Securing Sensitive Data in a Financial Application with AWS KMS and Cloud HSM

Team Name: DataDefenders

Takala

Awa: CloudTrail

Jaelin

Mariana: Security Validation Checklist; Knowledge Transfer: Key Management Operations Guide + Training Materials

KMS CloudTrail Integration

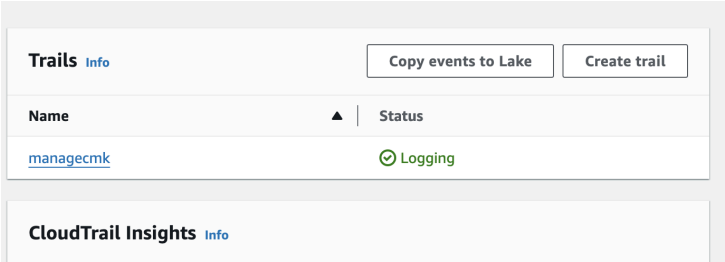
1. **CloudTrail Configuration Guide**: Provide a step-by-step guide on how to configure CloudTrail to log all AWS KMS actions, including:
    - Creating a dedicated CloudTrail trail for KMS events
    - Specifying the S3 bucket and KMS key for log file delivery and encryption
    - Enabling log file validation to ensure integrity
    - Configuring CloudWatch Logs integration for real-time monitoring
2. **CloudTrail Event Parsing Script**: Develop a script or tool to parse and analyze CloudTrail logs for KMS events, extracting relevant information such as:
    - Key usage (encrypt, decrypt, generate data key)
    - Access attempts (successful and failed)
    - Key management operations (create, disable, delete, rotate)
    - User identities and source IP addresses
3. **CloudTrail Log Analysis Report**: Provide a sample report demonstrating how to interpret and analyze CloudTrail logs for KMS events, including identifying potential security incidents or misconfigurations.

Setting up AWS CloudTrail for AWS KMS involves creating a dedicated trail, specifying an S3 bucket for logs with encryption, enabling log validation, and integrating with CloudWatch Logs for real-time monitoring. Developing a script to analyze CloudTrail logs helps extract key usage, access attempts, management operations, and user identities.

Step-by-Step Guide for CloudTrail Configuration

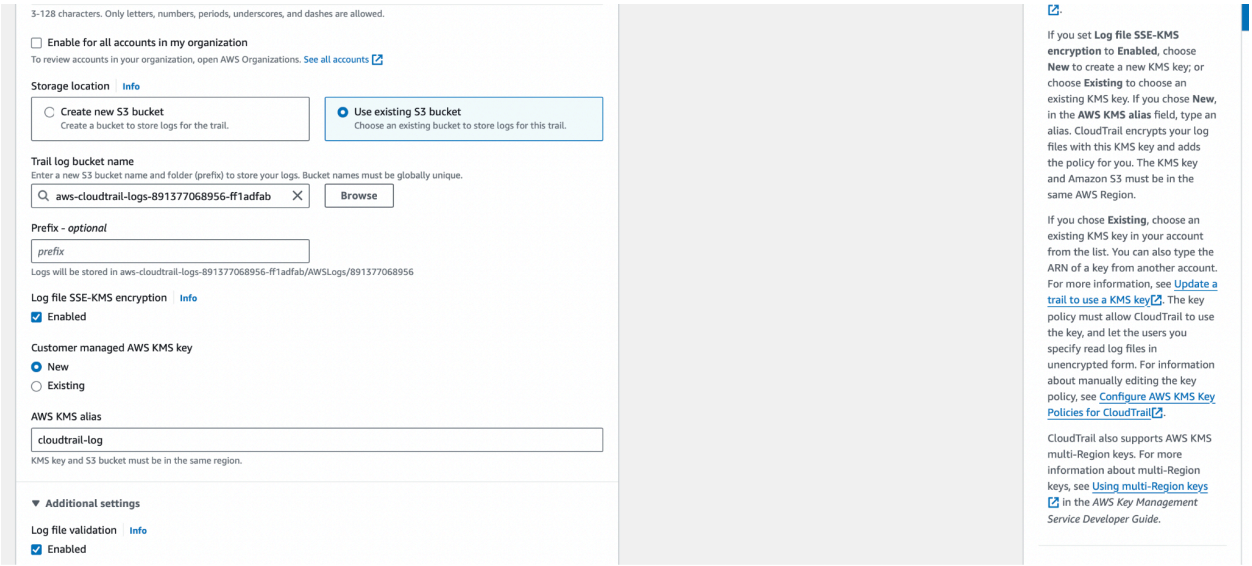
1. Setting Up a Dedicated CloudTrail Trail for KMS Events

1. **Log in to AWS Management Console:**
    - Navigate to the AWS CloudTrail console.
2. **Create a New Trail:**
    - Click on "Trails" in the left-hand menu.
    - Click on "Create trail".

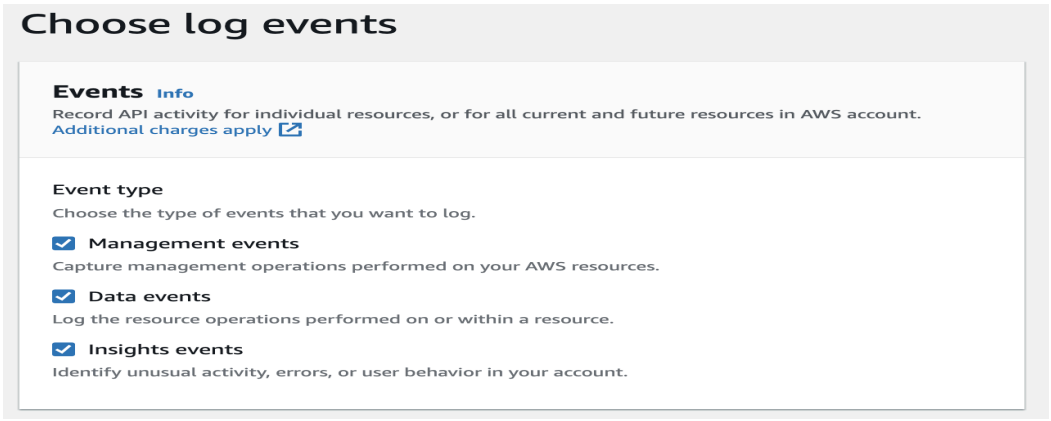


    - Enter a trail name (e.g., "KMS-CloudTrail").
    - Choose "Create a new S3 bucket" or select an existing bucket where you want CloudTrail to store logs.
        i. Here I used an existing bucket as seen below.

