Ecole Normale Supérieure de l'Enseignement
Technique Mohammedia
Université Hassan II de Casablanca

ENSET

المدرسة العليا لأساتذة التعليم التقني
المحمدية
جامعة الحسن الثاني بالدار البيضاء

Département Mathématiques et Informatiques

Compte Rendu de Devoir 1
Big Data Processing

Filière d'ingénieur :

Ingénierie Informatique, Big Data et Cloud Computing

Détection de Fraudes en Temps Réel avec Kafka Streams et Tableau de Bord Grafana

Réalisé par : Abdellatif HASSANI

Professeur: Mr. Abdelmajid BOUSSELHAM

Année Universitaire : 2024-2025

Sommaire

1) Création des conteneurs docker pour : Kafka, Influxdb et Grafana:

Le fichier **docker-compose** définit les services requis pour déployer une architecture complète incluant :

- **Kafka** avec **Zookeeper** pour la coordination du cluster,
- **InfluxDB** pour le stockage des transactions,
- **Grafana** pour visualiser des tableaux de bord en temps réel.

Cette configuration facilite la mise en place et la gestion de l'infrastructure nécessaire à l'analyse et au suivi des données.

```
version: '3'

services:

 zookeeper:

   image: confluentinc/cp-zookeeper:latest

   environment:

     ZOOKEEPER_CLIENT_PORT: 2181

   ports:

     - "2181:2181"


 kafka:

   image: confluentinc/cp-kafka:latest

   depends_on:

     - zookeeper

   ports:

     - "9092:9092"

   environment:

     KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181

     KAFKA_ADVERTISED_LISTENERS: PLAINTEXT://localhost:9092
```

```
    KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR: 1


influxdb:
  image: influxdb:latest
  ports:
    - "8086:8086"
  environment:
    - DOCKER_INFLUXDB_INIT_MODE=setup
    - DOCKER_INFLUXDB_INIT_USERNAME=admin
    - DOCKER_INFLUXDB_INIT_PASSWORD=strongpassword123
    - DOCKER_INFLUXDB_INIT_ORG=myorg
    - DOCKER_INFLUXDB_INIT_BUCKET=frauddetection


grafana:
  image: grafana/grafana:latest
  ports:
    - "3000:3000"
  depends_on:
    - influxdb
```

