Cloud based automated encryption approach to prevent S3 bucket leakage using AWS Lambda

MSc Research Project
MSc in 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:	MSc in Cloud Computing	**Year:**	2022
Module:	MSc Research Project		
Supervisor:	Dr. Shivani Jaswal		
Submission Due Date:	15-12-2022		
Project Title:	Cloud based automated encryption approach to prevent S3 bucket leakage using AWS Lambda		

WordCount:5885 **Page Count:**22

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 Program Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	

Cloud based automated encryption approach to prevent S3 bucket leakage using AWS Lambda

Chetan Baviskar

21166374

Abstract

AWS is a target for cybercriminals throughout the world, much like other public cloud services. One of the most important concerns in cloud computing is security and privacy. The AWS S3 (Simple Storage Service) bucket has been the subject of investigation in terms of data security and privacy issues. This report presents, the quickest method to reduce the risk of data loss and data concealment on the S3 bucket. The technique used in this report will make sure privacy and integrity of the data are maintained in S3 bucket. However public clouds like Amazon AWS, Google Cloud Platform, and Microsoft Azure are frequently noticed as unreliable. In general, users apply web-based dashboards and REST interfaces to upload and download data from S3 buckets. Particularly when a user outsources their information to public cloud platform, they typically miss control over the data to a local storage system. As size and complexity increase, handling and controlling admission becomes more challenging. This frequently happens when the access policies for an S3 buckets are incorrect, thus making data vulnerable to privacy attacks. The goal of this report is to find a way to protect the data in the bucket and to discuss a method for automating S3 bucket encryption to enhance data privacy in cloud platforms. Privacy is a crucial topic for cloud computing and needs to be seriously considered in terms of user belief and legal acceptance. This work offers a few data security methods that can be applied in serverless computing to protect S3 bucket.

1. Introduction

The cloud is a computing infrastructure made up of hardware, software, database and its associated operations that can be used as a service. Due to various benefits like flexibility to scale and be cost-effective, they are widely employed in many remote workplaces. These distributed storage services are practical because they make use of APIs, which eliminates the obligations between system administrators and software developers to disagree when configuring important storage server machines. For example, as demonstrated in figure 1, user interact with S3 bucket using AWS APIs as per access management role defined in IAM (Identity Access Management) policies. Adopting cloud service provider AWS has many possibilities and advantages.

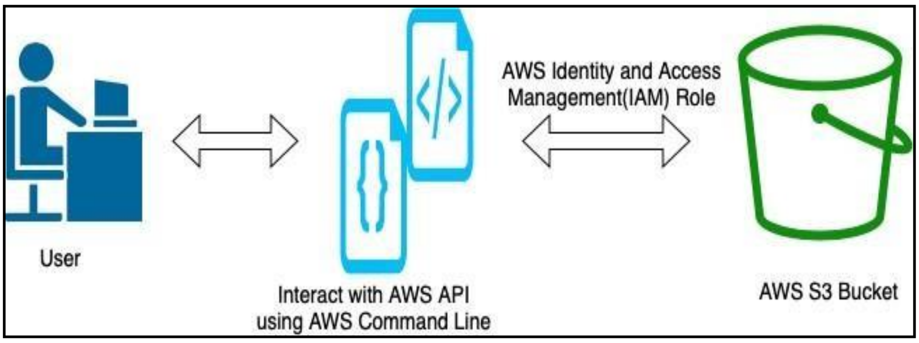


Figure 1: User Interaction with AWS S3 Bucket

AWS cloud services offer many benefits, including data compliance with regulations, governance, data measurability, data privacy law protection, adaptability and scalability, economic viability and affordability, a variety of scalable storage capacity, auto-scaling based upon demand, 24/7 data accessibility, and higher efficiency processing capability. Data Storage solutions, such Amazon S3 Bucket, Microsoft Azure Storage, Google Cloud Storage, and Digital Ocean Space, are fundamentally important operations of major cloud service suppliers. Amazon S3(Simple Storage Service) bucket is a most frequently used and famous. Here, users decide data containerization for S3 bucket, which may include unstructured objects like various diverse files, folders and crucial project information. Furthermore, object- and bucket-level access control policies are both applicable and applied on S3 bucket. Considering this, a lot of freshly created buckets and items are restricted as private, and only the bucket's administrator has permission to access them. Although these rules are followed naturally by default as a good practice still there have been several high-profile occasions where faulty application of permission privileges rules has raised security vulnerabilities. The cloud storage facilities offered by cloud computing environments and in-depth assessment of these cloud storage services in global contexts assist in determining the underlying cause of such configuration failures.

Similarly, author Kumar, A. et al (2012), believed introducing versioning as a fix remedy to provide additional extra security protection. Versioning option is used to keep track of data, recover lost data and recreate every single release of the data objects stored in an AWS S3 bucket. This makes it simple for consumers to rebuild from vulnerabilities and application failure. But yet, there are most common cause of malfunction is a misconfiguration of the Amazon S3 bucket's access control policy, which permits intruders to delete, modify and upload irregular files. Such occurrences can easily cause the leakage of sensitive data such as secret private login details, system dumps, and dump sites. They can also provide mass surveillance access to change already available information assets. Accordingly, from the viewpoint of a researcher observing Amazon S3 while considering this fact this report proposes a new technique and methods to mitigate the incidence of such a system failure and misconfigurations in the Amazon S3 bucket. These techniques have also been proven to be able to monitor security threats, discover publicly susceptible buckets and prevent vulnerable malfunctions and misconfigurations by auto encrypting S3 buckets and also alerts bucket owner for any malicious activity performed.

1.1 Motivation & Objective

This report study discussed the following research question.

> Making the AWS S3 bucket private, will it be breach proof? And is it possible to automate S3 bucket encryption while creating S3 bucket to ensure data privacy?

This report demonstrated positive resolutions to these challenges. In this report, this is first time ever discovered that possibility of hole in a private S3 bucket because of misconfiguration malfunction in the control of access and authorizations of Amazon S3 bucket. Hence, following points are some improvements to evade such challenges.

- Applying an encryption automatically on AWS S3 bucket on its creation itself making it even more secure because if the user forgot to apply encryption, still S3 bucket will be protected. To accomplish this technique, AWS Lambda is configured using python programming.
- Earlier in this report Honeytoken was introduced, as per author Bourke, D. and Grzelak, D., 2018, which sets up an alert alarm every time when deploying a new microservice or connecting a new laptop to a workstation system. However, sharing AWS access key over Git repository or any public platform is not recommended from AWS. As it creates security threat to user's AWS eco-system.
- Hence, AWS SNS (Simple Notification Service) is implemented to achieve same task in a more quickly while also safeguarding the AWS eco-system at a same time. Here, user will receive notification over registered email regarding object insertion, deletions and modification in S3 bucket.

1.2 Background

As per author Buyya et.al (2016) stated that early in 2015, Verizon investigated the shocking data leak instances that periodically happened during the recent few years. Furthermore, In the year 2013, it was reported that data damage or tampering also happened in cloud infrastructure as a result of incorrect configuration, hardware problems, power outages, and software defects. Additionally, the Cloud Security Alliance (CSA) reported that data privacy concerns and attacks are among the most widely discussed topics in the cloud age in 2013. Additionally, privacy concerns result from unwanted interference from internal and external competitors while utilizing cloud storage services like AWS S3. For instance, surveilling particular health information, it became visible that cloud services are completely insecure enough to maintain such confidential data. If there is no strong safety and privacy framework, using the cloud would just be difficult and problematic from cloud computing perspective to belive in cloud eco system to maintained users' information. However, cloud datacenters are useful to maintain cost efficiency and scalability of storage services. Particularly, when a user deploys a collection of data to a public cloud platform, they often miss control over the data compared to a local storage system. Author Ozer, M. et al. (2020), Elastic Cloud Computing (EC2) technology from Amazon Web Services first entered the corporate share market in 2006 to offer database storage and data processing services to specific companies. Following Amazon, other significant technological titans emerged, such as Google Cloud in 2008, Alibaba Cloud in 2009, Microsoft Azure in 2010, DigitalOcean in 2013 and IBM Cloud in 2014.

