Document de Conception du Data Lake

1. Objectifs

Pour ce projet, le **Data Lake** nous servira à **centraliser, transformer et rendre nos données exploitables** pour notre site e-commerce. Ces données proviennent de différentes sources. Nos principaux objectifs à l'issue du projet seront :

- **Avoir une vue unique** sur l'ensemble des données que l'on a récoltées.
- **Avoir la possibilité d'analyser et de prendre des décisions en temps réel.**
- **Avoir la possibilité de garantir la scalabilité** (gérer de plus gros volumes de données).

2. Sources des Données

Nos données proviennent de différents types. Pour commencer, nous avons les **transactions clients**, dont le format de données est **SQL**. La source est le **CRM qui gère les commandes**. On extrait les données de manière **quotidienne**, c'est-à-dire une fois par jour, avec un processus qui va extraire toutes les **nouvelles transactions effectuées durant la journée**. Ces données vont être chargées dans le **Data Lake**.

Nous avons également les **logs de serveurs web**. Ils sont stockés dans un **fichier texte** ; de ce fait, le format n'est pas structuré. La source est les **serveurs Apache** et les **API** de notre site e-commerce. Ce sont des **informations brutes** qui sont stockées : **route appelée**, **adresse IP**, **statuts HTTP**, etc. La fréquence est une **ingestion continue**, car un log est généré **à chaque requête envoyée au serveur**. Par exemple, lorsqu'un utilisateur visite une page, un log est directement enregistré dans le fichier texte.

Ensuite, nous avons les **données des médias sociaux**, comme **Twitter** ou **Facebook**. Ce sont des données disponibles dans le format **JSON**, avec le texte que l'utilisateur a posté ainsi que son **nom d'utilisateur**, par exemple. On intègre les données dès lors qu'elles sont générées dans les réseaux sociaux. C'est donc une ingestion en temps réel. Le but étant d'analyser immédiatement les données. Ça permet également de réagir immédiatement à des tendances,

Pour finir, il y a les flux publicitaires. Ils incluent les impressions, les clics, le cout etc. le format est en JSON ou CSV. On utilise Google ADS pour récupérer les données. La fréquence est la même que celle des médias sociaux c'est-à-dire de l'ingestion en temps réel.

3. Architecte Logique du Data Lake

Le Data Lake est structuré selon une architecture à plusieurs zones, permettant la gestion et la transformation progressive des données.

3.1. Raw Zone

Les données sont stockées sous format brut tel qu'elles sont récupérées depuis les différences sources. Leurs formats diffèrent, on a CSV, JSON et texte brut. Leurs structures c'est par source et par date.

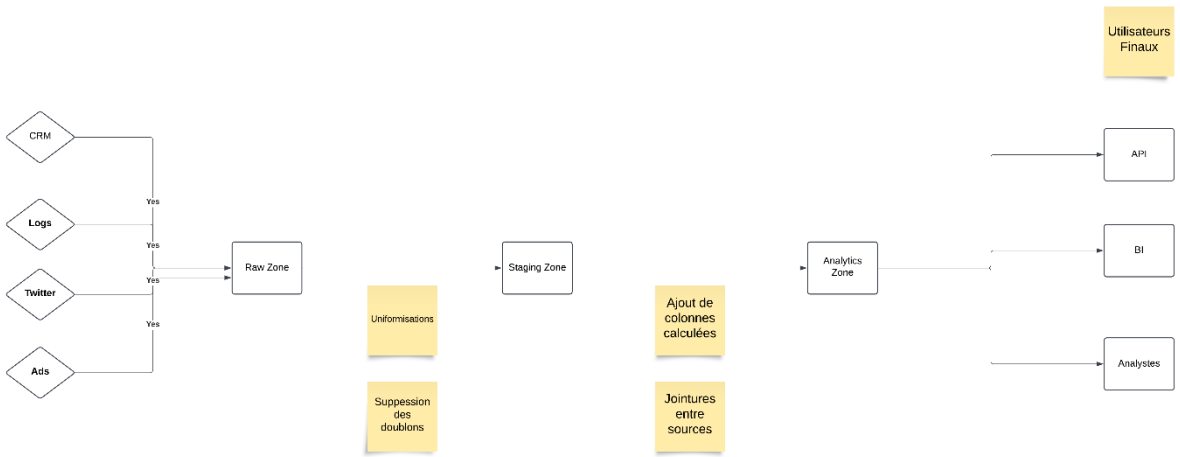
3.2. Staging Zone

Les données sont nettoyées et standardisées afin de les rendre exploitables. Le format que nous allons utiliser ce sera CSV ou Parquet dans certains cas. On va les organiser dans une entité logique, par noms et dates. Nous allons supprimées les doublons et les valeurs inutiles ou manquantes. On va également standardiser les colonnes pour permettre des jointures futures et uniformiser les formats. A la fin, les données seront prêtes pour des transformations complexes ou pour des analyses et les erreurs seront réduites pour la suite.

3.3. Analytics Zone

L'Analytics Zone contient les données enrichies et agrégées, optimisées pour les analyses et les visualisations métiers. Le format des fichiers ce sera parquet ou formats BI pour les visualisations. L'organisation se fait par cas d'usage. Cela permet d'avoir une traçabilité complète des données brutes. Nous allons enrichir les données, ajout de colonnes calculées et nous allons faire une jointure entre différences sources.
Nous pensons à faire un résumé par période et un calcul des indicateurs clés.

4. Diagramme des Flux de Données



5. Choix des technologies

5.1 Stockage

Nous allons stocker les données directement sur la machine dans des répertoires organiser par zone car c'est plus simple pour la mise en place et la gestion des données. On va d'abord procéder à un batch pour les transactions car c'est plus rapide et on peut écrire nos scripts sur mesure pour extraire les données une fois par jour.

5.2 Ingestion des Données

Ensuite, pour le streaming, on va utiliser Apache Kafka en local pour gérer les logs serveurs données publicitaires et médias sociaux. Cela va nous permettre un flux en temps réel pour tester l'ingestion continue.

5.3. Traitement des Données

Nous allons utiliser PySpark avec Python car Spark est l'outil de référence pour gérer les volumes et faire du traitement distribué. PySpark est une extension de Python donc on n'aura pas besoin d'apprendre un nouveau langage. En ce qui concerne, la Pipeline ETL, nous allons utiliser AirFlow afin de les orchestrer. Avec cet outil, on peut planifier les tâches comme le nettoyage des données tous les jours.

5.4. Formatage des données

Nous avons tout d'abord la Raw Zone dans laquelle, nous allons garder les données brutes. Pour la Staging Zone, on va utiliser Parquet ou CSV si besoin de compatibilité.

5.5. Visualisation et API

Nous allons utiliser Power BI pour créer les Dashboard pour une rapidité à cerner les indicateurs et pour l'API nous allons utiliser Flask.