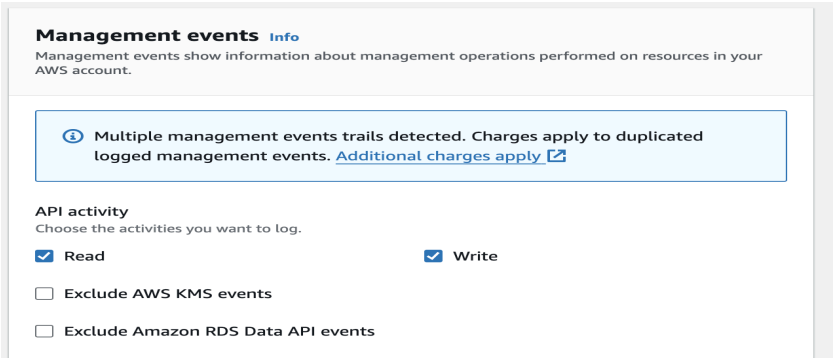
ii. Also enabled CloudTrail log encryption and created a new cmk KMS key named cloudtrail-log for encryption of logs.



3. **Configure Trail Settings:**

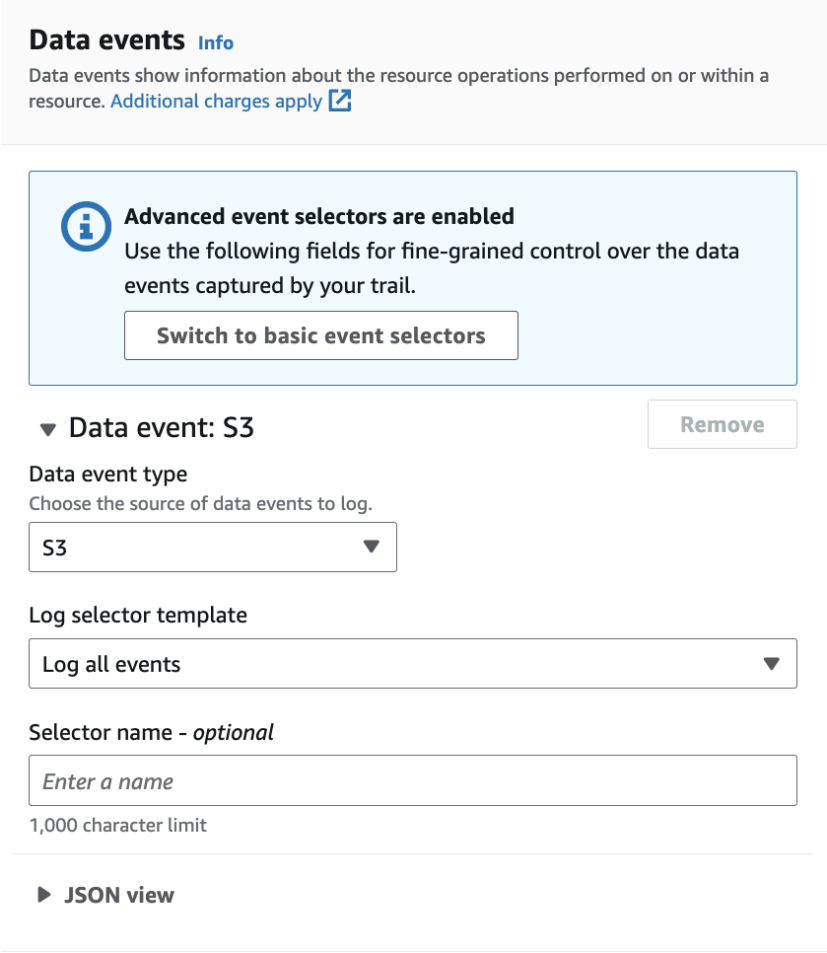


-
- Under "Management events", select "Read/Write events" to capture all KMS API actions.



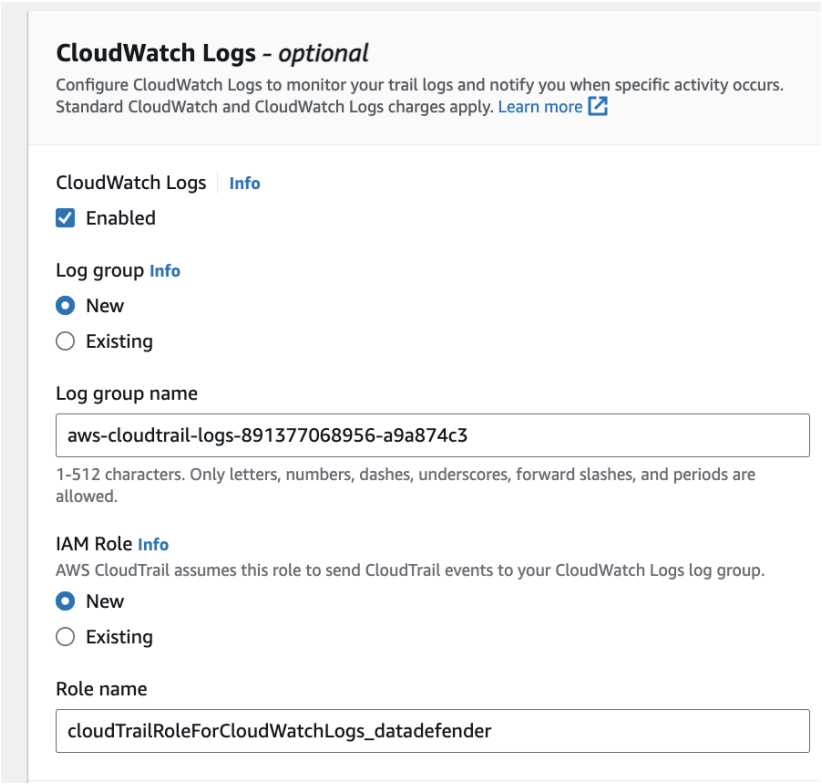
- ***Optionally, you can specify data events (e.g., KMS key usage, including encrypt/decrypt) depending on your requirements.***

