AWS SECURE FILE STORAGE WITH IAM AND CLOUDTRAIL MONITORING

A MINIPROJECT REPORT

Submitted by

RESHMA PM (73772221163)

SUVETHA S (73772221209)

VARSINI M (73772221213)

in partial fulfillment of the requirement

for the award of the degree

of

B.TECH

in

INFORMATION TECHNOLOGY

K.S. RANGASAMY COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

TIRUCHENGODE – 637 215

MAY 2025

K.S. RANGASAMY COLLEGE OF TECHNOLOGY
TIRUCHENGODE - 637 215

BONAFIDE CERTIFICATE

Certified that this project report titled **“AWS SECURE FILE STORAGE WITH IAM AND CLOUDTRAIL MONITORING ”** is the bonafide work of **RESHMA PM(73772221183)**, **SUVETHA S(73772221209)**, **VARSINI M (73772221213)** who carried out the project under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE
S. B. THAMARAI SELVI M.E.,
SUBJECT HANDLER
Assistant Professor
Department of Computer Science and Engineering
K.S. Rangasamy College of Technology
Tiruchengode - 637 215

SIGNATURE
DR.S. MADHAVI M.E., PH.D.,
HEAD OF THE DEPARTMENT
Professor
Department of Computer Science and Engineering
K.S. Rangasamy College of Technology
Tiruchengode - 637 215

DECLARATION

We jointly declare that the project report on **“AWS SECURE FILE STORAGE WITH IAM AND CLOUDTRAIL MONITORING”** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **“ANNA UNIVERSITY CHENNAI”** for the requirement of Degree of B.Tech. This project report is submitted on the partial fulfilment of the requirement of the award of Degree of B.Tech.

Signature

RESHMA PM

SUVETHA S

VARSINI M

Place :Tiruchengode

Date :

ABSTRACT

In today’s digital landscape, With the increasing demand for secure data management, organizations require efficient solutions for protecting sensitive information. This project, AWS Secure File Storage with IAM and CloudTrail Monitoring, aims to provide a scalable, cost-effective, and secure storage solution using Amazon Web Services (AWS). It leverages Amazon S3 for durable file storage, AWS Identity and Access Management (IAM) for fine-grained access control, and AWS CloudTrail for real-time monitoring and auditing. The system enforces strict access policies, ensuring that only authorized users can access critical data while maintaining a comprehensive log of all access attempts. This approach enhances data security, prevents unauthorized access, and provides detailed insights into data access patterns, making it ideal for businesses, educational institutions, and government organizations handling sensitive data. The project also includes automated notifications using Amazon SNS for real-time alerts on suspicious access attempts, ensuring proactive security management.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

In today’s digital landscape, secure data management is a critical requirement for organizations of all sizes. As businesses continue to generate and store vast amounts of sensitive information, ensuring data privacy and protection has become a top priority.To address this, Amazon Web Services (AWS) offers a comprehensive suite of tools and services, including Amazon S3, AWS Identity and Access Management (IAM), and AWS CloudTrail, which together provide a highly secure, scalable, acost-effective file storage solution, access control to prevent unauthorized data breaches.

Amazon S3 serves as the backbone of this project, offering durable, highly available, and scalable object storage for critical data. With features like versioning, lifecycle policies, and server-side encryption, S3 provides a robust platform for securely storing sensitive information. This is where AWS IAM plays a crucial role, enabling fine-grained permissions and user authentication to ensure that only authorized personnel can access critical files.

By To further enhance security, AWS CloudTrail is integrated to provide detailed logging and monitoring of all file access activities. This helps organizations maintain a complete audit trail, detect unauthorized access attempts, and comply with regulatory requirements. Additionally, real-time alerts through Amazon SNS can notify administrators of suspicious activity, allowing for proactive threat response.

CHAPTER 2

TOOL USED - AWS SERVICES

This project utilized a variety of AWS services and supporting technologies to implement a Secure File Storage solution with real-time monitoring and fine-grained access control.Below is a brief description of each tool and its role in the project:

1. Amazon S3 (Simple Storage Service)

Amazon S3 provides scalable, durable, and secure object storage for files. It is used to store sensitive data with server-side encryption, versioning, and lifecycle management for efficient file management. S3 buckets are configured with secure access policies to prevent unauthorized access.