And therefore, even though there are many benefits of using cloud services, there are more privacy leakage issues and security risks when using such unreliable cloud settings. Different security methods have been developed by top cloud providers including Amazon, Microsoft Azure, and Google Cloud to address these concerns. In this report discusses about one such security solution that protects the widely utilized AWS S3 bucket cloud storage service.

2. Literature review

In this part of the report, research work from designated scholars of cloud computing is studied. A detailed discussion of previous research work on different topics such as, debate on cloud computing framework in terms of privacy preservation in AWS S3 bucket is reviewed. Different risk on cloud computing is analyzed. Process of restoration of security mechanism are inspected and DDOS attacks and privacy prevention using encryption techniques are evaluated.

2.1 Privacy preservation in AWS Cloud S3 Bucket

The author Mosca et al. (2014) demonstrated state-of-the-art security options for protecting external data and data privacy. The author had covered some of the major guidelines that provide cloud storage services with data privacy, integrity, and protection. The author claims that encryption is the simplest and most reliable method of providing data privacy before it is uploaded to a cloud platform. To provide appropriate data search and computing methodology, new encryption techniques that are clickable encryption and privacy preservation provided. This avoids the old method of encryption, which increases work challenges and ambiguity. The author has noted few common setup errors that might unintentionally grant unrestricted access to reading, writing, or listing S3 buckets, posing security risks to sensitive data. The term "Cloud Security Posture Management" refers to a useful cloud security technique that has recently been proposed to address these issues (CSPM).

2.2 Cloud Security Risk on Amazon

As per author, K.A. Torkura et al. (2021) original security systems are indeed essential to solve these unresolved security issues. The author presented CSBAuditor, a risk analysis solution that complies with the above standards by works thorough analytics and actively monitoring the whole Cloud Storage Broker (CSB) system for suspicious activity and illegal modifications in the CSB cloud accounts.

Having these benefits, there are also a lot of security-related issues and vulnerabilities. This article clearly explains some significant security risks when using AWS services as a practical example for regular cloud expressions. According to the authors Mosca et.al (2014) and Mukherjee, S., 2019, the following three key benefits become significant roadblocks and concerns when developing a trustworthy and safe cloud platform:

- Multiple users across the world share the same cloud infrastructure, which is also known as a multi-tenancy architecture.
- Massive amounts of data are stored and delivered via cloud systems, including concentrated high-performance compute (AWS S3 bucket).

Cloud platforms offered outsourcing of computing resources to reduce cost usage and operational costs.

However, Wang, et al. (2010) suggested that users need to strengthen the integrity and privacy of security services in a way to settle these issues over cloud infrastructure.

2.3 Preserving Privacy and DDoS Attacks in the Cloud environment

As per author Wang, et al. (2010), highlighted a few key ideas to protect privacy in cloud computing. There are different cloud providers worldwide, and they provide a range of offerings that can connect to the same network infrastructure and other clouds for support. It will be difficult to preserve data integrity and privacy in a circumstance like this if data is being sent across clouds. According to the author, academics focus more on cloud environments' security for cloud data storage services (AWS S3 buckets). In cloud contexts, they employ identification technologies to address these privacy issues.

Additionally, author presented anonymity technology that can be applied to enable the direct usage of anonymous data without a key and its restoration on cloud platforms by sending it to cloud service providers. This makes the separation of individual privacy in cloud settings more receptive and secure.

Similarly, as per author, Buyya, et al. (2017) has initiated progression in cloud platform DDoS attack mitigation techniques. In the history, Denial of Service (DOS) hackers would try to overwhelm one of the active servers in order to target a certain main server, causing the active server to become inaccessible owing to an excessive number of requests in the service queue. As illustrated in figure 2, Distributed DoS, often known as DDoS, is a newer sort of denial-of-service attack that targets a amount of workstations assaulting a single core service in a cloud environment. More than 20% of organizations worldwide, according to the author, have acknowledged at least one announced DDoS assault in their environment.Author Buyya, et al. (2017) Outlined, a crucial assault in cloud settings that affects resource management, the economy, and the quality of storage services (AWS S3) is the Distributed Denial of Service (DDoS) attack. Reduced contention at the operating system (OS) control level will help the DDoS attacker, which is a resource-focused attack, enhance the overall mitigation strategy functioning on the target server. Furthermore, it speeds up both treatment and recovery. According to the cyber expert attacks get more harmful if resource conflicts reach to their critical point.

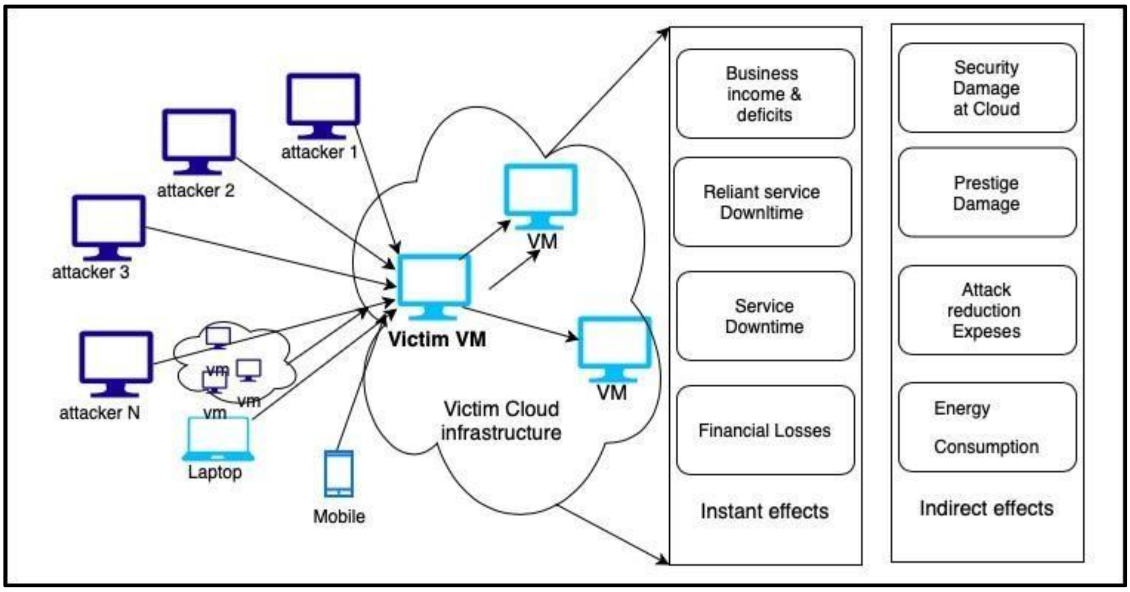


Figure 2: Distributed DoS assaults on the target host and their effects on the cloud business model, both directly and indirectly

2.4 Encrypting On-Premises Data Present on AWS S3 Bucket

As per author, Beer, K et al. (2014), presented encryption / encoding methods in an AWS white paper. According to the author, encryption requires the following three basic elements

- Encrypt Data objects.
- Algorithm to encrypt data Cryptographically.
- Algorithm to use Encryption key.

