Configuration Manual

MSc Research Project
Cloud Computing

Chetan Baviskar
Student ID: 21166374

School of Computing
National College of Ireland

Supervisor: Dr. Shivani Jaswal

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Chetan Rajendrakumar Baviskar		
Student ID:	21166374		
Programme:	Cloud Computing	**Year:**	2022
Module:	MSc Research Project		
Lecturer:			
Submission Due Date:	15-12-2022		
Project Title:	Configuration Manual		
Word Count:	1450	**Page Count:**	**13**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……

Date: 15-12-2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Chetan Baviskar
21166374

1 Overview

1.1 Purpose of this document

This section of configuration manual describes step-by-step procedure for implementing automation approach for encrypting S3 bucket using AWS Lambda. This manual will contain detailed explanation of each AWS service to achieve this automation. This research project includes AWS services like CloudFormation, Lambda, S3 Bucket, CloudTrail, CloudWatch and so on.

2 Requirement

- AWS Cloud Knowledge.
- Basic Python programming knowledge.
- AWS IAM role access to create user role and update polices if required.

This project is configured on AWS account. Steps for building automation infrastructure are mentioned in further sections below. To create new AWS account, need to visit below mentioned URL.

- AWS Account URL: https://aws.amazon.com

3 Procedure

3.1 Create AWS CloudTrail

First, in order to capture every detail of this process. User needs to enable CloudTrail.

CloudTrail allows user to capture each and every log during the operation in form of API gateway logs.

To enable AWS CloudTrail log, follow below steps:

- Go to CloudTrail from AWS console and click on ‘create a trail’.
- Provide a name to trail as shown in figure 1.

- Click apply to all multi region trail, in order to collects all API requests.

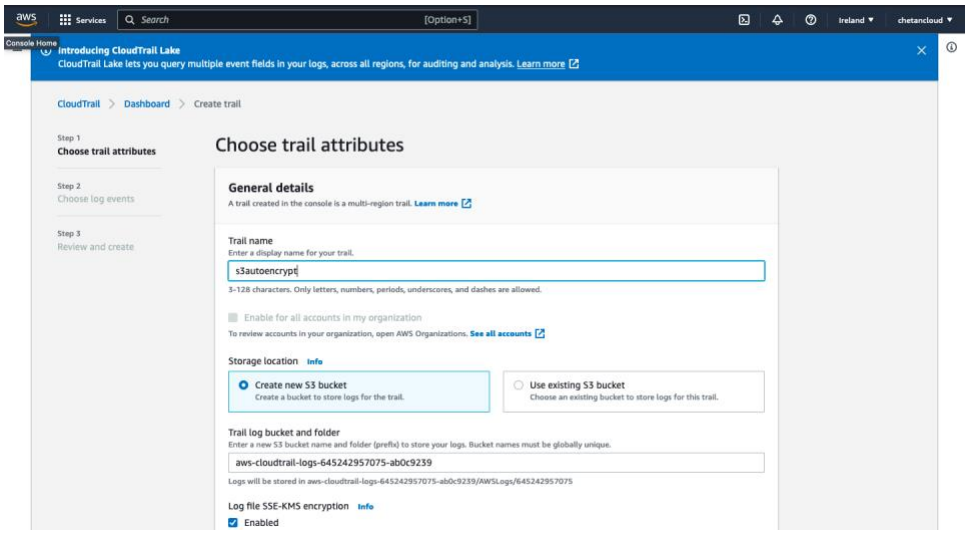

Figure 1: AWS CloudTrail

3.2 AWS CloudFormation

Now, once CloudTrail log option is enabled, Go to CloudFormation on AWS console and create a stack as shown in figure 2.

AWS CloudFormation works based on template file. User can provide all AWS resources configuration along with role to be created, event to be added and their dependencies are added in this template .yaml file so that user can deploy and set up them together in separate stack. Template file help to provision all required resources. This technique also known as 'Infrastructure as Code'.

- Upload the CloudFormation template file and click 'Next'

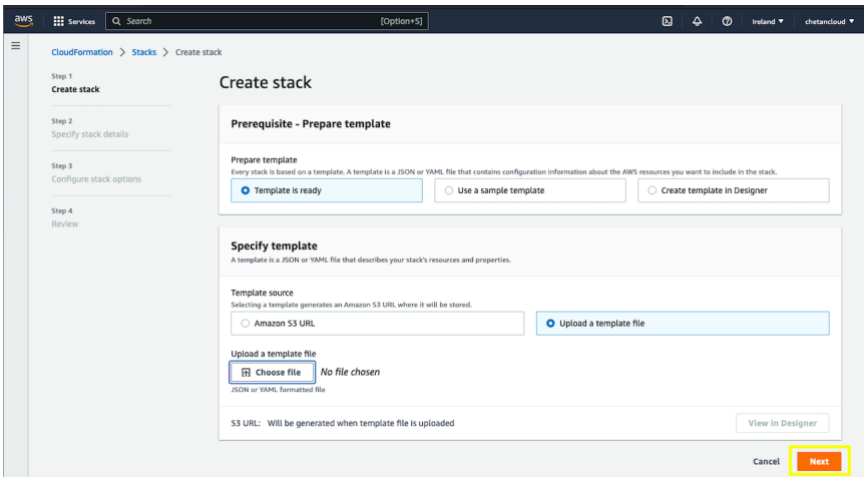

Figure 2: AWS CloudFormation template upload

- Provide name to CloudFormation stack as depicted in figure 3 and click 'Next'.