2. AWS IAM (Identity and Access Management)

IAM enables fine-grained access control by allowing administrators to create and manage user roles, policies, and multi-factor authentication (MFA). This ensures that only authorized users can access critical files, reducing the risk of data breaches.

3. AWS Cloudtrail

CloudTrail captures and logs every API call and file access activity within the AWS environment, providing a comprehensive audit trail for security analysis and compliance. It helps track user actions and identify potential security threats.

4. Amzon SNS (Simple Notification Service)

SNS is used for real-time alerts, sending notifications for unauthorized access attempts or critical security events. It supports multiple communication channels, including email, SMS, and mobile push notifications, ensuring immediate response to security incidents.

5. Amazon CloudWatch

CloudWatch provides real-time monitoring, custom metrics, and alarms to track file access patterns and detect anomalies. It offers dashboards and automated responses to improve security visibility and operational efficiency.

6. AWS KMS (Key Management System)

KMS is used for encrypting sensitive data stored in S3, utilizing customer-managed keys for enhanced data protection. It integrates seamlessly with IAM policies for fine-grained access control.

7. Amazon Lambda

Lambda is used for automated file processing, such as virus scanning, metadata extraction, or triggering alerts based on file activity. It eliminates the need for managing servers and supports event-driven workflows.

8. Apache Web Server

AWS S3 Object Lock is a powerful feature that helps protect your data from accidental or malicious deletion. It allows you to enforce write-once-read-many (WORM) policies, ensuring that objects remain immutable for a specified retention period. This is particularly useful for compliance requirements where data integrity must be preserved for various security needs.

CHAPTER 3

IMPLEMENTATION STEPS

This section outlines the step-by-step process followed to deploy a Secure File Storage System on AWS using services like Amazon S3, IAM, CloudTrail, SNS, and CloudWatch. The objective was to create a highly secure, scalable, and monitored file storage architecture.

3.1 S3 Bucket Creation and Configuration

- **Bucket Name:** A globally unique name was selected for the S3 bucket.
- **Region:** Chose a region close to the primary user base for low latency.
- **Versioning:** Enabled to maintain multiple versions of the same object for added data protection.
- **Encryption:** Configured server-side encryption using S3 Managed Keys (SSE-S3) or AWS KMS for advanced security.
- **Access Control:** Applied strict bucket policies and ACLs to limit access to authorized users only.
- **Bucket Policy:** Added policies to restrict public access and enforce encryption.

3.2 IAM Role and Policy Setup

- **Role Creation:** Created an IAM role for the S3 bucket to control access securely.
- **Policy Attachment:** Attached a custom policy that grants the necessary permissions for S3 read and write operations.
- **MFA Requirement:** Enabled multi-factor authentication (MFA) for additional security on critical actions.
- **Fine-Grained Permissions:** Used IAM policies to enforce the principle of least privilege.

3.3 Enabling CloudTrail for Audit Logging

- **Trail Creation:** Created a CloudTrail trail to log all API activity in the AWS account.
- **Log Storage:** Configured the trail to store logs in a dedicated S3 bucket.
- **Encryption:** Enabled encryption for the CloudTrail logs for added security.
- **Multi-Region Logging:** Enabled global service events for comprehensive logging.
- **Event History:** Configured the retention period for log files as per compliance.

3.4 Setting Up Real-Time Alerts with SNS

- **Topic Creation:** Created an SNS topic for security alerts.
- **Subscription:** Added email and SMS subscribers for immediate notifications.
- **Integration:** Linked the SNS topic with CloudWatch to receive alerts on unauthorized access attempts.
- **Message Filtering:** Configured message filters to reduce noise and focus on critical security events.

3.5 CloudWatch Configuration for Monitoring

- **Custom Metrics:** Created custom CloudWatch metrics to track S3 access patterns and object size.
- **Alarms:** Set up alarms for unusual file access or high read/write activity.
- **Dashboard Creation:** Built real-time dashboards for better visibility into file access.
- **Automated Responses:** Linked CloudWatch alarms with SNS for automated incident response.

