AWS Key Management Service (KMS)

Overview

- AWS KMS is a managed service that creates and controls encryption keys. It performs cryptographic operations on sensitive data.
- Instead of manually creating, storing, rotating, and protecting encryption keys yourself, AWS does all of this securely for you.
- We can use KMS to encrypt variety of things like your Logs, API keys, DB Details, S3 Objects etc.
- Envelop Encryption is a mechanism of encrypting Data using a Key and re-encrypting the Key used for Encryption using a different key. It only manages keys, not the data.
- AWS offers us three types of **Customer Master Keys** (CMK) for flexible operations —
    - Customer Managed CMK: Keys that **we can create and fully control** in your AWS account.
        - What we can do:
            - Create and delete the key
            - Define and edit key policies
            - Enable or disable the key
            - Rotate cryptographic material
            - Schedule key deletion
            - Control who can use the key
        - We use it when we need maximum control, security, and compliance.
        - Example: Encrypting production databases or sensitive customer data.

    - AWS Managed CMK: Keys that are **created and managed by AWS** on your behalf.
        - What we can do:
            - View the key in our account.
            - Use the key through AWS services.
        - What we cannot do:
            - Edit key policies
            - Rotate keys manually
            - Delete the key

AWS Key Management Service (KMS)

        - AWS services perform cryptographic operations automatically.
        - We use it when we want **encryption without managing keys**.
        - Example: Default encryption for S3, EBS, or RDS.

    - AWS Owned CMK: Keys that are **fully owned and managed by AWS**, not by our account. These keys are **shared across multiple AWS accounts**.
        - What you can do:
            - Use the service that uses the key.
        - What you cannot do:
            - View the key
            - Manage it
            - Audit it
            - Track its usage
        - We use it when we don't care about key management and just want the service to work securely.
        - Example: Encryption used internally by AWS services like CloudFront or DynamoDB (service-managed).

Why AWS KMS Is Needed

- Without KMS:
    - We must create encryption keys ourselves.
    - Securely store them.
    - Rotate them regularly. Key rotation means periodically changing the encryption key material to reduce the risk of compromise.
    - Ensure no one steals or misuses them.
- Problems solved by KMS:
    - Centralized key management.

AWS Key Management Service (KMS)

- Strong security backed by AWS HSMs. AWS HSM is a dedicated hardware device in AWS where encryption keys are stored and never exposed.
- Automatic key rotation.
- Fine-grained access control using IAM.
- Compliance (PCI-DSS, HIPAA, SOC, ISO).

Common Use-cases

- Encrypt EBS volumes – Protect EC2 disk data at rest.
- Encrypt S3 objects – Secure files stored in S3.
- Encrypt RDS databases – Protect database storage and backups.
- Application-level encryption – Encrypt sensitive fields like Aadhaar or credit card numbers.
- Secrets encryption – Encrypt credentials stored in Secrets Manager.

Key Terms

- CMK (Customer Managed Key) - Encryption key created and controlled by you.
- AWS Managed Key - AWS-created key for a service (e.g., aws/s3).
- Symmetric Key - Same key used for encrypting and decrypting (most common).
- Asymmetric Key - Public/private key pair (rare use cases).
- Key Policy - Resource policy that controls key access.
- Envelope Encryption - Data is encrypted using a data key, not the master key.
- HSM - Hardware device that securely stores keys.

Architecture Flow

- User or AWS service requests encryption.
- KMS generates a **data key.**
- Data is encrypted using the data key.
- Data key is encrypted using the KMS key.

AWS Key Management Service (KMS)

- Encrypted data + encrypted data key are stored.
- For decryption, KMS validates permissions and decrypts the data key.
- KMS never exposes the plaintext key.

Execution Steps

- Create a Customer Managed Key (CMK).
    - Open **AWS Console**
    - Go to **KMS**
    - Click **Create key**
    - Select **Symmetric**
    - Choose **Encrypt and decrypt**
    - Set key alias (example: **app-prod-key**)
    - Assign **Key Administrators**
    - Assign **Key Users**
    - Review and create key
- Encrypt an S3 Bucket with KMS.
    - Go to **S3**
    - Open your bucket
    - Click **Properties**
    - Enable **Default encryption**
    - Choose **KMS**
    - Select your CMK
    - Save changes
- Encrypt EC2 EBS Volume Using KMS
    - Launch EC2 instance
    - During storage configuration:
        - Enable **Encryption**
        - Choose **KMS key**
    - Attach IAM role to EC2 with:

AWS Key Management Service (KMS)

        - kms:Encrypt
        - kms:Decrypt
        - kms:GenerateDataKey
    - EC2 now stores encrypted data at rest.
    - EC2 app calls KMS Encrypt API.
    - KMS encrypts sensitive data.
    - Encrypted data stored in the database.
    - On read, EC2 calls Decrypt.

Best Practices

- Use **Customer Managed Keys** for production.
- Enable **automatic key rotation.**
- Follow **least privilege** in key policies.
- Separate keys by environment (dev, test, prod).
- Monitor usage using **CloudTrail.**
- Disable unused keys instead of deleting.