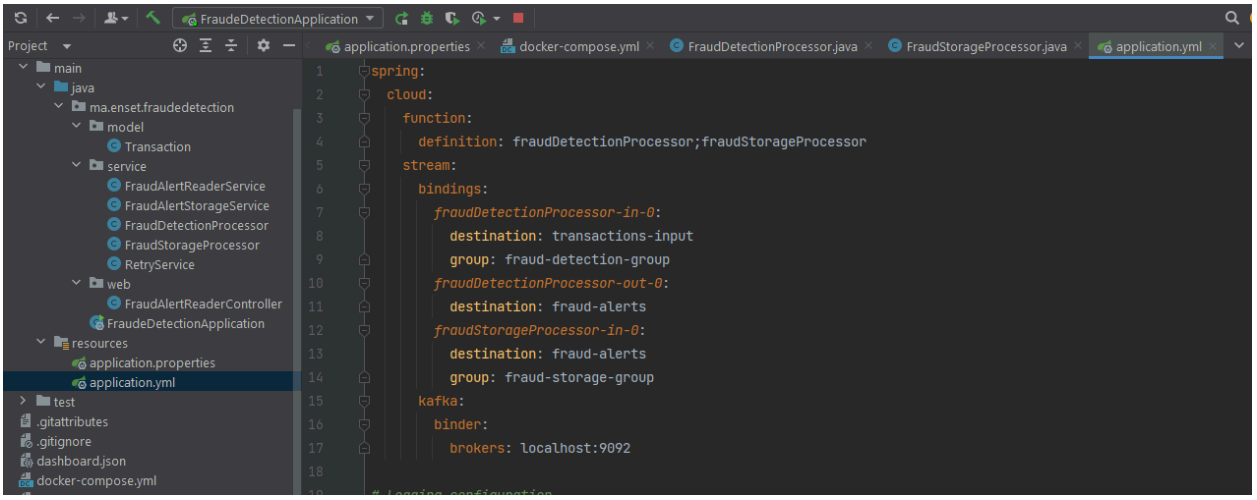
2) Configuration des topics kafka

```
C:\Users\Abdellatif>docker ps
CONTAINER ID   IMAGE                            COMMAND                  CREATED        STATUS        PORTS
             NAMES
2741b4be7c03   grafana/grafana:latest           "/run.sh"                2 days ago     Up 5 hours    0.0.0.0:3000->3000
/tcp         fraude-detection-grafana-1
ecf25117a534   influxdb:latest                  "/entrypoint.sh infl…"   2 days ago     Up 5 hours    0.0.0.0:8086->8086
/tcp         fraude-detection-influxdb-1
88e1e4866347   confluentinc/cp-kafka:7.3.0      "/etc/confluent/dock…"   2 months ago   Up 5 hours    0.0.0.0:9092->9092
/tcp         broker-enset
7be94b0d1d1a   confluentinc/cp-zookeeper:7.3.0  "/etc/confluent/dock…"   2 months ago   Up 5 hours    2181/tcp, 2888/tcp
, 3888/tcp   zookeeper-enset

C:\Users\Abdellatif>docker exec -ti broker-enset bash
[appuser@88e1e4866347 ~]$ kafka-topics --list --bootstrap-server localhost:29092
__consumer_offsets
fraud-alerts
fraudStorageProcessor-out-0
test-topic
test-topic1
test-topic2
test-topic3
transactions-input
[appuser@88e1e4866347 ~]$ |
```

```
@88e1e4866347:~   ×   Input Producer   ×   Input Consumer   ×   Output Consumer   ×   Command Promp

C:\Users\Abdellatif>docker exec -ti broker-enset bash
[appuser@88e1e4866347 ~]$ kafka-console-consumer --bootstrap-server localhost:29092 --topic
^CProcessed a total of 0 messages
[appuser@88e1e4866347 ~]$ kafka-console-consumer --bootstrap-server localhost:29092 --topic
ginning

serId": "12345", "amount": 5000, "timestamp": "2024-12-04T15:00:00Z"}
serId": "12345", "amount": 5000, "timestamp": "2024-12-04T15:00:00Z"}
serId": "12345", "amount": 5000, "timestamp": "2024-12-04T15:00:00Z"}
{"userId": "12345", "amount": 5000, "timestamp": "2024-12-04T15:00:00Z"}
{"userId": "67890", "amount": 15000, "timestamp": "2024-12-04T15:05:00Z"}
{"userId": "67890", "amount": 784115000, "timestamp": "2024-12-04T15:05:00Z"}
{"userId": "678890", "amount": 15000, "timestamp": "2024-12-04T15:05:00Z"}
{"userId": "678890", "amount": 15000, "timestamp": "2024-12-04T15:05:00Z"}
{"userId": "678890", "amount": 15000, "timestamp": "2024-12-04T15:05:00Z"}
{"userId": "678890", "amount": 15000, "timestamp": "2024-12-04T15:05:00Z"}
```

```
@88e1e4866347:~   ×   Input Producer   ×   Input Consumer   ×   Output Consume   ×
{"userId": "678890", "amount": 15000, "timestamp": "2024-12-04T15:05:00Z"}
{"userId": "678890", "amount": 15000, "timestamp": "2024-12-04T15:05:00Z"}
{"userId": "678890", "amount": 15000, "timestamp": "2024-12-04T15:05:00Z"}
{"userId": "678890", "amount": 15000, "timestamp": "2024-12-04T15:05:00Z"}
{"userId": "678890", "amount": 15000, "timestamp": "2024-12-04T15:05:00Z"}
{"userId": "678890", "amount": 15000, "timestamp": "2024-12-04T15:05:00Z"}
{"userId": "678890", "amount": 15000, "timestamp": "2024-12-04T15:05:00Z"}
{"userId": "678890", "amount": 15000, "timestamp": "2024-12-04T15:05:00Z"}
{"userId": "678890", "amount": 15000, "timestamp": "2024-12-04T15:05:00Z"}
{"userId": "678890", "amount": 15000, "timestamp": "2024-12-04T15:05:00Z"}
{"userId": "678890", "amount": 15000, "timestamp": "2024-12-04T15:05:00Z"}
{"userId": "678890", "amount": 15000, "timestamp": "2024-12-04T15:05:00Z"}
```

