

### S3 - Compartiments

C'est l'endroit où nous stockons notre base de donnée. Chaque fichier chargé dans le compartiment correspond à une table. Pour faire référence au table, nous devons faire référence au fichier dans le code pour y avoir accès. Il est possible d'avoir plusieurs compartiment.

### S3 - Objets

Les objets sont le contenu de notre base de données, stockées dans les compartiments d'Amazon S3. Une fois stockées on peut les manipuler via des fonctions lambda. Les objets possèdent des attributs qui sont les colonnes de la base de donnée (titre, genre, ect...).

### Lambda Déclencheur

Lambda déclencheur est un moyen de créer une API en lien avec la fonction lambda créée précédemment. Une fois que nous allons sur cette API, le code de la fonction liée à cette API s'exécutera. Le déclencheur est l'exécution de la fonction au moment où on l'a appelé.

### AWS Lambda

Lambda est un service de calcul proposé par AWS, il permet de créer des fonctions js permettant de manipuler la base de donnée. Pour utiliser la fonction, il faut créer un déclencheur et une API. Ainsi, nous pouvons héberger nos fonctions dans AWS et utiliser l'api faisant appel à la fonction voulue dans le code. Souvent, les fonctions lambda que nous créons correspondent au fonction nécessaire dans notre DAO principal.

### API Gateway

C'est un outil de gestion des interfaces et de programmation d'applications (API). Elle se positionne entre un client et une collection de services back-end. Cette passerelle sert également de protection de nos API car permet de dissocier l'interface client de la mise en œuvre back-end

### API Gateway CORS

Le partage de ressources cross-origin (CORS) est une fonctionnalité de sécurité du navigateur qui restreint les requêtes HTTP initiées à partir de scripts exécutés dans le navigateur. Si les ressources de notre API REST reçoivent des requêtes HTTP cross-origin non simples (autre que GET, HEAD, POST), on doit alors activer la prise en charge de CORS.

### Point de terminaison API

Un point de terminaison c'est l'URL du point d'entrée d'un service web AWS. C'est grâce à ce lien qu'on peut utiliser les API de notre application. Avec le point de terminaison de l'API nommé "lister" on peut récupérer tous les animés qui étaient présents dans notre fichier "liste-animé.json"

### API REST

Les API REST sont celles qu'on va utiliser pour envoyer des données. C'est également elle qui prennent en charge les intégrations Lambda personnalisées

### Rôle IAM

Un rôle IAM est un rôle est regroupement de stratégie ou politique IAM qu'on peut lier à nos fonctions lambda. Pour cela, il faut créer un rôle et attribué à celui-ci les stratégies nécessaires à son bon fonctionnement. Néanmoins, il faut donner une stratégie à une fonction de confiance pour ne pas endommager, par mégarde, la base de donnée dans le compartiment. Pour éviter cela il faut minutieusement bien choisir les stratégies de chaque fonction lorsqu'on les crée. Si une mauvaise manipulation a été faite nous pouvons quand même modifier les rôles et modifier les stratégies liées à celle-ci.

### IAM (Identity and access management)

Ce service permet à l'utilisateur de gérer les permissions des fonctions pour un compartiment. On peut par exemple autoriser la lecture dans un compartiment et interdire l'écriture, la fonction lambda liée à ce rôle possédera uniquement les accès qui lui sont données.

### Stratégie/Politique IAM

Les stratégies/politiques autorisent ou refusent l'accès aux services de AWS ou aux actions API qui fonctionnent avec IAM. Elles correspondent également au permission que nous pouvons attribuer aux rôles. Il existe de nombreuses permissions basiques comme complexe et il faut une stratégie pour chaque permission permettant ainsi d'agir précisément sur les permission des fonctions

### API HTTP

Les API HTTP sont des une API qui utilise le protocole de transfert hypertexte comme protocole de communication entre deux systèmes. C'est avec elle que nous récupérerons les données de notre BDD