

Enterprise Data Protection and Encryption Management on AWS

Author: Adedayo

Specialization: Cloud Security & Data Protection

Platform: Amazon Web Services (AWS)

1. Problem Statement

A financial services organization handling sensitive financial records required consistent encryption governance across storage, databases, and logs. Inconsistent encryption configurations created regulatory risk.

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'. The 'Account and organization settings' section is expanded, showing 'Block Public Access settings for this account' and 'AWS Organizations settings for Storage Lens'. A green success message at the top right says 'Block Public Access settings for this account successfully updated.' The 'Block all public access' section has several options checked under 'On': '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'. The 'AWS Organizations settings for Storage Lens' section shows an 'Organization ID' of 'o-pr0spcswbk' and a 'Status' of 'Disabled'. There is also a link to 'Edit trusted access'.

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 'Buckets', 'Access management and security', 'Storage management and insights', and 'Account and organization settings'. Under 'Access management and security', the 'Encryption type' section is selected, showing 'Server-side encryption with AWS Key Management Service keys (SSE-KMS)' and an 'Encryption key ARN' of 'arn:aws:kms:us-east-1:879381257906:key/f9bc03c2-6857-4950-b1a0-c2f22628e590'. The 'Default encryption' section indicates that server-side encryption is automatically applied to new objects stored in the bucket. A note about 'Upcoming change to default encryption' states that in 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 (0)' section shows a 'Create configuration' button. There are also 'View details', 'Edit', and 'Delete' buttons for managing configurations.

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 'Encryption manager' tab. The left sidebar includes sections for Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), Load Balancing (Load Balancers, Target Groups, Trust Stores), and Auto Scaling (Auto Scaling Groups). The main content area displays the 'EBS encryption' section, which is set to 'Always encrypt new EBS volumes' (Enabled) and uses a default encryption key (arn:aws:kms:us-east-1:879381257906:key/17d94a5b-c5bb-420c-acfb-7c6db3c12a42). Below this is the 'Block public access for AMIs' section, which shows 'Public access' is blocked and there are currently no publicly shared AMIs. The final section is 'Block public access for EBS snapshots', which also shows 'Public access' is blocked.

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 for managing keys. On the left, there's a navigation sidebar with 'Key Management Service (KMS)' selected under 'Customer-managed keys'. The main area displays the configuration for a key named 'securedataKMSKEY'. The 'General configuration' section includes fields for 'Alias' (securedataKMSKEY), 'Status' (Enabled), 'ARN' (arn:aws:kms:us-east-1:879381257906:key/f9bc03c2-6857-4950-b1a0-c2f22628e590), 'Description' (empty), 'Creation date' (Dec 21, 2025 12:59 GMT), and 'Regionality' (Single region). Below this, the 'Key policy', 'Cryptographic configuration', 'Tags', and 'Key material and rotations' tabs are present, with 'Key material and rotations' being the active tab. Under 'Key material and rotations', the 'Automatic key rotation' section is shown with 'Status' (Enabled) and 'Rotation period' (365 days). The 'Date of last automatic rotation' is listed as '-' and the 'Next rotation date' is 'Dec 21, 2026'. There's also an 'Edit' button and a 'Rotate now' button. A large watermark 'AWO' is overlaid on the right side of the screenshot.