As per author Beer, K et al. (2014), Numerous modern programming languages include libraries containing data and algorithms that include inspection of security mechanisms, compliance requirements, and performance presentation tailored to applications. Key management infrastructure (KMI) in AWS is responsible for managing security key management. The database storage level that protects plaintext keys and the management layer that verifies the keys make up the two fundamental building blocks of KMI. Here in figure 3, demonstrates that how a secure net solution may be implemented in an Amazon S3 bucket to store encrypted data. AWS provides encryption methods that allow users to manage the security of their data because they are the only ones with access to the keys for each individual data item that is used. Users that use the on-premises AWS S3 bucket encryption have no method of giving AWS access to their user keys or any cleartext data. To ensure that anonymity is maintained, it is customary to shift all keys from KMI using Secure Socket Layer (SSL) or Secure Shell (SSH) if a user application is presented on S3 bucket or EC2 (Elastic Cloud Computing) instance. Also, as per author Sailakshmi, V., 2021, The necessity for confidentiality is to make sure that information is released to only those who have been granted access.

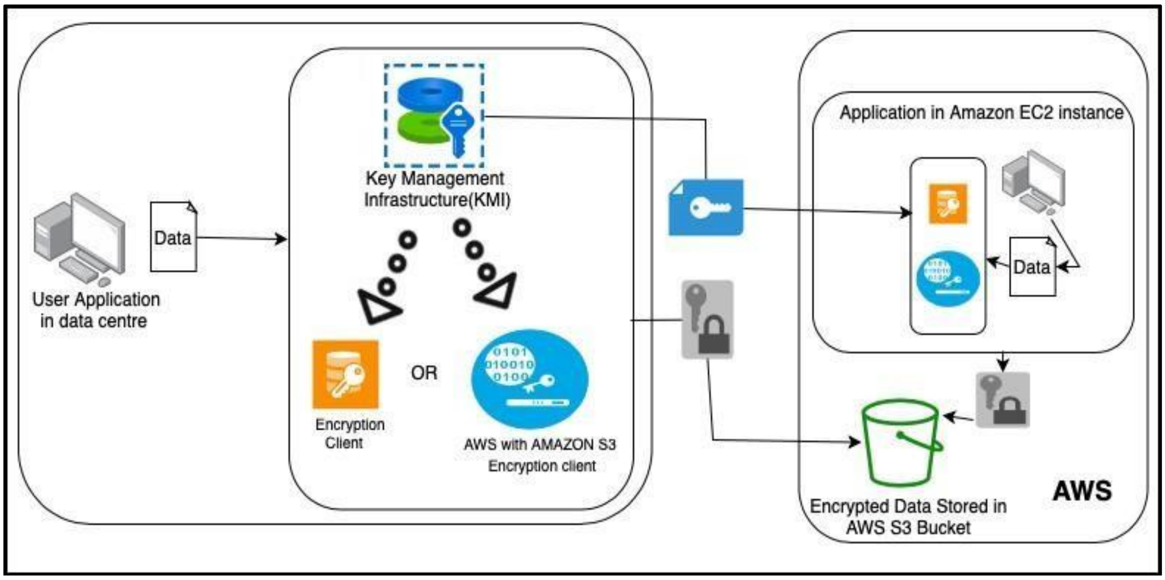


Figure 3: On-premises data encryption of Amazon S3 from Amazon EC2 infrastructure. Beer, K et al. (2014)

2.5 Levels of Privacy Protection

As per author I-Hsun Chuang, et al., (2011), each user composition has a privacy level set according to user requirements so that the best security solution may be used. Primarily, there are three different levels of privacy.

The author Hofman, D. (2017) stated that While the GDPR (General Data Protection Regulation) does not directly address the privacy issues raised by big data (besides, technology-neutral regulation), it still does give European users more confidence that intelligent data processors won't compromise their privacy. They are listed as follows:

- Privacy Grade 1(Speed): The essential meaning for this grade suggests that there should be no sensitive information in the data. To optimize the performance of cloud computing, users require weak or simple encryption.

- Privacy Grade 2(Hybrid): Here, some sensitive data must be included to meet Grade 2 privacy requirements. Users presume good cloud service performance at the same time.

- Privacy Grade 3(Security): In this category, data needs to be extremely sensitive. Here users sacrifice their right to keep their identities private in order to secure such a data.

2.6 Comparative study of the Literature research

References	Algorithm /Framework	Approach	Advantages	Limitation
[Continella et al. (2018)]	AWS S3 bucket Misconfiguration on prevention	Corruption in web setup and a way to forging server authorization requests are discussed.	This method can prevent S3 bucket outflow.	Since data encryption is not permitted, privacy of data can be endangered.
[Mosca et al. (2014)]	Analysis of privacy in cloud computing environment.	It depicts how the advantages of cloud computing become the justification for the violation of security privacy.	Emphasized on critical flaws in the security concept for cloud computing	The suggested solutions are generally costly.
[Wang, et al. (2010)]	Identity management Algorithm	Systems for managing identities are based on a variety of security and privacy features, ranging from simple one-factor authentication for passwords to complex multi-factor authentication.	Able to protect privacy by being anonymous.	Since data is being sent across clouds, maintaining data integrity and privacy will be difficult. will be a problematic to privacy and integrity of information.
[Buyya,et al.(2017)]	Algorithm of prevention of DDoS attacks	Sending a lot of queries to the vulnerable server to the point that it become unavailable.	Effectively protect DDoS from several hosts virtual server.	This method is not economically efficient and overall energy use is more.

[Beer, K et al.(2014)]	Framework of Key management infrastructure	Key administration infrastructure controls and manages security key management in AWS (KMI)	Keeping privacy maintained and integrated is beneficial.	Loss of keys might result in security breaches.

3. Research Methodology

In this section, methodology approach and process behavior are demonstrated to gain an perfect resolution to restrict dataflow of delicate data information from AWS S3 bucket.

3.1 Leakage in Private S3 Bucket

This report states that even after keeping S3 buckets private, bucket names might still reveal sensitive information. The majority of security problems have arisen as a result of IAM (Identity Access Management) misconfigurations and user-deployed mistakes. The process of creating an AWS S3 bucket on Amazon Web Services is fairly straightforward and uncomplicated; it won't take more than a few seconds.

As per author Cable, J., (2021), The hackers can apply the scripts to access the contents from buckets and predict the name of an S3 bucket. The widely accepted naming convention used across organizations makes S3 buckets more susceptible to these types of threats. In this research, "S3Scanner" security tools are used to show how to secure S3 buckets from such attacks in order to prevent such a leakage of information caused by AWS S3 bucket naming regulations. It is found that S3 bucket naming conventions are widely used, uniform across bucket names, and unexpectedly anticipated. As per article, S3Scanner
- Scan For Open S3 Buckets and Dump - GeeksforGeeks, (2022), S3Scanner is an automated cyber security tool designed to scan and transfer data from the target domain's accessible buckets. The GitHub platform hosts this tool, which was created in the Python programming language.

3.2 Employing Encryption on AWS S3 Buckets

Encoding is a method that encrypts the decoded data. It aids bucket owner to safeguard sensitive private data and defend information from being used in any vicious operation. Here there are 2 types of encryption methods are discussed, in general, user has the choice to have the encrypt bucket option when establishing an AWS. S3 bucket from the AWS dashboard, and each bucket owner should pick this alternative option to encrypt their data information in cloud computing platforms. Data safeguard relates to preserving user data both at rest or when it is saved inside S3 bucket and while in transit when it is transferred to and from Amazon S3 buckets. Information can be secure while transit mode with two types:

- Server-side encryption.
- Client -side encryption.

Server-side encryption take place when the program or service that collects the data it encrypts before transmitting it to its destination. Whenever Amazon S3 writes users information or data to disks in data storage, it encodes it at the object level and decrypts it for user at the time retrieving the information.

