Projet étudiant
MASTER II

DATA SCIENCE & ANALYSE DE DONNEES

DIFFUSION EN CONTINU STRUCTURÉE AVEC APACHE SPARK

Proposé par :

M. MOUNGOLE ARISTIDE

Réalisé par :

HYONTA KENGAP BLERIOT

Année : 2024/2025

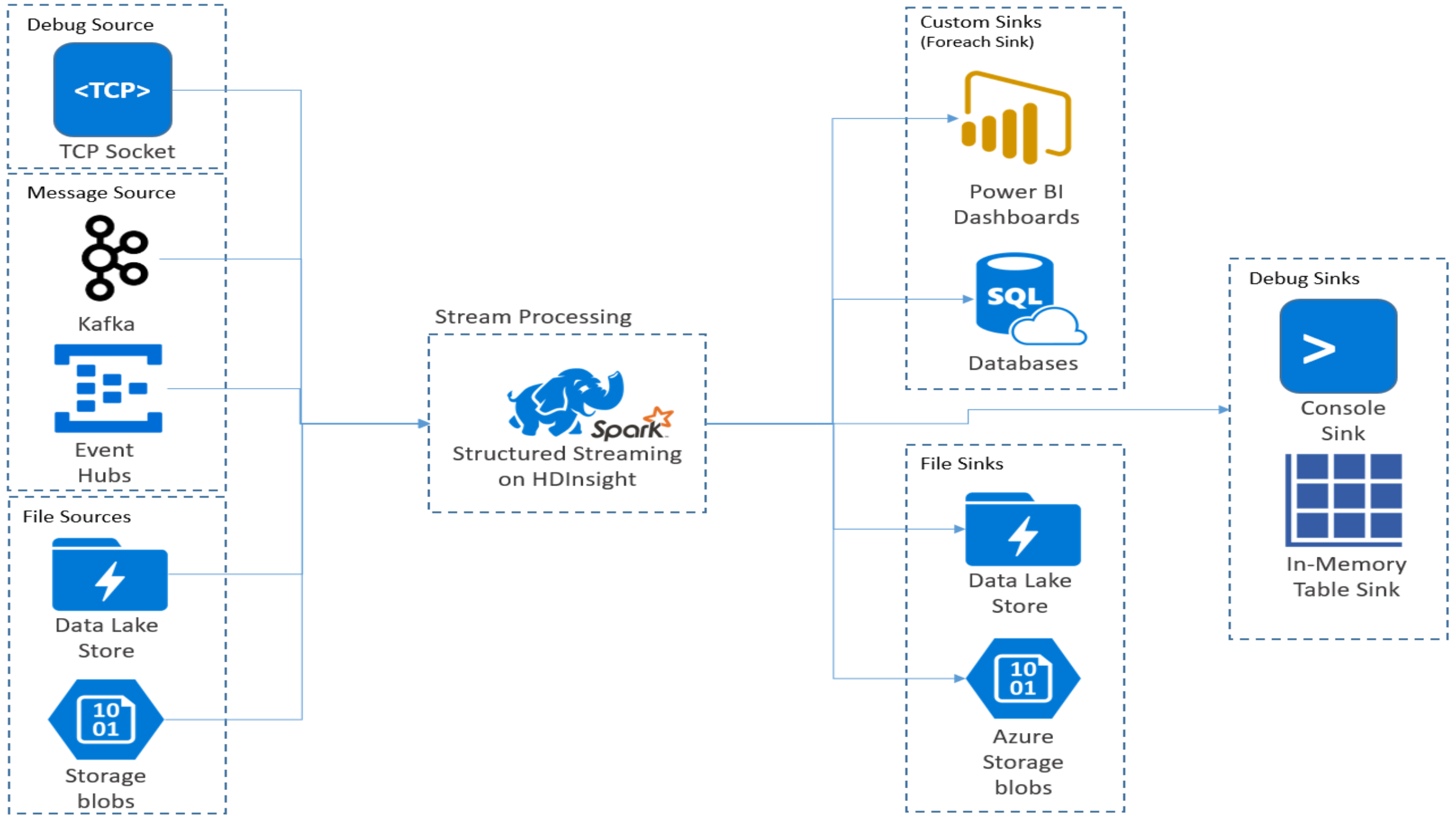

Debug Source
<TCP>
TCP Socket
Message Source
Kafka
Event Hubs
File Sources
Data Lake Store
Storage blobs
Stream Processing
Spark
Structured Streaming on HDInsight
Custom Sinks (Foreach Sink)
Power BI Dashboards
SQL
Databases
File Sinks
Data Lake Store
Azure Storage blobs
Debug Sinks
Console Sink
In-Memory Table Sink

