ICT & Infra S3 S/NO week 16: Capturing and analysing logs with different AWS native tools

Date: Sep 2020
Version 1.0

Class: CB01

Student numbers: 4961854

Student names: Heiko Morales

Introduction

S/NO: Following these exercises you will learn how to:

- Capture events with CloudTrail and store them in CloudWatch
- Viewing logs in CloudWatch
- Creating alerts with CloudWatch Alarms
- Searching Logs with Athena

How to deliver your assignments?

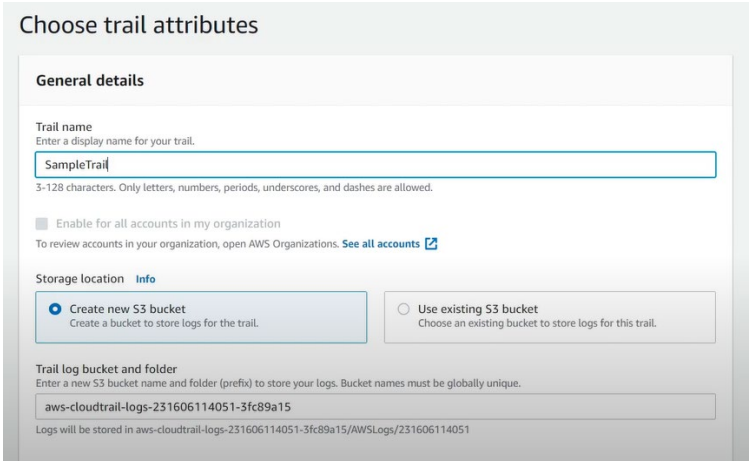
Fill in this document with required information. Answer questions and upload the document to Canvas at most one week after the assignment is given.

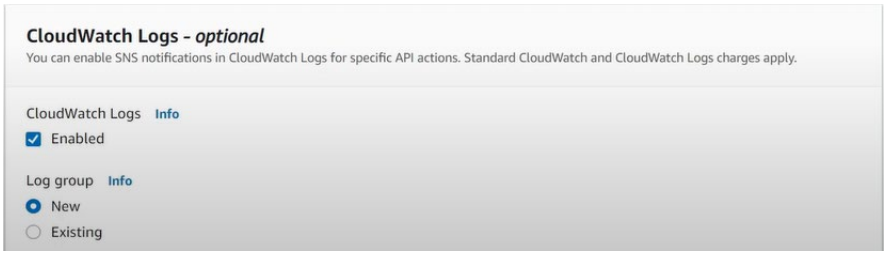
Assignment: Demonstrate security controls in your case-study project

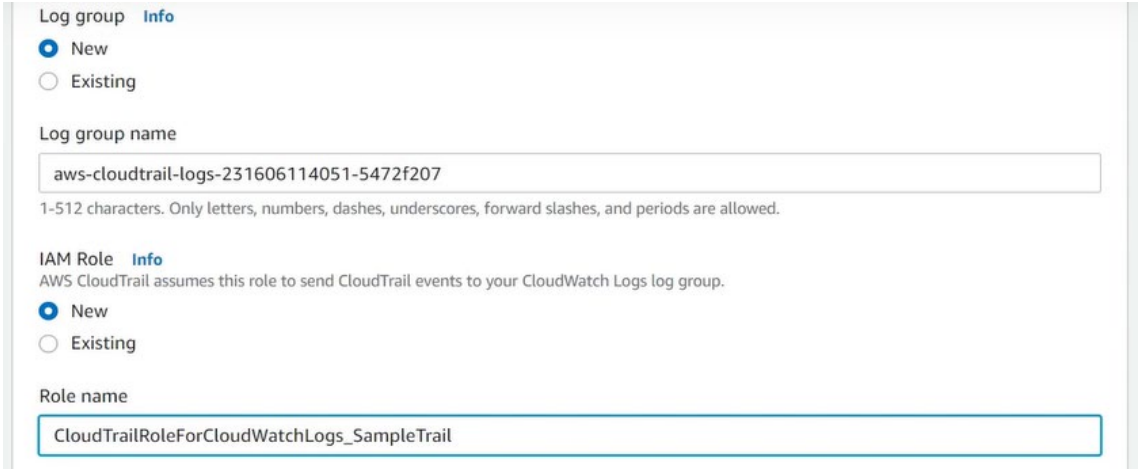
- Brainstorm together with your case-study fellow students about what kind of CloudTrail events in AWS you are going to capture. Think about the management or data events, relevant for your case-study security controls, security incident and/or problem managements analysis?
- Provide the analysis of these events either using CloudWatch or Athena tools.
- Think about alarms, notifications you can set up based on certain conditions, CloudWatch metrics.
- Think about if you can deploy Lambda functions from these alarms/notifications generated (e.g. automatic processing of the events).
- Bonus: How can you generate alarms/notifications from the results returned by Athena?