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

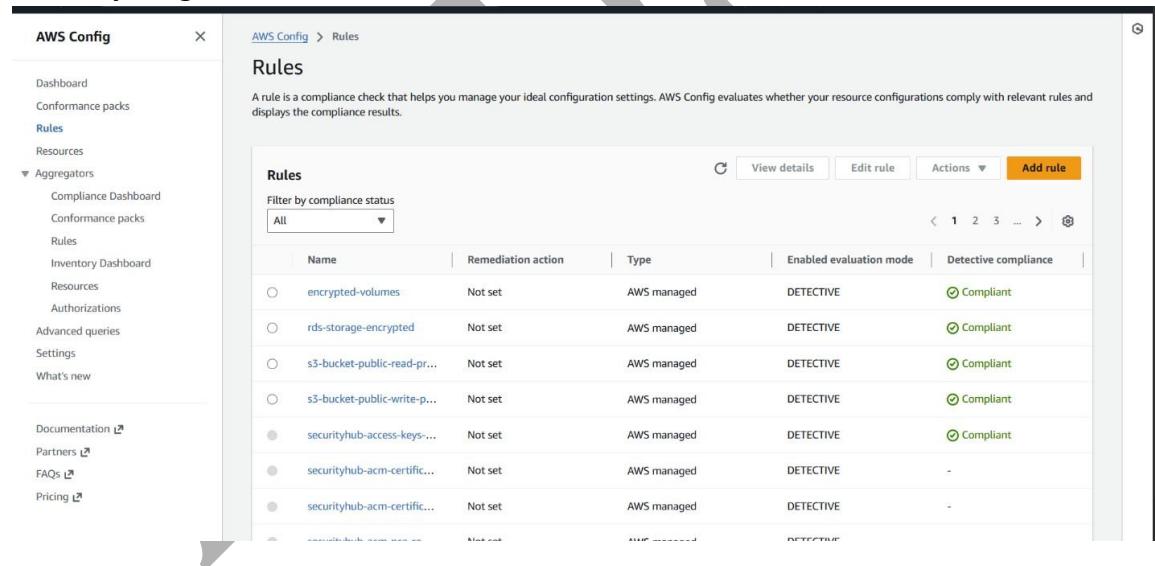
This process ensured secure rotation without service disruption.

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 has a breadcrumb trail: AWS Config > Rules. A large title 'DAYO' is overlaid on the top right of the page. The central part is a table titled 'Rules' with columns: Name, Remediation action, Type, Enabled evaluation mode, and Detective compliance. The table lists several rules, all of which are marked as 'Compliant'. The 'Enabled evaluation mode' column shows 'DETECTIVE' for all rules.

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.

The screenshot shows a Gmail inbox with the following details:

- Inbox:** 221 messages
- Starred:** 0
- Snoozed:** 0
- Sent:** 0
- Drafts:** 91 messages
- Purchases:** 25 messages
- More:** 0
- Labels:** +
- Unwanted:** 0

AWS Notifications (4 messages):

- 8:47PM (33 minutes ago) [Redacted]
- 8:47PM (33 minutes ago) [Redacted]
- 8:48PM (32 minutes ago) [Redacted]
- 8:53PM (27 minutes ago) [Redacted]

Each message is a JSON object representing an AWS CloudTrail event. The last message is partially visible:

```
[{"version": "0", "id": "502670f4-0094-8db1-070d-fe63849fe527", "detail-type": "Config Configuration Item Change", "source": "aws.config", "account": "879381257906", "time": "2026-01-06T20:53:06Z", "region": "us-east-1", "resources": []}, {"detail": {"recordVersion": "1.3", "messageType": "ConfigurationItemChangeNotification", "configurationItemDiff": {"changedProperties": [{"Configuration.blockPublicAccess": {"previousValue": false, "updatedValue": true, "changeType": "UPDATE"}, "Configuration.ignorePublicACIs": {"previousValue": false, "updatedValue": true, "changeType": "UPDATE"}, "Configuration.restrictPublicBuckets": {"previousValue": false, "updatedValue": true, "changeType": "UPDATE"}, "notificationCreationTime": "2026-01-06T20:53:06.135Z", "configurationItem": {"relatedEvents": [], "relationships": []}, "configuration": {"blockPublicACIs": true, "ignorePublicPolicy": true, "restrictPublicBuckets": true}, "supplementaryConfiguration": []}, "tags": [{"configurationItemVersion": "1.3", "configurationItemCaptureTime": "2026-01-06T20:53:04.794Z", "configurationStaleId": "1.767732784704E12", "awsAccountId": "879381257906", "configurationItemStatus": "OK", "resourceType": "AWS-S3 Account PublicAccessBlock", "resourceId": "879381257906", "awsRegion": "us-east-1", "availabilityZone": "NotApplicable", "configurationItemMd5Hash": ""}}]}}
```

Below the inbox are standard Gmail controls: Reply, Forward, and a trash icon.

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.

ADDEDAYO