

**Master of Science**

**M.Sc1**

# **Rapport de Projet de fin de l'année**

**Année académique 2020/2021**

**Réalisé par :**

- **Sene Mohamed Fadel**
- **Binalounga Erick Celestin**
- **Babaka Toure**
- **Wilfridwareljunior**
- **Honore Tokou**

# Table des matières

---

I.	Introduction .....	3
II.	Architecture AWS : Choix techniques .....	3
1.	Infrastructure.....	3
2.	Architecture générale .....	4
3.	Sécurité .....	5
a)	Chiffrement des données.....	5
b)	Sécuriser l'accès aux données.....	6
c)	Pare-feu IP.....	6
d)	Gestion des identités et des accès.....	6
e)	Détection avancée des menaces .....	6
4.	Sauvegarde.....	7
5.	Extensibilité .....	8
6.	Haute Disponibilité.....	8
III.	Les applications : Choix techniques .....	10
1.	Développement de l'Application web .....	10
2.	Développement de l'Application backend de Data Processing.....	11
3.	Développement de l'Application mobile.....	12

# **I. Introduction**

Dans ce document seront présentées les différentes solutions choisies, les motivations de ces choix. Les parties concernées pour ce projet sont : l'infrastructure, la base de données, le développement des applications web et mobile.

## **II. Architecture AWS : Choix techniques :**

Il existe sur le marché plusieurs fournisseurs de services cloud tels qu'Azure de Microsoft, Amazon Web Services, GooglePlatform ..., nous avons choisi AWS pour des raisons que nous allons expliquer ci-dessous pour satisfaire au mieux les besoins du projet.

AWS est la plateforme de cloud public de Amazon Web Service. Comme les autres fournisseurs de services Cloud, AWS permet de profiter de ressources de Cloud Computing à la demande. Il permet aux entreprises de faire d'importantes économies en leur évitant d'avoir à ériger un centre de données sur site, de le maintenir, de le mettre à jour, de le refroidir et de payer l'électricité.

AWS offre un éventail de services, nous pouvons citer : base de données (Dynamo DB), la sécurité et la haute disponibilité des données, ...

### **1. Infrastructure :**

Notre infrastructure est basée sur AWS, Dynamo DB est le service de base de données multi-modèle de Amazon distribué à l'échelle mondiale. Dynamo DB permet de générer des applications hautement disponibles et très réactives dans le monde entier. DynamoDB réplique en toute transparence nos données où que soient les utilisateurs, afin que ces derniers puissent interagir avec la réplique des données la plus proche.

L'image suivante donne un résumé sur la façon dont les données sont stockées, extraites, traitées et utilisées par Dynamo DB et les applications.

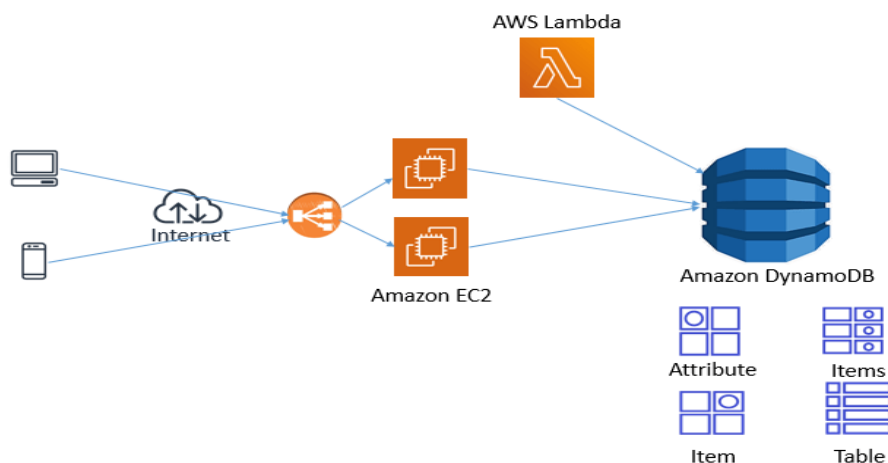


Figure 1: Relation entre Dynamo DB et les instances

## 2. Architecture générale

Dans le cadre du dimensionnement du système phygital de notre entreprise, nous avons opté pour une solution cloud avec AWS, accessible via internet. Localement pour l'utilisation de l'application web ou mobile par les clients des points d'accès sont déployés ceux-ci connectés à une passerelle comme l'indique la figure suivante.

Un administrateur aura accès à la console AWS via internet et un VPN pour la sécurité, de plus AWS fournit un pare-feu pour sécuriser le réseau dans le cloud.