Provide screenshots and descriptions of the steps above

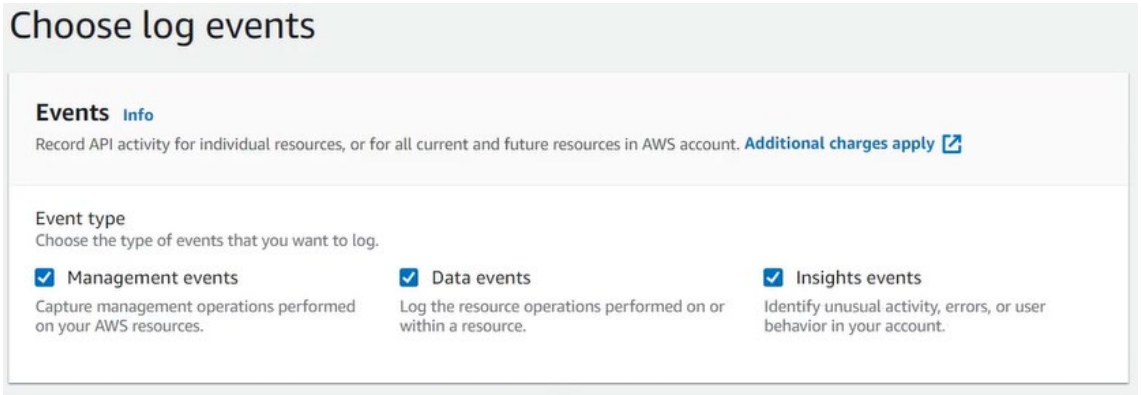
first, we will go to CloudTrail and fill in the form.