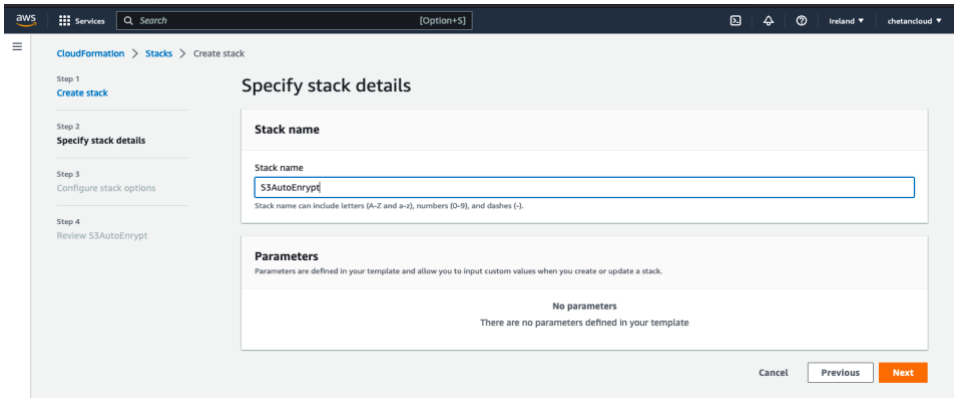

Figure 3: AWS CloudFormation stack

- Now, keep all default values as it is and mark it checked the acknowledge as illustrated in figure 4, to allow CloudFormation to create role if required.

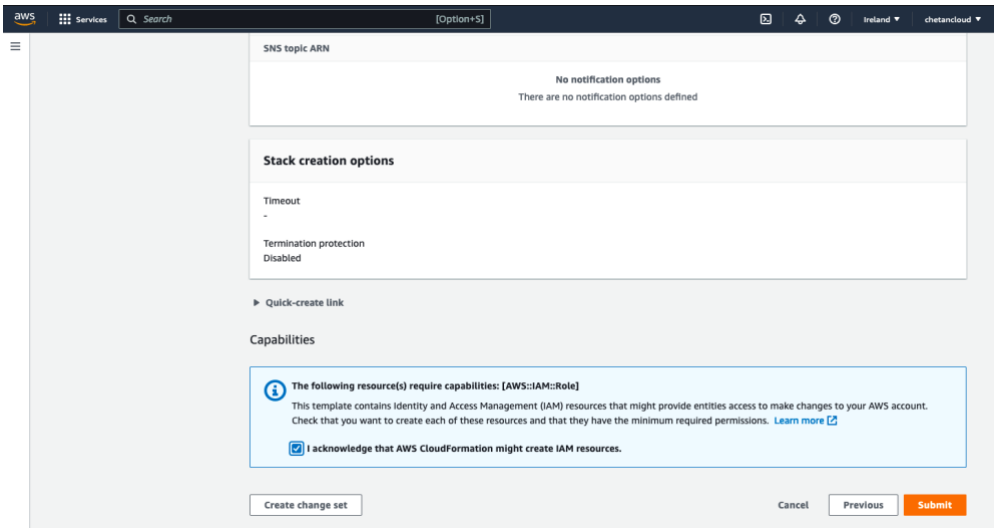

Figure 4: AWS CloudFormation submission

- Then click on ‘Submit’ and wait for stack deployment to finish.

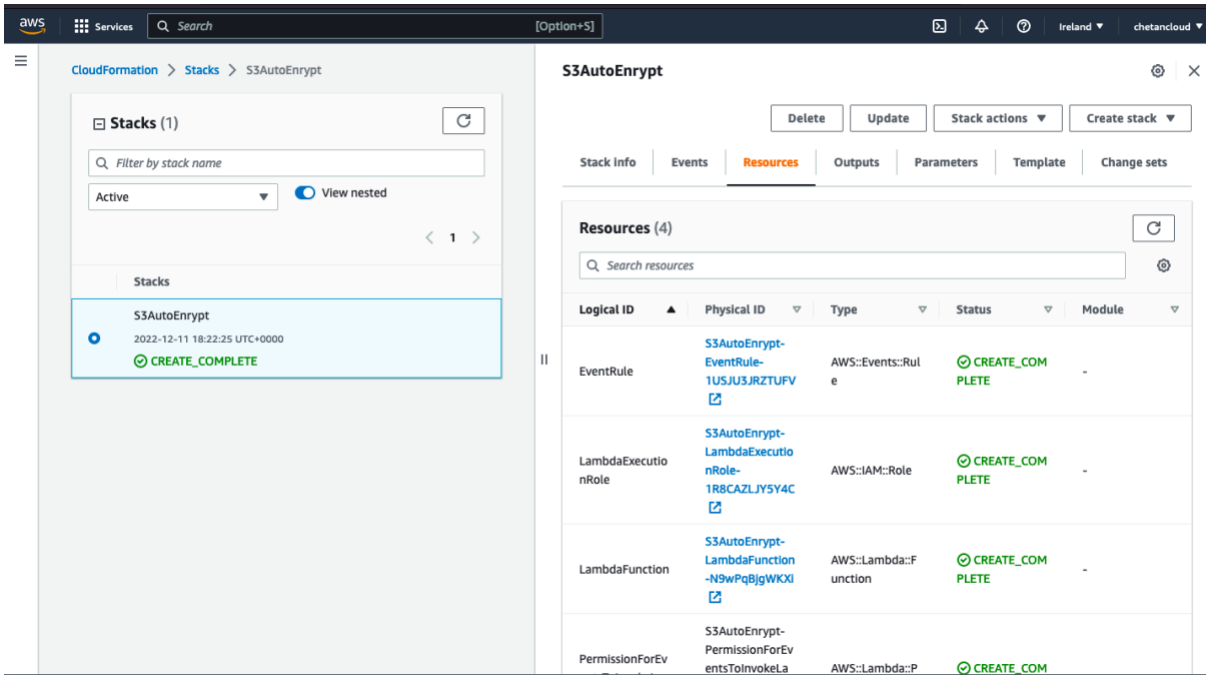

Figure 5: CloudFormation Resource Status.

