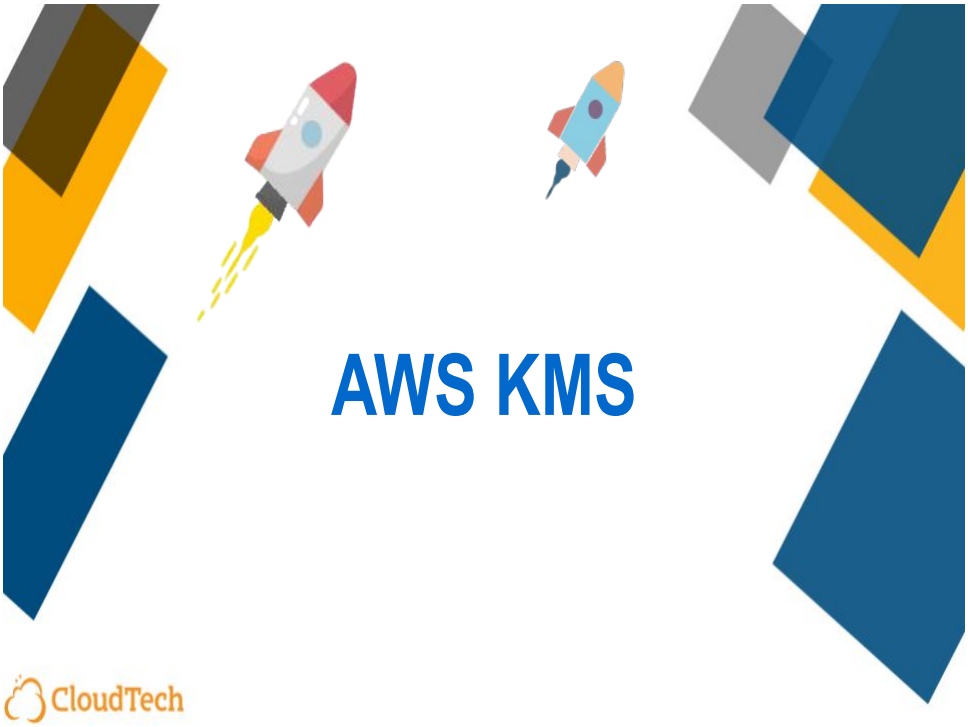



- KMSは暗号化キーの作成、管理を行うサービスです

## KMS管理の3種類のキーとCDK

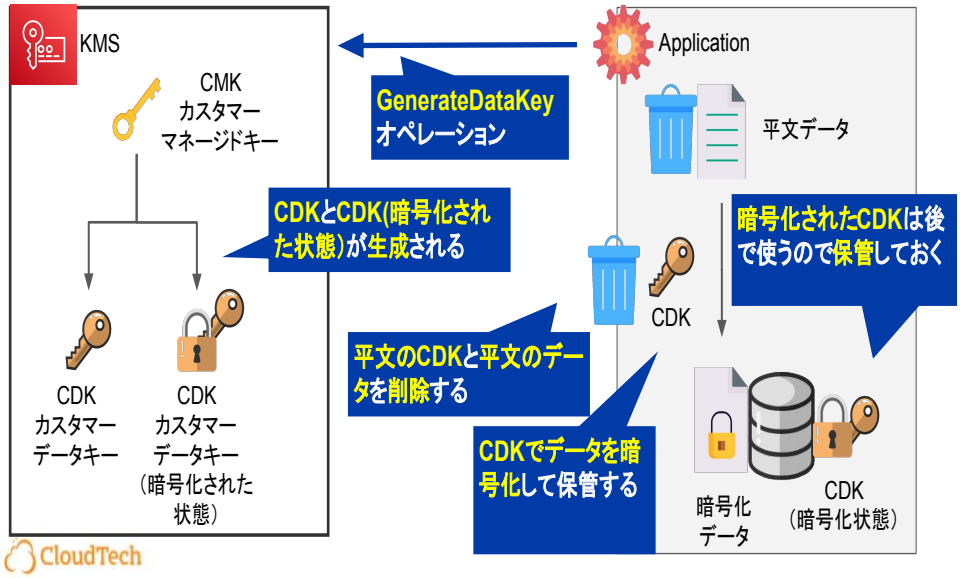

- まず、KMSには3種類のキーがあります。
  - カスタマーマネージドキー（我々利用者が管理し、我々利用者側が作成したアプリケーションで暗号化、復号する鍵です。）
  - AWSマネージドキー（AWSが管理し、AWSサービスで暗号化、復号する鍵です。）
  - AWS所有のキー
    - これはあまり気にしなくてOKです。
      - アカウント関係なく、AWSサービスの裏側で使用しているAWSサービスとして管理しているキーです。
        - 以上がKMSで管理している3種類のキーですが、
  - もうひとつ大事なキーがあります。それが「カスタマーデータキー」です。
      - CDK、またはデータキー）とも呼ばれます。

        - KMSの3種類のキーとして取り扱われていませんが、大変重要な役割があります。
- KMSの仕組みで混乱される方も多いですが、混乱しないためには、