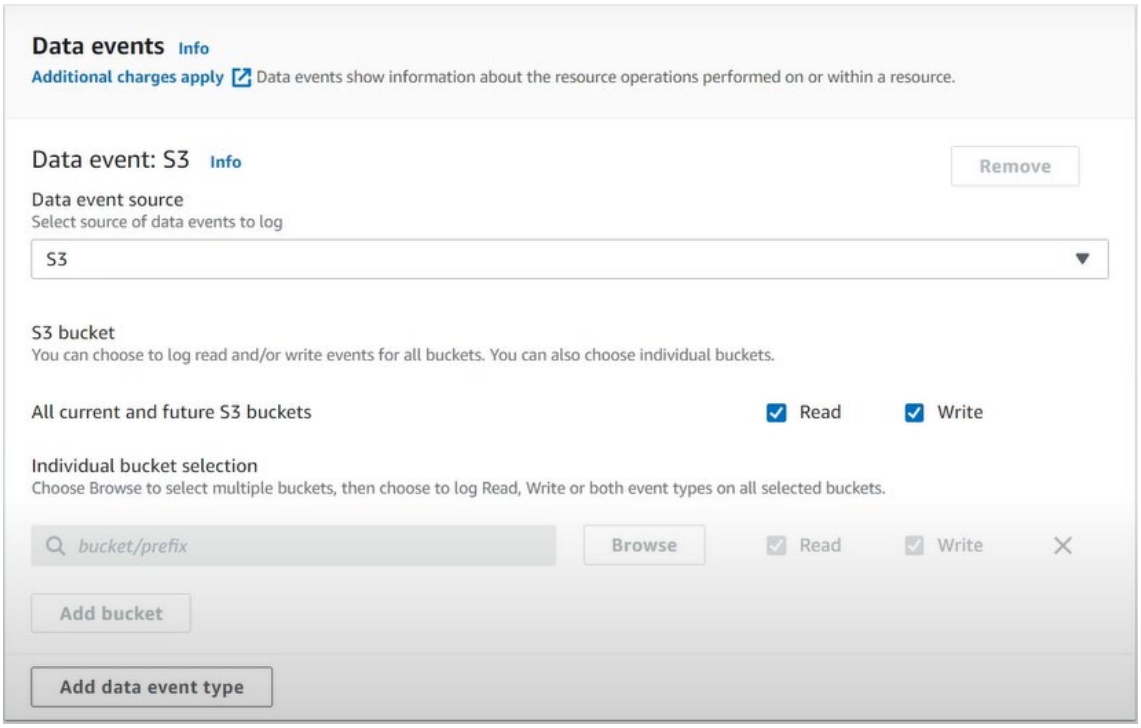




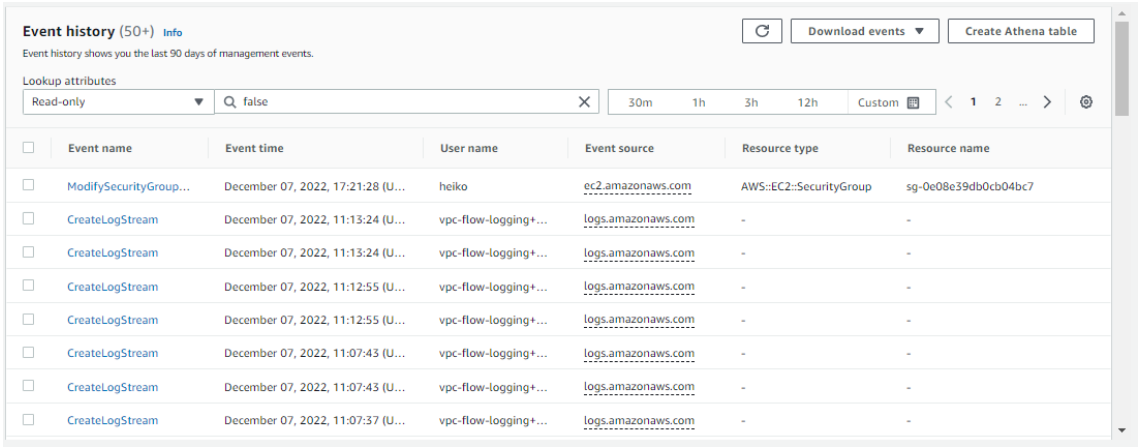


in the next section we will choose what we want to monitor.