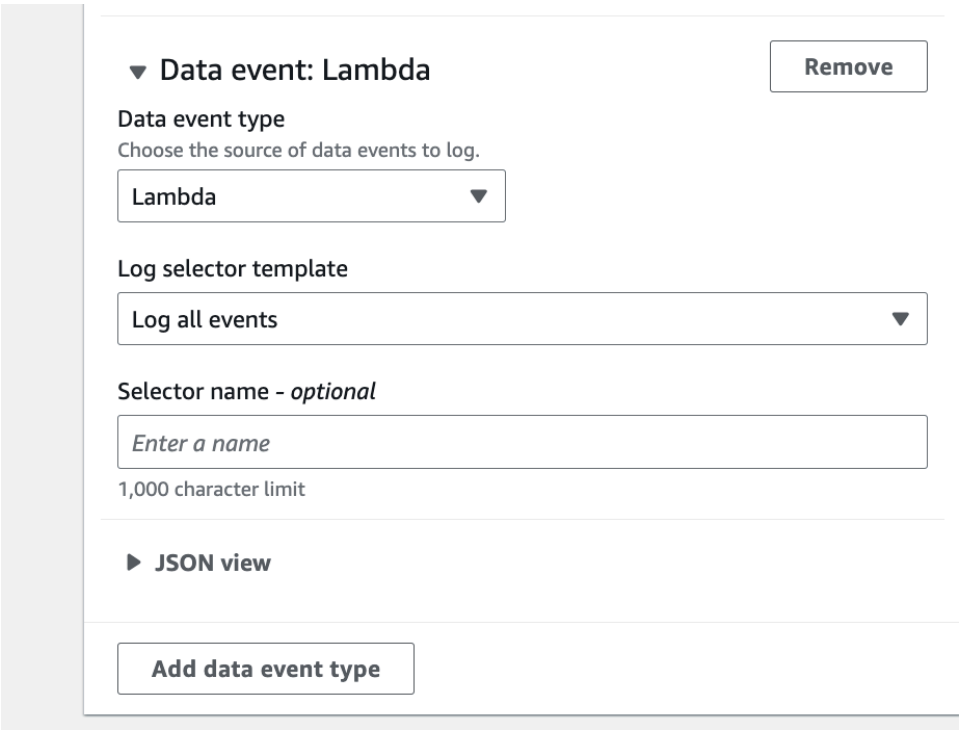
4. **Choose Log File Encryption:**
   - Select the KMS key to encrypt your log files stored in S3 for added security.

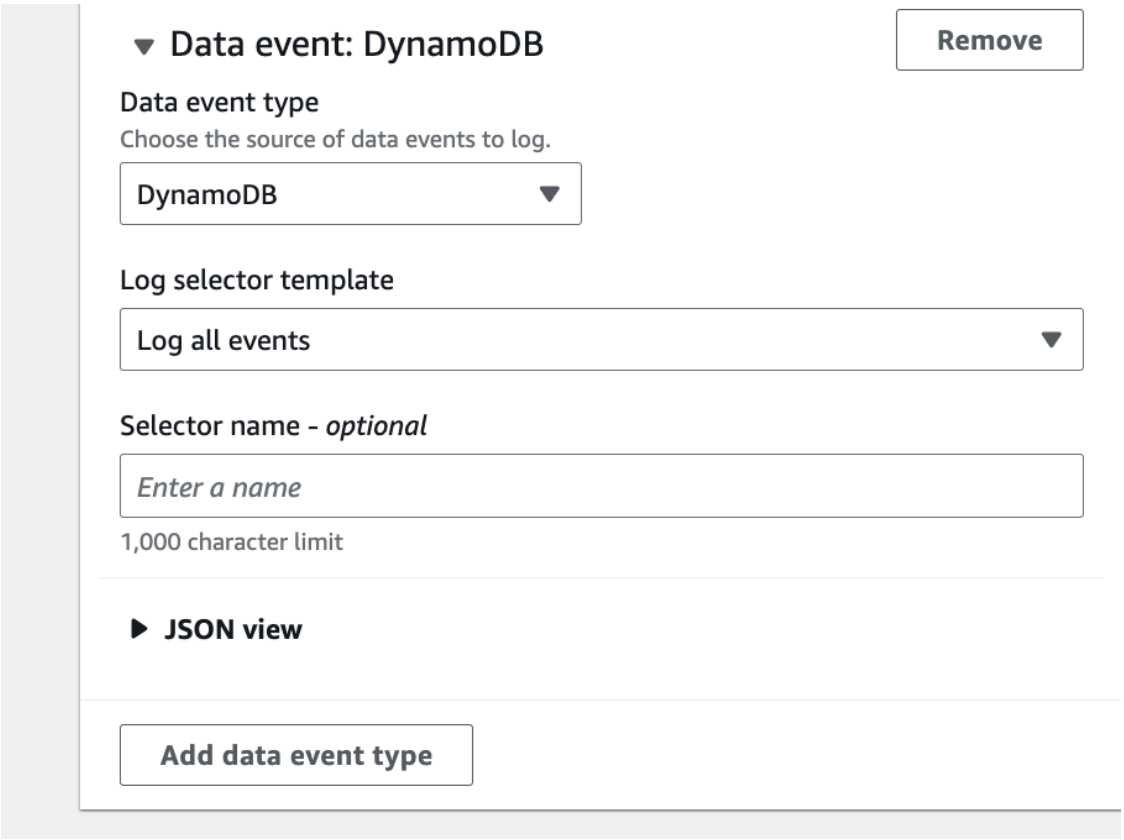


5. **Enable Log File Validation:**
   - Enable log file validation to ensure the integrity of your CloudTrail logs.

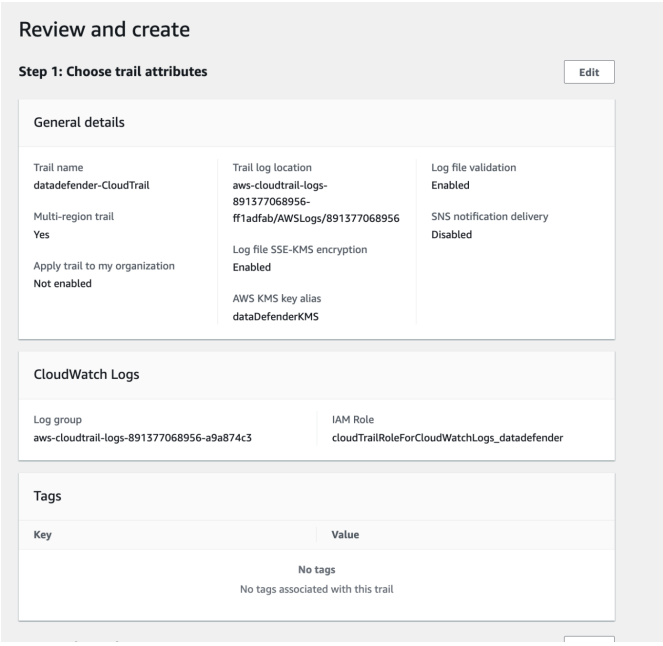
6. **Set up CloudWatch Logs Integration:**
    - Choose an existing CloudWatch Logs group or create a new one to receive CloudTrail logs for real-time monitoring.