    - カスタマーマネージドキーとカスタマーデータキーを利用した暗号化と復号のフローをしっかり理解するのが重要です。
        - 本講座も大部分の解説をここに置いています。
    - まずはこの2つのキーからみていきましょう！

# CMKとCDK
# エンベロープ暗号化

CloudTech

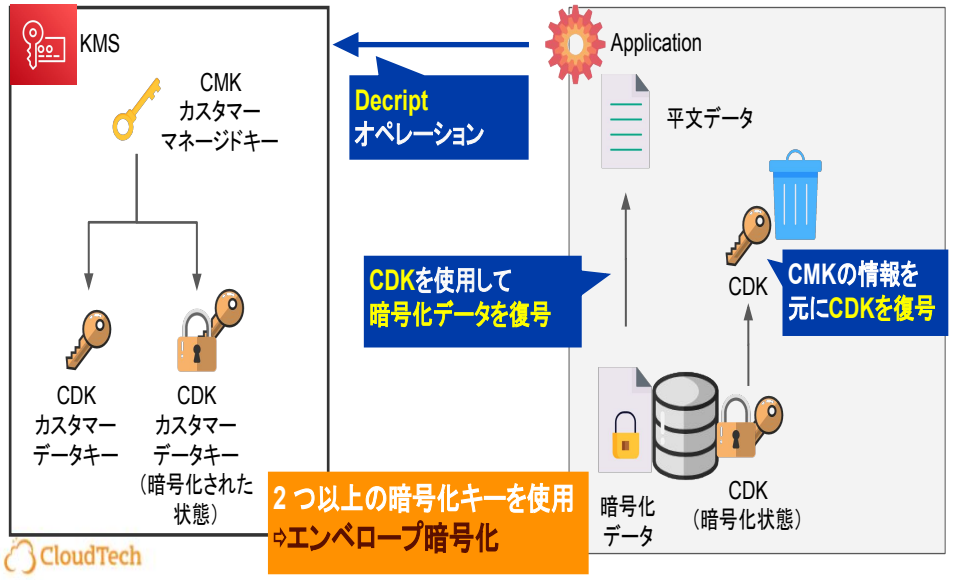
## KMS（CMKでのデータ暗号化フロー）




- カスタマーマネージドキー（CMK）は、我々利用者が管理し、我々利用者側が作成したアプリケーションで使用する鍵です。
  - 開発したアプリケーションで、平文データを暗号化して、暗号化したデータをデータベースに保管する、
    - この流れの中でCMKがどう使われるのか見ていきましょう。
- まずはじめに、AppicationがKMSサービスに対してGenerate DataKey（データキーの作成依頼）を出します
  - そうすると、CMKに紐づく、CDK（カスタマーデータキー）が作成されます。
    - KMSの3種類のキーとして取り扱われていませんが、大変重要な役割があると説明しました。
    - もうひとつ、暗号化された状態の CDKが作成されます。
      - 暗号化されているので、この CDKは何