Client-side encryption, it is the procedure of remotely or locally encoding user data to preserve its security and confidentiality of system while it goes into an Amazon S3 bucket. The most significant thing to understand is that the Amazon S3 bucket service only accepts user data; it is not involved in encrypting or decrypting user information. In this study, server-side encryption is applied by Amazon S3 server-side encryption to encrypt data using AWS Lambda. It utilizes one of the most efficient block ciphers, 256-bit Advanced Encryption Standard (AES-256).

3.3 Proposed Approach

A quick recap of the research question is that, this report aiming for a mechanism that would allow users to create AWS S3 buckets without needed to worry about data encryption, and that would automatically encrypt the S3 buckets to ensure data privacy.

The adoption of the "AWS Lambda" service, which precisely suits the description and use-case for this report, has a direct impact on the question. To make this process automatic, the AWS service which commonly known for its ‘infra-as-code’ methodology “AWS CloudFormation” is used.

And, cherry on the top is that here user will also get notified about object insertion, deletion and modification in S3 bucket using service called SNS (Simple Notification service) of AWS.

4. Design Specification

This section describes the main design involved. The design goes hand-in-hand with main objective. As illustrated in figure 4, it has been clear that implementation focus on AWS cloud infrastructure having Lambda and CloudFormation plays vital role. It is important to choose the ideal solution by examining the different options at hand.

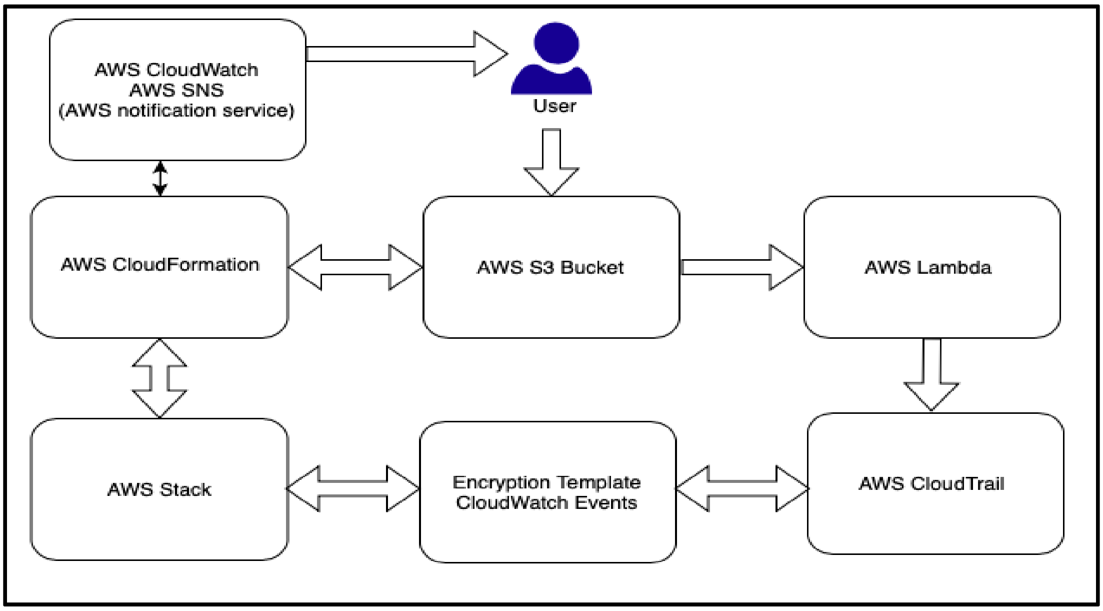


Figure 4: Dataflow diagram

AWS CloudFormation: The obvious choice is AWS CloudFormation. Not only because is AWS being well-known cloud provider, but CloudFormation is also ideal for this use-case and it also allows any third-party installations if needed. The versatility of Terraform from Hashicorp, which eliminates the constraint particular to clouds, makes it the second-best alternative. However, AWS CloudFormation still has the advantage because it can be used to work on more than 250+ services by itself from single source and template describing all requirements.

AWS Lambda: The serverless compute service AWS Lambda executes user's code in response to events and maintains the underlying compute resource. As per author Quiles, C. (2021), User can design their own backend services that runs at AWS scope, accomplishment and protection using AWS Lambda to insert traditional functionality to different AWS services.

AWS CloudTrail- By selecting Event history in the CloudTrail console, users may quickly view events. As per CloudTrail, A. (2022), Users may see, search, and download the last 90 days of activity in their AWS account using Event History. To store, evaluate, and react to changes in AWS resources, user can also set a CloudTrail trail. A trail is a setup that permits event delivery to a user-specified Amazon S3 bucket.

AWS SNS: Amazon SNS is used by Amazon CloudWatch to send email. As per SNS, A. (2022), Build and subscribe to an SNS topic initially. When a CloudWatch alarm is established, the user can include this SNS topic to deliver an email alert whenever the bucket's states updated.

4.1 Design Workflow

Serverless computing infrastructure presented by AWS Lambda is an event driven structure as per authors McGrath, G et al. (2017) and Krishnamoorthy (2022), using lambda, logic of the encryption functions gets executed in whenever, create s3 bucket event occurrences.

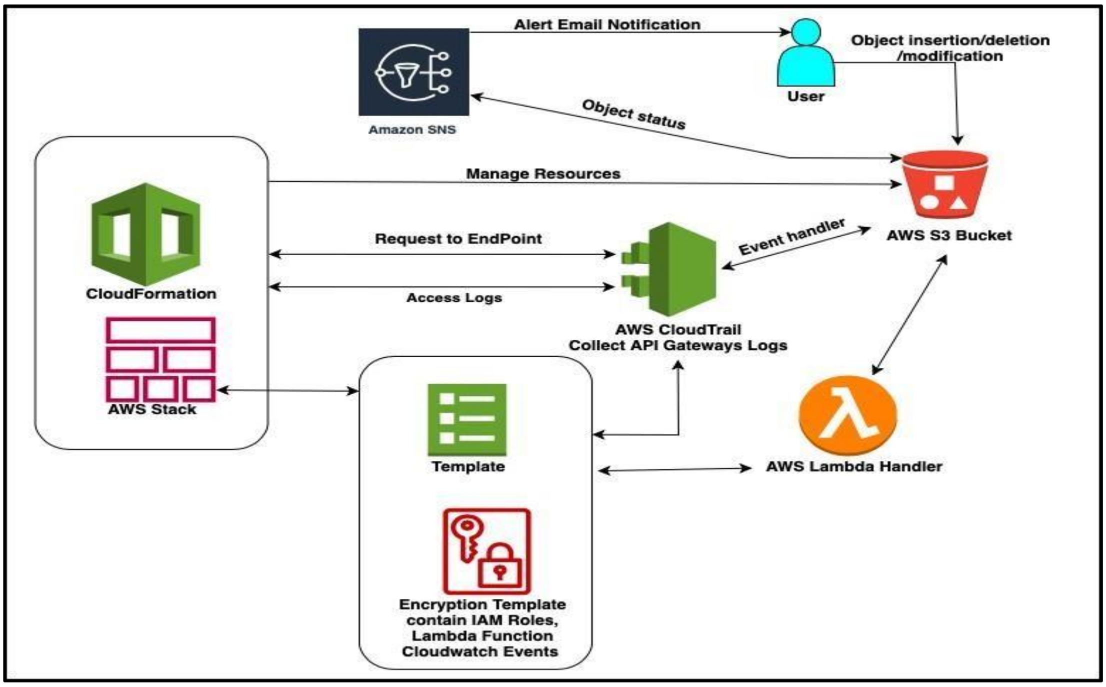


Figure 5: Automatic Encryption of AWS S3 Bucket using Lambda

As presented in figure 5, Lambda is event-driven and commonly known for its serverless computing platform offered by AWS. To permit encryption of S3 bucket automatically it launches encryption as it triggers Lambda function which is in template file coded using python programming language.