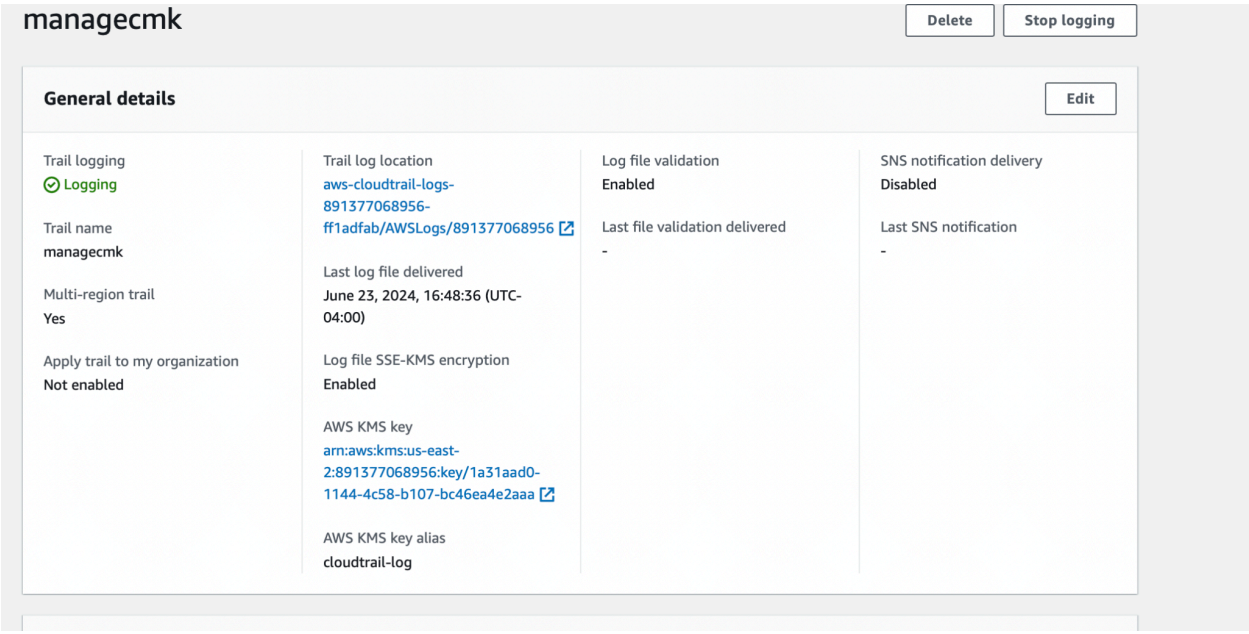


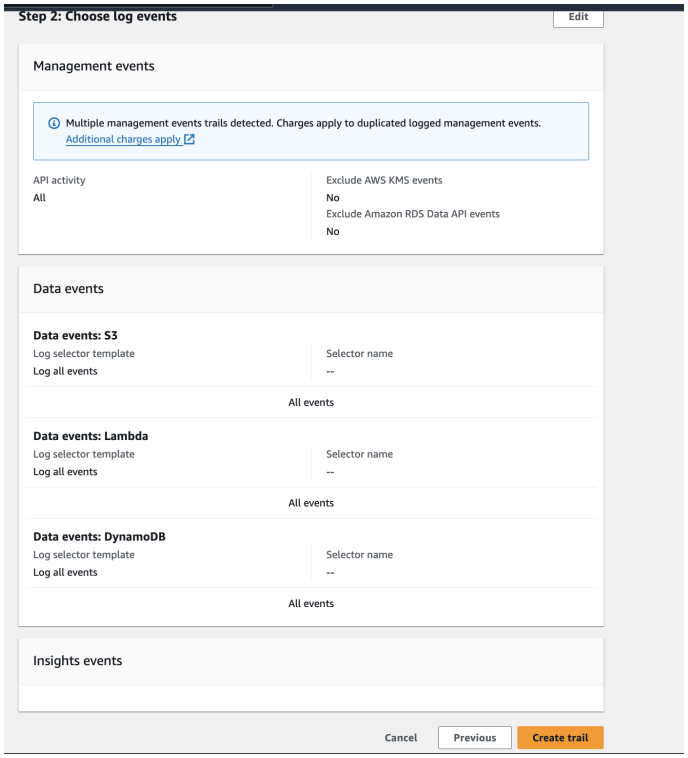


7. **Review and Create:**



- Review your settings and click "Create trail" to activate the CloudTrail configuration.





2. Developing a CloudTrail Event Parsing Script

To parse and analyze CloudTrail logs for KMS events, you can use AWS CLI commands or AWS SDKs (e.g., boto3 for Python). Here's a basic outline of what the script should do:

- **Retrieve CloudTrail Logs:**
  - Use AWS CLI (`aws cloudtrail lookup-events`) or SDK to fetch CloudTrail logs.
- **Parse Event Data:**
  - Extract relevant fields such as event name, event time, user identity, source IP, and details about KMS key operations (e.g., key usage, management actions).
- **Filter KMS Events:**
  - Filter events related to KMS actions (e.g., `Encrypt`, `Decrypt`, `GenerateDataKey`, `CreateKey`, `DisableKey`, `DeleteKey`, `RotateKey`).
- **Store or Analyze Data:**
  - Store parsed data in a structured format (e.g., JSON) or analyze it directly for security auditing or monitoring purposes.

3. Sample CloudTrail Log Analysis Report

A CloudTrail log analysis report should provide insights into KMS events, highlighting potential security incidents or misconfigurations. Here's an example format:

Conclusion

By following these steps, you can configure AWS CloudTrail to effectively log KMS events, develop a script to parse and analyze these logs, and create reports to monitor and audit KMS key usage within your AWS environment. This approach helps in maintaining security and compliance with logging and monitoring best practices.

Security Automation for Key Management

1. **AWS Lambda Function Code**: Develop AWS Lambda functions to automate security checks and alerts for key management, such as:
    - Monitoring for disabled or compromised keys
    - Detecting unusual key usage patterns (e.g., high volume, unexpected sources)
    - Validating key rotation and expiration policies
    - Checking for insecure key configurations (e.g., lack of key rotation, permissive key policies)
2. **Lambda Deployment Guide**: Document the process for deploying and configuring the Lambda functions, including:
    - Creating the necessary IAM roles and policies
    - Setting up event sources (e.g., CloudWatch Events, S3 events)
    - Configuring Lambda environment variables and function settings
3. **Alert and Notification Configuration**: Provide instructions on how to configure alerts and notifications for the Lambda functions, such as:
    - Setting up Amazon SNS topics for email or SMS notifications

    - Integrating with third-party notification services (e.g., PagerDuty, Slack)
    - Defining alert thresholds and conditions