                - の役にも立ちません。
                - たとえこれが盗まれても問題ありません、CMKがなければ復号できませので。
                    - ここまでOKですかね？ AppicationのGenerateDataKeyというAPIの実行により、
                        - CMKからCDKが2つ作成される。ひとつは暗号化されていない平文状態、もうひとつは暗号化されている。
- アプリケーションはこの 2つのCDKを取得します。
    - CMKは大切な鍵なので取得しません。安全な KMSの領域内から外には出しません。
        - 暗号化されたCDKは後の復号で使うので、保管しておきます。
            - で、平文のCDKの方で平文データを暗号化して、保管します。
- これでデータは暗号化されました。以上といきたいところですが、
    - アプリケーションの中に平文の CDKと平文データが残っています。
        - これは盗まれてしまったら大変ですので、アプリケーション側で削除処理が必要です。
            - 処理の主体を整理すると、
                - KMS側で鍵の生成
                - アプリケーション側でCDKを使ったデータの暗号化
                - 保管
                - データ削除
                    - を担当します。

## KMS（CMKでのデータ復号フロー）




- では、暗号化されたデータに対し、取り出して復号するフローを見てみましょう。
  - AppicationがKMSサービスに対してDecript（復号依頼）を出します
    - そうすると、アプリケーションは CMKの情報をもとに、保管されている暗号化 CDKを復号します。
      - このCDKでデータを復号する、といった流れです。
  - 複合したCDKは、このまま残しておくとセキュリティ上よろしくないので、削除します。

このように、2 つ以上の暗号化キーを使用すること（ CMKでCDKを暗号化する）をエンベロープ暗号化とも呼びます。
エンベロープとは封筒とも訳されますが、包み込むようなニュアンスですね。
パフォーマンスやアルゴリズムの強度などの面からメリットがあ

る手法です。

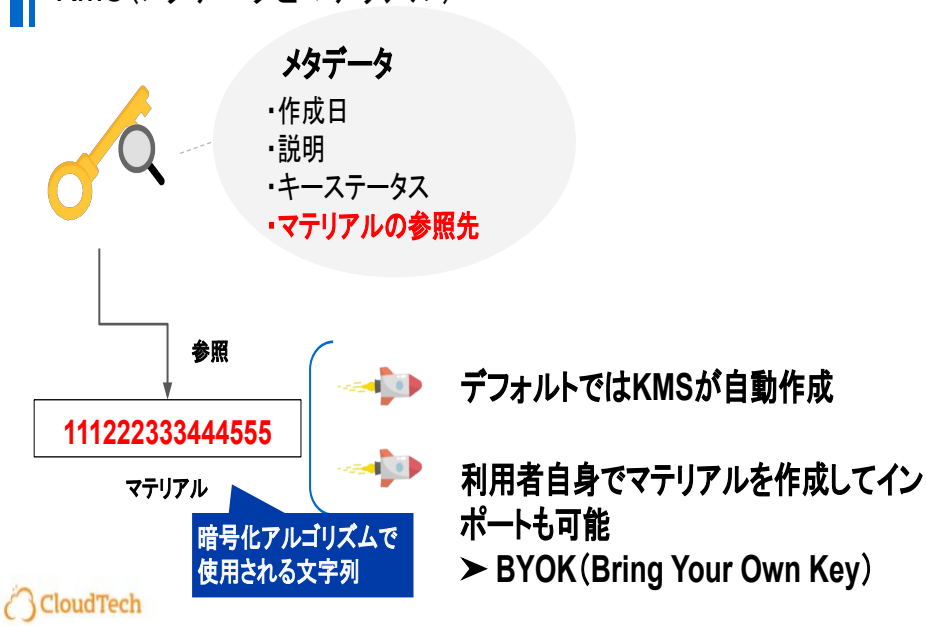
CMKとCDK、混同しないようにしましょう。
まずはこのフローを抑えていただければ OKです。