- As per figure 5, In the Resource tab user can monitor status of each resource like EventRule, ExecutionRole, LambdaFuncation and PermissionForEventsToInvokeLambda if status is 'CRETE_COMPLETE' means deployment is successful.

- Once template deployment is successful. As figure 6, user can check template on CloudFormation where Lambda function written using python programming to encrypt S3 bucket automatically as soon as it is created by user.

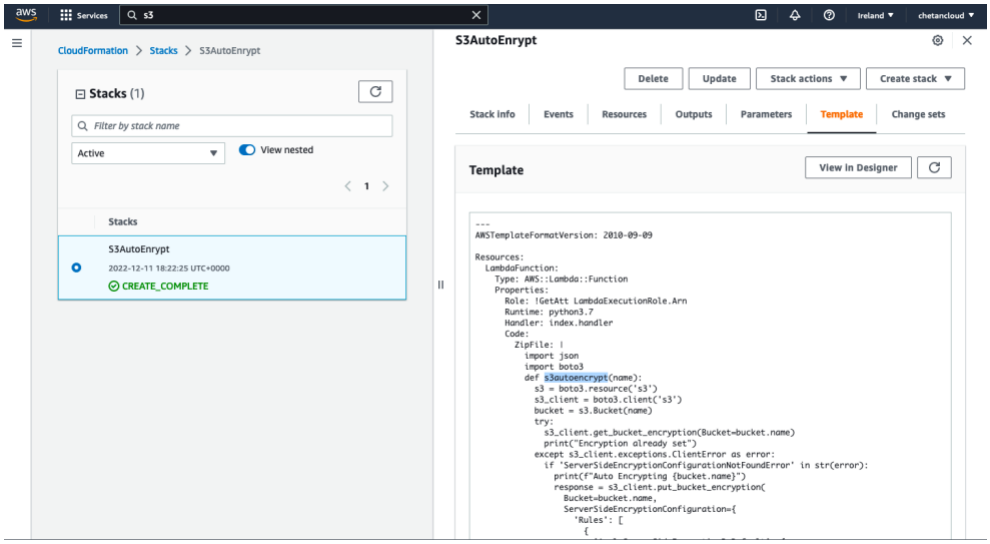

Figure 6: AWS Lambda function and other resources on CloudFormation template

3.3 Create AWS S3 Bucket.

- Now go to S3 bucket from AWS console to create S3 bucket.
- As presented in figure 7, provide bucket name and keep all other details unchanged.

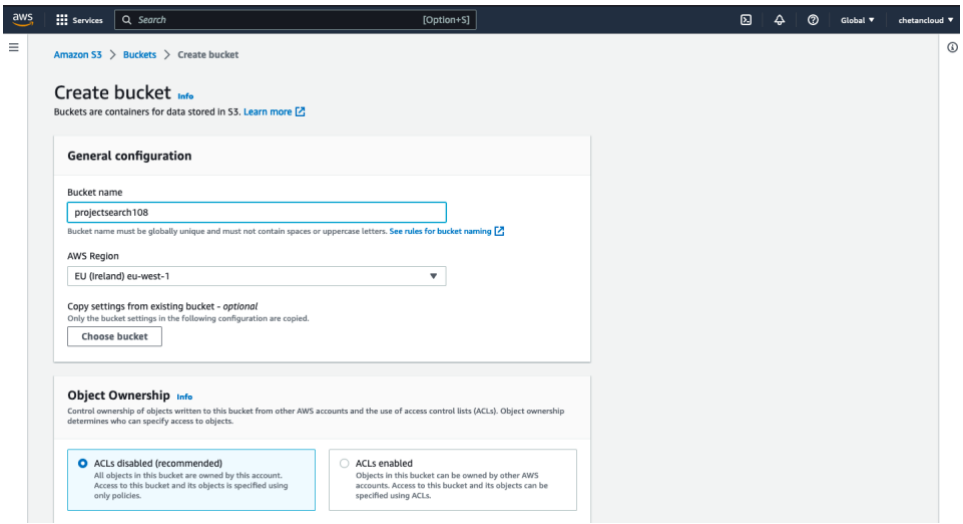

Figure 7: S3 Bucket Creation.

