

## Détection de Fraude en Streaming avec Kafka & Spark

### Structure du projet :

```
flux_donnee/
├── .vscode/          # Configurations VS Code
├── app/
│   ├── spark_streaming.py # Script Spark pour détecter les fraudes en temps réel
│   ├── kafka_producer.py  # Script Python pour générer des transactions fictives
│   ├── Dockerfile.producer # Dockerfile du service Producer Kafka
│   ├── docker-compose.yml  # Lancement des services (Kafka, Spark, etc.)
│   └── README.md          # (Ce fichier)
```

### Objectif :

Ce projet met en place une chaîne de traitement de données en streaming à l'aide de Kafka et Spark Structured Streaming pour détecter différentes formes de fraude financière à partir de transactions synthétiques générées en continu.

### **Fichiers :**

#### *kafka\_producer.py*

- Génère des transactions simulées (user\_id, montant, lieu, devise, etc.)
- Envoie les données vers le topic Kafka *transactions*
- Paramètres :
  - *rate\_per\_second* : nombre de transactions générées par seconde
  - *total\_transactions* : nombre total de transactions à envoyer

#### *spark\_streaming.py*

Ce script, déployé dans un conteneur Spark, consomme les données du topic Kafka *transactions* et effectue la détection de fraudes en temps réel.

### Règles de détection :

1. Montant élevé supérieur à 45 000 € : *HIGH\_VALUE*
2. Transactions fréquentes (3 ou plus en 5 minutes) *HIGH\_FREQUENCY*
3. Changement géographique rapide *GEO\_SWITCH*
4. Utilisation de plusieurs devises en peu de temps *CURRENCY\_SWITCH*
5. Petits montants chez plusieurs marchands différents *CAROUSEL\_FRAUD*

Les alertes détectées sont :

- affichées dans la console
- enregistrées en fichiers Parquet dans *output/parquet\_alerts*

- envoyées vers le topic Kafka *fraud-alerts*

### Dockerfile.producer

Image Docker pour exécuter le producteur Kafka écrit en Python.

### docker-compose.yml

Orchestre les services suivants :

- zookeeper : coordination pour Kafka
- kafka : broker Kafka
- producer : génère les transactions fictives
- spark : traitement en streaming avec détection de fraude

## Lancement du projet

Pour lancer tous les services (Kafka, Producer, Spark) :

```
docker-compose up --build
```

## Dossiers générés automatiquement

Ces dossiers sont créés lors de l'exécution de Spark :

```
output/
├─ parquet_alerts/      # Fichiers Parquet des fraudes détectées
├─ checkpoint_parquet/  # Checkpoints Spark pour l'écriture Parquet
└─ checkpoint_kafka/    # Checkpoints Spark pour l'écriture vers Kafka
```

## Visualisation des résultats

### Console Kafka

Pour afficher les alertes détectées en temps réel :

```
docker exec -it flux_donnee-kafka-1 kafka-console-consumer \
  --bootstrap-server localhost:9092 \
  --topic fraud-alerts \
  --from-beginning
```

**Fichiers Parquet** : Les alertes sont également disponibles dans le dossier *output/parquet\_alerts*.

**Topic Kafka** : Les alertes sont aussi publiées dans le topic Kafka *fraud-alerts*.

## Dashboard Streamlit

Nous avons développé une interface de visualisation en Streamlit pour suivre la détection de fraudes en temps réel, avec un rafraîchissement automatique toutes les 5 secondes.

L'interface est accessible depuis n'importe quel navigateur à l'adresse <http://localhost:8501>.

