes. After this, I did 5 more port scans to trigger both alarms.

  
*Afbeelding met tekst, schermopname, lijn, Perceel

Door AI gegenereerde inhoud is mogelijk onjuist.  
Here you can see both alarms went off and I received email notifications*