Penetration Testing & Security Validation

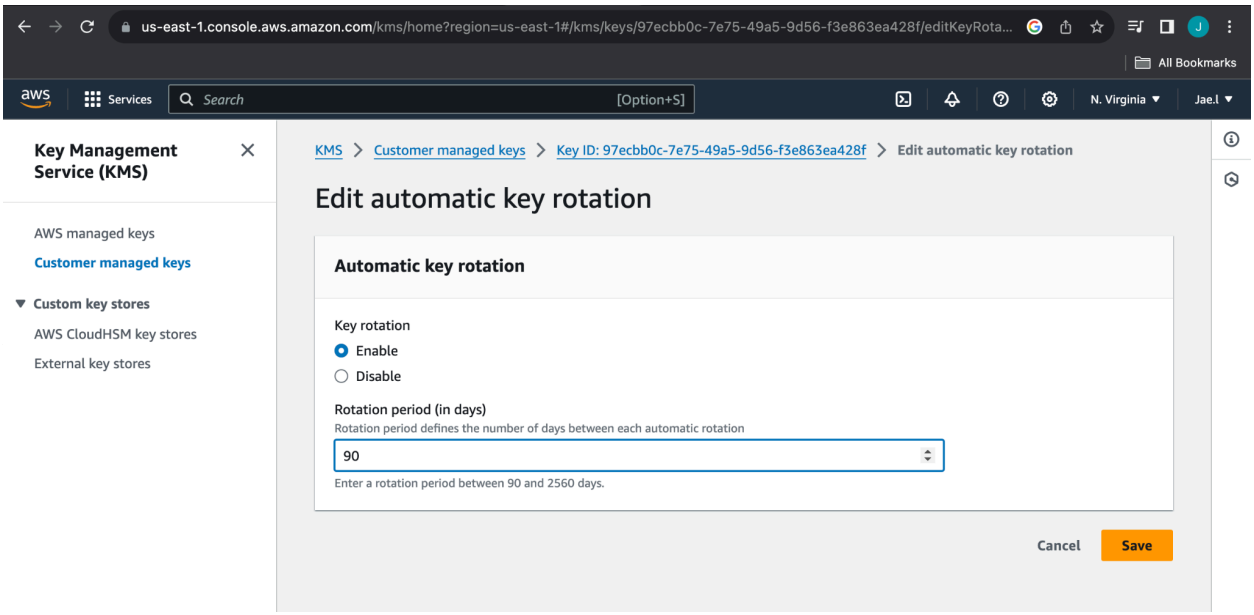
1. **Penetration Testing Plan**: Develop a comprehensive plan for conducting penetration testing on the key management infrastructure, including:
    - Scope and objectives
    - Testing methodologies (e.g., black-box, gray-box, white-box)
    - Testing tools and techniques
    - Reporting and remediation processes
2. **Penetration Testing Report**: Document the findings and results of the penetration testing, including:
    - Identified vulnerabilities and their severity
    - Recommendations for remediation and mitigation
    - Evidence of successful exploitation attempts (if applicable)

3. **Security Validation Checklist**: Create a checklist to validate the effectiveness of the data encryption strategy and key management practices, covering aspects such as:

    - Key rotation and expiration policies

A best practice is to enable key rotation and configure a rotation period; this needs to be done after creation of a customer managed KMS key as this is disabled by default.

Here we set a rotation period of 90 days.



Here, our key is enabled to have IAM user permissions and has access for key administrators to use the designated key. (see below)

Tags

Key	Value
datadefenders	cmk

Key policy

To change this policy, return to previous steps or edit the text here.

```
{
    "Id": "key-consolepolicy-3",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::891377068956:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "Allow access for Key Administrators",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::891377068956:role/aws-reserved/sso.amazonaws.com
```

- Encryption context and key usage patterns

Key policy for encryption of our CloudTrail logs for KMS events/activity:

Key policy | Cryptographic configuration | Tags | Key rotation | Aliases

Key policy Edit

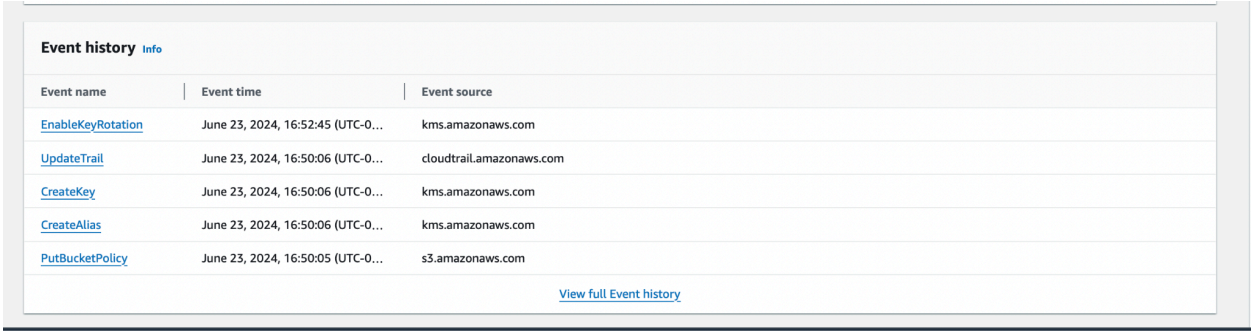
```
{
    "Version": "2012-10-17",
    "Id": "Key policy created by CloudTrail",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:sts::891377068956:assumed-role/AWSReservedSSO_AdministratorAccess_877c90735e14ae2f/Mariana-TKH",
                    "arn:aws:iam::891377068956:root"
                ]
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "Allow CloudTrail to encrypt logs",
            "Effect": "Allow",
            "Principal": {
                "Service": "cloudtrail.amazonaws.com"
            },
            "Action": "kms:GenerateDataKey*",
            "Resource": "*",
            "Condition": {
```

The KMS key (seen above) that we designated for encryption of CloudTrail logs tracking KMS events/activity has a key policy configured with principals being only the AWS SSO administrative access role (provisioned through IAM) of Mariana-TKH along with the root user (which is an account that should generally not be used for any regular operations so in effect, only the Mariana-TKH SSO identity role will have access to this key used to encrypt CloudTrail logs for KMS activity).

Another important part of monitoring KMS keys involves manually monitoring key usage/status/activity. You can customize the AWS managed keys and Customer managed keys pages of the AWS KMS console to display the following information about each KMS key:

- Key ID
- Status
- Creation date
- Expiration date (for KMS keys with imported key material)
- Origin
- Custom key store ID (for KMS keys in custom key stores)

We can see and monitor KMS events and key usage patterns from our CloudTrail dashboard.