- まとめと、その他補足情報です。
  - CMKはCDKを生成することができ、CDKを暗号化、そして復号することができます。
  - CDKは実際のデータを暗号化、復号する役割で使用されます。
    - CMKは4kbyteまでしかデータを暗号化できないという成約があります。
      - なかなか、少ない容量ですよね。
  - 一方で、CDKは大容量のデータを暗号化できます。
  - CMKはKMSの鍵（つまりKMSが管理する対象）であるのに対し、
  - CDKはユーザー側の管理となります。
    - CDKはアプリケーションで取り扱う必要があり、KMSは管理したりはしません。

# その他用語の解説

CloudTech

## KMS（メタデータとマテリアル）



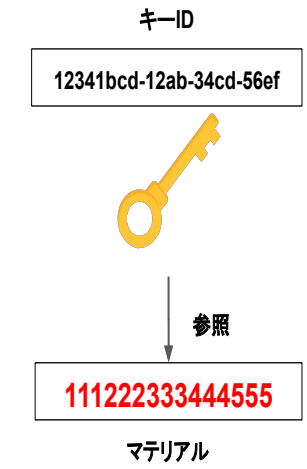

その他用語

- キーにはメタデータという情報が含まれます。
- メタデータには
  - 作成日
  - 説明情報
  - キーステータス
    - 作成中、有効、無効など、キーの状態を表すものです。
  - マテリアルの参照先
    - などの情報が含まれます。
      - マテリアルについて、英語では「材料」と約される単語ですが、、
        - これは、暗号化アルゴリズムで使用される文字列です。
        - メタデータの中には、このキーマテリアルへの参照情

- 報が含まれます。
  - マテリアルとはこのランダムな文字列をイメージしていただき、キーごとにユニークな（他と重複しない）マテリアルを持ちます。
    - このマテリアルは、暗号化アルゴリズムで使用される材料という感じです。
  - デフォルトでは、KMSは、CMKのマテリアルを自動作成しますが、利用者自身でのカスタマイズも可能です。
    - マテリアルが無い空のキーを生成しておき、
    - 利用者自身がサードパーティーのツールなどでマテリアルを作成し、インポートすることも可能です。
      - この場合、BYOK（Bring