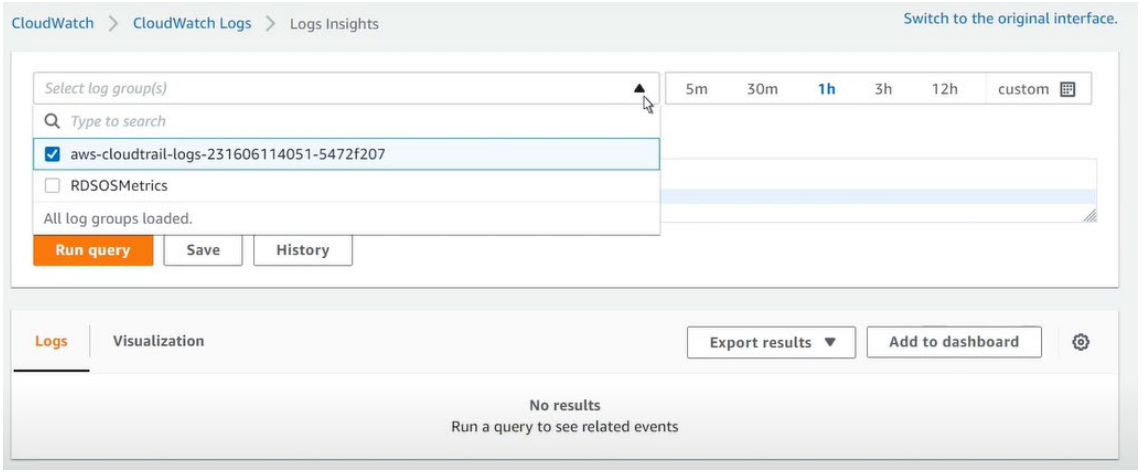




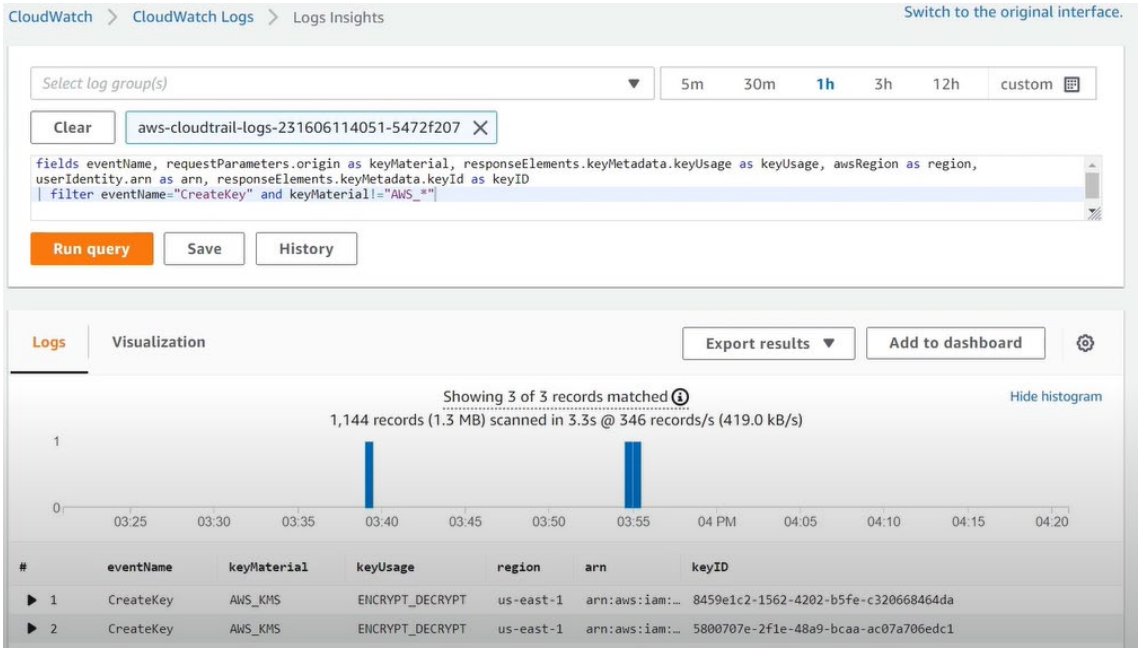
once created we can see after a while that we start to receive logs.



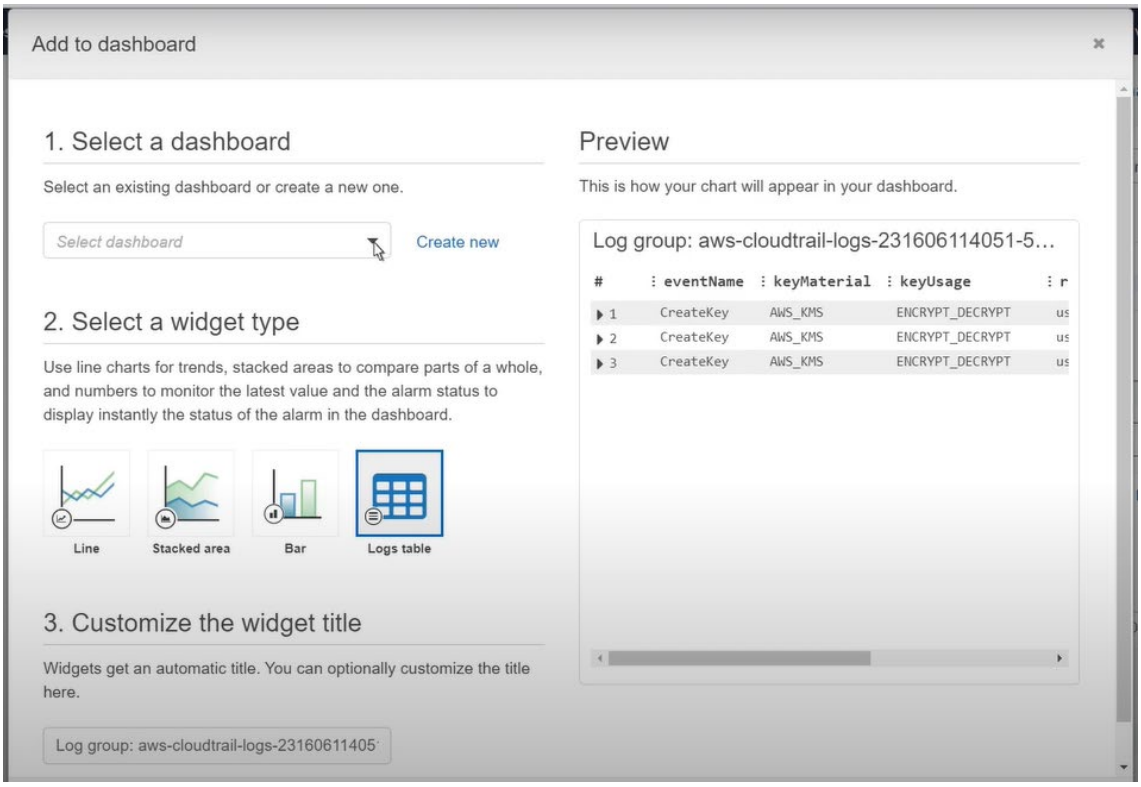
Next, we go to cloudwatch logs and in insights we select our already created cloudtrail.