Event history Info

Event name	Event time	Event source
EnableKeyRotation	June 23, 2024, 16:52:45 (UTC-0...	kms.amazonaws.com
UpdateTrail	June 23, 2024, 16:50:06 (UTC-0...	cloudtrail.amazonaws.com
CreateKey	June 23, 2024, 16:50:06 (UTC-0...	kms.amazonaws.com
CreateAlias	June 23, 2024, 16:50:06 (UTC-0...	kms.amazonaws.com
PutBucketPolicy	June 23, 2024, 16:50:05 (UTC-0...	s3.amazonaws.com

View full Event history

- Compliance with industry standards and regulations

We set up and integrated CloudHSM clusters and key stores for use with our AWS KMS keys.

This allows us to align with enterprise security standards by using AWS CloudHSM to manage private keys that protect highly confidential data. The HSMs provided by AWS CloudHSM are FIPS 140-2 level 3 certified and comply with PCI DSS. Additionally, AWS CloudHSM is PCI PIN compliant and PCI-3DS compliant. As such, by setting up CloudHSM clusters integrated with our AWS KMS keys, we can ensure our cryptographic keys and operations that protect our sensitive data are compliant with industry standards and regulations.

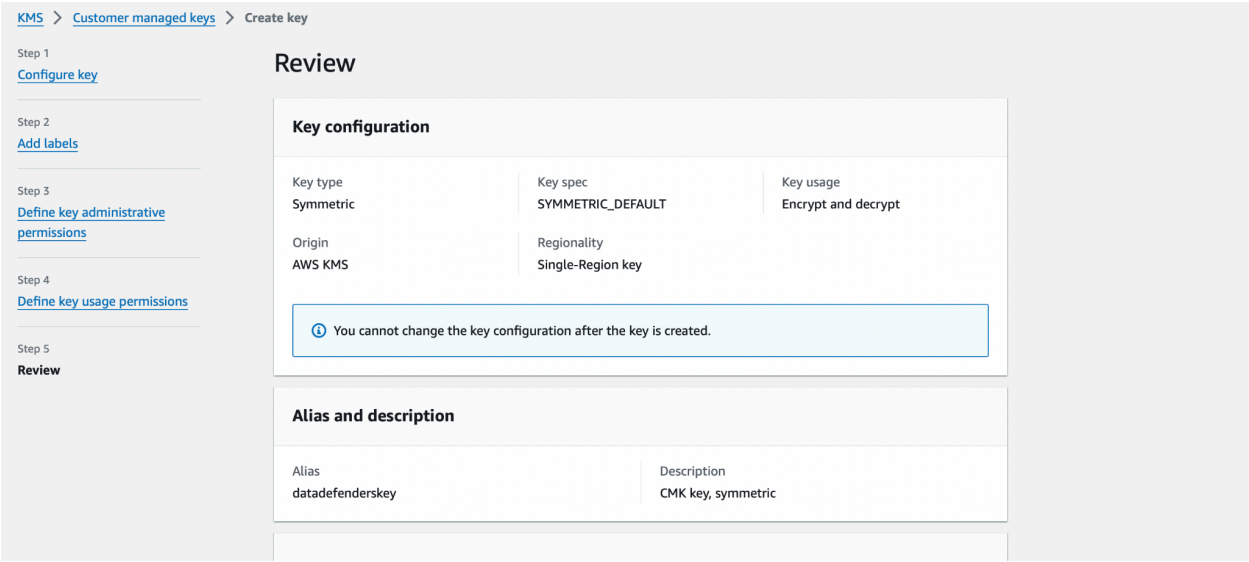
Knowledge Transfer & Handover Package

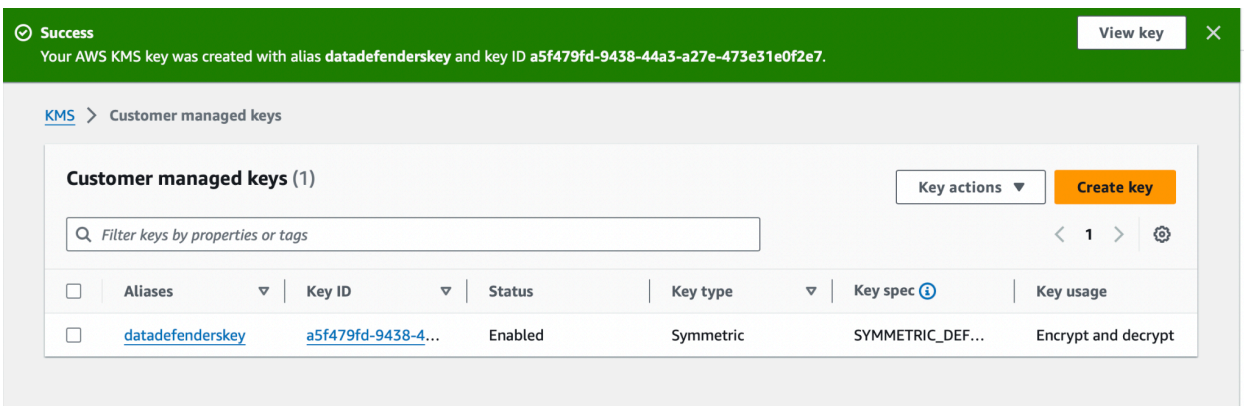
1. **Key Management Operations Guide:** Develop a comprehensive guide for managing and maintaining the key management infrastructure, including:
    - Key lifecycle management (creation, rotation, deletion)

Key Creation:

Configure your key:

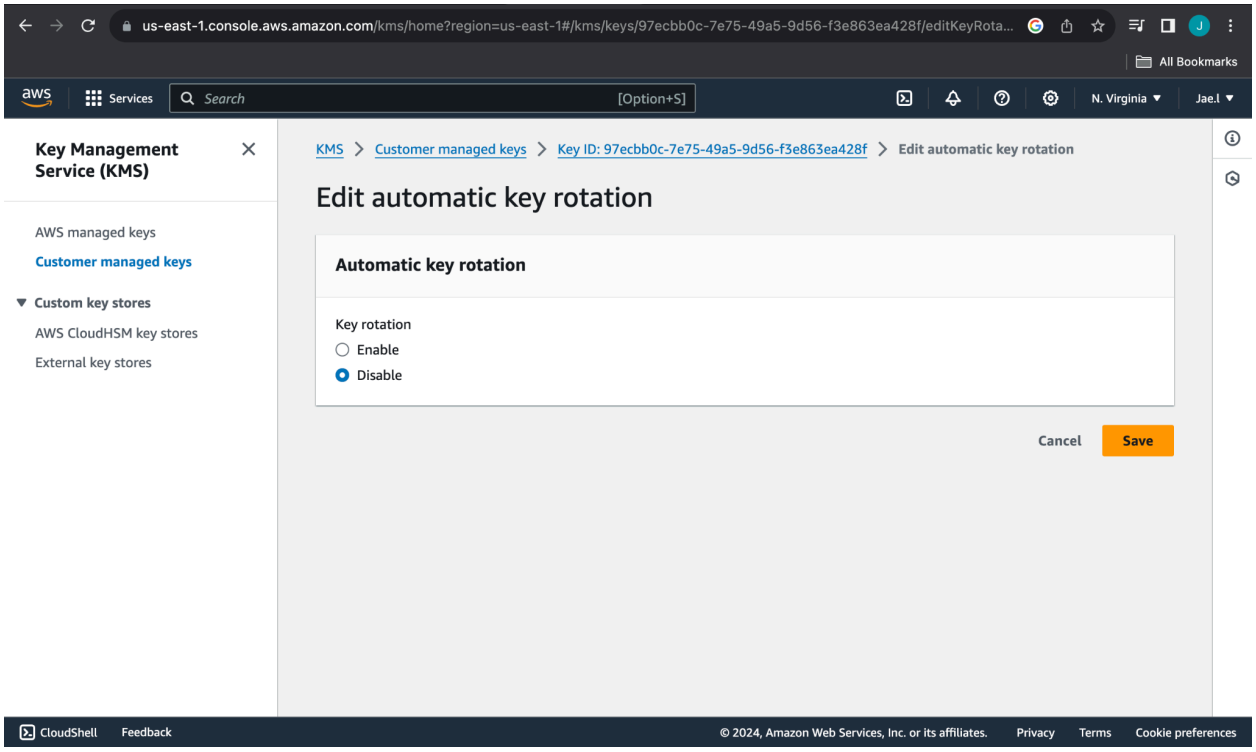
- Set key type such as symmetric or asymmetric
- Give your key a name
- Set your key to be usable for single or multi-region
- Set key's usage for either encryption operations, decryption operations, or both.
- Review your key configuration.



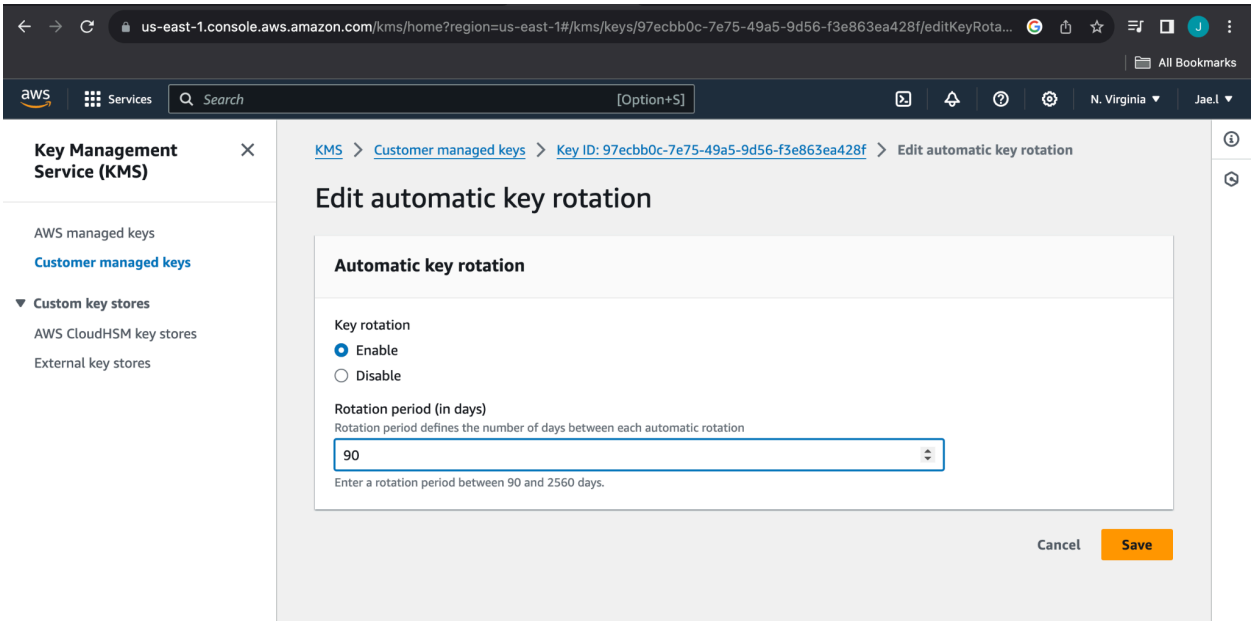


Key Rotation:

Key rotation isn't automatically enabled by default in AWS KMS so you need to edit this after creating a key. It is actually disabled as seen below:



We enable automatic key rotation as a best practice and then set the rotation period in days. The most frequent rotation period you can set is a 90 day period (aka keys will be rotated every 3 months, 90 days to be exact).



- Key policy management and access control

When using cryptographic keys and performing encryption, you need a mechanism for managing access to keys that is different from the one you use for managing access to your data.

As such, a best practice for key management is effectively ensuring separation of duties by assigning one set of administrators who can only manage our keys and a different set of administrators who can only manage access to the underlying encrypted data. Configuring the key management process like this helps us achieve a separation of duties needed to avoid accidentally escalating privilege to decrypt data to unauthorized users.

For even further separation of control, AWS CloudHSM offers an independent policy mechanism to define access to keys.

- Backup and disaster recovery procedures

AWS CloudHSM helps organizations meet corporate, contractual, and regulatory compliance requirements for data security. CloudHSM is a fully-managed service that automates time-consuming administrative tasks for AWS customers, such as hardware provisioning, software patching, high-availability, and backups.

AWS CloudHSM can be used to generate and use encryption keys using FIPS 140-2 Level 3 validated hardware security modules (HSMs) on the AWS Cloud. CloudHSM integrates with client applications using industry-standard APIs, such as PKCS#11, Java Cryptography Extensions (JCE), and Microsoft CryptoNG (CNG).

- Troubleshooting and incident response processes

For security monitoring, threat detection, and incident response capabilities, we can use AWS CloudTrail.

AWS KMS can be used with AWS CloudTrail to log the use of organizational KMS keys. This is used to audit key usage and ensure compliance. Additionally, it can be used to increase visibility of KMS activity to provide additional security and detection capabilities. By default, all AWS KMS actions are logged as CloudTrail events. If a customer wants, they can exclude AWS KMS actions from a CloudTrail trail but that requires further action in changing a trail’s configuration.

CloudTrail logs all AWS KMS actions, including read-only, such as ListAliases and GetKeyRotationStatus, operations that manage KMS keys, such as CreateKey and PutKeyPolicy, and cryptographic operations, such as GenerateDataKey and Decrypt.

Additionally for security reasons, there are fields omitted from AWS KMS log entries, such as the Plaintext parameter of an Encrypt request, and the response to GetKeyPolicy or any cryptographic operation.