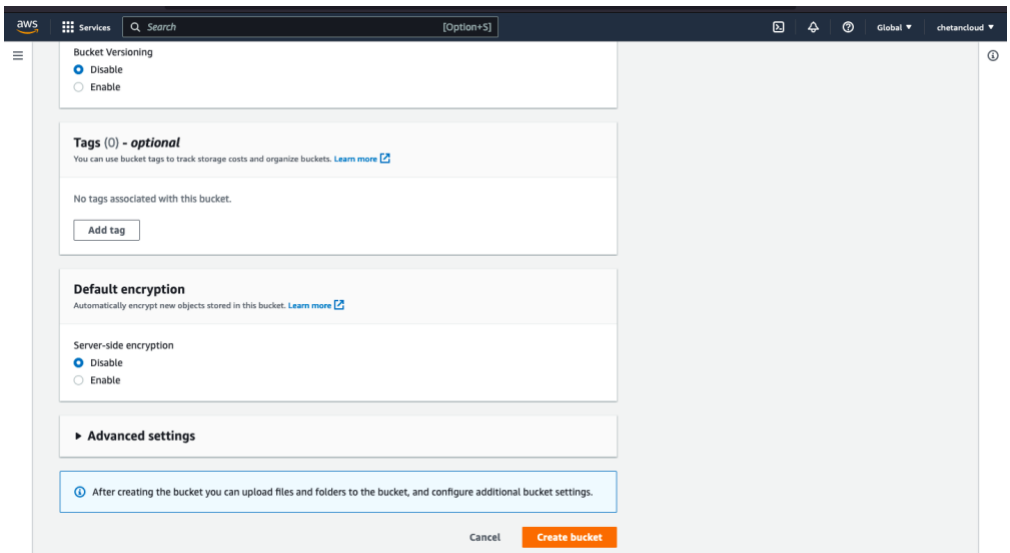

Figure 8: Encryption disable while creating bucket

- As illustrated in figure 8, maintain default encryption as disable so that bucket encryption can be happen through AWS Lambda automatically. Now click 'Create bucket'.
- Now, once S3 bucket is created. Head over to CloudFormation console and go to Resource tab click on Lambda function as shown in figure 9.

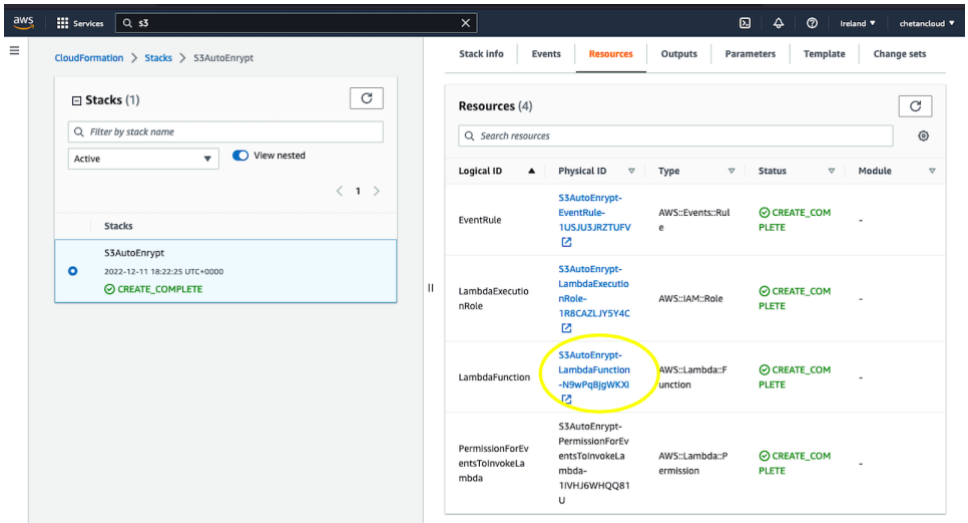

Figure 9: Lambda Function link on CloudFormation stack

- As presented in figure 10, it is clearly visible that EventBridge (CloudWatch Events) is triggering our Lambda function 'S3AutoEncrypt'.
- It means that, S3 bucket 'projectreseach108' creation event captured by CloudWatch events that leads to Lambda function 'S3AutoEncrypt' to run.

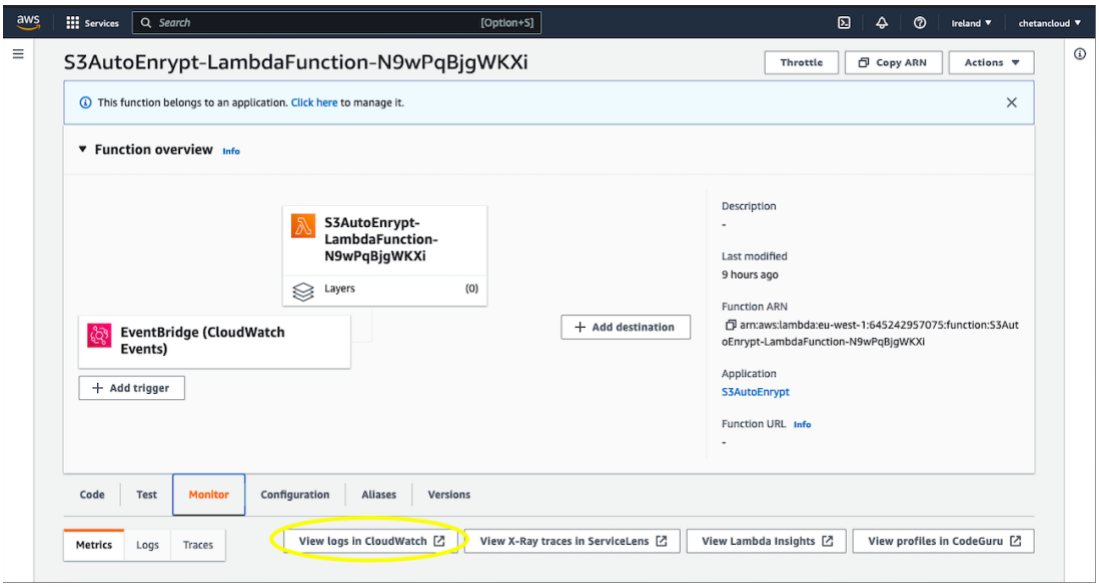

Figure 10: EventBridge (CloudWatch Events) cause Lambda Function to run.

- To gain more information about Lambda function and to check encryption validation. User need to click on 'view logs in CloudWatch' as depicted in figure 10.
- As illustrated in figure 11, user can scroll down and see logs that bucket name 'projectresearch108' is encrypted automatically.
- As per CloudFormation template, method for lambda function is added to trigger lambda function and provide becket name through API request.

- Also, user can see response logs where status code for http is 200 which describe as a successful response. Hence, objective of this research work to encrypt S3 bucket automatically using AWS Lambda function is achieved.