3.6 Automating File Processing with Lambda

- **Function Creation:** Created a Lambda function to automate tasks like virus scanning or metadata extraction.
- **Trigger Setup:** Configured S3 event triggers for real-time processing.
- **IAM Role:** Assigned a role to the Lambda function with the necessary S3 permissions.
- **Test Execution:** Validated the function by uploading test files to the S3 bucket.

3.7 Final Validation

- **Access Testing:** Verified access control by attempting unauthorized actions.
- **Log Verification:** Checked CloudTrail logs for accurate event tracking.
- **Alert Testing:** Simulated unauthorized access to confirm SNS notifications.
- **Data Integrity:** Validated data encryption and decryption using KMS keys.

CHAPTER 4

TESTING

This section outlines the various testing procedures performed to validate the configuration and functionality of the AWS Secure File Storage System. The objective was to ensure high security, access control, monitoring, and data integrity across the deployed architecture.

4.1 S3 Bucket Test

After creating the S3 bucket, its accessibility and security settings were verified:

- Attempted to access the bucket from an unauthorized IAM user.
- Verified that access was denied due to the bucket policy and IAM restrictions.
- Uploaded test files to confirm correct write permissions for authorized users.
- Confirmed that the bucket enforced server-side encryption for all uploaded files.

This confirmed that:

- The S3 bucket was secure against unauthorized access.
- The bucket policy and IAM roles were correctly implemented.
- Data encryption settings were correctly applied.

4.2 IAM Role and PolicyValidation

To ensure the IAM roles were correctly configured:

- Logged in as an IAM user with restricted S3 access.
- Attempted to perform unauthorized actions (e.g., deleting a file without proper permissions).
- Verified that actions were blocked based on the principle of least privilege.
- Tested MFA-enforced access to confirm enhanced security.

This confirmed that:

- IAM roles had the correct permissions and were securely configured.
- MFA requirements were correctly enforced for sensitive actions.

4.3 CloudTrail Logging Verification

To validate the CloudTrail configuration:

- Performed various S3 actions (upload, download, delete) and checked the CloudTrail logs.
- Verified that all API calls and user actions were accurately captured.
- Checked that log files were stored securely in the designated S3 bucket.
- Confirmed that event logs were encrypted for data security.

4.4 SNS Notification Test

To confirm real-time alerts were working:

- Triggered unauthorized access attempts to generate SNS notifications.
- Verified that emails or SMS alerts were received promptly.
- Tested message filtering to ensure only critical alerts were forwarded.

This confirmed that:

- SNS was correctly integrated for real-time security alerts.
- Notifications were effectively reaching the designated recipients.

4.5 CloudWatch Monitoring and Alarm Test

To validate CloudWatch Monitoring:

- Uploaded large volumes of data to the S3 bucket to trigger alarms.
- Monitored custom metrics like data access rates and unauthorized access attempts.
- Verified that alarms were generated as expected and integrated with SNS for instant alerts.

This confirmed that:

- CloudWatch was effectively monitoring file storage activity.
- Automated responses to security incidents were functioning correctly.

4.6 Final Security Audit

To validate CloudWatch Monitoring:

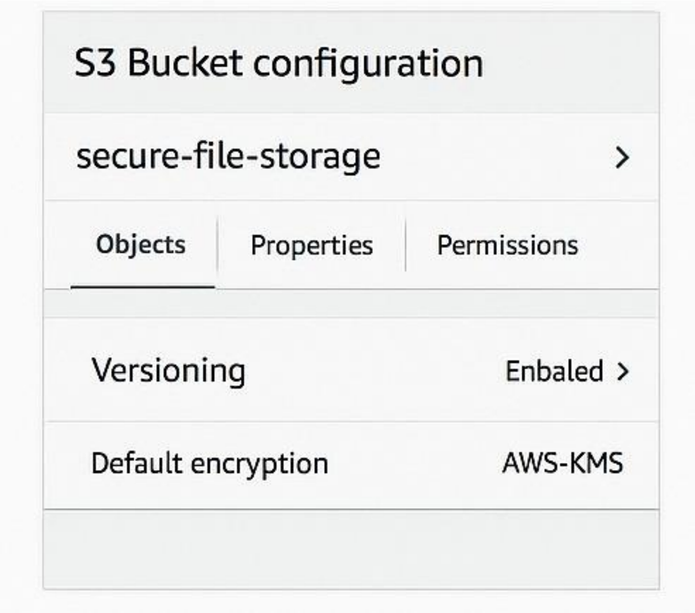
- Performed a comprehensive review of IAM policies, bucket permissions, and CloudTrail logs.
- Tested for common security vulnerabilities, including excessive permissions and misconfigurations.
- Conducted penetration tests to assess the overall security posture.