- Your Own Key）と呼びます。
- CMKにキーマテリアルをインポートすると、CMKはそのキーマテリアルと永久に関連付けられます。

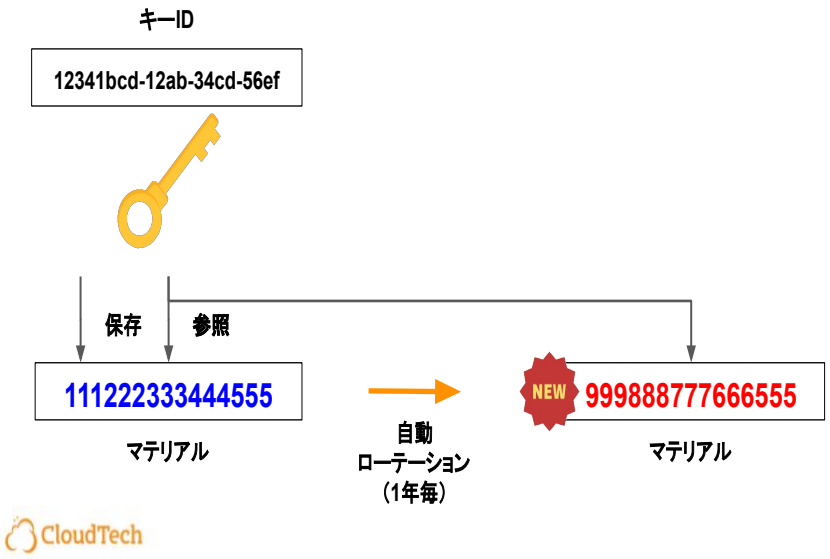
## KMS（キーIDとローテーション）




CloudTech

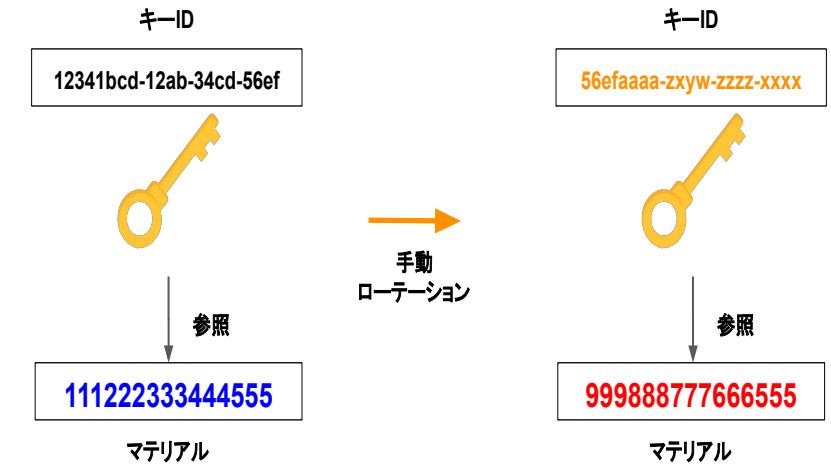
- キーID
  - これはキーの名前のようなものです。
    - AWSコンソール画面などで KMS キーを識別するのに役立ちます。

## KMS（自動ローテーション）




- ローテーション
  - KMS ではキーの自動ローテーション機能があり、これを有効にすると
    - KMSがキーの新しい暗号化マテリアルを生成して1年ごとに更新します。
      - キー IDなどの基本的なKMS キーのプロパティは、キーがローテーションされても変更されません。
        - 要は、中身のマテリアルだけ変わるイメージです。
          - ただし、古い暗号化マテリアルはすべて永続的に保存されるため、KMS キーで暗号化されたすべてのデータを復号することができます。