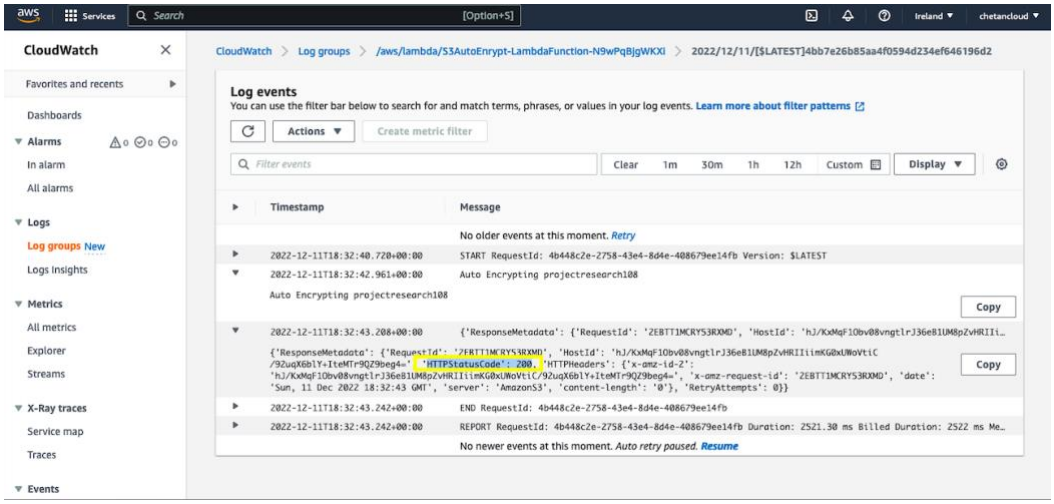

Figure 11: Encryption result of S3 bucket.

- Now, Bucket ‘projectreseach108’ is encrypted. On top this, user can even receive notification of S3 object insertion or deletion over mail using Simple Notification Service of AWS.

3.4 S3 SNS (Simple Notification Service) Notification

- Go to Simple Notification Service (SNS) from AWS console.
- Click on ‘Next Step’ this will lead to create topic page as depicted in figure 12.

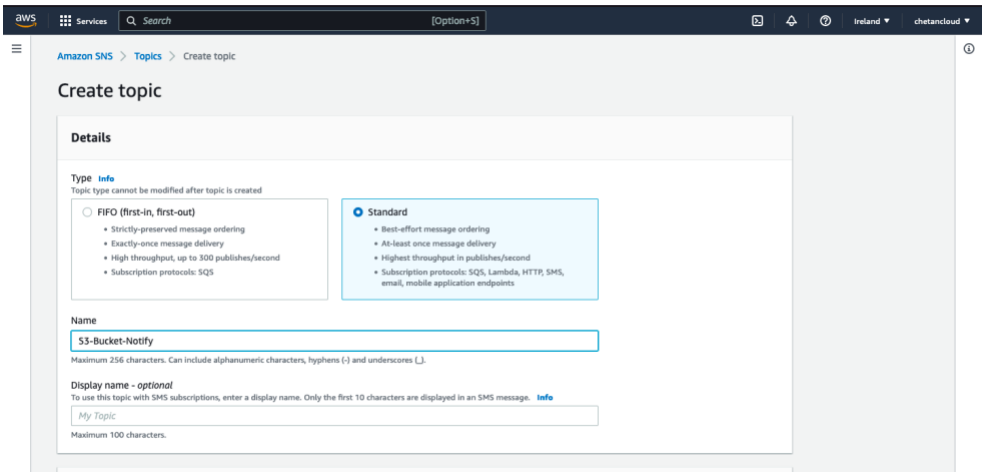

Figure 12: AWS SNS Notification

- User can provide topic name and scroll down to click on ‘Create topic’
- Once topic is created then user need to create subscription to avail email notification service. Therefore, click on ‘Create subscription’ as shown in figure 13.

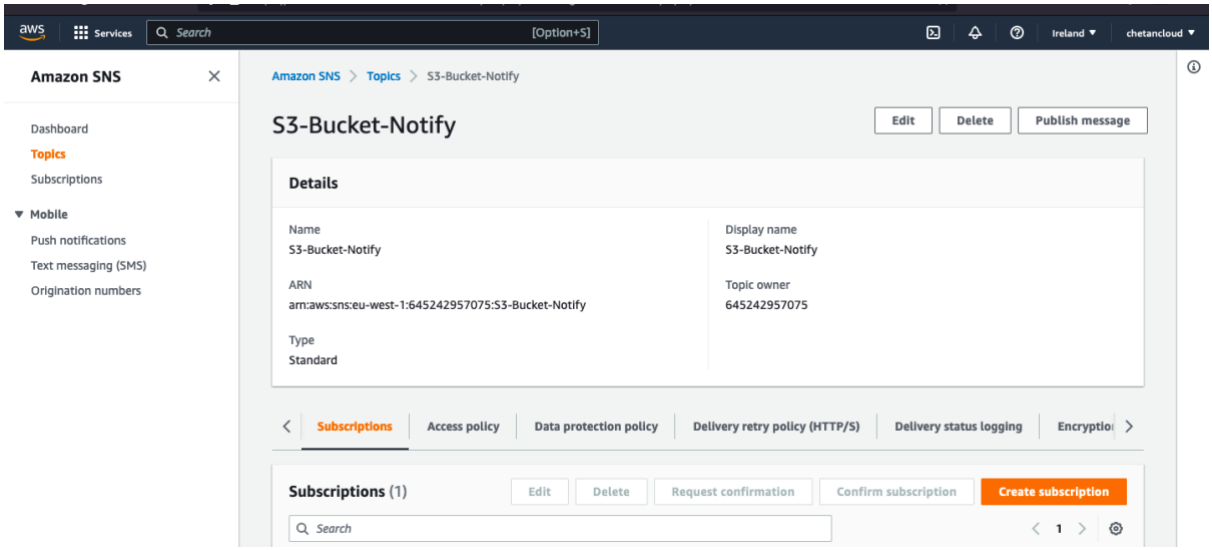

Figure 13: Create subscription to topic

- As illustrated in the figure 14, select protocol as 'Email' from drop-down menu.
- User needs to provide email address on which notification can be sent from AWS and then scroll down to click 'Create subscription'.