This confirmed that:

- The overall architecture was secure and compliant with best practices.
- Sensitive data was protected against unauthorized access.

CHAPTER 5

RESULTS



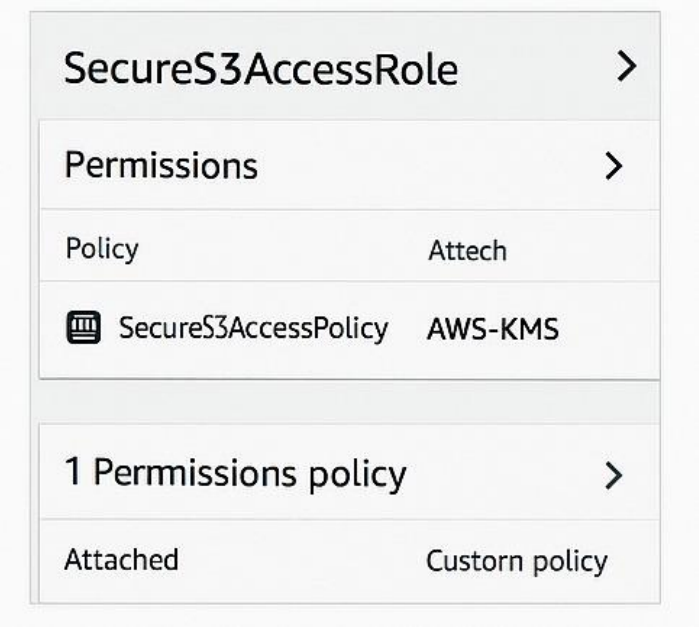
S3 Bucket configuration

secure-file-storage >

Versioning Enbaled >

Default encryption AWS-KMS

Figure 5.1 S3 Bucket Configuration



SecureS3AccessRole >

Permissions >

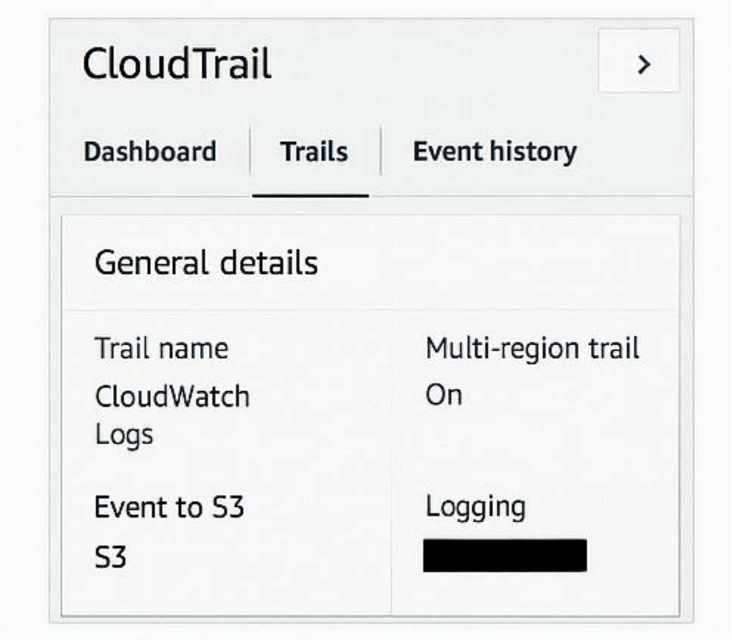
Policy Attech

SecureS3AccessPolicy AWS-KMS

1 Permissions policy >

Attached Custorn policy