Table des matières

Introduction à la Diffusion en Continu Structurée	01
Qu'est-ce que la Diffusion en Continu Structurée ?	02
Technologies Utilisées	03
Contexte Camerounais et Applications	04
Secteur Financier : Détection de Fraudes	05
Architecture Globale du Système	06
Exemple de Code : Producteur Kafka en Python	07
Traitement avec Apache Spark en Scala	08
Démonstration : Détection de Fraude en Temps Réel	09
Conclusion	10

Introduction à la Diffusion en Continu Structurée

Introduction à la Diffusion en Continu Structurée

La diffusion en continu structurée est une approche moderne du traitement des flux de données en temps réel.

Cette approche gère efficacement les données de différentes sources.

Applications Potentielles

Dans le domaine de l'éducation et de la formation, elle peut être utilisée pour fournir des contenus d'apprentissage en temps réel.

En marketing et communication, elle permet d'analyser les comportements des consommateurs et d'ajuster les stratégies en conséquence.

Dans la recherche et développement, elle facilite l'accès aux données et aux résultats d'expériences en temps réel.

Importance de la Diffusion

Elle facilite la circulation des idées et des informations entre les utilisateurs.

Elle renforce la collaboration entre les individus et les organisations, favorisant ainsi un environnement de travail dynamique.

Elle contribue à l'innovation et à l'amélioration des processus en permettant une analyse rapide des données.

Conclusion

La diffusion en continu structurée représente une avancée significative dans le traitement des données, offrant des solutions adaptées à divers secteurs.

Son adoption peut transformer la manière dont les organisations gèrent et analysent les informations.

Qu'est-ce que la Diffusion en Continu Structurée ?

Définition de la diffusion en continu structurée

La diffusion en continu structurée est un moteur de traitement de flux de données en temps réel introduit dans Apache Spark à partir de la version 2.0.

Elle repose sur un modèle de programmation déclaratif utilisant Spark SQL.

Comparaison avec d'autres technologies de traitement de flux

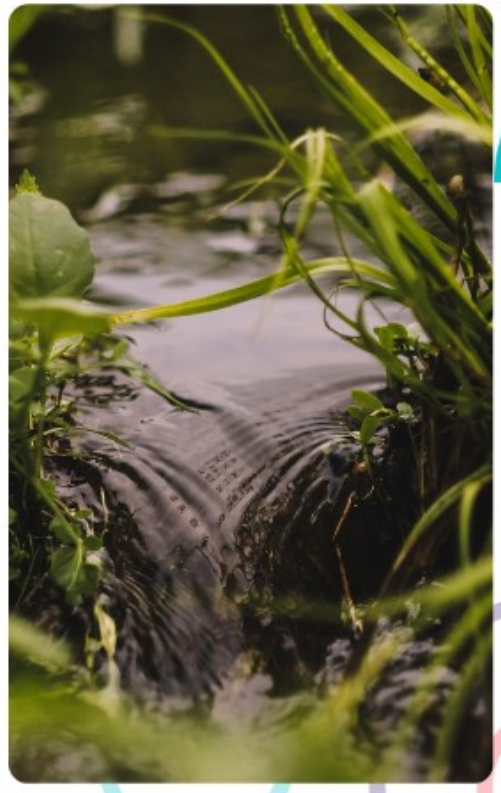
Contrairement aux approches traditionnelles comme Apache Storm ou Flink, la diffusion en continu structurée utilise un modèle de micro-lots.

Cette méthode permet une gestion efficace des états et une intégration native avec l'écosystème Spark.

Avantages spécifiques de l'approche

Elle permet de traiter les flux de données de manière similaire aux bases de données relationnelles.

La diffusion en continu structurée est particulièrement adaptée pour des applications nécessitant une détection de fraudes en temps réel.



Technologies Utilisées

Technologies Utilisées

Présentation des technologies utilisées dans le projet.

Apache Spark

Apache Spark est un moteur de traitement de flux utilisé pour le traitement en temps réel des données.