CloudFormation creates the stake with template. This template includes IAM roles , Lambda function and CloudWatch events. S3 bucket encryption code is placed inside this same template. In the same template, Lambda handler call encryption function, this handler allow user to pull out the bucket and encrypt it. While this process going on, S3 bucket also parallelly interacting with AWS SNS (Simple Notification Service) services to update status of object insertion/deletion/modification. As per subscribed topic to an SNS service, it sends an email notification to user when event modify the state of bucket.

5. Implementation

This section describes the implementation of AWS cloud services and process of configuring key components using CloudFormation. AWS Lambda function is written inside stack template of CloudFormation using python language. The main idea behind using CloudFormation is to include as many as cloud services as possible to facilitates developer and DevOps team exempted from any kind of resource provisioning besides some general configuration and authentication setup.

The general steps include:

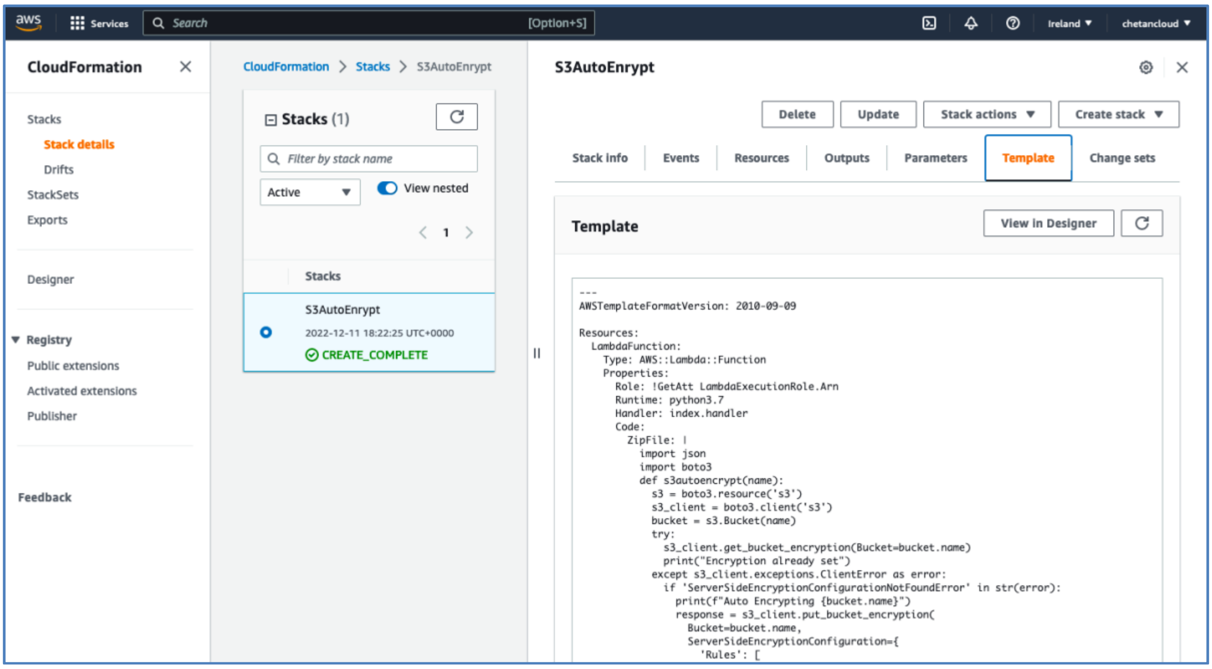
- Login to AWS account.
- Create CloudFormation stack template to configure AWS Lambda encryption function and other event rule, roles and permission level configurations.
- Upload that template file in CloudFormation stack and wait until finished deploying.
- Then, Create S3 bucket with SNS configuration.

6. Evaluation and Results

This section takes through overall implementation and evaluation part to better understand the working of each AWS resources and services used in this report. Each phase provides extensive inspection of the results to achieve main objective of encrypting S3 bucket automatically.

6.1 Template Design and upload to CloudFormation Stack

AWS CloudFormation streamlines the resource provisioning and help administration of AWS services. Metadata can be provided in template to give further statistics using JSON and YAML objects. To achieve objective of this research, as depicted in figure 6, S3AutoEncrypt template has been designed. Template can be modified as per requirement using parameters. These parameters help to update custom or user values at runtime. In this research, AWS Lambda encryption function and IAM role are embedded into the template itself. Apart from that, CloudTrail event rule and pattern and role policies are included in this template.

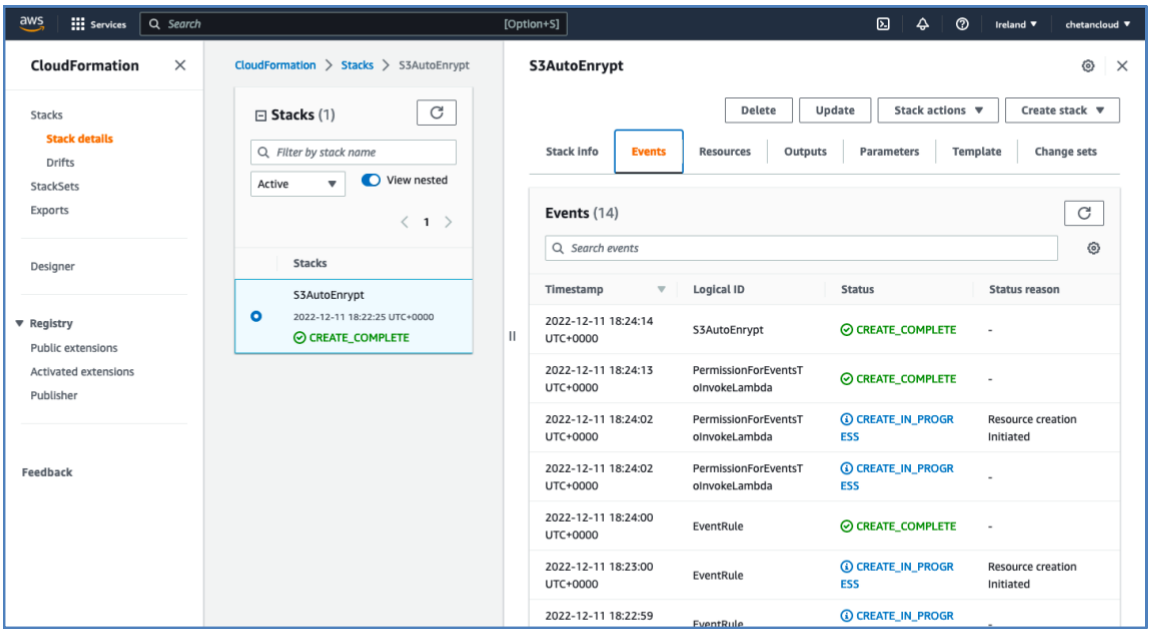


```
---
AWSTemplateFormatVersion: 2010-09-09

Resources:
  LambdaFunction:
    Type: AWS::Lambda::Function
    Properties:
      Role: !GetAtt LambdaExecutionRole.Arn
      Runtime: python3.7
      Handler: index.handler
      Code:
        ZipFile: |
          import json
          import boto3
          def s3autoencrypt(name):
            s3 = boto3.resource('s3')
            s3_client = boto3.client('s3')
            bucket = s3.Bucket(name)
            try:
              s3_client.get_bucket_encryption(Bucket=bucket.name)
              print("Encryption already set")
            except s3_client.exceptions.ClientError as error:
              if 'ServerSideEncryptionConfigurationNotFoundError' in str(error):
                print(f"Auto Encrypting {bucket.name}")
                response = s3_client.put_bucket_encryption(
                  Bucket=bucket.name,
                  ServerSideEncryptionConfiguration={
                    'Rules': [
```

Figure 6: S3AutoEncrypt CloudFormation Template

Once all required AWS services are provisioned and configured in template.it is saved with .yaml extension and uploaded into AWS CloudFormation. As shown in figure 7, after deployment wait for event to complete.