Figure 5.2 IAM Role and Policy Attatchment



CloudTrail

General details

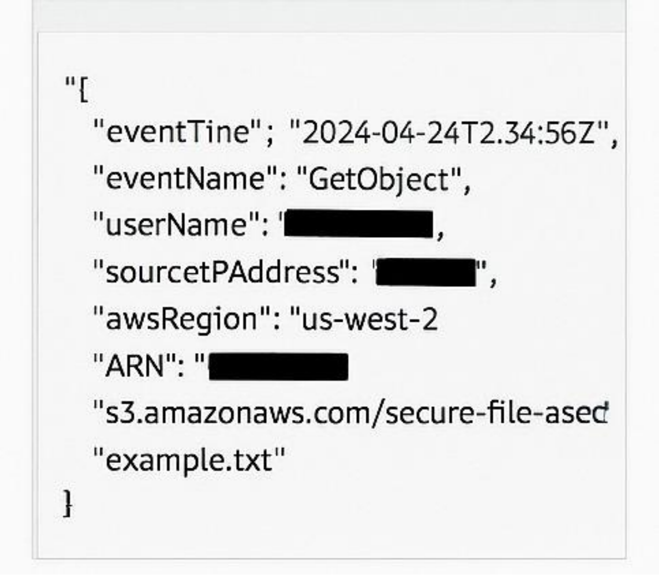
Trail name
CloudWatch Logs

Multi-region trail
On

Event to S3
S3

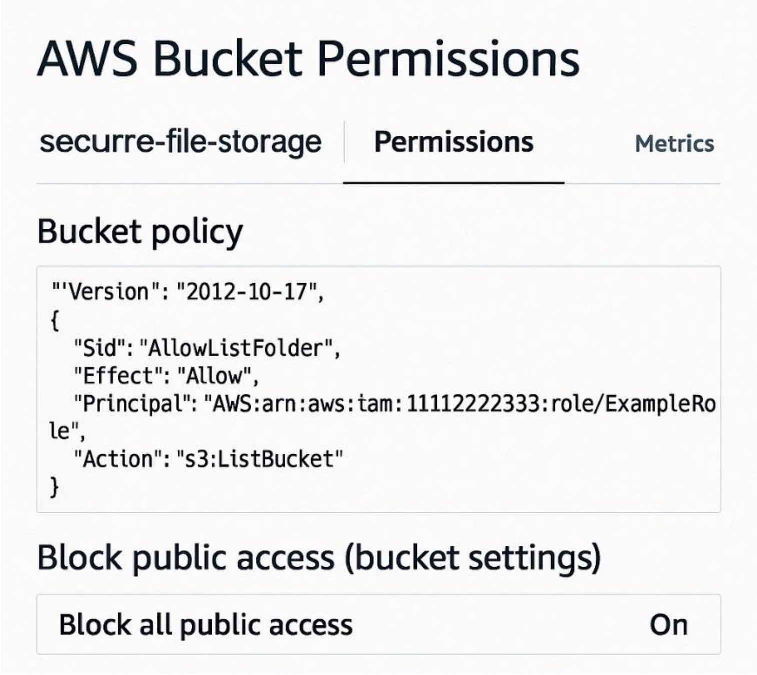
Logging
[redacted]

Figure 5.3 CloudTrail Configuration



```
"[
  "eventTine"; "2024-04-24T2.34:56Z",
  "eventName": "GetObject",
  "userName": '[redacted],
  "sourcetPAddress": '[redacted]",
  "awsRegion": "us-west-2
  "ARN": "[redacted]
  "s3.amazonaws.com/secure-file-ased
  "example.txt"
}
```