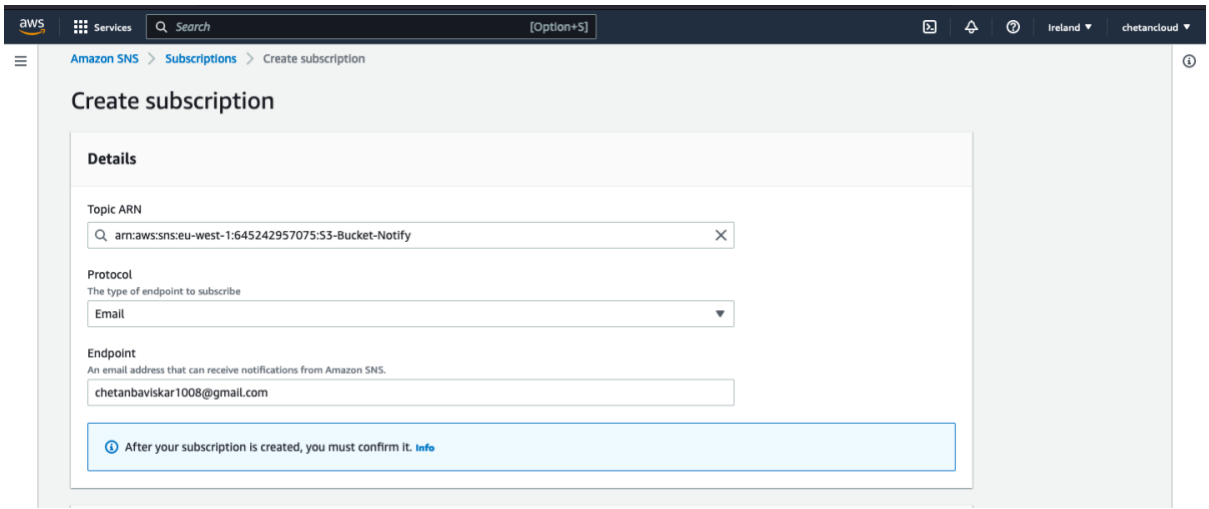

Figure 14: Provide email address to receive notification.

- Once, subscription is created, AWS will send subscription confirmation email to email address provided.

- As shown in figure 15, user need to provide confirmation by clicking 'confirm subscription'

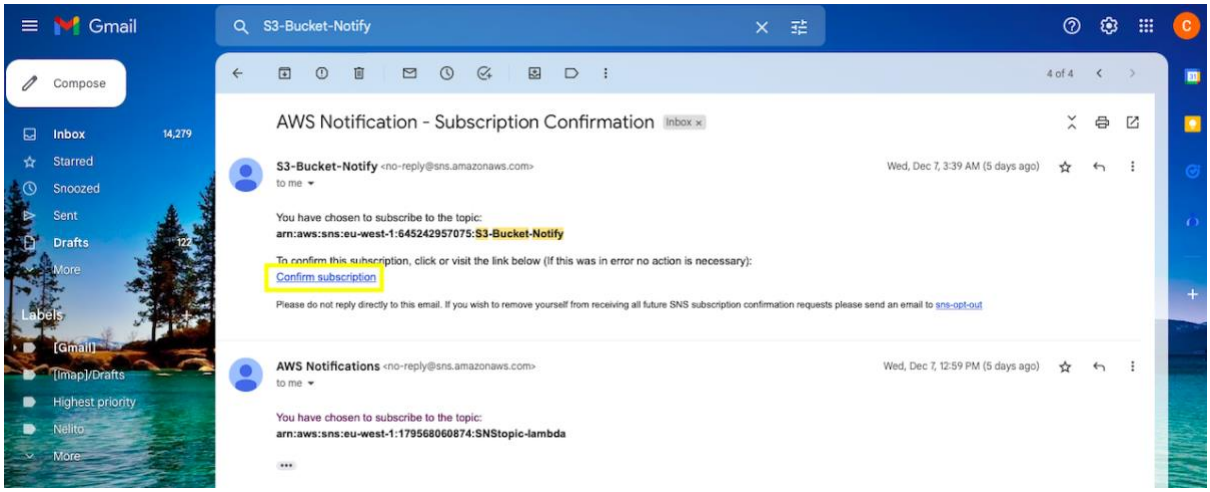

Figure 15: Subscription Confirmation

- Now, here in this research work another lambda function 'SNS_Lambda' is used to avail SNS service on particular S3 bucket. Need to create role for this lambda function.
- Go to IAM (Identity and Access Management) to create role named as 'SNS_Role' as shown in figure 16.

