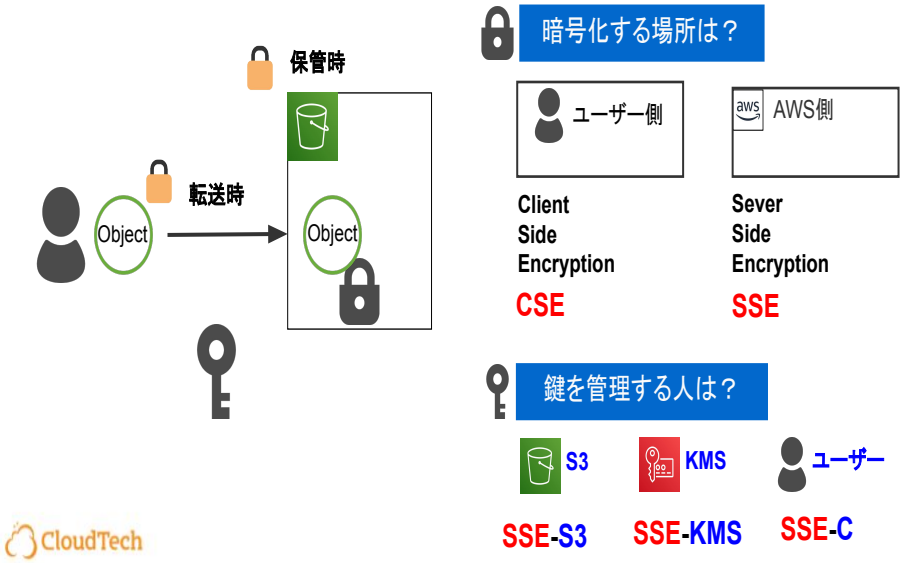
S3
保管時の暗号化

CloudTech

S3(保管時の暗号化)



- S3における暗号化は2つの側面があります。
  - オブジェクトをS3に転送する際の暗号化
    - これはHTTPS通信などで安全に通信をします。
  - オブジェクトを保管する時の暗号化
    - ファイル自体を暗号化して保管することで、今回はこの保管時の暗号化の話となります。

- 保管時の暗号化で重要なのはこの 2つ
  - 暗号化する場所はどこで行うのか？
  - データを暗号化するためには鍵が必要です。
    - この鍵を管理する人は誰か？
      - です。

- 暗号化する場所はユーザー側か、AWS側かで大きく分けられます。
  - ユーザー側の場合

        - Clientサイドで暗号化する、略して CSE
            - ユーザー側がデータの暗号化を行ってから、S3へデータをアップロードします
    - 暗号化する場所がAWS側の場合
        - サーバーサイドで暗号化、略して SSE
            - と呼びます
        - S3へデータをアップロードしてから暗号化を行う方法です

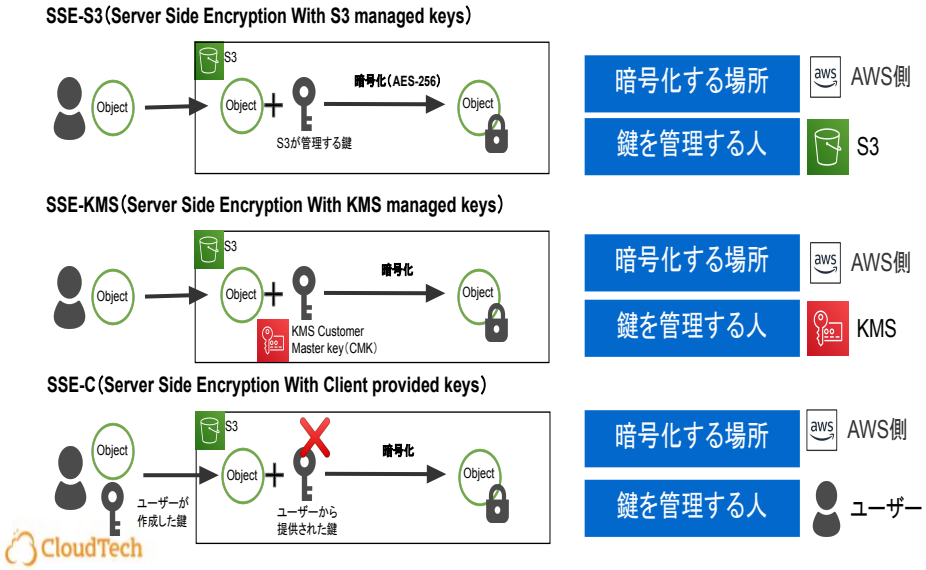
- 鍵を管理する人についてですが、
    - AWS側で鍵管理を任せるパターンとして 2つ
        - S3が鍵を管理するパターン
        - KMSが鍵を管理するパターン
    - そして、ユーザーが鍵を管理するパターンがあります。

