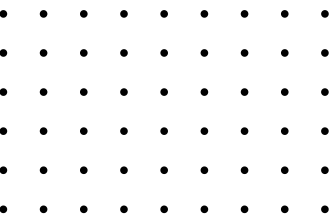




EFREI
paris panthéon assas université



Rendu du projet

DATA **LAKES**

Le 21/11/2024

Professeur: Lionel Brice SOUOP PEKAM

Rédigé par :
BRUDER Louis
BETTAIEB Nabil
BAFFOUN Kenza





TABLES DES MATIÈRES

Introduction

- 1.1. Contexte et objectifs du projet

Choix des technologies pour le stockage

- 2.1. Présentation des outils sélectionnés
- 2.2. Justification des choix
- 2.3. Avantages spécifiques de chaque technologie

Documentation technique sur les choix de technologies et solutions Cloud

- 3.1. Technologies Cloud utilisées
- 3.2. Rôle de chaque outil dans le pipeline
- 3.3. Justification des décisions de conception (Cloud vs local)
- 3.4. Développement de l'API

Documentation sur les transformations appliquées aux données

- 4.1. Définition des transformations de données
- 4.2. Travail effectué : Dédoublonnage des données

Conclusion et perspectives

- 6.1. Synthèse des travaux réalisés
- 6.2. Prochaines étapes possibles pour améliorer la solution





INTRODUCTION

1.1. Contexte et objectifs du projet

Dans un monde où les données numériques jouent un rôle central dans la prise de décision, les entreprises de commerce en ligne doivent exploiter efficacement leurs informations pour rester compétitives. Chaque interaction avec un client, chaque campagne publicitaire ou encore chaque visite sur un site web génère des données précieuses. Cependant, ces données proviennent de sources variées :

- Transactions clients (bases relationnelles) qui offrent des insights sur les comportements d'achat.
- Logs des serveurs web (non structurés) qui permettent de comprendre l'activité et la performance des plateformes numériques.
- Données des médias sociaux (semi-structurées en JSON) qui reflètent la perception des clients et leurs attentes.

Malgré leur richesse, ces données ne peuvent être exploitées de manière optimale sans une infrastructure adaptée. L'entreprise fait face à plusieurs défis :

1. **Diversité des formats**: Structurées, semi-structurées ou non structurées, les données nécessitent des traitements spécifiques.
2. **Volumes croissants**: Les quantités générées chaque jour dépassent la capacité des systèmes traditionnels.
3. **Besoins d'analyse rapide**: Les décisions commerciales doivent s'appuyer sur des insights obtenus en temps réel.
4. **Sécurisation et gouvernance**: Il est essentiel de protéger ces données sensibles tout en assurant leur qualité et leur traçabilité.





C'est dans ce contexte que le projet vise à concevoir un Data Lake, une solution moderne et évolutive permettant de centraliser, traiter et analyser toutes ces données.

Les objectifs principaux du projet sont :

- Centralisation des données dans un environnement unique, flexible et scalable.
- Traitement et transformation pour nettoyer, enrichir et préparer les données à des analyses avancées.
- Analyse et exploitation des données via des API et des outils de visualisation.
- Sécurisation et gouvernance pour garantir la conformité, la qualité et la traçabilité des données tout au long de leur cycle de vie.

En offrant une architecture capable de répondre à ces enjeux, le Data Lake permettra à l'entreprise d'améliorer sa prise de décision, de renforcer sa compétitivité et d'ouvrir la voie à des usages innovants tels que l'intelligence artificielle ou l'analyse prédictive.



2. CHOIX DES TECHNOLOGIES POUR LE STOCKAGE

2.1. Présentation des outils sélectionnés

Dans le cadre de ce projet de création d'un Data Lake, plusieurs technologies ont été sélectionnées pour répondre aux besoins d'ingestion, de stockage et de transformation des données. Les technologies choisies sont les suivantes:

- Apache Kafka : Une plateforme de streaming distribuée utilisée pour l'ingestion de données en temps réel.
- MongoDB Atlas : Une base de données NoSQL entièrement gérée dans le Cloud, idéale pour stocker des données semi-structurées au format JSON.
- Kafka Connect : Un outil intégré à Kafka pour connecter facilement des sources et des cibles de données (par exemple, bases de données ou systèmes de fichiers). Mais l'implémentation de cet outil est encore en exécution.
- Apache Spark : Une technologie de traitement distribué puissante utilisée pour effectuer des transformations et des analyses sur des données volumineuses.