AWS KMS adds the key ARN of the affected KMS key to the responseElements field in CloudTrail log entries for some AWS KMS key management operations, even when the API operation doesn't return the key ARN. This key ARN is included by KMS to certain CloudTrail log entries for key management events to make it easier to search for log entries for specific KMS keys. As such, in the event of an incident or if needing to troubleshoot a specific KMS key, we are able to audit our CloudTrail logs efficiently to see events, activity history, and actions involving that specific key.

2. **Training Materials**: 🗀 DATA DEFENDERS TRAINING MATERIALS
    - Create training materials, such as presentations, videos, or interactive tutorials, covering topics like:
        - AWS KMS and CloudHSM fundamentals
        - Key management best practices
        - Security monitoring and incident response
        - Compliance and regulatory requirements

*** 🗀 DATA DEFENDERS TRAINING MATERIALS

Access to training materials presentation provided to George Robbins and Emilie Dionisio by Mariana Espinoza.

***Text Link: https://docs.google.com/presentation/d/1yTPhugCsB3Y7matzpMswg2XIHXvvzVRPreX0mo7iAVw/edit?usp=sharing

3. **Source Code Documentation**: Document the source code for any custom scripts, tools, or automation developed during the project, including:
    - Inline comments and code documentation
    - Architecture diagrams and data flow diagrams

    - API documentation (if applicable)
4. **Handover Checklist**: Prepare a handover checklist to ensure a smooth transition of the key management infrastructure to the cloud security and data security teams, including:
    - Outstanding tasks or issues
    - Future enhancement plans
    - Contact information for support and escalation

Glossary

Access Control Mechanisms: Mechanisms to control who or what can access data and resources.

Alert: A notification that something important has happened, such as a potential security threat.

API (*Application Programming Interface*): A set of instructions and standards that allow applications to communicate with each other.

Architecture diagram: A visual representation of the components of a system and how they interact.

AWS (*Amazon Web Services*): A cloud computing provider offered by Amazon, providing a vast collection of on-demand IT resources and services over the internet.

Backup: Creating a copy of your data or system that can be used to recover it in case of an outage or disaster.

Best practice: A recommended way to perform a task or configure a system that is considered secure and effective.

CloudHSM (*Hardware Security Model*): A specialized physical device that safeguards and manages encryption keys, often used for high-security applications.

Cloud Providers: Companies offering cloud computing services.

CloudTrail: An AWS service that logs API calls made to your AWS account. This can help you track who is doing what in your account and identify any suspicious activity.

CloudTrail event: A single entry in a CloudTrail log file that represents an API call made to your AWS account.

CloudTrail trail: A configuration within CloudTrail that specifies which AWS API calls to log and where to store the logs.

CloudWatch Logs: An AWS service that allows you to store, monitor, and analyze log files from your AWS resources.

CMK (Customer Managed Key): An encryption key that you create and manage yourself within AWS KMS.

Compliance: Adhering to regulations and standards.

Data Classification: Categorizing data by sensitivity levels to implement appropriate security measures.

Data flow diagram: A visual representation of how data moves through a system.

Data key: A key used to encrypt and decrypt specific data objects. These keys are often generated and managed by KMS.

Decryption: The process of transforming encrypted data back into its original form using a decryption key.

Disaster recovery: The process of recovering your data and systems after a disaster or outage.

DynamoDB: Fully managed, serverless, NoSQL key-value database offered by AWS, designed to provide fast, predictable performance with seamless scalability.

Encryption: The process of scrambling data to make it unreadable without a decryption key.

Encryption context: Additional information associated with an encryption operation that can be used for security or auditing purposes.

Encryption Key: A string of code used to encrypt and decrypt data.

Financial application: A software application used to manage financial transactions and data.

Handover checklist: A list of tasks that need to be completed to transfer ownership or responsibility for something.

Handover Package Terms: Refer to Key lifecycle management, Key creation, Key rotation, Key policy management.

IAM (*Identity and Access Management*): A framework of policies, processes and technologies that help organizations manage digital identities and control what users can do within their computer systems or online applications.

IAM Roles: A way to grant permissions to access AWS services and resources specifying what can or cannot be performed within an AWS account, allowing for enhanced security, flexibility, and service integration between AWS and other services and resources.

Incident response: The process of identifying, containing, and recovering from a security incident.

KMS (*Key Management Service*): A cloud-based service for generating, storing, and managing encryption keys securely.

Key access control: Who has permission to use a KMS key and what they can do with it.

Key lifecycle management: The process of creating, rotating, and deleting KMS keys throughout their lifetime.

Key policy: A set of rules that determines who can use a KMS key and how they can use it.

Key policy management: Creating and maintaining the policies that control who can use a KMS key and how they can use it.

Key rotation: Regularly changing the encryption key used to protect your data to mitigate the risk of someone compromising the key.

KMS key: A digital key used to encrypt and decrypt data. There are two main types: AWS managed keys (pre-configured by AWS) and customer managed keys (created and managed by you).

Lambda function: A serverless compute service offered by AWS that lets you run code without provisioning or managing servers.

Log file encryption: Encrypting log files to protect sensitive information they may contain, like user activity or system events.

Log file validation: Ensuring the integrity of log files by verifying they haven't been tampered with.

Misconfiguration: An incorrect or insecure setting on a system or service.

Notification: A message sent to inform someone about something, like an alert or update.

Outstanding task: A task that has not yet been completed

Penetration Testing (*Pen Testing*): The practice of simulating a cyber attack to identify vulnerabilities in your systems and security posture.

Penetration Testing Terms: Refer to Scope, Objective, Methodology.

Regulatory requirement: A rule or standard that you must comply with, often set by a government or industry body.

Remediation: The process of fixing a vulnerability.

S3 bucket (*Simple Storage Service bucket*): A storage location within Amazon S3 for storing objects (like files) in the cloud.

Security incident: An event that compromises the confidentiality, integrity, or availability of your data or systems.

Security monitoring: Continuously monitoring your systems and data for security threats.

Security Validation Terms: Refer to Key usage pattern, Key access control, Encryption context.

Sensitive Data: Information that, if compromised, could cause harm or damage.

SNS topic (*Simple Notification Service topic*): A service that allows you to send notifications to multiple subscribers, like email addresses or other services.

Source code documentation: Comments and explanations added to code to make it easier to understand and maintain.

Tamper-Resistant: Describes a device designed to resist physical attacks aimed at extracting sensitive information.

Threat Modeling: Identifying potential threats and developing countermeasures.

Training Material Terms: Refer to Best practice.

Vulnerability: A weakness in a system or service that can be exploited by an attacker.

Vulnerability Assessment: Process of identifying security weaknesses in a system.