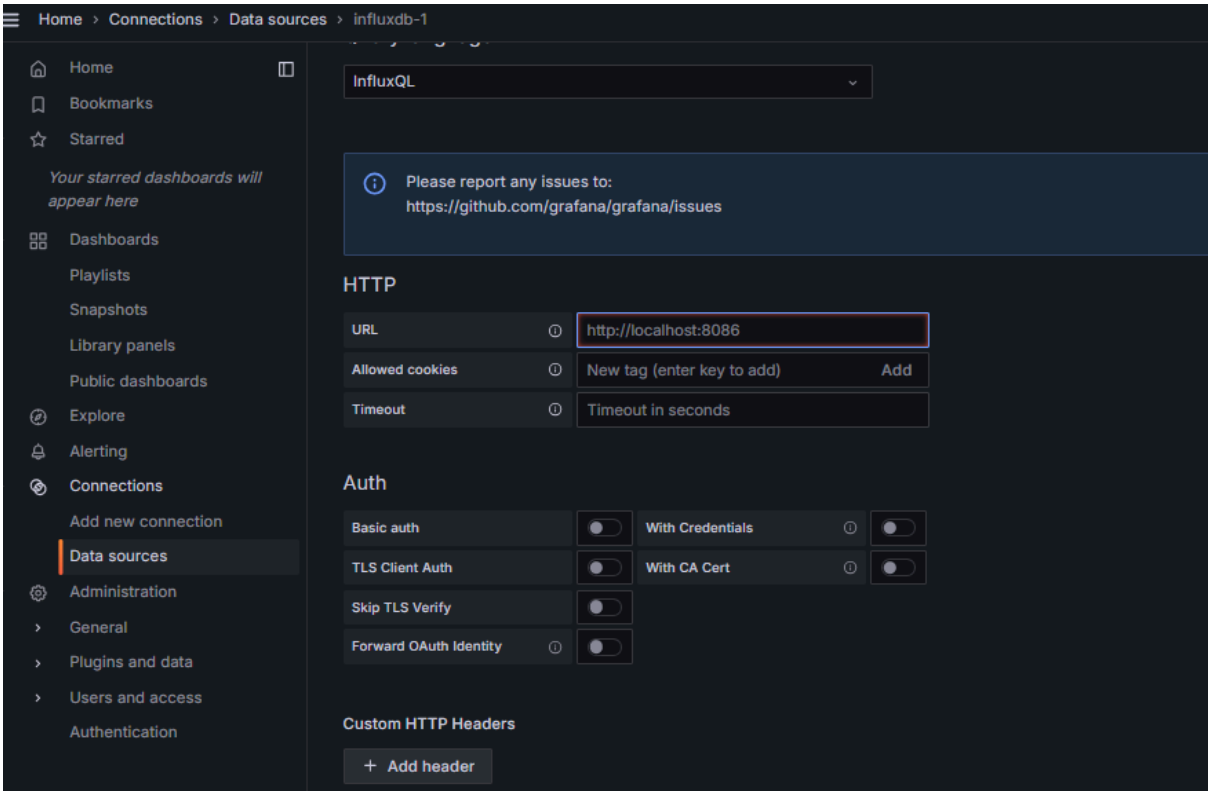
3) Développement d'une application Spring boot avec Spring Cloud Streams pour analyser les transactions en temps réel:



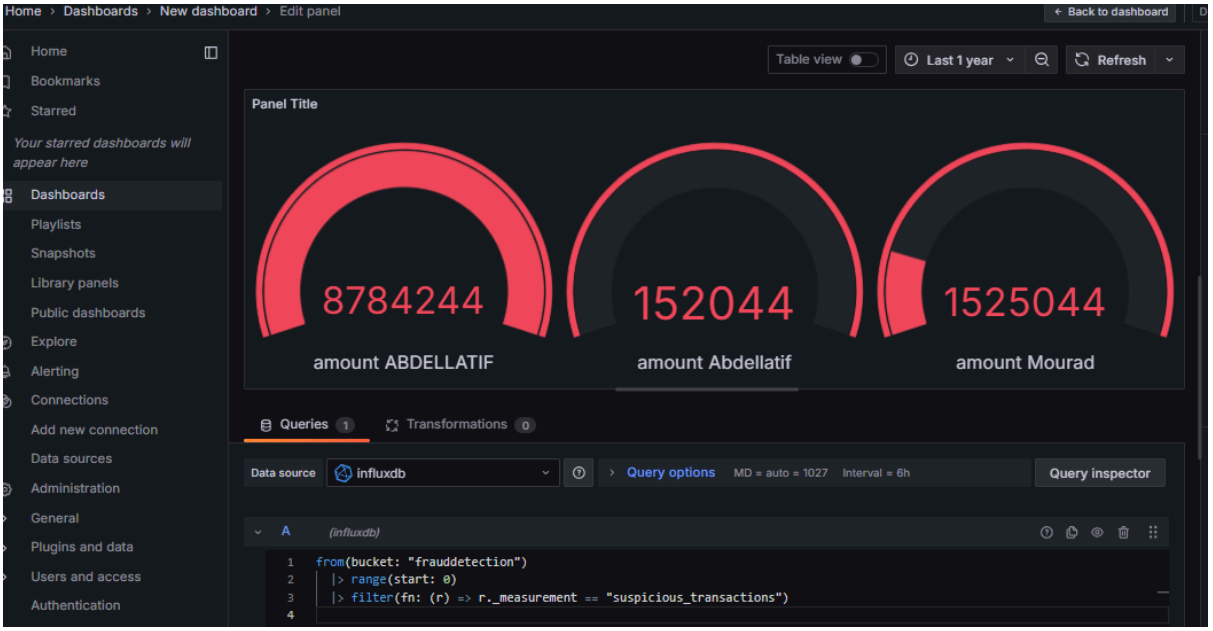

```java
@Service
public class FraudDetectionProcessor implements Function<String,
Transaction> {
   private static final Logger LOGGER =
Logger.getLogger(FraudDetectionProcessor.class.getName());
   private final ObjectMapper objectMapper = new ObjectMapper();
   @Override
   public Transaction apply(String message) {
       try {
           Transaction transaction = objectMapper.readValue(message,
Transaction.class);
           boolean isSuspicious = isFraudulent(transaction);
           if (isSuspicious) {
               LOGGER.info("Suspicious transaction detected: " +
message);
               return transaction;
           } else {
               LOGGER.info("Non-suspicious transaction detected: " +
message);
               return null;
           }
       } catch (Exception e) {
           LOGGER.severe("Error processing transaction: " +
e.getMessage());
           return null;
       }
   }
```

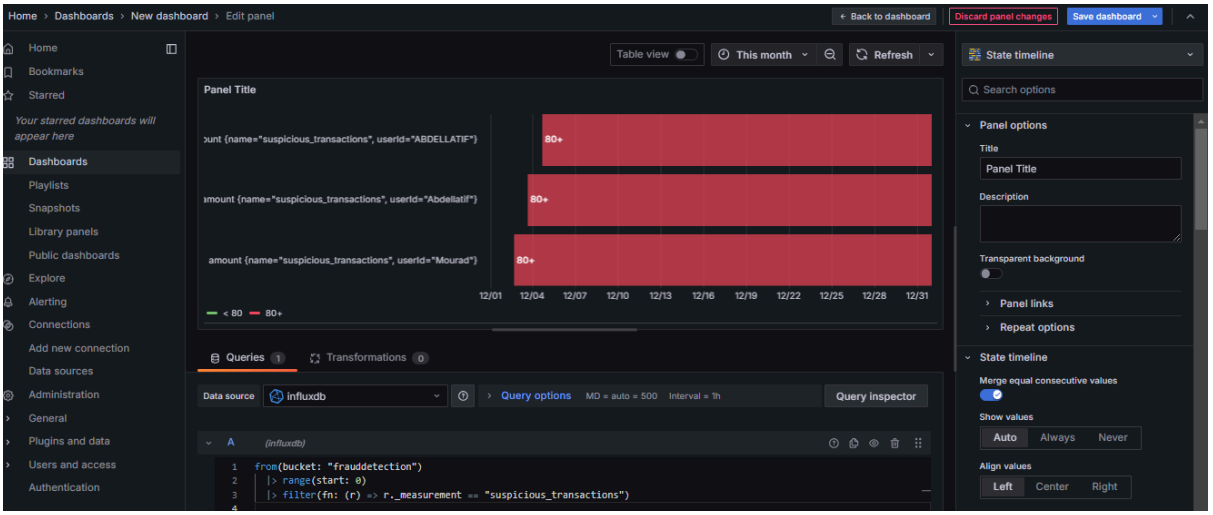
```java
@Service
public class FraudStorageProcessor implements Consumer<String> {
   private static final Logger LOGGER =
Logger.getLogger(FraudStorageProcessor.class.getName());
   private final ObjectMapper objectMapper = new ObjectMapper();

   @Autowired
   private FraudAlertStorageService storageService;

   @Override
   public void accept(String message) {
       try {
           Transaction transaction = objectMapper.readValue(message,
Transaction.class);
           storageService.storeFraudAlert(transaction);
           LOGGER.info("Transaction stored in InfluxDB for user: " +
transaction.getUserId());
       } catch (Exception e) {
           LOGGER.severe("Error storing transaction: " +
e.getMessage());
       }
   }
}
```