Apache Kafka

Apache Kafka est un système de messagerie distribué pour la diffusion des données en temps réel.

PostgreSQL

PostgreSQL est une base de données relationnelle pour stocker les résultats du traitement.

Scala

Scala est le langage de programmation utilisé pour développer l'application Spark.

Contexte Camerounais et Applications

Contexte Camerounais et Applications

La diffusion en continu structurée est essentielle pour le traitement des données en temps réel au Cameroun.

Elle a des applications significatives dans le secteur bancaire, notamment pour la détection de fraudes.

Secteur Financier : Détection de Fraudes

Mécanisme de Détection de Fraudes

Les transactions dépassant un seuil de 10 000 XAF sont considérées comme suspectes et stockées pour une analyse approfondie.

Ce système permet aux banques camerounaises de prendre des mesures immédiates pour enquêter et éventuellement bloquer une transaction.

Traitement en Temps Réel

L'utilisation d'Apache Spark permet un traitement en temps réel des données, essentiel pour améliorer la sécurité et l'efficacité des systèmes financiers.

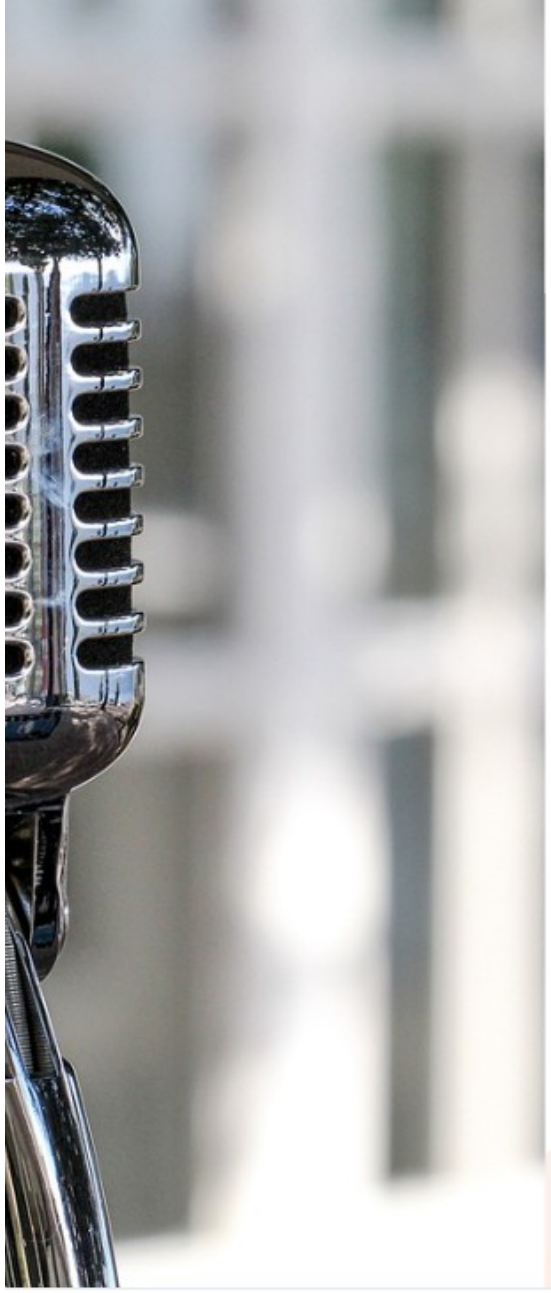
Les données sont analysées instantanément pour détecter des activités frauduleuses.

Surveillance des Transactions

La diffusion en continu structurée permet de surveiller les transactions en temps réel et d'identifier des anomalies.

Des critères spécifiques, tels que la fréquence anormale et la localisation suspecte, sont utilisés pour détecter les fraudes.

Architecture Globale du Système



Exemple de Code : Producteur Kafka en Python

Exemple de Code : Producteur Kafka en Python

Ce script envoie des transactions à Kafka.

```
from kafka import KafkaProducer
import json
import time
import random

def generate_transaction():
    return {
        'id': random.randint(1000, 9999),
        'transaction_id': random.randint(10000, 99999),
        'amount': round(random.uniform(100, 20000), 2),
        'currency': 'XAF',
        'timestamp': time.strftime('%Y-%m-%d %H:%M:%S')
    }

producer = KafkaProducer(
    bootstrap_servers='localhost:9092',
    value_serializer=lambda v: json.dumps(v).encode('utf-8')
)

while True:
    transaction = generate_transaction()
    producer.send('transaction', transaction)
    print(f'Transaction envoyée : {transaction}')
    time.sleep(1)
```

Traitement avec Apache Spark en Scala

Filtrage des Transactions en Temps Réel

Le script lit les flux de transactions à partir de Kafka.