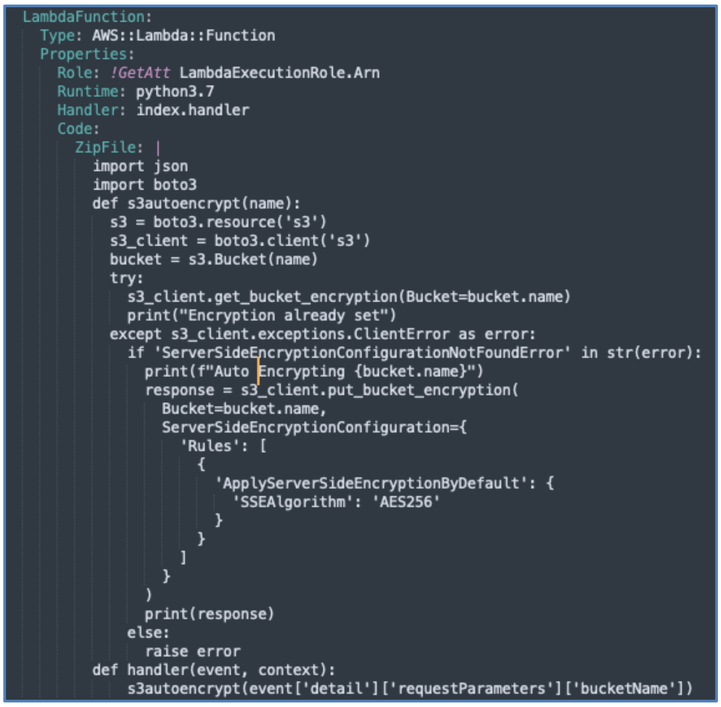


Timestamp	Logical ID	Status	Status reason
2022-12-11 18:24:14 UTC+0000	S3AutoEnrypt	CREATE_COMPLETE	-
2022-12-11 18:24:13 UTC+0000	PermissionForEventsToInvokeLambda	CREATE_COMPLETE	-
2022-12-11 18:24:02 UTC+0000	PermissionForEventsToInvokeLambda	CREATE_IN_PROGRESS	Resource creation Initiated
2022-12-11 18:24:02 UTC+0000	PermissionForEventsToInvokeLambda	CREATE_IN_PROGRESS	-
2022-12-11 18:24:00 UTC+0000	EventRule	CREATE_COMPLETE	-
2022-12-11 18:23:00 UTC+0000	EventRule	CREATE_IN_PROGRESS	Resource creation Initiated
2022-12-11 18:22:59	EventRule	CREATE_IN_PROGR	-

Figure 7: CloudFormation Events

There are possibilities and situations happened in the past that user forgot to enable the encryption while creating bucket and all confidential critical data become vulnerable and used for malicious activity; Hence to solve this problem, AWS Lambda function is used to facilitate encryption to each and every bucket while user creates the S3 bucket.

6.2 AWS Lambda function for automatic encryption



```
LambdaFunction:
  Type: AWS::Lambda::Function
  Properties:
    Role: !GetAtt LambdaExecutionRole.Arn
    Runtime: python3.7
    Handler: index.handler
    Code:
      ZipFile: |
        import json
        import boto3
        def s3autoencrypt(name):
          s3 = boto3.resource('s3')
          s3_client = boto3.client('s3')
          bucket = s3.Bucket(name)
          try:
            s3_client.get_bucket_encryption(Bucket=bucket.name)
            print("Encryption already set")
          except s3_client.exceptions.ClientError as error:
            if 'ServerSideEncryptionConfigurationNotFoundError' in str(error):
              print(f"Auto Encrypting {bucket.name}")
              response = s3_client.put_bucket_encryption(
                Bucket=bucket.name,
                ServerSideEncryptionConfiguration={
                  'Rules': [
                    {
                      'ApplyServerSideEncryptionByDefault': {
                        'SSEAlgorithm': 'AES256'
                      }
                    }
                  ]
                }
              )
              print(response)
            else:
              raise error
        def handler(event, context):
            s3autoencrypt(event['detail']['requestParameters']['bucketName'])
```

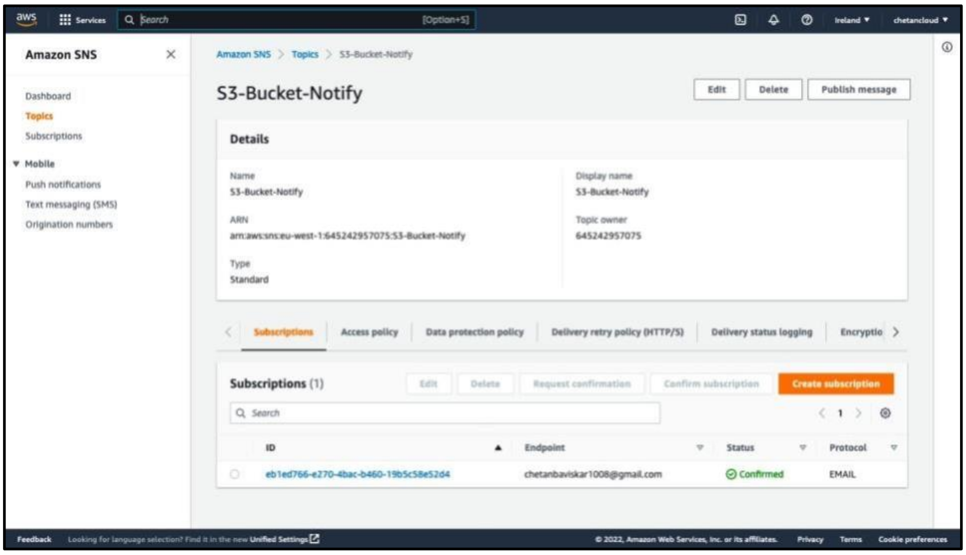
Figure 8: AWS Lambda Function

As displayed in figure 8, AWS Lambda function ‘s3autoencrypt’ is written in python programming language hence boto3 library is imported so that Amazon S3 bucket can utilize in the program. This lambda function is designed such way that as soon as user create S3 bucket, this function automatically gets trigger and check if encryption is applied to the S3 bucket or not; if the encryption is not used then it will impose and apply the encryption to that particular S3 bucket.

In Lambda function, s3_client represent low level client which denoted as independent client which does not depends on any request or response. This s3_client calls get_bucket_encryption method which returns and check whether server-side encryption is applied to S3 bucket or not. If not this s3_client calls put_bucket_encryption method which configure new server-side encryption (or substitute an existing one, if present). Container of server-side encryption set up rule as ‘ApplyServerSideEncryptionByDefault’ which denotes that by default server-side encryption is applied to new objects in the bucket. Even if create object request does mention any server-side encryption, this by default encryption will be applied.

Server-side encryption follows default algorithm that is SSEAlgorithm. The SSEAlgorithm uses AES-256 (Advance Encryption standards). It is extremely secure as it takes around 2^{39} times to recover complete 256 bit key.