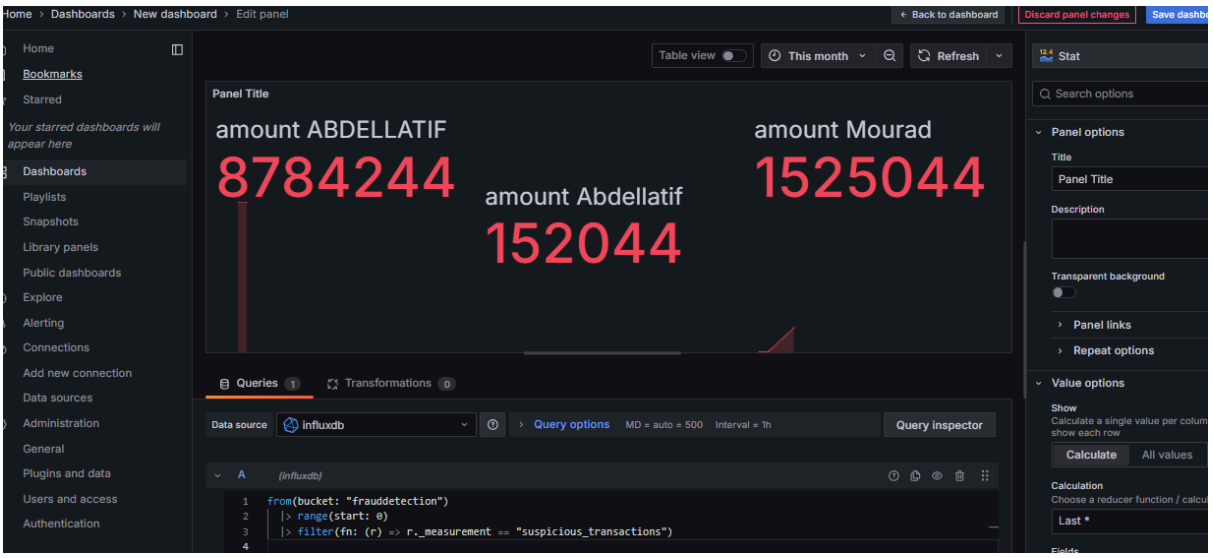
4) Configuration d'InfluxDB en tant que source de données pour Grafana



5) Création de Dashboard:




Home › Dashboards › New dashboard › Edit panel
Back to dashboard
Discard panel changes
Save dashboard
Table view
This month
Refresh
State timeline
Panel Title
amount {name="suspicious_transactions", userId="ABDELLATIF"}
amount {name="suspicious_transactions", userId="Abdellatif"}
amount {name="suspicious_transactions", userId="Mourad"}
80+
12/01 12/04 12/07 12/10 12/13 12/16 12/19 12/22 12/25 12/28 12/31
< 80 80+
Queries 1
Transformations 0
Data source
influxdb
Query options
MD = auto = 500
Interval = 1h
Query inspector
A
(influxdb)
from(bucket: "frauddetection")
|> range(start: 0)
|> filter(fn: (r) => r._measurement == "suspicious_transactions")
Search options
Panel options
Title
Panel Title
Description
Transparent background
Panel links
Repeat options
State timeline
Merge equal consecutive values
Show values
Auto
Always
Never
Align values
Left
Center
Right
Home
Bookmarks
Starred
Your starred dashboards will appear here
Dashboards
Playlists
Snapshots
Library panels
Public dashboards
Explore
Alerting
Connections
Add new connection
Data sources
Administration
General
Plugins and data
Users and access
Authentication


Home › Dashboards › New dashboard › Edit panel
Back to dashboard
Discard panel changes
Table view
This month
Refresh
Stat
Panel Title
amount ABDELLATIF
8784244
amount Abdellatif
152044
amount Mourad
1525044
Queries 1
Transformations 0
Data source
influxdb
Query options
MD = auto = 500
Interval = 1h
Query inspector
A
(influxdb)
from(bucket: "frauddetection")
|> range(start: 0)
|> filter(fn: (r) => r._measurement == "suspicious_transactions")
Search options
Panel options
Title
Panel Title
Description
Transparent background
Panel links
Repeat options
Value options
Show
Calculate
All values
Calculation
Last *
Home
Bookmarks
Starred
Your starred dashboards will appear here
Dashboards
Playlists
Snapshots
Library panels
Public dashboards
Explore
Alerting
Connections
Add new connection
Data sources
Administration
General
Plugins and data
Users and access
Authentication