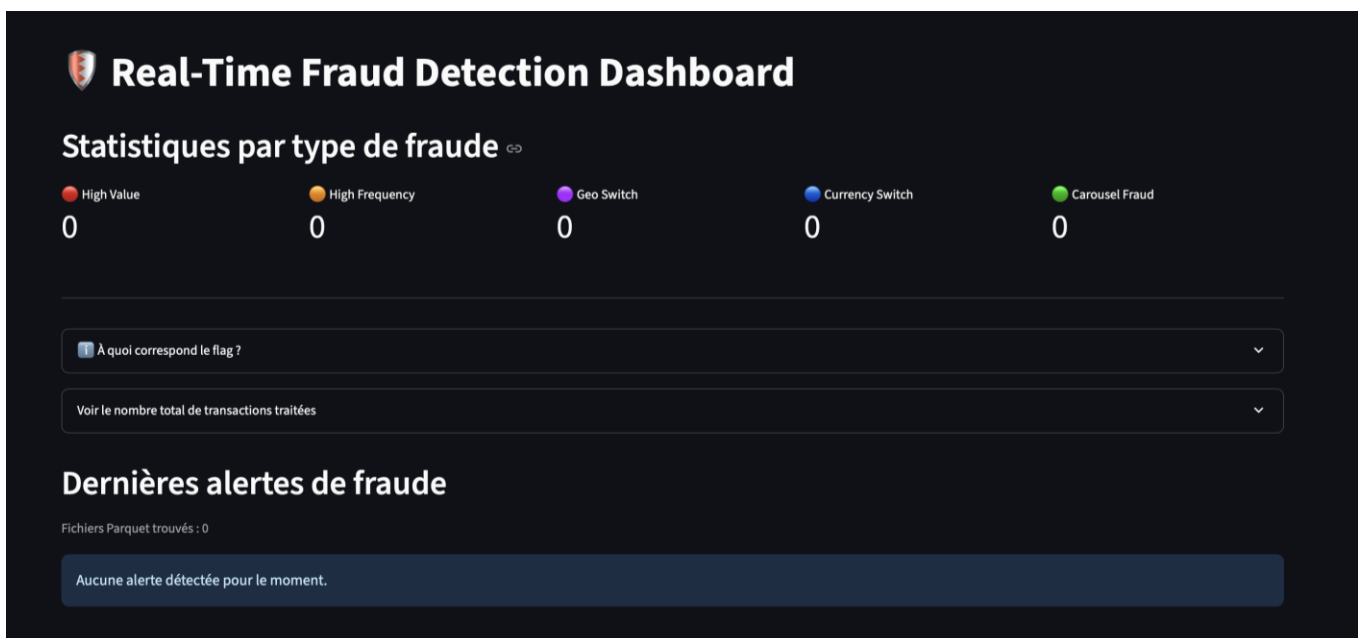
### Fonctionnalités de l'interface :

- Compteur par type de fraude détectée
- Descriptions détaillées de chaque type de fraude simulée
- Compteur global du nombre total de fraudes détectées depuis le lancement
- Tableau des fraudes alimenté dynamiquement à chaque nouvelle détection, incluant :
  - transaction\_id
  - user\_id
  - type de fraude
  - flag associé