Les transactions dont le montant dépasse 10 000 XAF sont filtrées.

Logique de Détection de Fraude

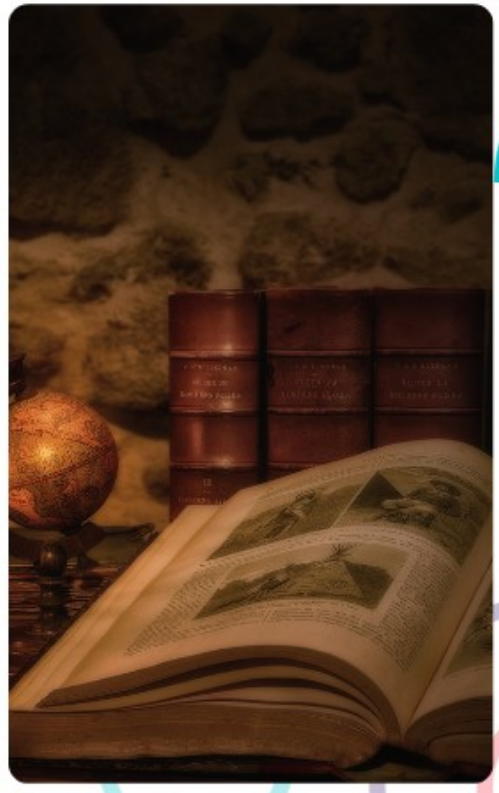
Les transactions suspectes sont stockées pour une analyse approfondie.

Le système permet aux banques de prendre des mesures immédiates.

Stockage des Données dans PostgreSQL

Les transactions filtrées sont écrites dans une base de données PostgreSQL.

Cela facilite le suivi et l'analyse des transactions suspectes.