### 3. Sécurité

La sécurité est la priorité dans le cloud, nous avons trouvé des informations précises et à jour sur la sécurité AWS. L'une des meilleures raisons de l'utilisation de AWS pour notre plateforme est de tirer parti de **sa large gamme de fonctionnalités et outils de sécurité**. Ces outils et fonctionnalités permettent de créer des solutions sécurisées sur la plateforme AWS sécurisée. AWS assure la confidentialité, l'intégrité et la disponibilité des données client, tout en permettant la gestion transparente des responsabilités.

#### a) Chiffrement des données

Dynamo DB stocke ses bases de données primaires sur des SSD. Ses sauvegardes sont stockées dans le service de AWS. Avec l'apparition du chiffrement au repos pour Dynamo DB, notre base de données, nos documents et sauvegardes sont désormais chiffrés pendant le transit c'est-à-dire sur le réseau et au repos ce qui nous permet un chiffrement de bout en bout.

AWS, en tant que service PaaS, est très facile à utiliser. Comme toutes nos données stockées dans Dynamo DB sont chiffrées, nous n'avons aucune action à effectuer. En d'autres termes, le **chiffrement au repos est « activé » par défaut. Il n'existe aucun contrôle à activer ou à désactiver**. Dynamo DB utilise le chiffrement AES-256 dans toutes les régions où le compte est exécuté. AWS fournit cette fonctionnalité pour respecter le contrat SLA de **disponibilité et de performances**.

- Les données stockées dans notre console AWS sont chiffrées automatiquement et de façon fluide avec des clés gérées par AWS (clés gérées par le service).

Le flux de base d'une demande d'un utilisateur est le suivant :

- ✓ Le compte de base de données de l'utilisateur est préparé et les clés de stockage sont récupérées via une demande au fournisseur de ressources du service de gestion.
- ✓ Un utilisateur crée une connexion à Dynamo DB via le transport HTTPS/sécurisé. (Les SDK permettent de s'abstraire des détails.)
- ✓ L'utilisateur envoie un document à stocker sur la connexion sécurisée créée précédemment.
- ✓ Périodiquement, les données sont lues à partir du stockage sécurisé et sauvegardées dans le magasin d'objets blob chiffré d'AWS.

## b) Sécuriser l'accès aux données

AWS Dynamo DB utilise deux types de clés pour authentifier les utilisateurs et permettre d'accéder à ses données et à ses ressources.

- Les clés principales fournissent un accès à toutes les ressources administratives du compte de base de données. Clés principales :
  - ✓ Fournissent un accès aux comptes, aux bases de données, aux utilisateurs et aux autorisations.
  - ✓ Ne peuvent pas être utilisées pour fournir un accès précis aux conteneurs et aux documents.
  - ✓ Sont créées lors de la création d'un compte.
  - ✓ Peuvent être régénérées à tout moment.
- Les jetons de ressource fournissent un accès aux ressources d'application au sein d'une base de données, ils sont liés à **une période de validité personnalisable**. La durée de validité par défaut est d'une heure. La durée de vie du jeton peut cependant être définie de manière explicite (cinq heures maximum). **Offrent une alternative sûre pour céder la clé principale.**

Permettent aux clients de lire, d'écrire et de supprimer des ressources dans le compte AWS Dynamo DB en fonction des autorisations qui leur ont été accordées.

## c) Pare-feu IP

Avec des pare-feux, nous pouvons configurer notre compte Cosmos pour qu'il soit accessible uniquement à partir d'un ensemble d'ordinateurs et/ou de services cloud approuvés.

Par défaut, notre compte AWS Dynamo DB est accessible depuis Internet, tant que la demande est accompagnée d'un jeton d'autorisation valide. Pour configurer le contrôle d'accès basé sur **la stratégie IP**, nous devons fournir le jeu d'adresses IP ou des plages d'adresses IP pour être ajouté à la liste d'adresses IP clientes autorisées. Une fois cette configuration appliquée, toutes les demandes provenant d'ordinateurs qui ne figurent pas sur cette liste autorisée reçoivent une réponse **403 (Interdit)**.

L'accès est requis pour autoriser l'utilisation de l'Explorateur de données ainsi que pour récupérer des métriques pour votre compte qui s'affichent sur le portail AWS.

## d) Gestion des identités et des accès

Le volet Contrôle d'accès (IAM) du Portail AWS permet de configurer le contrôle d'accès en fonction du rôle sur les ressources AWS. Les rôles sont appliqués aux utilisateurs, aux groupes, aux principaux de service et aux identités gérées dans IAM. Nous avons utilisé des rôles intégrés ce qui répond à nos besoins, mais AWS nous donne également la possibilité de créer nos propres rôles personnalisés pour les individus et les groupes.