### Interface à l'état initial

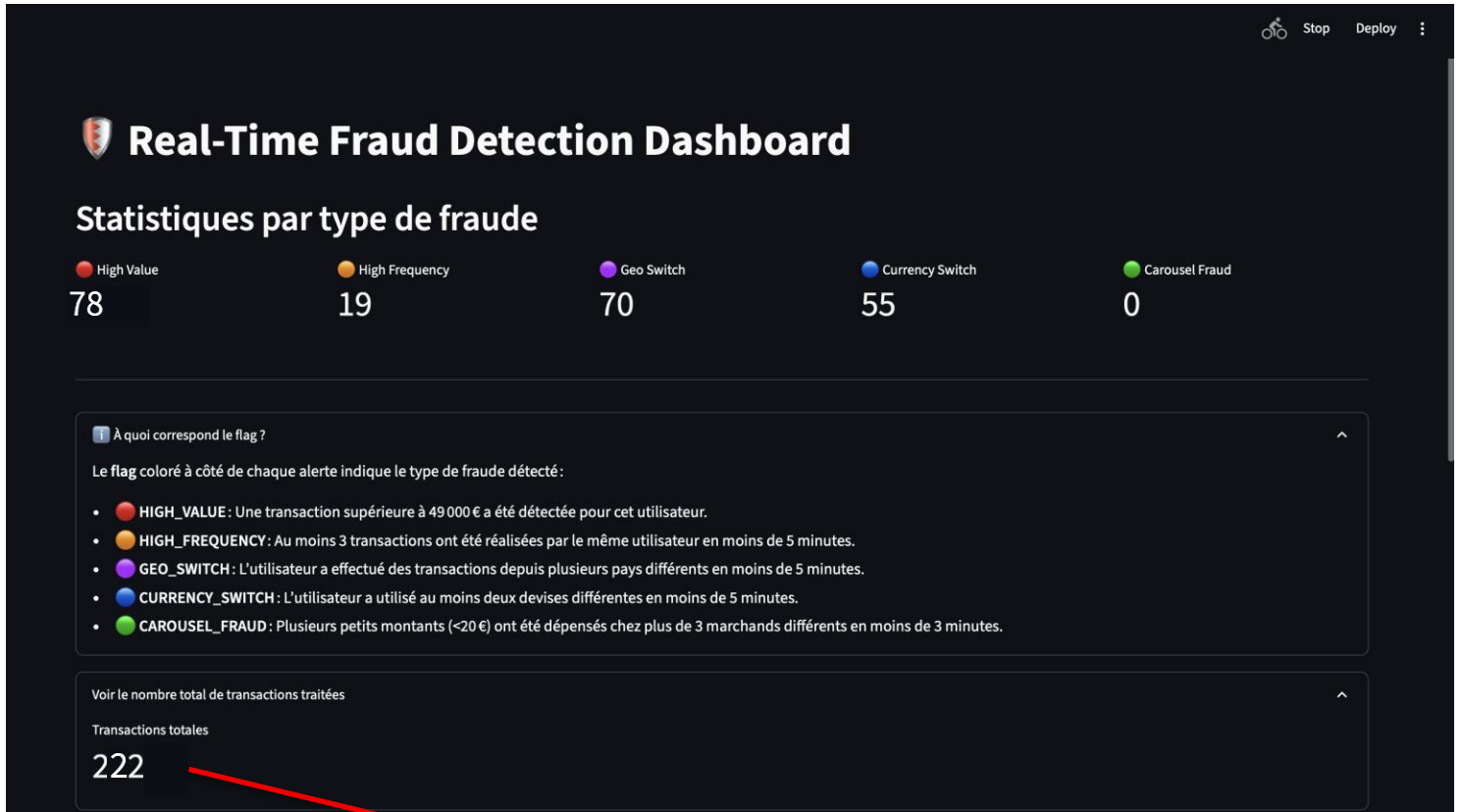
Lorsque Spark n'a pas encore traité de transactions, l'interface affiche des compteurs à zéro et le tableau est vide.

```
-----  
Batch: 0  
-----  
+-----+-----+-----+-----+  
|user_id|transaction_id|timestamp|fraud_type|  
+-----+-----+-----+-----+  
+-----+-----+-----+-----+
```



## Rafraîchissement en temps réel

Toutes les 5 secondes, les compteurs sont automatiquement mis à jour afin d'indiquer les dernières fraudes détectées.



Ici on compte le nombre total de transactions qui ont été détectées en tant que fraudes financières











Batch: 1

user_id	transaction_id	timestamp	fraud_type
u425	t-0000801	2025-06-01 00:13:21	HIGH_VALUE
u5268	t-0000814	2025-06-01 00:13:34	HIGH_VALUE
u6675	t-0000815	2025-06-01 00:13:35	HIGH_VALUE
u2050	t-0000820	2025-06-01 00:13:40	HIGH_VALUE
u5997	t-0000833	2025-06-01 00:13:53	HIGH_VALUE
u3880	t-0000849	2025-06-01 00:14:09	HIGH_VALUE
u5288	t-0000853	2025-06-01 00:14:13	HIGH_VALUE
u2831	t-0000862	2025-06-01 00:14:22	HIGH_VALUE
u9260	t-0000871	2025-06-01 00:14:31	HIGH_VALUE
u571	t-0000877	2025-06-01 00:14:37	HIGH_VALUE
u6705	t-0000883	2025-06-01 00:14:43	HIGH_VALUE
u3976	t-0000899	2025-06-01 00:14:59	HIGH_VALUE
u4716	t-0000902	2025-06-01 00:15:02	HIGH_VALUE

Le tableau au bas de la page Streamlit permet de consulter le détail des dernières fraudes détectées