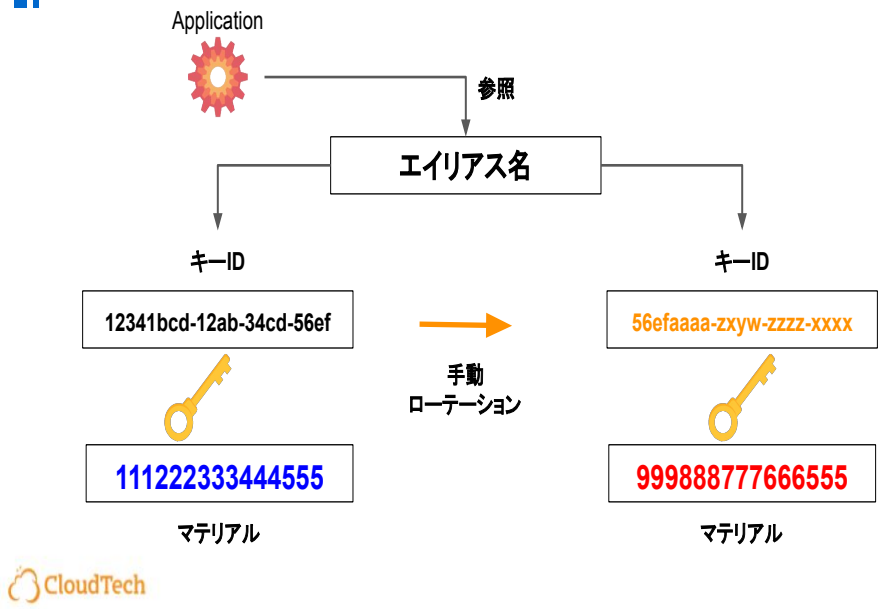
## KMS（手動ローテーション）




CloudTech

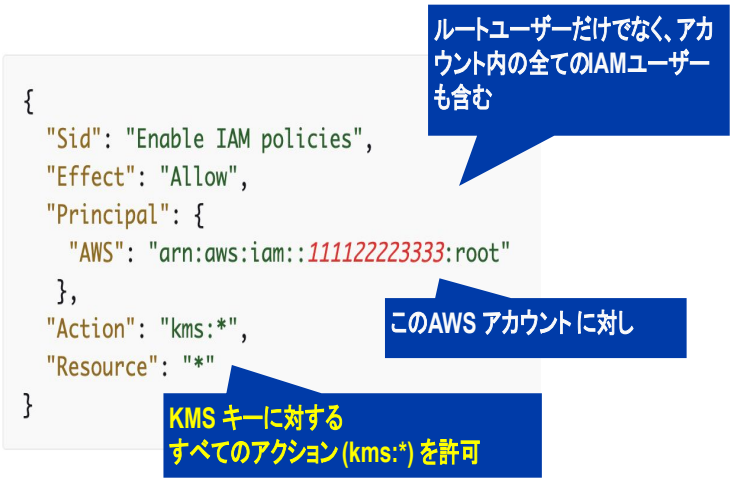
- セキュリティ要件的に1年更新では都合が悪い場合、手動でのローテーションも可能です。
  - これは、全く新しいキーを作成する行為を指します。
  - もちろん、キーIDやマテリアルなども、全く新しいものとなります。
    - 全く新しいものになってしまうと、アプリ側の設定が変更になってしまいますよね。
    - この手間を軽減する方法として、エイリアスがあります。

## KMS（エイリアス）




- エイリアスとは、
  - キーに付与する別名のことです。
  - アプリ側で、エイリアスの名前を参照させておくことで、
    - キーが全く新しいものになったとしても、エイリアスの参照先を新しいキーに付け替えるだけでよくなり、
      - アプリ側の設定変更が不要となります。

## KMS（キーポリシー）




CloudTech

キーポリシーは、KMS キーへのアクセスを制御する方法です。
KMS キーには 1 つのキーポリシーが必要となります。

S3のリソースベースポリシーのように、

- IAMポリシーは誰が操作できるのか、使う側に対して許可するのに対し、
- キーポリシーは誰が使って OKなのかを許可します。

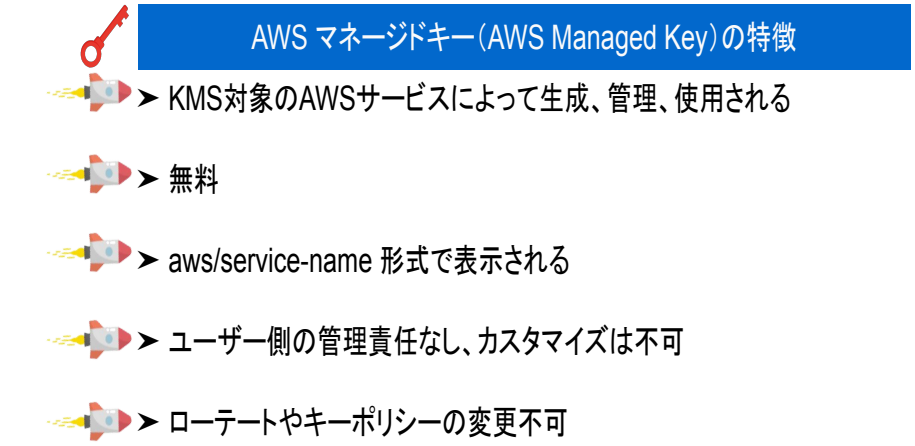
上記のキーポリシーステートメントは、
このAWS アカウント に対し、KMS キーに対するすべてのアクション (kms:*) を許可する権限を付与しています。