## e) Détection avancée des menaces

Amazon GuardDuty et AWS Security Hub fournissent une visibilité continue, une conformité et une détection des menaces pour les comptes AWS et les charges de travail ce qui offre une sécurité supplémentaire qui détecte les tentatives d'accès ou d'exploitation inhabituelles et potentiellement dangereuses des comptes AWS Dynamo DB. Cette couche de protection, permet de traiter

efficacement les menaces sans pour autant être un expert en sécurité et de les intégrer aux systèmes centraux de supervision de la sécurité.

Les alertes de sécurité sont déclenchées lorsque des anomalies se produisent dans l'activité. Ces alertes de sécurité sont intégrées avec AWS Security Hub et sont également envoyées par e-mail aux administrateurs d'abonnement, avec les détails des activités suspectes, ainsi que des recommandations sur la façon d'examiner et de corriger les menaces.

### Quels types de menaces ?

- **Accès à partir d'emplacements inhabituels** : Cette alerte est déclenchée en cas de modification du modèle d'accès à un compte AWS Dynamo, où un utilisateur s'est connecté au point de terminaison AWS Dynamo DB **à partir d'un emplacement géographique inhabituel**. Dans certains cas, l'alerte détecte une action légitime indiquant une nouvelle application ou une opération de maintenance du développeur. Dans d'autres cas, l'alerte détecte une action malveillante, par exemple, d'un attaquant externe, d'un ancien employé,, etc.
- **Extraction de données inhabituelles** : Cette alerte est déclenchée quand un client extrait une quantité inhabituelle de données d'un compte AWS Dynamo DB. Il peut s'agir d'un **symptôme d'exfiltration de données effectuée pour transférer toutes les données stockées** dans le compte vers un magasin de données externe.

## 4. Sauvegarde :

AWS Dynamo DB sauvegarde automatiquement nos données à intervalles réguliers. Les sauvegardes automatiques sont effectuées **sans affecter les performances ou la disponibilité des opérations de base de données**. Toutes les sauvegardes sont stockées séparément dans un service de stockage, et ces sauvegardes sont répliquées globalement pour garantir la résilience contre les sinistres régionaux. Les sauvegardes automatiques sont utiles dans les scénarios où nous supprimons ou mettons à jour accidentellement notre compte, base de données ou conteneur Dynamo et où nous avons besoin ultérieurement de récupérer les données.

### Sauvegardes automatiques et en ligne :

Avec AWS DynamoDB, nos données et leurs sauvegardes sont rendues hautement redondantes et résilientes aux sinistres régionaux. Les étapes suivantes montrent comment AWS Dynamo DB sauvegarde les données :

- ✓ Dynamo DB sauvegarde automatiquement notre base de données **toutes les 6 heures** et à tout moment, seules les **dernières 2 sauvegardes sont stockées**. Toutefois, si le conteneur ou la base de données est supprimé, AWS DynamoDB conserve les captures instantanées existantes du conteneur ou de la base de données **pendant 30 jours**.

- ✓ Les sauvegardes sont effectuées sans affecter les performances ou la disponibilité de notre application. **La sauvegarde des données s'effectue en arrière-plan sans consommer de débit supplémentaire et sans affecter les performances ou la disponibilité de la base de données.**
- ✓ Si nous avons **accidentellement supprimé ou endommagé nos données**, il faut contacter le support Amazon le plus rapidement possible idéalement dans l'heure afin que l'équipe AWS Dynamo DB puisse nous aider à les restaurer à partir des sauvegardes.

## 5. Extensibilité

Dynamo DB offre une extensibilité élastique sans précédent pour nos écritures et lectures, partout dans le monde. Nous pouvons passer de façon élastique de milliers à des centaines de millions de requêtes par seconde dans le monde, en un seul appel d'API, et payer uniquement le débit et le stockage dont nous avons besoin. Cette fonctionnalité nous aide à gérer les pics inattendus dans nos charges de travail sans devoir effectuer un sur-provisionnement.