Figure 5.4 CloudTrail Log Entry



AWS Bucket Permissions

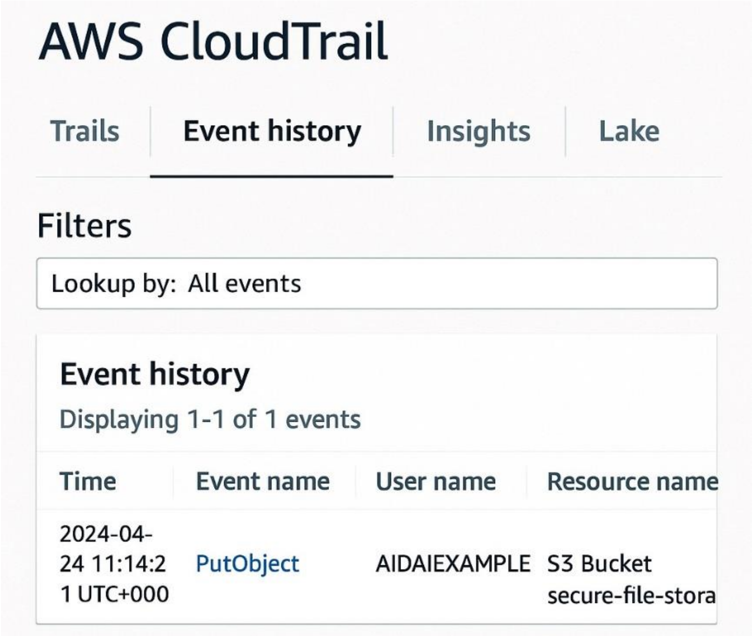
Bucket policy

```
"'Version": "2012-10-17",
{
  "Sid": "AllowListFolder",
  "Effect": "Allow",
  "Principal": "AWS:arn:aws:tam:11112222333:role/ExampleRo
le",
  "Action": "s3:ListBucket"
}
```

Block public access (bucket settings)

Block all public access On

Figure 5.3 Bucket Permissions



AWS CloudTrail

Filters

Lookup by: All events

Event history

Displaying 1-1 of 1 events

Time	Event name	User name	Resource name
2024-04-24 11:14:21 UTC+000	PutObject	AIDAIEXAMPLE	S3 Bucket secure-file-stora

Figure 5.3 CloudTrail

CHAPTER 6

FUTURE SCOPE

While the current project establishes a robust foundation for secure file storage and access management using AWS services, several enhancements can be explored in future iterations to improve scalability, security, and automation.

1. Advanced Encryption and Key Management
   - Integrate AWS KMS (Key Management Service) for fine-grained control over encryption keys.
   - Implement envelope encryption for an added layer of data security.
2. Automated File Lifecycle Management
   - Use S3 lifecycle policies to automatically archive old files to Amazon S3 Glacier.
   - Implement intelligent tiering to reduce storage costs for infrequently accessed data.
3. Enhanced Real-Time Monitoring and Alerts
   - Set up advanced CloudWatch dashboards for real-time monitoring of file access patterns.
   - Use AWS EventBridge for more sophisticated alerting and automated responses to suspicious activity.
4. Multi-Region Disaster Recovery
   - Implement cross-region replication (CRR) for higher data durability and faster recovery.
   - Use AWS Backup for comprehensive, automated data protection.
5. Integration with Machine Learning
   - Use Amazon Macie for automated data classification and sensitive data discovery.
   - Integrate AWS Lambda for real-time file scanning and anomaly detection.

6. Global Load Balancing and Multi-Region Support
   - Expand deployment to multiple AWS regions using Route 53 and Global Accelerator.
   - Ensures better latency and disaster recovery readiness.
7. Serverless File Processing
   - Use AWS Lambda for automated file processing, such as virus scanning or metadata extraction.
   - Leverage S3 event triggers for real-time processing without managing servers.
8. Enhanced Access Control and MFA
   - Enforce IAM policies with context-based access control (e.g., IP restrictions).
   - Use AWS SSO for centralized user management and MFA enforcement.

CONCLUSION

In this project, a comprehensive and secure file storage system was successfully implemented using AWS services such as Amazon S3, IAM, CloudTrail, SNS, and CloudWatch. The architecture demonstrated effective access control, real-time monitoring, and automated event notifications, ensuring data integrity and security.

The project effectively addressed critical aspects like fine-grained access management, data encryption, and detailed audit logging, making it a robust solution for secure file management. The integration of multi-layered security features, including IAM policies and CloudTrail logging, provided a strong foundation for maintaining data confidentiality and regulatory compliance.

Overall, this project highlights the potential of AWS cloud services in building scalable, secure, and efficient file storage architectures, laying the groundwork for future enhancements such as automated lifecycle management, serverless processing, and machine learning integration.

REFERENCES

1. https://docs.aws.amazon.com/s3/index.html
2. https://docs.aws.amazon.com/iam/index.html
3. https://docs.aws.amazon.com/cloudtrail/index.html
4. https://docs.aws.amazon.com/cloudwatch/index.html