6.3 AWS Simple Notification Service (SNS)



S3-Bucket-Notify

Details

Name
S3-Bucket-Notify

Display name
S3-Bucket-Notify

ARN
arn:aws:sns:eu-west-1:645242957075:S3-Bucket-Notify

Topic owner
645242957075

Type
Standard

Subscriptions (1)

ID	Endpoint	Status	Protocol
eb1ed766-e270-4bac-b460-19b5c58e52d4	chetanbaviskar1008@gmail.com	Confirmed	EMAIL

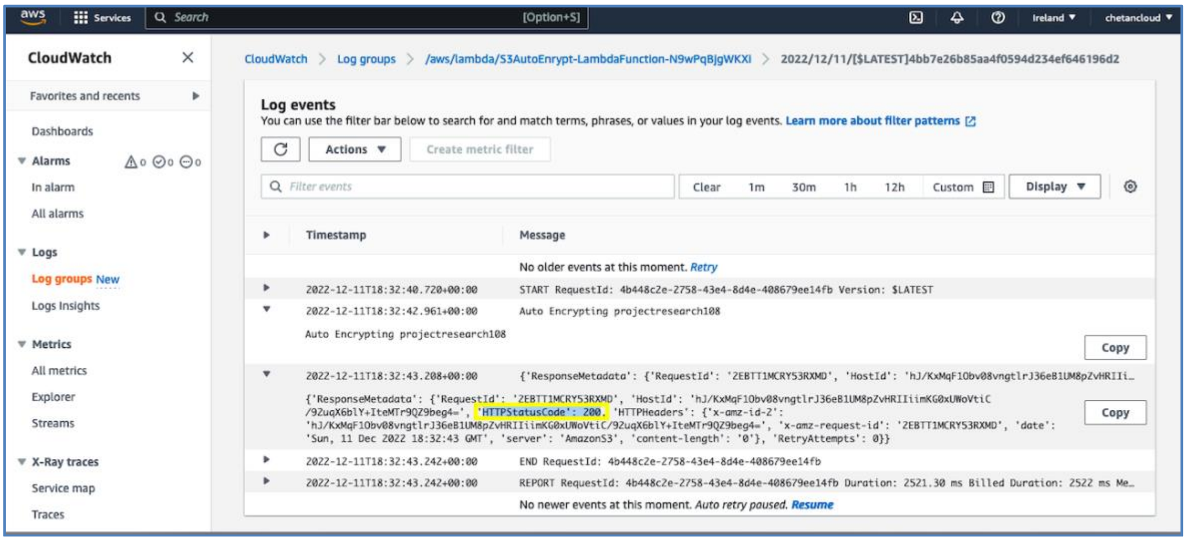
Figure 9: SNS

As per figure 9, AWS Simple Notification Service (SNS) creates topic to subscribed to email service endpoint at receiver end to get warnings about S3 object insertion and deletion from S3 bucket.

6.4 CloudWatch logs

Once CloudFormation template is deployed successfully ('Create Complete'). AWS Lambda generates logs of request and response in the CloudWatch. As depicted in figure 10, S3 bucket name 'projectresearch108' is being encrypted automatically through AWS lambda.

In CloudWatch response logs, HTTPStatusCode:200 states successful completion of request.



CloudWatch > Log groups > /aws/lambda/S3AutoEnrypt-LambdaFunction-N9wPqBjgWKXI > 2022/12/11/[$LATEST]4bb7e26b85aa4f0594d234ef646196d2

Log events

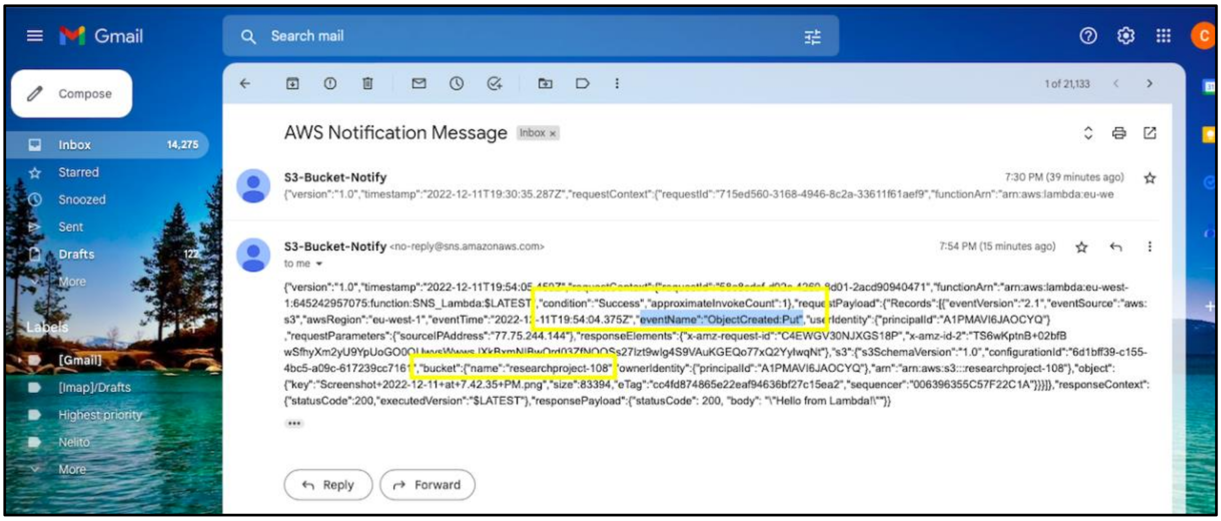
You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

```
Timestamp                       Message
                                No older events at this moment. Retry
2022-12-11T18:32:40.720+00:00   START RequestId: 4b448c2e-2758-43e4-8d4e-408679ee14fb Version: $LATEST
2022-12-11T18:32:42.961+00:00   Auto Encrypting projectresearch108
Auto Encrypting projectresearch108
2022-12-11T18:32:43.208+00:00   {'ResponseMetadata': {'RequestId': '2EBTT1MCRY53RXMD', 'HostId': 'hJ/KxMqF1Obv08vngtlrJ36eB1UM8pZvHRIIi…
{'ResponseMetadata': {'RequestId': '2EBTT1MCRY53RXMD', 'HostId': 'hJ/KxMqF1Obv08vngtlrJ36eB1UM8pZvHRIIiimKG0xUWoVtiC
/92uqX6blY+IteMTr9QZ9beg4=', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amz-id-2':
'hJ/KxMqF1Obv08vngtlrJ36eB1UM8pZvHRIIiimKG0xUWoVtiC/92uqX6blY+IteMTr9QZ9beg4=', 'x-amz-request-id': '2EBTT1MCRY53RXMD', 'date':
'Sun, 11 Dec 2022 18:32:43 GMT', 'server': 'AmazonS3', 'content-length': '0'}, 'RetryAttempts': 0}}
2022-12-11T18:32:43.242+00:00   END RequestId: 4b448c2e-2758-43e4-8d4e-408679ee14fb
2022-12-11T18:32:43.242+00:00   REPORT RequestId: 4b448c2e-2758-43e4-8d4e-408679ee14fb Duration: 2521.30 ms Billed Duration: 2522 ms Me…
                                No newer events at this moment. Auto retry paused. Resume
```

Figure 10. CloudWatch logs

6.5 Notification to user

Using Simple Notification Service (SNS) user receives alerts on email about S3 bucket operation of object insertion or deletion using another lambda function. As per figure 11, ‘S3-Bucket-Notify’ is representing SNS topic that indicate bucket name and event name ‘ObjectCreated:Put’ is indication of object being uploaded in S3 Bucket. A quick, adaptable, and fully managed messaging solution and it can also use as a corporate messaging network or as a cloud-oriented mobile app alert notification service to transmit push notifications, emails, and mobile messages.