## 6. Haute Disponibilité

AWS Dynamo DB réplique en toute transparence nos données vers toutes les régions AWS associées à notre compte. Dynamo DB emploie plusieurs couches de redondance pour nos données, comme le montre l'image suivante :



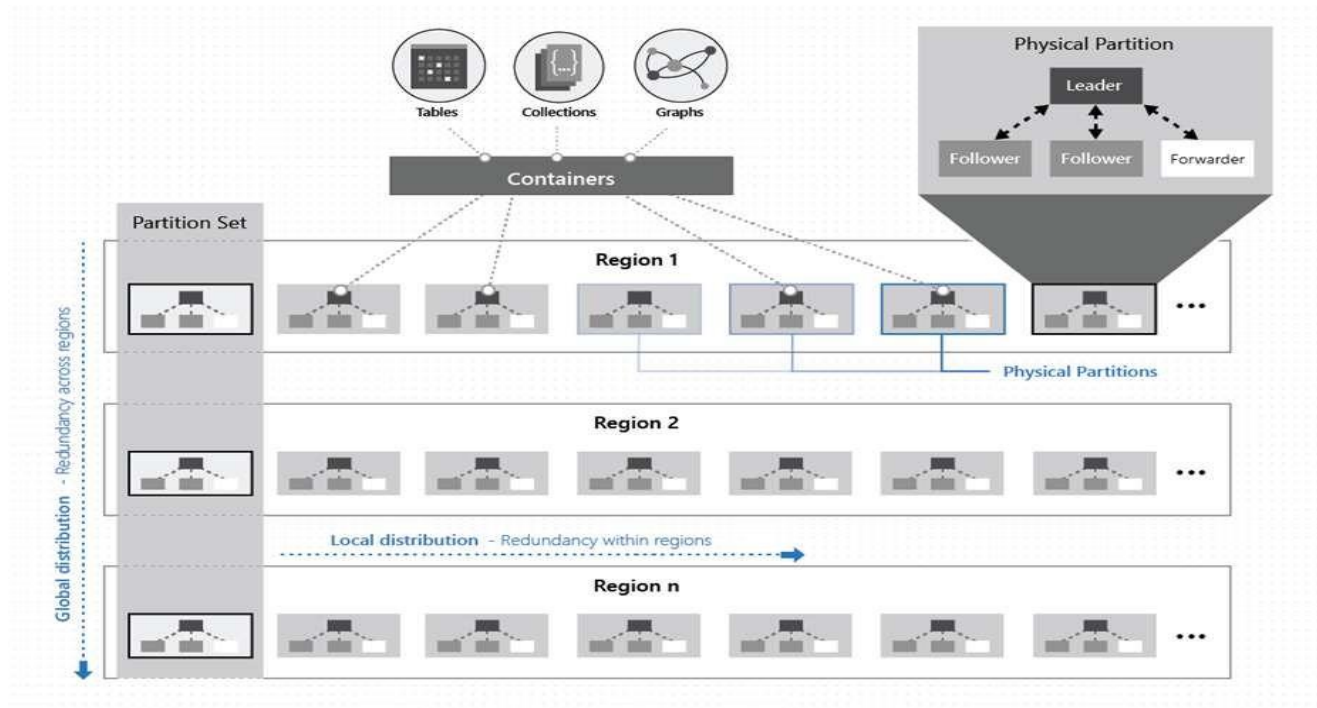


Figure 3: Redondance dans Dynamo DB

- Les données dans les conteneurs sont partitionnées horizontalement.
- Dans chaque région, chaque partition est protégée par un jeu de réplicas avec toutes les écritures répliquées et durablement validées par une majorité de réplicas. Les réplicas sont répartis entre 10 et 20 domaines d'erreur.
- Chaque partition est répliquée dans toutes les régions. Chaque région contient toutes les partitions de données d'un conteneur Cosmos et peut accepter des écritures et traiter les lectures.

**En cas de panne, Dynamo DB permet de s'assurer que notre base de données est toujours hautement disponible.** Les détails suivants permettent de capturer le comportement de Dynamo DB pendant une panne, en fonction de la configuration de notre compte :

- Avant la reconnaissance d'une opération d'écriture auprès du client alors que Dynamo DB est utilisé, les données sont validées durablement par un quorum de réplicas se trouvant la région qui accepte les opérations d'écriture.
- Les comptes multi région configurés avec plusieurs régions d'écriture sont hautement disponibles pour les écritures et les lectures. Les basculements régionaux sont instantanés et ne nécessitent aucune modification à partir de l'application.

- En plus Dynamo DB fournit des contrats SLA complets qui englobent le débit, la cohérence et une haute disponibilité.

### III. Les applications : Choix techniques

#### 1. Développement de l'Application web

Pour développer l'application **nous** avons :

- ✓ Utiliser le Django pour la partie présentation (client)
- ✓ Découper le projet en deux modules.
- ✓ Programmer par couche.
- ✓ Utiliser Spring pour l'injection de dépendances.
- ✓ Créer une base de données sur DynamoDB de AWS pour la collecte et le stockage des données.
- ✓ Utiliser GIT pour la gestion de version.