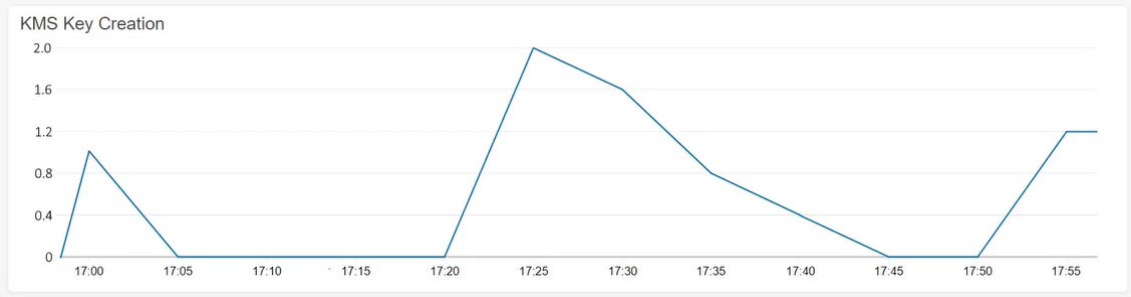


at this point we can now filter the logs we are receiving

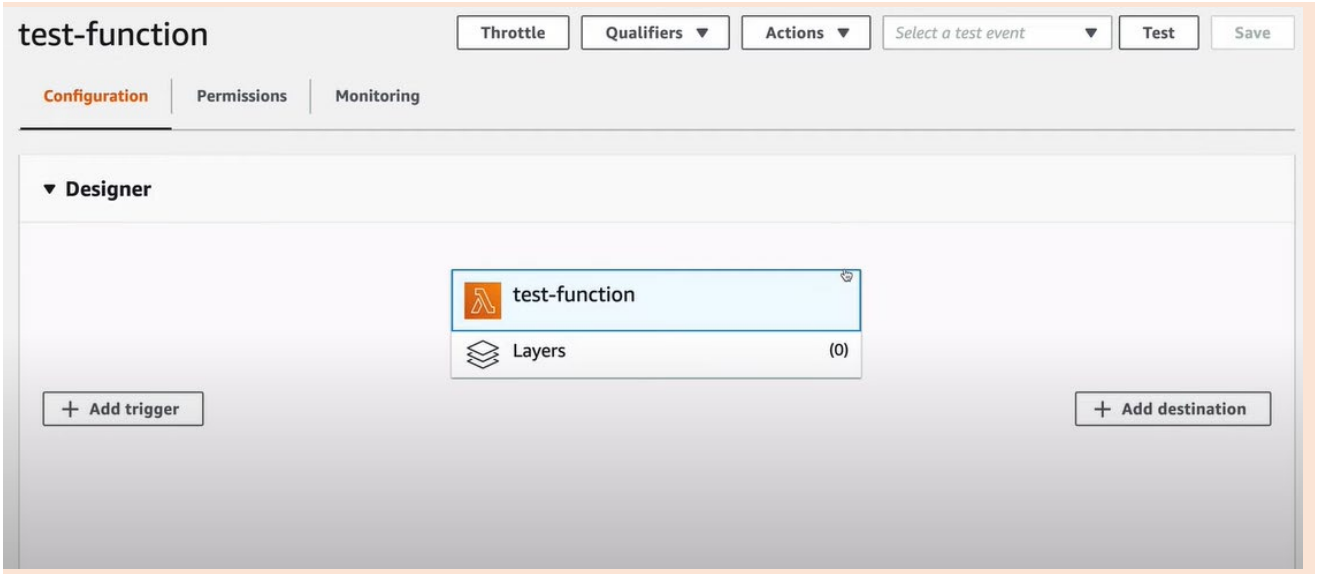


We can also add this new log filtering to a dashboard for better visualisation.