arn:aws:iam::account-id:root は、AWS アカウントのルートユーザーだけでなく、アカウント内の全ての IAMユーザーも含んでいます。

# AWS マネージドキー

CloudTech

## AWS マネージドキー



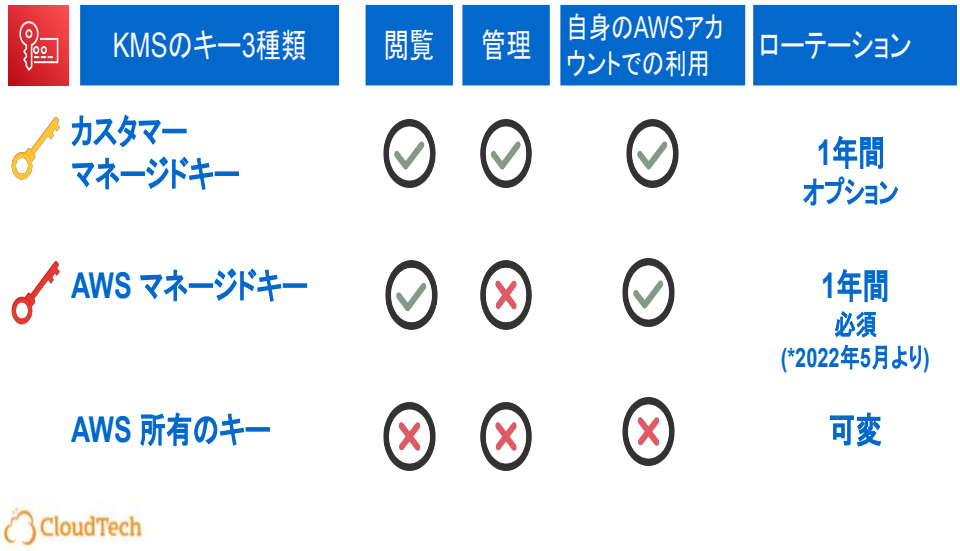

AWS　Managed CMK

- AWS managed CMKはAWSサービス（KMSで統合（対象）となっているもの）によって生成、管理、使用されるものです。
- 月額料金はまったく発生しません。
- AWS マネージドキー は KMS の Management Console で aws/redshift のような aws/service-name 形式で表示されます。
- マネージドされてるものですので、管理責任はありませんが、同時にカスタマイズもできません
- このキーのローテートやキーポリシーを変更はできません。

# KMSのキー
# 3種類の比較

CloudTech

## KMS管理の3種類のキーの比較




最後にKMSのそれぞれの鍵の違いをまとめます

- Costomaer managed CMK（カスタマー管理CMK） は
  - マネジメントコンソールで閲覧可能
  - 管理可能
  - AWSアカウント内において利用可能
  - ローテートは1年です。
    - 有効無効が選択できます
- AWS Managed CMK（AWS管理CMK）
  - マネジメントコンソールで閲覧可能ですが
  - 管理はできません。
  - AWSアカウント内において利用可能
  - ローテートは1年です。
    - 必須です。
    - 2022年5月に3年から1年に変更されました。
- AWS owned CMK（AWS所有CMK）
  - AWSサービスの裏側で勝手に動作しているものですので、

- 閲覧や管理は不可、
- 他のAWSアカウントでも利用され、
- ローテーションもその仕様は可変となっており、我々が意識することはありません。

■参考ドキュメント（公式サイト）

・AWS KMS の概念

https://docs.aws.amazon.com/ja_jp/kms/latest/developerguide/concepts.html#aws-owned-cmk

・デフォルトのキーポリシー

https://docs.aws.amazon.com/ja_jp/kms/latest/developerguide/key-policy-default.html