2.2. Justification des choix technologiques

1. Apache Kafka

- Scénario d'utilisation : Permet l'ingestion de flux de données en temps réel provenant de sources variées, comme les logs serveurs ou les campagnes publicitaires.
- Avantages :
 - Haute scalabilité, capable de gérer de gros volumes de données en temps réel.
 - Résilience élevée grâce à la réplication et au stockage distribué.
 - Faible latence, adaptée aux applications nécessitant une faible réponse dans les flux de données.

2. MongoDB Atlas

- Scénario d'utilisation : Idéal pour le stockage des données semi-structurées issues des médias sociaux (format JSON).
- Avantages :
 - Gestion efficace des documents JSON grâce à son architecture orientée document.
 - Service Cloud entièrement géré, réduisant l'effort de maintenance.
 - Support de fonctionnalités avancées telles que les indexations complexes et les recherches full-text.



3. Kafka Connect

- Scénario d'utilisation : Simplifie l'intégration entre Kafka et MongoDB Atlas pour l'ingestion de données structurées ou non structurées.
- Avantages :
 - Déploiement rapide grâce à une bibliothèque riche de connecteurs prêts à l'emploi.
 - Transformations légères sur les données avant leur arrivée dans la base cible.
 - Évolutivité pour gérer des pipelines complexes.

4. Apache Spark

- Scénario d'utilisation : Transformation et enrichissement des données dans le Data Lake.
- Avantages :
 - Traitement distribué pour gérer de grands volumes de données.
 - Compatibilité avec différents formats de données (JSON, Parquet, etc.).
 - Framework unifié pour le traitement batch et le streaming.

2.3. Impact des choix technologiques sur le projet

- Flexibilité et évolutivité : Les outils choisis permettent de gérer des volumes de données croissants et de s'adapter aux besoins futurs de l'entreprise.
- Simplicité d'intégration : Kafka Connect facilite l'interconnexion entre les systèmes, et Spark offre une interface unifiée pour le traitement batch et streaming.
- Réduction des coûts de maintenance : MongoDB Atlas étant une solution Cloud gérée, elle diminue l'effort nécessaire pour gérer et surveiller les bases de données.
- Performance optimisée : Grâce à Kafka et Spark, le projet bénéficie d'une ingestion rapide et d'un traitement efficace des données.





3. DOCUMENTATION TECHNIQUE SUR LES CHOIX DE TECHNOLOGIES ET SOLUTIONS CLOUD

3.1. Choix des solutions Cloud (Travail en cours d'exécution)

L'infrastructure de ce Data Lake a été déployée dans un environnement cloud pour garantir scalabilité, flexibilité et disponibilité. Le choix du cloud est motivé par les avantages suivants :

- **Évolutivité**: Le cloud permet de faire face à des volumes de données croissants sans avoir à investir dans des infrastructures physiques.
- **Flexibilité**: Les ressources peuvent être ajustées dynamiquement en fonction des besoins.
- **Coût maîtrisé**: Les solutions cloud fonctionnent selon un modèle pay-per-use, ce qui permet de ne payer que pour ce qui est consommé.
- **Haute disponibilité**: Le cloud garantit une réplication et une redondance des données, ce qui minimise les risques de pannes et assure la disponibilité continue.

Pour ce projet, Google Cloud Platform (GCP) sera choisi comme environnement cloud principal en raison de ses services de stockage performants, de ses outils d'orchestration et de ses solutions de sécurité robustes.

3.2. Solutions de stockage et de gestion des données

1. MongoDB Atlas :

Pour la gestion des données semi-structurées, MongoDB Atlas a été choisi en raison de sa capacité à stockage flexible et sa gestion cloud native.

- **Multi-régions** : MongoDB Atlas offre une réplication mondiale, permettant d'assurer une haute disponibilité et une récupération rapide en cas de panne.
- **Gestion automatique** : La gestion de la base de données, de la mise à l'échelle et des backups est entièrement automatisée, ce qui réduit les coûts opérationnels.





3.3. Solutions de traitement des données

1. Apache Kafka et Kafka Connect :

- Apache Kafka est utilisé pour le streaming des données en temps réel, permettant une ingestion rapide des flux de données tels que les logs des serveurs web, les données des médias sociaux et les campagnes publicitaires. Kafka est déployé en utilisant les services managés de GCP, ce qui permet de bénéficier d'une infrastructure gérée tout en conservant la flexibilité de personnalisation.
- Kafka Connect est utilisé pour l'intégration avec diverses sources de données externes (bases de données SQL, fichiers CSV, etc.). Cette solution réduit la complexité de l'intégration et permet une ingestion fluide des données en provenance d'autres systèmes.

Par contre, pour Kafka Connect on a eu du mal à l'implémenter.