Add to dashboard
1. Select a dashboard
Select an existing dashboard or create a new one.
Select dashboard
Create new
2. Select a widget type
Use line charts for trends, stacked areas to compare parts of a whole, and numbers to monitor the latest value and the alarm status to display instantly the status of the alarm in the dashboard.
Line
Stacked area
Bar
Logs table
3. Customize the widget title
Widgets get an automatic title. You can optionally customize the title here.
Log group: aws-cloudtrail-logs-23160611405
Preview
This is how your chart will appear in your dashboard.
Log group: aws-cloudtrail-logs-231606114051-5...
eventName keyMaterial keyUsage r
1 CreateKey AWS_KMS ENCRYPT_DECRYPT us
2 CreateKey AWS_KMS ENCRYPT_DECRYPT us
3 CreateKey AWS_KMS ENCRYPT_DECRYPT us


KMS Key Creation
2.0
1.6
1.2
0.8
0.4
0
17:00
17:05
17:10
17:15
17:20
17:25
17:30
17:35
17:40
17:45
17:50
17:55

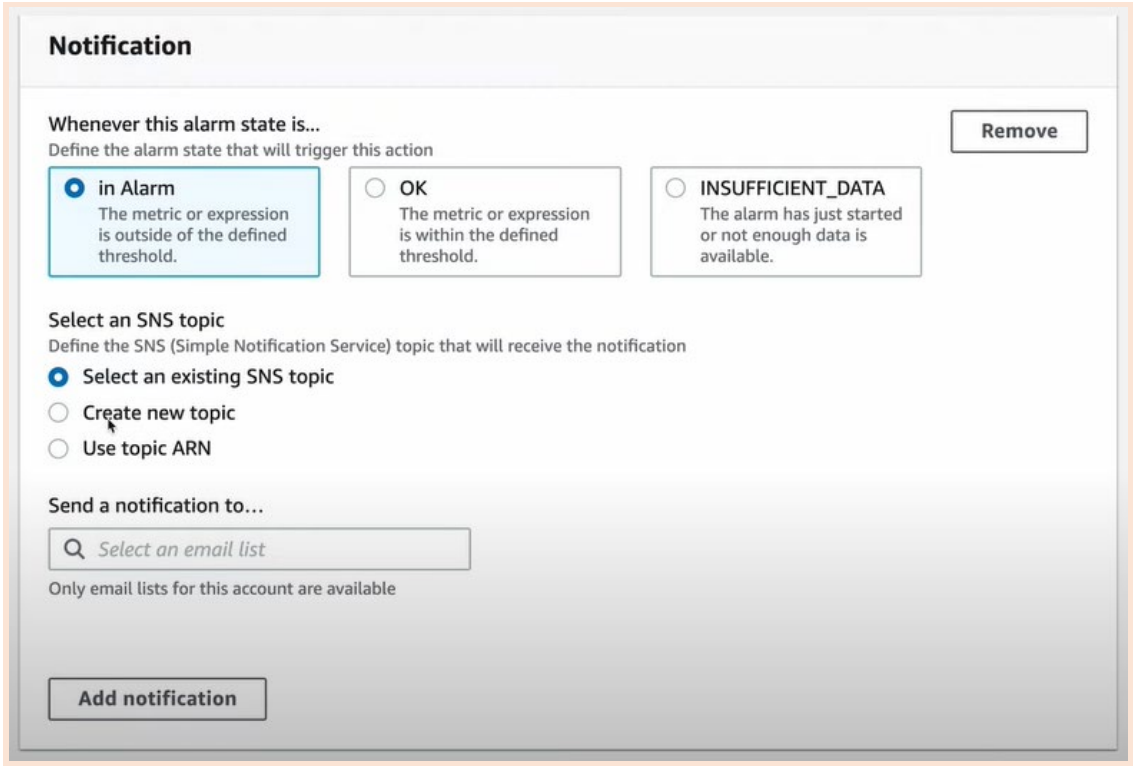
for landa and notifications first we will go to landa



