

Enterprise Data Protection and Encryption Management on AWS

Author: Adedayo

Specialization: Cloud Security & Data Protection

Platform: Amazon Web Services (AWS)

## 1. Introduction

This document describes the design and implementation of a comprehensive data protection and encryption framework in an AWS environment.

I implemented this framework to ensure that sensitive data stored in S3, EBS, and RDS is protected using strong encryption, centralized key management, and continuous compliance monitoring.

## 2. Objectives

The primary objectives of this implementation were to:

- Prevent public exposure of sensitive data
- Enforce encryption at rest across all storage services
- Centralize key management using AWS KMS
- Enable automatic and manual key rotation
- Maintain continuous compliance visibility
- Reduce the risk of data leakage

## 3. S3 Data Protection

### 3.1 Account-Level Public Access Controls

I enabled S3 Block Public Access at the account level to prevent accidental public exposure of any bucket.

The screenshot shows the 'Amazon S3' service in the AWS console. On the left, the navigation pane includes 'Buckets', 'Access management and security', 'Storage management and insights', and 'Account and organization settings'. Under 'Account and organization settings', there is a link to 'AWS Marketplace for S3'. The main content area is titled 'Account and organization settings' and contains a green success message: 'Block Public Access settings for this account successfully updated.' Below this, the 'Block Public Access settings for this account' section is shown, with all five options ('Block all public access', 'Block public access to buckets and objects granted through new access control lists (ACLs)', 'Block public access to buckets and objects granted through any access control lists (ACLs)', 'Block public access to buckets and objects granted through new public bucket or access point policies', and 'Block public and cross-account access to buckets and objects through any public bucket or access point policies') set to 'On'. At the bottom, the 'AWS Organizations settings for Storage Lens' section is visible, showing an 'Organization ID' of 'o-prn0spcswbk' and a 'Status' of 'Disabled'.

This ensured that all existing and newly created buckets were protected by default.

### 3.2 S3 Encryption

All S3 buckets were encrypted using AWS SSE-KMS with customer-managed keys.

The screenshot shows the 'Amazon S3 > Buckets > aws-cloudtrail-logs-securephere' page. The left sidebar includes 'General purpose buckets', 'Access management and security', 'Storage management and insights', and 'Account and organization settings'. The main content area shows a green success message: 'Successfully edited default encryption. Objects uploaded, modified, or copied into this bucket will inherit this encryption configuration unless otherwise specified.' Below this, the 'Default encryption' section is displayed, indicating 'Server-side encryption is automatically applied to new objects stored in this bucket.' It shows the 'Encryption type' as 'Info' (Server-side encryption with AWS Key Management Service keys (SSE-KMS)) and the 'Encryption key ARN' as 'arn:aws:kms:us-east-1:1879381257906:key/f9bc03c2-6857-4950-b1a0-c2f22628e590'. A note about 'Upcoming change to default encryption' states that from April 2026, server-side encryption with customer-provided keys (SSE-C) will be blocked by default for all new buckets. The 'Intelligent-Tiering Archive configurations' section is also present at the bottom.

This provided full control over encryption policies, auditing, and key lifecycle management.

## 4. EBS Encryption Management

## 4.1 Default Encryption

EBS encryption by default was enabled to ensure that all new volumes and snapshots are automatically encrypted.

The screenshot shows the AWS EC2 Settings page under the 'Security' tab. The left sidebar includes sections for Images, Elastic Block Store (selected), Network & Security, Load Balancing, Auto Scaling, and Settings. The main content area displays three configuration sections: 'EBS encryption' (Info: Manage the default encryption option for all new EBS volumes and copies of snapshots created in your account. Status: Always encrypt new EBS volumes is Enabled), 'Default encryption key' (Info: arn:aws:kms:us-east-1:879381257906:key/17d94a5b-c5bb-420c-acfb-7c6db3c12a42), and 'Block public access for AMIs' (Info: Block public access for AMIs at the account level to prevent the public sharing of your AMIs in this Region. Status: Public access is New public sharing blocked. Note: There are currently no publicly shared AMIs in this Region). There are also sections for 'Block public access for EBS snapshots' (Info: Block public access at the account level to prevent the public sharing of your snapshots in this Region. Status: Public access) and 'Block public access for RDS snapshots' (Info: Block public access at the account level to prevent the public sharing of your RDS snapshots in this Region. Status: Public access).

## 4.2 Migrating Unencrypted Volumes

An unencrypted EBS volume was identified on an existing EC2 instance.

To remediate this:

- A snapshot was created from the unencrypted volume
- The snapshot was copied and encrypted
- A new encrypted volume was created
- The instance was stopped
- The encrypted volume was attached

This process ensured encryption without data loss.

## 5. RDS Encryption Enforcement

All RDS databases were reviewed to confirm encryption at rest.

Encryption was enforced at creation time, as RDS encryption cannot be enabled after deployment.

Operational guidance was provided to ensure future databases follow this standard.

## 6. Key Management with AWS KMS

### 6.1 Customer-Managed Keys

A customer-managed symmetric KMS key was created for workloads.

This key was used to encrypt:

- S3 buckets
- EBS volumes
- RDS databases

### 6.2 Automatic Key Rotation

Automatic key rotation was enabled with a 365-day cycle.