2. Apache Spark :

Pour les opérations de traitement des données massives, Apache Spark est utilisé pour exécuter des pipelines ETL (Extract, Transform, Load). Spark permet de traiter de grands ensembles de données rapidement grâce à son moteur de traitement parallèle et son calcul en mémoire. Il est particulièrement adapté pour les processus de transformation complexes et le nettoyage des données avant leur ingestion dans des systèmes comme BigQuery ou MongoDB Atlas.

- Google Dataproc a été choisi comme solution managée pour déployer Spark sur GCP, facilitant ainsi l'installation et la gestion des clusters Spark.
- Les transformations sont exécutées en batch ou en temps réel via Spark Streaming.

3.4. Développement de l'API :

Nous travaillons actuellement sur le développement d'une API permettant d'interagir avec les données stockées dans le Data Lake. L'objectif principal est de fournir une interface centralisée et standardisée pour accéder, rechercher, et manipuler les données de manière efficace.





Fonctionnalités principales de l'API

1. Rechercher des données dans MongoDB Atlas :

L'API utilise la fonctionnalité Search Index de MongoDB Atlas, une solution puissante pour les recherches avancées sur les données semi-structurées.

- Search Index offre des capacités de recherche textuelle rapide, prenant en charge des fonctionnalités avancées comme :
 - La recherche par mots-clés avec une tolérance aux fautes d'orthographe (fuzzy matching).
 - Les filtres complexes combinant plusieurs critères.
 - La prise en charge des requêtes géospatiales pour des données localisées.
- Cela permet aux utilisateurs d'extraire rapidement des données spécifiques répondant à leurs besoins analytiques ou opérationnels.


2. Accès aux statistiques en temps réel :

L'API permettra de consulter des statistiques sur les données ingérées et traitées, telles que :

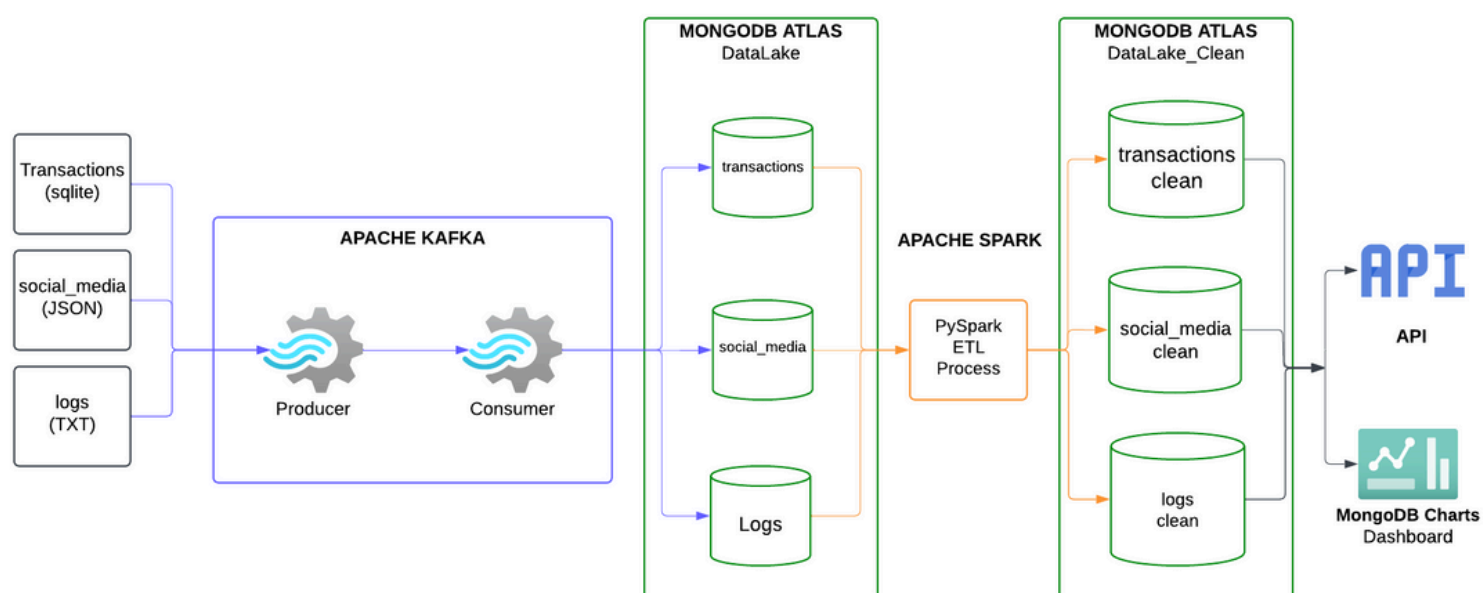
- Le volume de données ingérées quotidiennement.
- Les données rejetées ou conformes après validation.
- Les temps de traitement des pipelines ETL.
- Ces statistiques seront présentées via des endpoints optimisés pour minimiser les temps de réponse, aidant ainsi à surveiller la performance et la qualité du Data Lake.