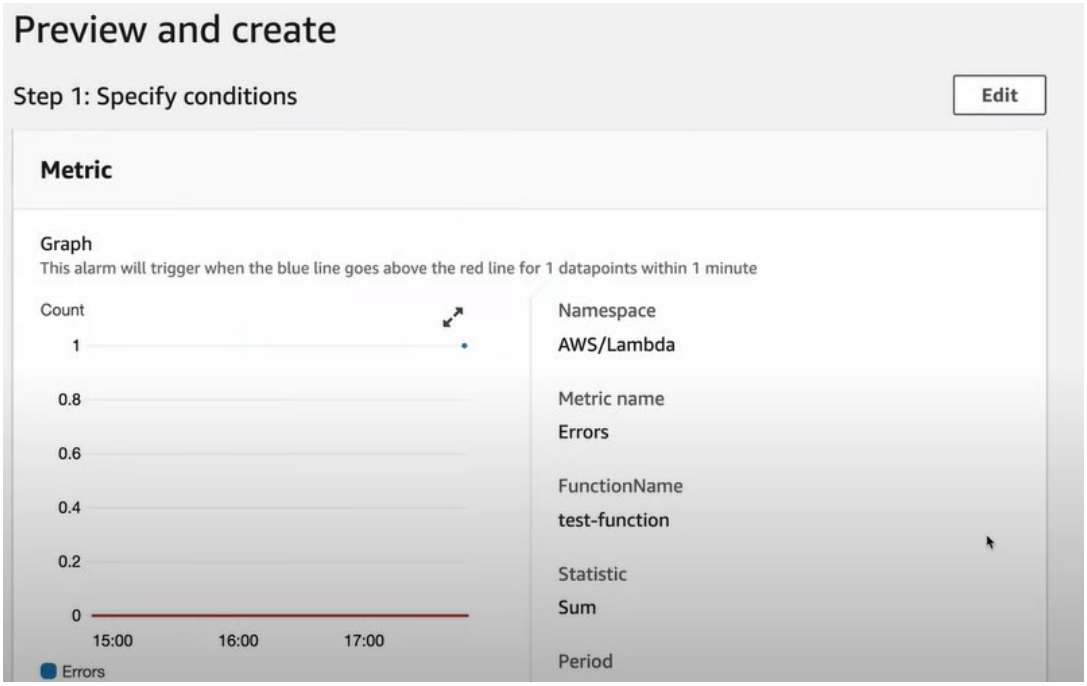
Here we will create a function that detects errors and gives an exception.

```
import json

def lambda_handler(event, context):

    if "error" in event:
        raise Exception(event["error"])

    # TODO implement
    return {
        'statusCode': 200,
        'body': json.dumps('Hello from Lambda!')
    }
```

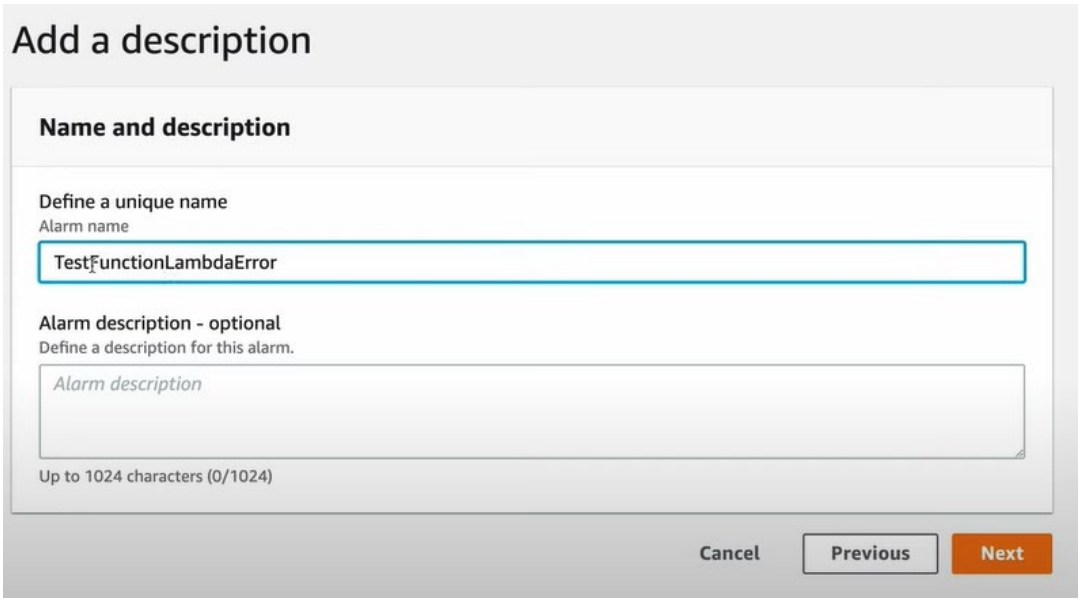
Then we will create an alarm notification and we will write the email to where the alert has to be sent.