The screenshot shows the AWS KMS console interface. On the left, there's a navigation sidebar with 'Key Management Service (KMS)' selected. Under 'Customer-managed keys', 'Custom key stores' is expanded, showing 'AWS CloudHSM key stores' and 'External key stores'. The main panel displays a 'General configuration' section for a key named 'securedataKMSKEY'. This section includes fields for 'Alias' (securedataKMSKEY), 'Status' (Enabled), 'Description' (empty), 'Creation date' (Dec 21, 2025 12:59 GMT), and 'Regionality' (Single region). Below this, there are tabs for 'Key policy', 'Cryptographic configuration', 'Tags', 'Key material and rotations' (which is currently selected), and 'Aliases'. The 'Key material and rotations' section contains a table with columns for 'Status' (Enabled), 'Rotation period' (365 days), 'Date of last automatic rotation' (empty), and 'Next rotation date' (Dec 21, 2026). There are also 'Edit' and 'Rotate now' buttons. At the bottom of this section, it says 'On-demand key rotation' and 'Immediately initiate key material rotation for this key. You can initiate on-demand key rotation a maximum of 10 times.'

This reduced the risk associated with long-lived cryptographic keys.

### 6.3 Manual Key Rotation Process

For keys that do not support automatic rotation, a documented manual rotation process was implemented:

Step 1: Identify the active key

Step 2: Create a new customer-managed key

Step 3: Update services to use the new key

Step 4: Monitor system behavior

Step 5: Disable the old key

Step 6: Retain the old key for recovery and compliance

This process ensured secure rotation without service disruption.

### 7. Compliance Monitoring and Alerting

#### 7.1 AWS Config Rules

AWS Config rules were created to monitor:

- S3 public access settings
- Storage encryption status
- KMS key usage

The screenshot shows the AWS Config Rules interface. On the left is a navigation sidebar with links like Dashboard, Conformance packs, Rules, Resources, Aggregators, Documentation, Partners, FAQs, and Pricing. The main area is titled 'Rules' and contains a table with the following data:

Name	Remediation action	Type	Enabled evaluation mode	Detective compliance
encrypted-volumes	Not set	AWS managed	DETECTIVE	Compliant
rds-storage-encrypted	Not set	AWS managed	DETECTIVE	Compliant
s3-bucket-public-read-pr...	Not set	AWS managed	DETECTIVE	Compliant
s3-bucket-public-write-p...	Not set	AWS managed	DETECTIVE	Compliant
securityhub-access-keys-...	Not set	AWS managed	DETECTIVE	Compliant
securityhub-acm-certific...	Not set	AWS managed	DETECTIVE	-
securityhub-acm-certific...	Not set	AWS managed	DETECTIVE	-

These rules provided real-time compliance visibility.

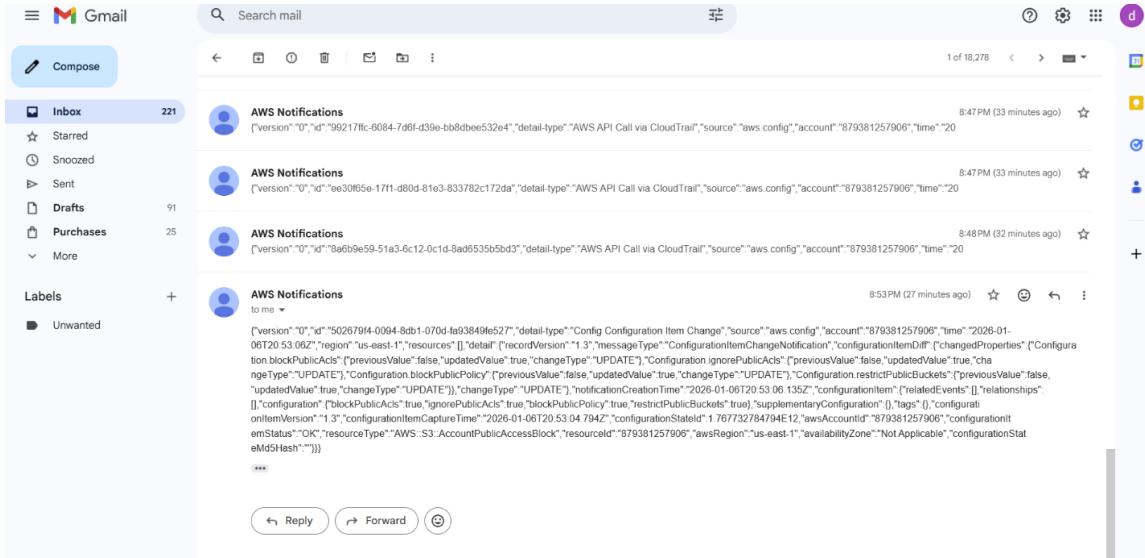
## 7.2 Alerting with SNS

Amazon SNS topics were configured to deliver alerts when non-compliant resources were detected.

## 7.3 Validation Testing

Public access settings were temporarily modified to validate alert delivery.

Successful notifications confirmed correct configuration.



## 8. Governance and Documentation

All encryption policies, key management procedures, and compliance controls were documented.

This supported:

- Security audits
- Regulatory compliance
- Incident investigations
- Operational continuity

## 9. Outcomes and Impact

This implementation delivered the following results:

- Eliminated public S3 exposure risk
- Enforced encryption across storage services
- Centralized cryptographic key management
- Reduced operational risk
- Improved audit readiness
- Enabled real-time compliance monitoring

## 10. Conclusion

I designed and implemented a secure, compliant, and centrally managed data protection framework on AWS.

Through strong encryption, effective key management, and continuous monitoring, this solution protects sensitive data and supports regulatory requirements.

ADEDAYO