3. Intégration avec des systèmes internes et externes :

L'API sera utilisée comme un point d'accès pour d'autres applications, permettant :

- La récupération des données nécessaires à des outils BI (Business Intelligence).
 - L'intégration facile avec des systèmes internes tels que des tableaux de bord ou des applications analytiques.
 - La création de rapports automatisés basés sur les données du Data Lake.
- 

ARCHITECTURE DE NOTRE SOLUTION





4. DOCUMENTATION SUR LES TRANSFORMATIONS APPLIQUÉES AUX DONNÉES

4.1. Définition des transformations de données

Les transformations de données désignent les processus utilisés pour modifier, nettoyer et enrichir les données brutes afin qu'elles soient prêtes à l'analyse. Cela inclut des actions comme la suppression des doublons, l'enrichissement des données, la conversion de formats ou encore l'agrégation. Ces transformations sont cruciales pour garantir la qualité des données, leur cohérence et leur adéquation avec les objectifs d'analyse. Elles sont souvent réalisées avec des outils comme Apache Spark, qui permet de traiter des volumes massifs de données de manière distribuée.

4.2. Travail effectué : Dédoublonnage des données

Dans ce projet, j'ai principalement effectué une transformation de dédoublonnage sur les données collectées. Cela consiste à identifier et supprimer les enregistrements en double dans les différentes sources de données (transactions, logs, etc.), afin de ne conserver que des entrées uniques. Le dédoublonnage a été réalisé en utilisant Apache Spark, qui est particulièrement adapté pour traiter de grandes quantités de données de manière distribuée. À l'aide de fonctions spécifiques dans Spark, j'ai pu comparer les enregistrements et éliminer ceux qui étaient identiques, ce qui a permis de nettoyer les données avant qu'elles ne soient stockées dans MongoDB Atlas ou Google Cloud Storage (GCS).

Cela a été l'une des transformations principales, car le nettoyage des doublons est essentiel pour assurer la fiabilité des données à analyser. Bien que d'autres types de transformations, comme l'enrichissement ou la normalisation, soient également nécessaires, le dédoublonnage a constitué l'étape clé de traitement pour garantir la qualité des données dans le Data Lake.





6. CONCLUSION

6.1. Synthèse des travaux réalisés

Au cours de ce projet, nous avons conçu et déployé une architecture de Data Lake pour centraliser et traiter les données issues de différentes sources (transactions, logs, médias sociaux, et données en temps réel). Nous avons choisi des technologies adaptées telles que Kafka, MongoDB Atlas, Kafka Connect et Apache Spark pour assurer une ingestion, une transformation et un stockage efficaces des données.

Nous avons mis en place des pipelines d'ingestion pour traiter les données structurées, semi-structurées et non structurées, ainsi que des transformations essentielles, comme le dédoublonnage, pour garantir la qualité des données.

6.2. Prochaines étapes possibles pour améliorer la solution

Malgré les réalisations, plusieurs pistes d'amélioration peuvent être envisagées pour renforcer la solution :

1. **Enrichissement des données** : Au-delà du dédoublonnage, des transformations supplémentaires telles que l'enrichissement des données à partir de sources externes ou l'agrégation des données en temps réel pourraient être intégrées pour offrir des analyses plus approfondies.
2. **Automatisation des pipelines de données** : L'automatisation complète des pipelines ETL (Extraction, Transformation, Chargement) en intégrant des outils comme Airflow ou Apache NiFi permettrait de garantir une gestion fluide et continue des flux de données.
3. **Amélioration de la gouvernance des données** : L'ajout de processus de gestion de la qualité des données, comme la détection d'anomalies ou la gestion des valeurs manquantes, permettrait d'assurer une meilleure intégrité des données à long terme.
4. **Scalabilité et performance** : Pour faire face à une augmentation des volumes de données, des optimisations sur la scalabilité et la performance du Data Lake, telles que le partitionnement des données ou l'utilisation de solutions de stockage à faible coût comme Amazon S3, pourraient être mises en œuvre.





5. **Visualisation et analyse avancée** : Intégrer des outils de business intelligence (comme Power BI ou Tableau) pour fournir des tableaux de bord interactifs et des rapports plus détaillés serait une manière de valoriser encore plus les données collectées.

Ces étapes permettront d'améliorer la solution de Data Lake, de renforcer sa résilience et d'optimiser les processus d'analyse, tout en offrant de nouvelles perspectives pour l'entreprise en matière de prise de décision basée sur les données.

