**Key Management Service(KMS)**

* KMS is a managed service that enables you eo easily encrypt your data
* KMS provides a highly available key storage, management and autiditing solution for you to encrypt data within your own applications and control the encryption of stored data across AWS services
* KMS allows you to centrally manage and securely store your keys. These are known as customer maaster keys or CMKs
* You can generate CMKSs in KMS, in a CloudHSM cluster, or import them from your own key management infrastructure
* These master keys are protected by hardware security modules(HSMs) and are only ever used within those modulees
* You can submit data directly to KMS to be encrypted or decrypted using these master keys
* You set usage policies on these keys that determine which user can use them to ecnrypt and decrypt data and under which conditions
* KMS is tightly integrated into many AWS services like Lambda, S3, EBS, EFS, DyanmoDB SQS etc
* KMS is integrated with AWS services and client-side toolkits that use a method known as envelope encryption to encrypt your data
* Under this method, KMS generates data keys which are used to encrypt data and are themselves encrypted using your master keys in KMS
* Data keys are not retained or managed by KMS
* AWS services encrypt your data and store an encrypted copy of the data key along with the data it protects
* When a service needs to decrypt your data they request KMS to decrypt the data key using your master key
* If a user requesting data form the AWS service is authorized to decrypt under your master key policy, the service will receive the decrypted data key from KMS with which it can decrypt the data and return it in plaintext
* All requests to use your master keys are logged in CT so you can understand who used which key under which context and when they used it
* You can control who manages and accesses keys via IAM users and roles
* You can audit the use of keys via CT
* KMS differes from Secrets Manager as its purpose-built for encryption key management
* KMS is validated by many compliance schemas
* You can perform the following key management functions in KMS
  + Create keys with a unique alias and description
  + Import your own key material
  + Define which IAM users and roles can manage keys
  + Define which IAM users and roles can use keys to encrypt and decrypt data
  + Choose to have AWS KMS automatically rotate your keys on an annual basis
  + Temporarily disable keys so they cannot be used by anyone
  + Re-enable disabled keys
  + Delete keys that you no longer use
  + Audit use of keys by inspecting logs in CT
  + Create custom key stores
  + Connect and disconnect custom key stores
  + Delete custom key stores
* Typically data is encrypted in one of the following three scenarios
  + You can use KMS APIs directly to encrypt and decrypt data using your master keys stored in KMS
  + You can choose to have AWS services encrypt your data using your master keys stored in KMS. In this case data is encrypted using data keys that are protected by your master keys in KMS
  + You can use the AWS Encryption SDK that is integrated with KMS to perform encryption within your own applications, whether they operate in AWS or not
* Custom Key Store
  + The AWS KMS custom key store feature combines the controls provided by CloudHSM with the integration and ease of use of KMS
  + You can configure your own CloudHSM cluster and authorize KMS to use it as a dedicated key store for your keys rather than the default KMS key store
  + When you create keys in KMS you can shose to generate the key material in your CloudHSM cluster. Master keys that are generated in your custom key store never leave the HSMs in the CloudHSM cluster in plaintext and all KMS operations that use those keys are only performed in your HSMs
  + In all other respects master keys stored in your custom key store are consistent with other KMS CMKs
* Key deletion
  + You can schedule a customer master key and associated metadata that you created in KMS for deletion, with a configurable waiting period from 7 to 30 days
  + This waiting period allows you to verify the impact of deleting a key on your applications and users that depend on it
  + The default waiting period is 30 days
  + You can cancel key deletion during the waiting period
* Limits:
  + You can create up to 1000 customer master keys per account per region
  + As both enabled and disabled customer master keys count towards the limit, AWS recommend deleting disabled keys that you no longer use
  + AWS managed master keys created on your behalf for use within supported AWS services do not count against this limit
  + There is no limit to the number of data keys that can be derived using a master key and used in your application or by AWS services to encrypt data on your behalf