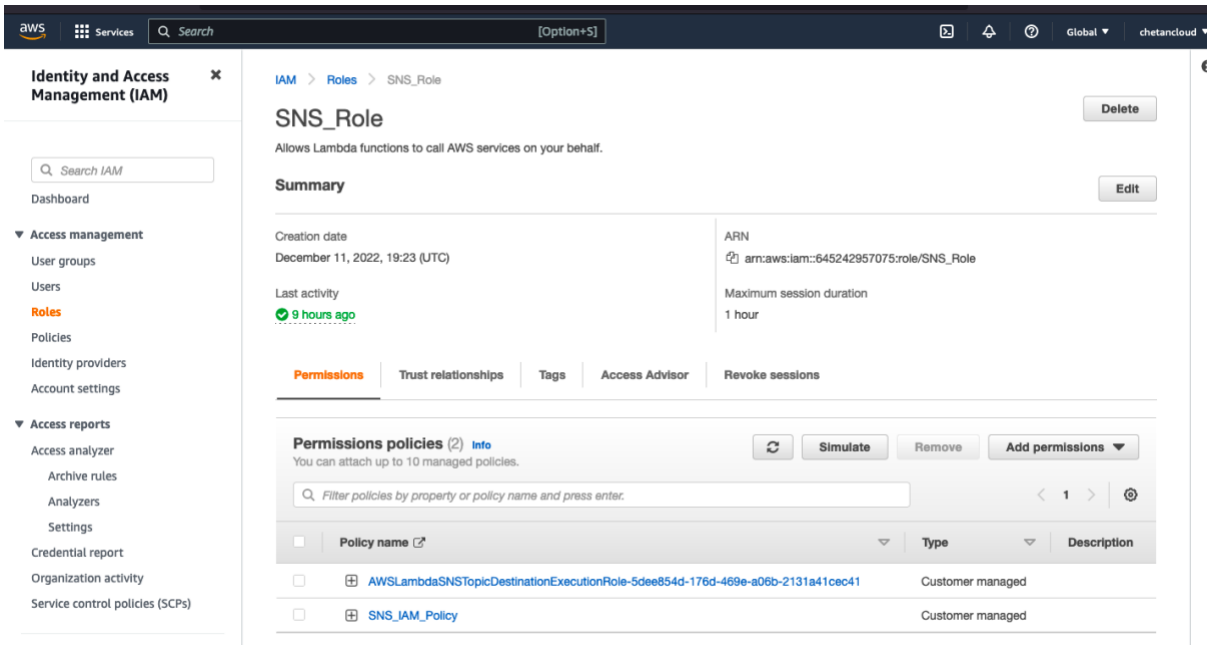

Figure 16: IAM Role creation for SNS Lambda function.

- Go to AWS Lambda from AWS console and create 'SNS_Lambda' with selecting SNS_Role. Lambda function code is attached into code artifact submission.

- Once Lambda is created. User can go to lambda and click on 'Add trigger' to use AWS resource as shown in figure 17. In this research work S3 bucket used hence select S3 bucket from drop-down menu.

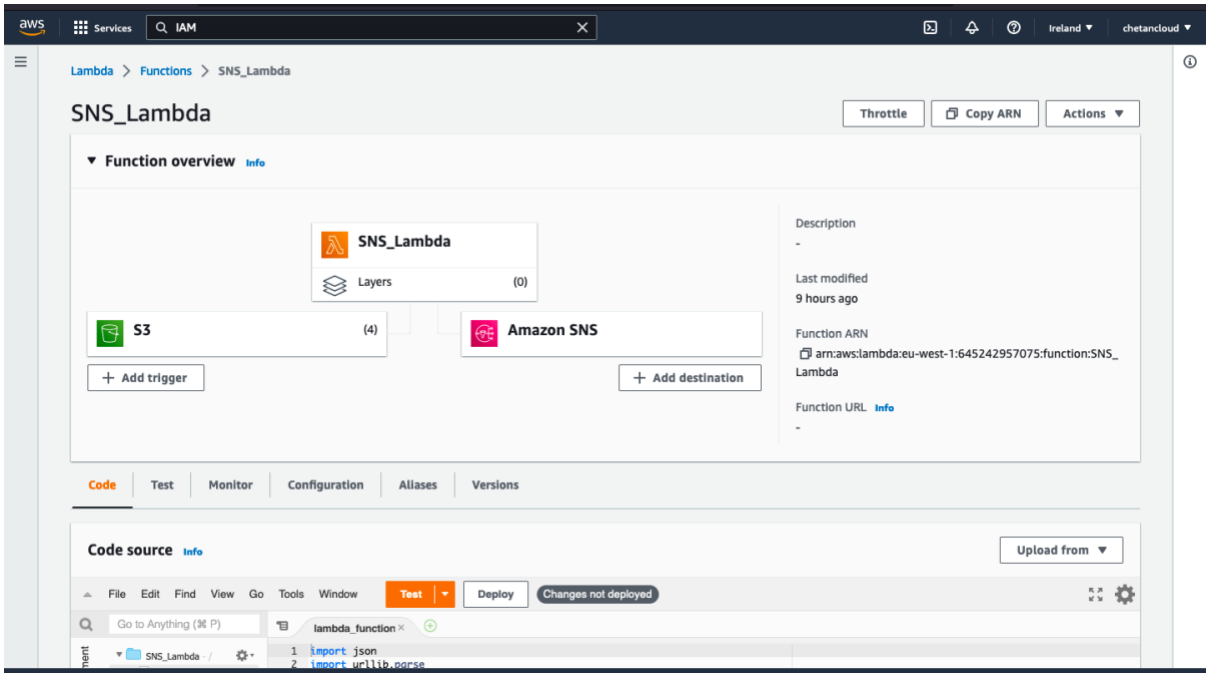

Figure 17: Lambda function for SNS service.

- Once S3 bucket is selected, user can choose particular bucket and event type as per shown in figure 18. Now, scroll down and click 'Add'.
- Here, in this research project Create and Delete event are used so that any object from S3 bucket created or deleted user will receive notification of the same.