## Dernières alertes de fraude

Fichiers Parquet trouvés : 133

timestamp	user_id	transaction_id	Flag
2025-06-01 06:58:04	u2678	t-0025084	 HIGH_VALUE
2025-06-01 06:57:52	u2513	t-0025072	 HIGH_VALUE
2025-06-01 06:57:47	u8122	t-0025067	 HIGH_VALUE
2025-06-01 06:57:40	u7571	t-0025060	 HIGH_VALUE
2025-06-01 06:57:30	u6684	t-0025050	 HIGH_VALUE
2025-06-01 06:57:28	u531	t-0025048	 HIGH_VALUE
2025-06-01 06:57:16	u6746	t-0025036	 HIGH_VALUE
2025-06-01 06:56:41	u2779	t-0025001	 HIGH_VALUE
2025-06-01 06:56:40	u4356	t-0025000	 HIGH_VALUE
2025-06-01 06:55:27	u4714	t-0024927	 HIGH_VALUE

timestamp	user_id	Flag	CURRENCY
2025-06-01 01:40:37	u5004	 HIGH_VALUE	1 of 55 results
2025-06-01 01:40:34	u4380	 HIGH_VALUE	
2025-06-01 01:40:28	u4943	 HIGH_VALUE	
2025-06-01 01:40:12	u5675	 HIGH_VALUE	
2025-06-01 01:40:00	u8559	 CURRENCY_SWITCH	
2025-06-01 01:40:00	u8559	 GEO_SWITCH	
2025-06-01 01:40:00	u1130	 GEO_SWITCH	
2025-06-01 01:40:00	u1130	 CURRENCY_SWITCH	
2025-06-01 01:39:54	u2613	 HIGH_VALUE	
2025-06-01 01:39:52	u1434	 HIGH_VALUE	
2025-06-01 01:39:52	u1434	 HIGH_VALUE	

### Remarques :

- High Value

Cette fraude correspond à toute transaction dont le montant dépasse **49 000 €**. Nous avons fixé ce seuil afin de simuler un cas de fraude réaliste. Par exemple, avec un seuil à 45 000 € (montant que nous avons choisi initialement), la probabilité d’avoir une transaction supérieure à cette valeur aurait été trop élevée.

Le montant de chaque transaction suit une loi uniforme discrète entre 1 et 50 000. On peut donc calculer exactement la probabilité de détecter cette fraude.

Soit  $X$  une variable aléatoire suivant une loi uniforme discrète sur l'ensemble  $\{1, 2, \dots, 50\,000\}$ .

Chaque valeur a la même probabilité :

$$\mathbb{P}(X = k) = \frac{1}{50\,000}, \quad \text{pour } k = 1, 2, \dots, 50\,000.$$

On cherche :

$$\mathbb{P}(X \geq 49\,000)$$

Le nombre d'entiers de 49 000 à 50 000 inclus est :

$$50\,000 - 49\,000 + 1 = 1\,001$$

Donc,

$$\mathbb{P}(X \geq 49\,000) = \frac{\text{nombre de valeurs possibles pour } X \geq 49\,000}{\text{nombre total de valeurs possibles}} = \frac{1\,001}{50\,000}$$

$$\mathbb{P}(X \geq 49\,000) = 0,02002$$

Ce résultat justifie qu'il est pertinent de considérer toute transaction supérieure à 49 000 € comme potentiellement frauduleuse.

- Carousel Fraud

Cette fraude est définie comme :

Plusieurs petits montants (< 20 €) ont été dépensés chez plus de 3 marchands différents en moins de 3 minutes.

Ce type de fraude, bien que rare dans notre simulation, est représentatif de techniques réelles. Dans la pratique, les fraudeurs commencent souvent par effectuer de petites transactions discrètes afin de tester si la carte volée est encore active (sans que le véritable propriétaire ne s'en rende compte immédiatement). Si ces tests passent inaperçus, ils enchaînent ensuite avec un retrait important.

C'est précisément pour simuler ce schéma réel de fraude que nous avons intégré cette règle dans notre système de détection. Cependant dans notre simulation, cette situation est totalement absente. Cela s'explique par deux facteurs :

1. Le comportement simulé est très spécifique (plus de 3 commerçants, montants < 20 €, en < 3 min), ce qui en fait une condition rare.
2. La génération aléatoire des transactions ne favorise pas ce type de schéma : les petits montants sont moins fréquents, et les identifiants de marchands sont dispersés.

Pour pouvoir vérifier que cette fraude est bien détectée par notre système, nous avons nous-mêmes généré de manière aléatoire des transactions correspondant à ce cas.

## Real-Time Fraud Detection Dashboard

### Statistiques par type de fraude

 High Value	 High Frequency	 Geo Switch	 Currency Switch	 Carousel Fraud
136	211	127	78	30

 À quoi correspond le flag ?