- 次のスライドから、登場する用語、SSE-S3、SSE-KMS、SSE-C、がありますが、
    - SSE-S3の場合は
        - AWS側でオブジェクトを暗号化して、その鍵は S3が保管しているよ、という意味です。
            - SSE-KMSでも同様に、AWS側でオブジェクトを暗号化して、その鍵はKMSが保管だよ、
                - SSE-Cは、鍵はクライアント（ユーザー）が管理してるよ、ということとなります。
                    - 詳しくみていきましょう

S3
サーバーサイド
暗号化

CloudTech

S3（サーバサイド暗号化）



- SSE-S3
  - ユーザーがオブジェクトをアップロードすると、S3が管理をする鍵でオブジェクトを暗号化するという方式です。
  - なので、
    - 暗号化する場所はAWS側で、
    - 鍵の管理はS3ですね。
      - データの暗号化に使われる鍵は S3で管理されるもの
      - S3は一意のキー（ユニークキー：重複しないキー）で暗号化する
        - 補足ですが。追加の安全策として、ユニークキー自体を別のマスターキーで暗号化し、マスターキーを定期的にローテーションできます。
      - 暗号オプションはAES-256です。強度が高い暗号化方式です
      - 無料で使えます。

SSE-KMS

- ユーザーがオブジェクトをアップロードすると、KMSが管理をする鍵でオブジェクトを暗号化するという方式です。
- なので、
    - 暗号化する場所はAWS側で、
    - 鍵の管理はKMSですね。

- こちらの方式はCloudTrailで証跡が取れるので、暗号化されたか・複合されたかどうかについての証跡を取っておきたい Objectがある場合に有用です。
    - 追加で料金がかかるのがデメリット

SSE-S3のデメリットですが

- 暗号化の追跡はしていない。CloudTrailにログ残らない
    - 鍵に対するアクセス制限できない

- なお、S3のバケットの設定で、自動的にデフォルトで暗号化する設定ができます。
    - バケットにデータを書き込むと自動的に暗号化され、
    - バケットからデータを読み出す時は自動的に複合される
        - バケットをデフォルトで暗号化に設定できる方式は、SSE-S3とSSE-KMSとなります。

SSE-C

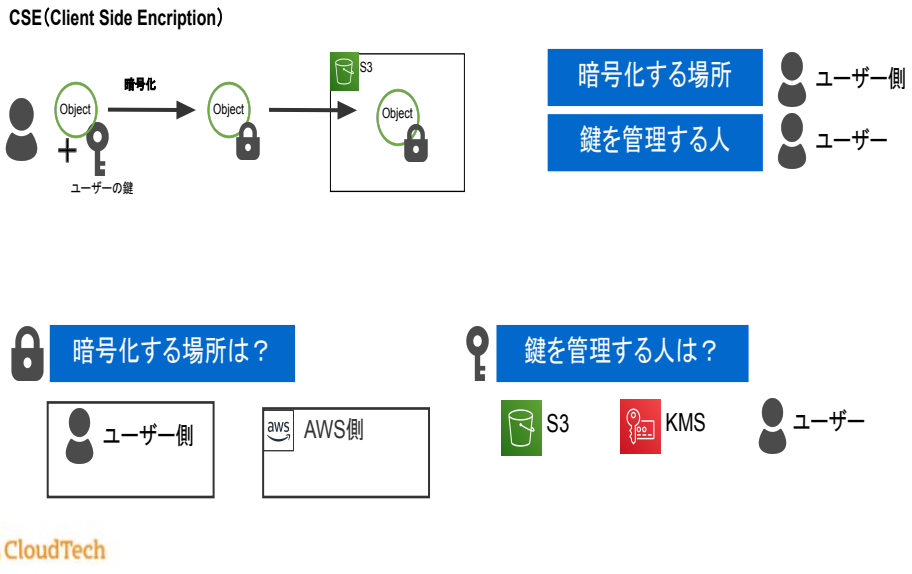
- これは、ユーザーが鍵を作成・管理します。AWSに鍵の保管はしないのが特徴
    - ヘッダー情報で鍵とオブジェクトをアップロードする。
        - S3は鍵情報を使って暗号化し、鍵情報を保管しないので破棄する
        - で、取り出す際には、再びユーザーの同じ鍵を使って取り出す。

        - 暗号化をする場所はどこでしょうか？
            - 暗号化、複合はAWS側ですね
        - では、鍵を管理するのは？
            - これはユーザーですね、
    - ユーザー側で鍵を自分で管理しないといけないので、ちょっと面倒
        - とは言え、AWS環境とキーの保管場所が別れるため、比較的セキュアになります。