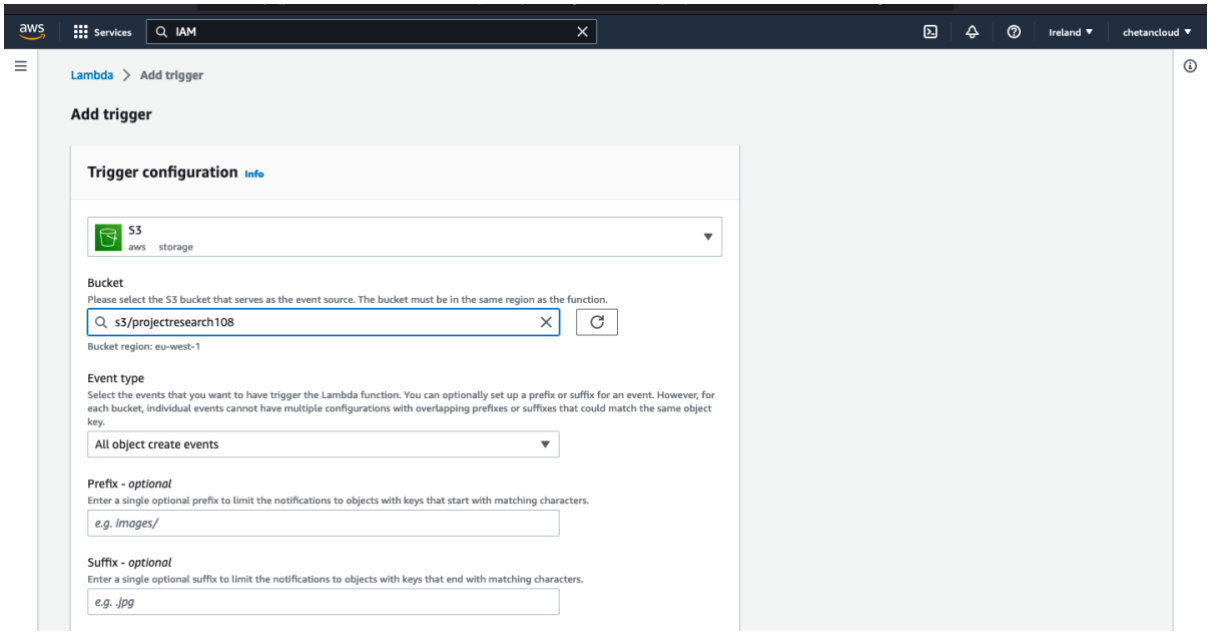

Figure 18: Adding object create event trigger for SNS service.

- Now, user can upload any object on mentioned bucket ‘projectresearch108’ to check SNS service working or not.
- As shown in figure 19, email notification of topic ‘S3-Bucket-Notify’ that object creation on S3 bucket ‘projectreseach108’ that means SNS service working successfully.

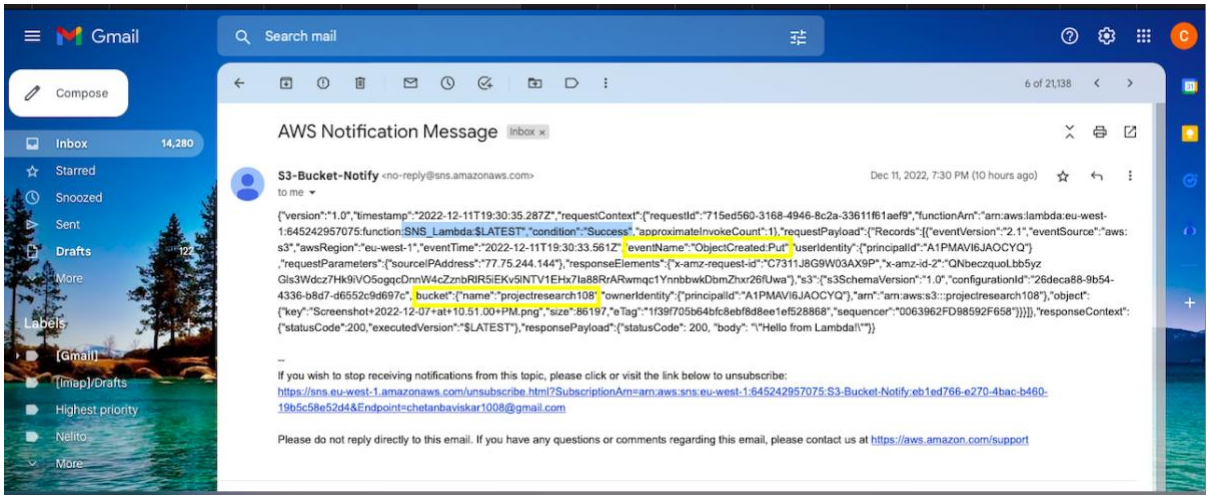

Figure 19: SNS topic S3-Bucket-Notify email notification to user

Therefore, even if user forgot to encrypt S3 bucket, using this approach bucket can secured as lambda will trigger immediately if S3 bucket is created and provide encryption to secure the bucket.

In this way, AWS Cloud based automated encryption approach and alert notification of S3 bucket using lambda is configured and implemented to protect not only security of S3 bucket but also privacy of S3 objects in the bucket.

References

CloudFormation, A. (2022) AWS CloudFormation Features, Amazon Web Services, Inc. Available at: https://aws.amazon.com/cloudformation/features/.

CloudTrail, A. (2022) AWS CloudTrail Features - Amazon Web Services, Amazon Web Services, Inc. Available at: https://aws.amazon.com/cloudtrail/features/.

SNS, A. (2022) Amazon Simple Notification Service (SNS) Features | Messaging Service | AWS, Amazon Web Services, Inc. Available at: https://aws.amazon.com/sns/features/.