AWS Notification Message

S3-Bucket-Notify 7:30 PM (39 minutes ago)

{"version":"1.0","timestamp":"2022-12-11T19:30:35.287Z","requestContext":{"requestId":"715ed560-3168-4946-8c2a-33611f61aef9","functionArn":"arn:aws:lambda:eu-we

S3-Bucket-Notify <no-reply@sns.amazonaws.com> 7:54 PM (15 minutes ago)
to me

{"version":"1.0","timestamp":"2022-12-11T19:54:05.450Z","requestContext":{"requestId":"58e8edef-d02e-4260-9d01-2acd90940471","functionArn":"arn:aws:lambda:eu-west-1:645242957075:function:SNS_Lambda:$LATEST","condition":"Success","approximateInvokeCount":1},"requestPayload":{"Records":[{"eventVersion":"2.1","eventSource":"aws:s3","awsRegion":"eu-west-1","eventTime":"2022-12-11T19:54:04.375Z","eventName":"ObjectCreated:Put","userIdentity":{"principalId":"A1PMAVI6JAOCYQ"},"requestParameters":{"sourceIPAddress":"77.75.244.144"},"responseElements":{"x-amz-request-id":"C4EWGV30NJXGS18P","x-amz-id-2":"TS6wKptnB+02bfB wSfhyXm2yU9YpUoGO0OUwysWwws IXkBxmNIBwOrd03ZfNOOSs27Izt9wlg4S9VAuKGEQo77xQ2YylwqNt"},"s3":{"s3SchemaVersion":"1.0","configurationId":"6d1bff39-c155-4bc5-a09c-617239cc7161","bucket":{"name":"researchproject-108","ownerIdentity":{"principalId":"A1PMAVI6JAOCYQ"},"arn":"arn:aws:s3:::researchproject-108"},"object":{"key":"Screenshot+2022-12-11+at+7.42.35+PM.png","size":83394,"eTag":"cc4fd874865e22eaf94636bf27c15ea2","sequencer":"006396355C57F22C1A"}}}]},"responseContext":{"statusCode":200,"executedVersion":"$LATEST"},"responsePayload":{"statusCode": 200, "body": "\"Hello from Lambda!\""}}

Figure 11: SNS Notification

7. Conclusion and Future Work

Data privacy is crucial part of any institution. When reading the news, it’s hard to pass a few days without reading about a huge data breach that might have given scammers access to the personal information of millions of users. In this report, it is successfully demonstrated that how information stored in AWS S3 bucket can be protected efficiently by applying automated encryption technique with AWS Lambda. Using this method, S3 bucket not only safeguarded but also shielded with SNS service that shoots the notification to user for any alteration of S3 bucket objects. AWS Lambda being event driven and plays important role in this practice. Using SNS service user can avail notification alert service not only email but also mobile SMS to get push notification. AWS CloudTrail and AWS CloudWatch are responsibly collecting the logs of all the cloud services used in this technique. Apart from that, it would be difficult without CloudFormation that managed and setup all cloud services under single roof and makes the automation very smooth.

In a future perspective, AWS CloudFormation and Lambda are going to be a game changer in cloud and cybersecurity industry. As it brings more of such automation in the system which makes system actionable quickly without any delay. AWS SQS (Simple Queue service) more likely to be used over SNS in such a scenario because of its message storing capability. Ultimately, security and privacy are linked and hence it’s good practice of protecting both.

References

Beer, K. and Holland, R., 2014. Encrypting Data at Rest. White Paper of amazon web services.

Bourke, D. and Grzelak, D., 2018. Breach detection at scale with aws honey tokens. Blackhat Asia, https://www. blackhat. com/asia-18/briefings. htmlbreach- detection-at-scale-withaws-honey-tokens, pp.20-23

Cable, J., Gregory, D., Izhikevich, L. and Durumeric, Z., 2021, October. Stratosphere: Finding vulnerable cloud storage buckets. In 24th International Symposium on Research in Attacks, Intrusions and Defenses (pp. 399-411).

Chuang, I.H., Li, S.H., Huang, K.C. and Kuo, Y.H., 2011, February. An effective privacy protection scheme for cloud computing. In 13th International Conference on Advanced Communication Technology (ICACT2011) (pp. 260-265). IEEE.

Cloudtrail, A. (2022) Secure Standardized Logging - AWS CloudTrail - Amazon Web Services, Amazon Web Services, Inc. Available at: https://aws.amazon.com/cloudtrail/ (Accessed: April 2, 2022).

Continella, A., Polino, M., Pogliani, M. and Zanero, S., 2018, December. There's a hole in that bucket! a large-scale analysis of misconfigured s3 buckets. In Proceedings of the 34th Annual Computer Security Applications Conference (pp. 702-711).

GeeksforGeeks (2022) S3Scanner - Scan For Open S3 Buckets And Dump - GeeksforGeeks, GeeksforGeeks. Available at: https://www.geeksforgeeks.org/s3scanner-scan-for-open-s3-buckets-and-dump/

Hofman, D., Duranti, L. and How, E., 2017. Trust in the balance: Data protection laws as tools for privacy and security in the cloud. Algorithms, 10(2), p.47.

krishnamoorthy (2022). Private Amazon S3 buckets can leak. [online]Medium. Available at: https://medium.com/towards-aws/private-amazon-s3-buckets-do-leak- 9bee13c775e1 [Published 29 May. 2022]

Kumar, A., Lee, H. and Singh, R.P., 2012, October. Efficient and secure Cloud storage for handling big data. In 2012 6th International Conference on New Trends in Information Science, Service Science and Data Mining (ISSDM2012) (pp. 162-166). IEEE

McGrath, G. and Brenner, P.R., 2017, June. Serverless computing: Design, implementation, and performance. In 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW) (pp. 405-410). IEEE.

Mosca, P., Zhang, Y., Xiao, Z. and Wang, Y., 2014. Cloud security: Services, risks, and a case study on amazon cloud services. Int'l J. of Communications, Network and System Sciences, 7(12), p.529

Mukherjee, S., 2019. Benefits of AWS in the modern cloud. arXiv preprint arXiv:1903.03219

Somani, G., Gaur, M.S., Sanghi, D., Conti, M. and Buyya, R., 2017. DDoS attacks in cloud computing: Issues, taxonomy, and future directions. Computer Communications, 107, pp.30-48.

Ozer, M., Varlioglu, S., Gonen, B., Adewopo, V., Elsayed, N. and Zengin, S., 2020, December. Cloud incident response: Challenges and opportunities. In 2020 International Conference on Computational Science and Computational Intelligence (CSCI) (pp. 49-54). IEEE.

Quiles, C. (2021). S3 Encryption. [online] Medium. Available at: https://medium.com/@quileswest/automating- aws-lambda-to-run-python-for-s3-encryption-559f624bf291 [Published 23 Feb. 2021]

Sailakshmi, V., 2021. Analysis of Cloud Security Controls in AWS, Azure, and Google Cloud.

SNS, A. (2022) Messaging Service – Amazon Simple Notification Service (SNS) – Amazon Web Services, Amazon Web Services, Inc. Available at: https://aws.amazon.com/sns/ (Accessed: November 2, 2022).

Tang, J., Cui, Y., Li, Q., Ren, K., Liu, J. and Buyya, R., 2016. Ensuring security and privacy preservation for cloud data services. ACM Computing Surveys (CSUR), 49(1), pp.1-39

Torkura, K.A., Sukmana, M.I., Cheng, F. and Meinel, C., 2021. Continuous auditing and threat detection in multi-cloud infrastructure. Computers & Security, 102, p.102124.

Torkura, K.A., Sukmana, M.I., Strauss, T., Graupner, H., Cheng, F. and Meinel, C., 2018, November. Csbauditor: Proactive security risk analysis for cloud storage broker systems. In 2018 IEEE 17th International Symposium on Network Computing and Applications (NCA) (pp. 1-10). IEEE.

Wang, J., Zhao, Y., Jiang, S. and Le, J., 2010, May. Providing privacy preserving in cloud computing. In 3rd International Conference on Human System Interaction (pp. 472-475). IEEE