S3
クライアントサイド
暗号化

CloudTech

S3（クライアントサイド暗号化）



- CSE
- これは図の通りです
  - ユーザー側で、オブジェクトを S3にアップロードする前に、オブジェクトを暗号化しておき、S3にアップする
  - そのデータを使用する時は暗号化状態のまま手元にダウンロードして、複合する
    - 全てユーザー側で完結させるのが特徴です。
    - データを送信する際の、通信経路から確実に暗号化したい場合はこちらが良いでしょう。
      - 場所も管理もユーザー側です
  - KMSで作成されたCMK（Customer Master Key）を使って暗号化するパターンもあるのですが、ここでは割愛します

サーバーサイド、クライアントサイド、みてきましたが、大切なのは

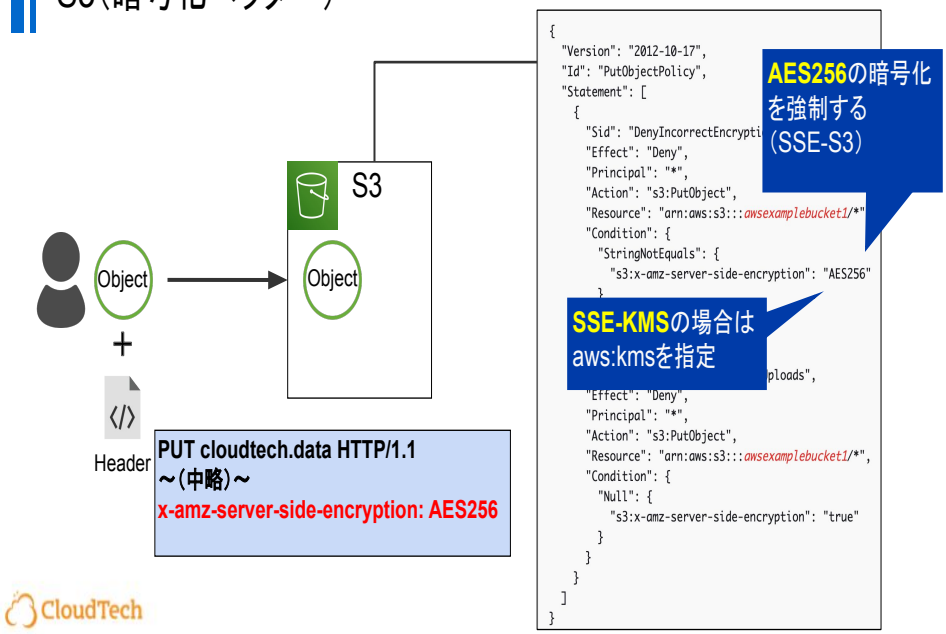
- 暗号化する場所はどっちか？
- 鍵を管理する人は誰か？

- を意識することです。

S3
暗号化ヘッダー

CloudTech

S3（暗号化ヘッダー）



暗号化ヘッダーについて説明します。

- 実際には、S3にオブジェクトをアップロードする際に、HTTPリクエストのヘッダー情報も飛んでいますが、ここに
  - x-amz-server-side-encryptionをセットしなければなりません。
    - 値はAES256かaws:kmsを指定します

次に、S3 へのオブジェクト保存時に暗号化を強制するために、S3のバケットポリシーをこのように設定する必要があります。

- こちらは AWS Documentation からの引用です。
  - EffectはDenyなので拒否ルール、
  - ヘッダー情報が正しくない場合に拒否するというものです。
    - どういう条件の時に拒否するかというと …
  - Principal（操作している主体）は全ての人を問答無用に対象とし、

- S3にオブジェクトをPUT（配置）するアクションに対して、
- 対象バケットを記載して、
  - 大切なのがココ、Conditionで
    - x-amz-server-side-encryptionの値がAES256
      - NOTイコール、の場合は拒否しますよ。
        - つまり、x-amz-server-side-encryptionの値はAES256を必須としてくださいね
        - そうでなければ拒否しますよ、ということです。
          - なお、AES256の暗号化を強制するSSE-S3の場合はこのような記述ですが、
          - SSE-KMSの場合はこのAES256を aws:kms と指定します。
- その下のルールは、暗号化されていないオブジェクトは拒否するというルールです。

https://docs.aws.amazon.com/ja_jp/AmazonS3/latest/userguide/UsingServerSideEncryption.html

S3
暗号化まとめ

CloudTech

S3（暗号化まとめ）

まとめ


➤ サーバーサイドで暗号化するSSE-S3, SSE-KMS, SSE-C


➤ クライアントサイドで暗号化するCSE


➤ バケットの設定によりデフォルトで暗号化される
（SSE-S3, SSE-KMS）


➤ x-amz-server-side-encryptionでコントロールする

CloudTech

- バケットを暗号化したら、それ以降にアップロードしたオブジェクトは自動で暗号化される
  - 注意点としては、バケットを暗号化する前にアップロードされているオブジェクトまで、一括でまるっと暗号化されるわけではないという点に注意しましょう。

- x-amz-server-side-encryptionでコントロールする
  - アップロード時のヘッダー、バケットポリシーで、x-amz-server-side-encryptionを使ったコントロールが発生することを覚えておきましょう。