Démonstration : Détection de Fraude en Temps Réel

{"id": 1334, "transaction_id": 157411, "amount": 3952.89, "currency": "GBP", "timestamp": "2025-02-26 03:31:35"}
{"id": 9081, "transaction_id": 378171, "amount": 18857.46, "currency": "EUR", "timestamp": "2025-02-26 03:31:37"}
{"id": 7348, "transaction_id": 622969, "amount": 5769.54, "currency": "EUR", "timestamp": "2025-02-26 03:31:39"}
{"id": 4021, "transaction_id": 331904, "amount": 12662.73, "currency": "GBP", "timestamp": "2025-02-26 03:31:41"}
{"id": 3906, "transaction_id": 594472, "amount": 7088.24, "currency": "USD", "timestamp": "2025-02-26 03:31:43"}
{"id": 9187, "transaction_id": 982740, "amount": 14591.37, "currency": "EUR", "timestamp": "2025-02-26 03:31:45"}
{"id": 4011, "transaction_id": 764490, "amount": 15501.92, "currency": "EUR", "timestamp": "2025-02-26 03:31:47"}
{"id": 8560, "transaction_id": 316545, "amount": 13660.73, "currency": "EUR", "timestamp": "2025-02-26 03:31:49"}
{"id": 4805, "transaction_id": 731621, "amount": 8147.23, "currency": "GBP", "timestamp": "2025-02-26 03:31:51"}
{"id": 6555, "transaction_id": 716213, "amount": 15981.36, "currency": "EUR", "timestamp": "2025-02-26 03:31:53"}
{"id": 8352, "transaction_id": 581938, "amount": 14781.34, "currency": "EUR", "timestamp": "2025-02-26 03:31:55"}
{"id": 6394, "transaction_id": 413415, "amount": 7102.13, "currency": "GBP", "timestamp": "2025-02-26 03:31:57"}
{"id": 1313, "transaction_id": 302745, "amount": 16061.04, "currency": "GBP", "timestamp": "2025-02-26 03:31:59"}
{"id": 2637, "transaction_id": 144903, "amount": 13105.75, "currency": "USD", "timestamp": "2025-02-26 03:32:01"}
{"id": 3200, "transaction_id": 434342, "amount": 16383.45, "currency": "EUR", "timestamp": "2025-02-26 03:32:03"}
{"id": 3748, "transaction_id": 556026, "amount": 11431.5, "currency": "EUR", "timestamp": "2025-02-26 03:32:05"}
{"id": 4120, "transaction_id": 740536, "amount": 12341.02, "currency": "USD", "timestamp": "2025-02-26 03:32:07"}
{"id": 6425, "transaction_id": 276286, "amount": 3414.7, "currency": "EUR", "timestamp": "2025-02-26 03:32:09"}
{"id": 4047, "transaction_id": 281913, "amount": 19204.72, "currency": "GBP", "timestamp": "2025-02-26 03:32:11"}
{"id": 5415, "transaction_id": 740258, "amount": 10088.04, "currency": "GBP", "timestamp": "2025-02-26 03:32:13"}
{"id": 1334, "transaction_id": 798598, "amount": 3879.19, "currency": "USD", "timestamp": "2025-02-26 03:32:15"}
{"id": 7857, "transaction_id": 414190, "amount": 8726.77, "currency": "USD", "timestamp": "2025-02-26 03:32:17"}
{"id": 6382, "transaction_id": 332469, "amount": 9276.29, "currency": "USD", "timestamp": "2025-02-26 03:32:19"}
{"id": 5590, "transaction_id": 876873, "amount": 15538.49, "currency": "USD", "timestamp": "2025-02-26 03:32:21"}
{"id": 9890, "transaction_id": 134630, "amount": 18255.96, "currency": "USD", "timestamp": "2025-02-26 03:32:23"}
{"id": 9575, "transaction_id": 176797, "amount": 14628.68, "currency": "EUR", "timestamp": "2025-02-26 03:32:25"}
{"id": 9264, "transaction_id": 955414, "amount": 7651.83, "currency": "GBP", "timestamp": "2025-02-26 03:32:27"}
{"id": 4867, "transaction_id": 396549, "amount": 6936.14, "currency": "EUR", "timestamp": "2025-02-26 03:32:29"}
{"id": 1825, "transaction_id": 799839, "amount": 824.52, "currency": "EUR", "timestamp": "2025-02-26 03:32:31"}
{"id": 4740, "transaction_id": 327190, "amount": 1932.12, "currency": "GBP", "timestamp": "2025-02-26 03:32:33"}
{"id": 8325, "transaction_id": 513274, "amount": 5272.91, "currency": "USD", "timestamp": "2025-02-26 03:32:35"}
{"id": 2367, "transaction_id": 795169, "amount": 14054.16, "currency": "USD", "timestamp": "2025-02-26 03:32:37"}
{"id": 8678, "transaction_id": 728844, "amount": 4252.37, "currency": "USD", "timestamp": "2025-02-26 03:32:39"}
{"id": 4388, "transaction_id": 638664, "amount": 18407.92, "currency": "EUR", "timestamp": "2025-02-26 03:32:41"}
{"id": 2987, "transaction_id": 935758, "amount": 2931.42, "currency": "USD", "timestamp": "2025-02-26 03:32:43"}
{"id": 5193, "transaction_id": 903476, "amount": 11572.76, "currency": "EUR", "timestamp": "2025-02-26 03:32:45"}
{"id": 8764, "transaction_id": 599182, "amount": 16249.61, "currency": "USD", "timestamp": "2025-02-26 03:32:47"}
{"id": 3869, "transaction_id": 605021, "amount": 15280.69, "currency": "EUR", "timestamp": "2025-02-26 03:32:49"}
{"id": 2936, "transaction_id": 319022, "amount": 16225.38, "currency": "USD", "timestamp": "2025-02-26 03:32:51"}

Conclusion

Résumé des points clés du projet

La diffusion en continu structurée avec Apache Spark permet de traiter des flux de données en temps réel, en particulier pour la détection de fraudes dans le secteur bancaire camerounais.

Avantages de la diffusion en continu structurée

Cette technologie offre une gestion efficace des données, permettant aux banques de détecter immédiatement des activités suspectes et de renforcer la sécurité des transactions.

Perspectives d'avenir

Avec l'augmentation des transactions numériques, l'adoption de solutions de traitement en temps réel est essentielle pour améliorer la sécurité et l'efficacité des systèmes financiers au Cameroun.

Mercie pour votre attention