Rapport: Traitement des Transactions avec Spark et MinIO

Lucas CHIPAN Lyes BOUMRAH

1 Introduction

L'objectif de ce projet est d'intégrer un **pipeline de traitement des données** en temps réel en utilisant :

- **Kafka** pour la réception des transactions.
- Apache Spark pour la transformation et le stockage des données.
- MinIO comme solution de stockage objet en format Parquet.

Ce document détaille l'implémentation Scala avec Spark Streaming et l'envoi des fichiers traités vers MinIO.

2 Architecture du Système d'Information (SI)

Le système est composé de :

- Kafka : Diffusion des transactions en temps réel.
- Spark Streaming: Transformation des données.
- MinIO (S3): Stockage des fichiers Parquet transformés.

3 Implémentation de l'Étape 3 - Traitement et Stockage

3.1 Traitement des transactions

- 1. Connexion à Kafka : Récupération des transactions en temps réel.
- 2. Transformation des données :
 - Conversion des montants USD en EUR.
 - Conversion des dates et application du fuseau horaire Europe/Paris.
 - o Suppression des transactions erronées et des valeurs nulles.
- 3. **Stockage sur MinIO**: Sauvegarde des transactions sous format **Parquet compressé** (Snappy).

3.2 Explication du Processus

- Lecture en continu des messages Kafka avec Spark Streaming.
- Transformation des données pour assurer la cohérence et la qualité des informations.
- Stockage dans MinIO sous format Parquet, optimisé pour l'analyse de données.

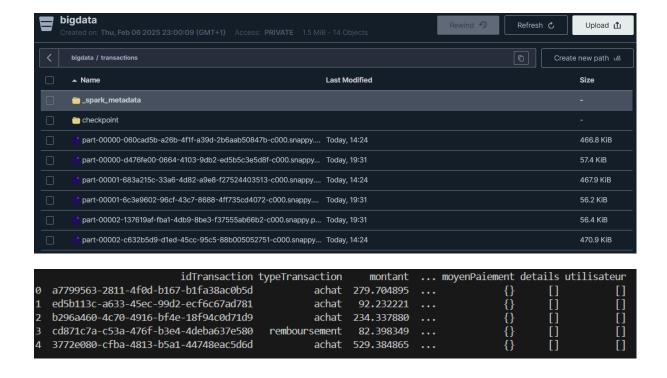
3.3 Résultats Attendus

Les transactions avant transformation contiennent des devises différentes et des dates au format texte.

Après transformation:

- Les montants sont convertis en EUR.
- Les dates sont standardisées avec le fuseau horaire Europe/Paris.
- Les transactions erronées sont filtrées.

Les fichiers sont stockés sur **MinIO** dans bigdata/transactions/ et peuvent être analysés avec Spark.



4 Conclusion

- Implémentation du pipeline Kafka → Spark → MinIO.
- Stockage des transactions en Parquet compressé (Snappy).
- Possibilité de relire les données avec Spark.