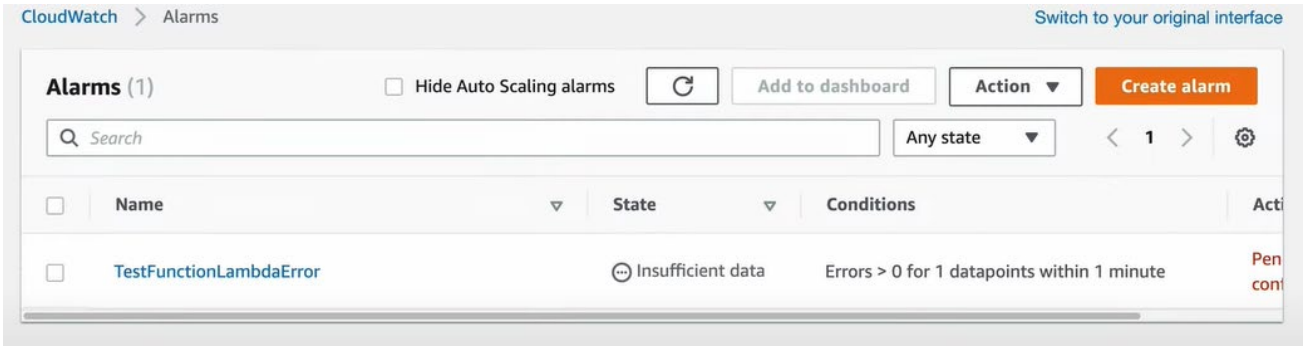
We create the notification with the specified conditions.

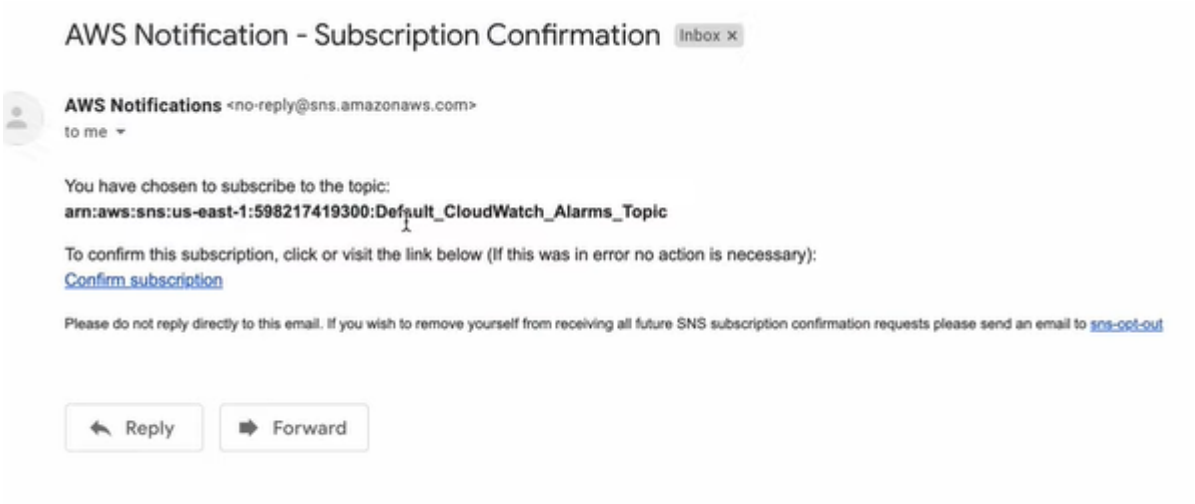


Add a description and create the alarm



As we can see, the alarm is created and I receive a notification by email





Later we execute the function lamda and we can see in the graph how it is registered.

Finally we see that in the mail comes a notification about the error and we already have the alerts configured with a lamda function.