Voir le nombre total de transactions traitées

### Dernières alertes de fraude

Fichiers Parquet trouvés : 411

timestamp	user_id	transaction_id	Flag
2025-06-01 05:34:16	u9979	t-0020056	 HIGH_VALUE
2025-06-01 05:33:40	u8302	t-0020020	 HIGH_VALUE
2025-06-01 05:30:50	u3776	t-0019850	 HIGH_VALUE
2025-06-01 05:30:43	u5085	t-0019843	 HIGH_VALUE
2025-06-01 05:30:37	u4508	t-0019837	 HIGH_VALUE

## Annexes :

```
producer-1 | Sent: t-0008691 - 14769.69 CNY
producer-1 | Sent: t-0008692 - 8199.37 DH
producer-1 | Sent: t-0008693 - 22461.69 USD
producer-1 | Sent: t-0008694 - 29564.1 DH
producer-1 | Sent: t-0008695 - 32360.62 GBP
producer-1 | Sent: t-0008696 - 3071.5 GBP
producer-1 | Sent: t-0008697 - 6114.17 CNY
producer-1 | Sent: t-0008698 - 30498.9 DH
producer-1 | Sent: t-0008699 - 41602.77 USD
producer-1 | Sent: t-0008700 - 15641.49 GBP
producer-1 | Sent: t-0008701 - 33724.27 EUR
producer-1 | Sent: t-0008702 - 48992.68 USD
producer-1 | Sent: t-0008703 - 25621.43 DH
producer-1 | Sent: t-0008704 - 28595.79 DH
producer-1 | Sent: t-0008705 - 1129.78 GBP
producer-1 | Sent: t-0008706 - 18124.01 GBP
producer-1 | Sent: t-0008707 - 8084.98 CNY
producer-1 | Sent: t-0008708 - 21531.56 CNY
producer-1 | Sent: t-0008709 - 8262.49 GBP
producer-1 | Sent: t-0008710 - 12011.55 GBP
producer-1 | Sent: t-0008711 - 24752.59 CNY
producer-1 | Sent: t-0008712 - 32196.98 USD
producer-1 | Sent: t-0008713 - 20276.35 DH
producer-1 | Sent: t-0008714 - 23232.92 EUR
producer-1 | Sent: t-0008715 - 19877.63 EUR
producer-1 | Sent: t-0008716 - 43732.47 DH
producer-1 | Sent: t-0008717 - 32611.99 EUR
producer-1 | Sent: t-0008718 - 5104.03 EUR
```

**Containers / spark**

**spark**  
ff38e43ee1ab bitnami/spark:3.5

**STATUS**  
Running (25 seconds ago)

**Logs** Inspect Bind mounts Exec Files Stats

Batch: 2

user_id	transaction_id	timestamp	fraud_type
u3218	t-0002852	2025-06-01 00:47:32	HIGH_VALUE
u7570	t-0002864	2025-06-01 00:47:44	HIGH_VALUE
u7752	t-0002869	2025-06-01 00:47:49	HIGH_VALUE
u5583	t-0002875	2025-06-01 00:47:55	HIGH_VALUE
u3103	t-0002881	2025-06-01 00:48:01	HIGH_VALUE
u9108	t-0002929	2025-06-01 00:48:49	HIGH_VALUE
u2644	t-0002936	2025-06-01 00:48:56	HIGH_VALUE
u2522	t-0002960	2025-06-01 00:49:20	HIGH_VALUE
u2526	t-0002969	2025-06-01 00:49:29	HIGH_VALUE
u3953	t-0002973	2025-06-01 00:49:33	HIGH_VALUE
u9018	t-0002984	2025-06-01 00:49:44	HIGH_VALUE
u5765	t-0003012	2025-06-01 00:50:12	HIGH_VALUE
u1487	t-0003022	2025-06-01 00:50:22	HIGH_VALUE
u5359	t-0003028	2025-06-01 00:50:28	HIGH_VALUE
u1068	t-0003029	2025-06-01 00:50:29	HIGH_VALUE
u6000	t-0003055	2025-06-01 00:50:55	HIGH_VALUE
u1907	t-0003067	2025-06-01 00:51:07	HIGH_VALUE

Engine running RAM 3.85 GB CPU 34.15% Disk: 8.24 GB used (limit 1006.85 GB)

Terminal New version available