Comme environnement nous avons utilisé :

- ✓ **Python** : C'est un langage de « haut niveau », la communauté Python est énorme, c'est simple et rapide. Python est en constante évolution du fait de sa masse importante d'utilisateur.
- ✓ **Modèle MVT (Modèle Vue Template)** : est une architecture et une méthode de conception pour le développement d'applications logicielles, c'est une collection de trois composants importants, elle s'inspire de l'architecture MVC, mais n'a pas de contrôleur s'éparer.

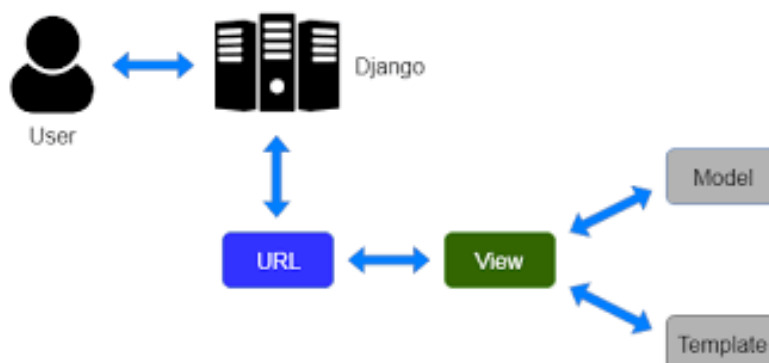


Figure 4: Le modèle MVT

- La couche **Modèle** se charge des traitements à effectuer sur les données et de leur stockage ;
- La couche **Vue** est utilisée pour exécuter la logique et interagir avec un modèle pour transporter les données et restituer un Template
- La couche **Template** gère la partie interface utilisateur.

➤ **Bootstrap :**

Est une collection de technologies utiles à la création du design (animation, graphisme, et interactions avec les pages des navigateurs, ça nous permettra de rendre l'application plus interactifs , et l'application pourra s'adapter facilement aux navigateurs des tablettes et aux smartphones même si le client ne passe pas par l'application mobile

## **2. Développement de l'Application backend de Data Processing :**

Pour développer l'application on a opté pour les technologies suivantes :

✓ **Visual Studio Code :**

Le VS code permet de passer plus de temps à coder et moins de temps à basculer entre les outils. On peut utiliser des fonctionnalités et extensions qui s'intègrent avec AWS et GitHub pour tout développer, déboguer et déployer à partir d'un seul emplacement.

Les extensions AWS incluent des émulateurs pour Stockage, AWS Dynamo DB et nous permet de développer notre application en localement avant de la déployer, en plus VS code offre la possibilité de Stocker et gérer des données.

**✓ Python :**

Plusieurs facteurs ont fait que de de langage un acteur incontournable dans beaucoup de domaines, notamment celui de Data science. Ici j'énumère quelques facteurs clés :

- **Un langage open source**
- **Une syntaxe facile**
- **Un langage de Programmation Orienté Objet**
- **La gamme de librairies**
- **Python comporte une large gamme de librairies pour Data science et Data analytics :**
- - Numpy: utile pour les calculs mathématiques comme la multiplication matricielle, opérations sur les tableaux.
  - SciPy: utile pour les calculs scientifiques avec des modules de visualisation, optimisation, algèbre linéaire et beaucoup d'autres choses.
  - Pandas : contient des outils et des fonctions qui rendent l'analyse des données rapide et moins complexe.

### **3. Développement de l'Application mobile :**

Pour développer l'application mobile on a utilisé les technologies suivantes :

**✓ Android studio :**

Android Studio 2.0 offre la solution la plus rapide pour développer des applications performantes et de qualité destinée aux appareils Android, y compris les téléphones et les tablettes, Android Auto, AndroidWear et Android TV. En tant qu'environnement de développement intégré officiel de Google, Android Studio inclut tout ce dont vous avez besoin pour développer une application : éditeur de code et débogueur intelligents, outils d'analyse des performances, émulateurs, et bien plus.

**✓ Java pour android:**

Cette langue de programmation Android aide à créer une app native. C'est le langage le plus populaire pour le développement sur systèmes embarqués et Android. Savoir programmer avec ce langage natif

ouvre des portes pour tout développeur. Il a facilité le développement d'applications natives pour plus de 3 milliards de smartphones. Langage open source, Java est partagé par une très grande communauté de développeurs. Techniquement, le développement pour Android avec Java favorise l'accès aux fonctionnalités natives du smartphone. Les applications pour Android spécifiquement développées mobilisent sans restriction

le GPS, l'accéléromètre, et bien d'autres features.