

RAPPORT BIG DATA & DATA ENGINEERING

ANALYSE DE LOGS EN TEMPS RÉEL

Gestion des Événements

Réalisée par :

Sara El-Otmani

Encadrée par :

M. Hassan BADIR

1. Introduction

Dans le cadre du module Big Data / Data Engineering, nous avons développé un projet de traitement et analyse de logs système en temps réel.

Objectifs du projet :

1. Mettre en pratique les concepts Spark (RDD, DataFrame, Spark SQL, Spark Streaming).
2. Comprendre le fonctionnement d'un pipeline Big Data complet : ingestion → stockage → traitement → visualisation.
3. Utiliser des technologies distribuées : Kafka, HDFS et Spark.

Cas d'usage choisi :

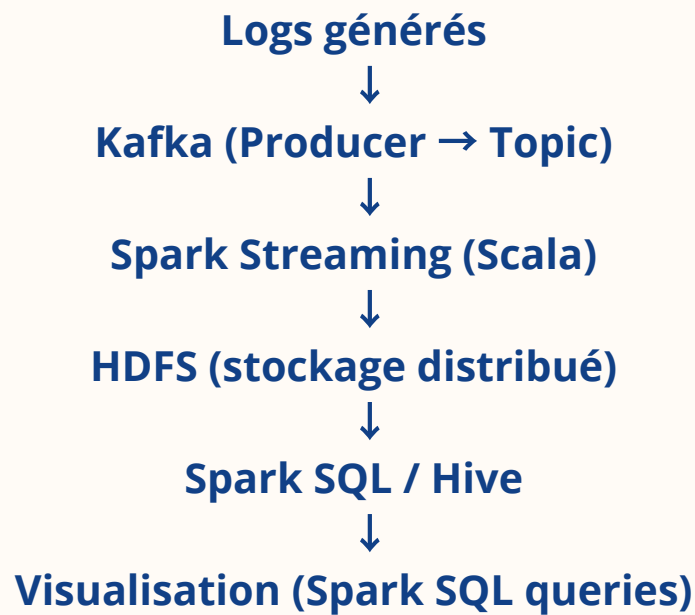
- Analyse des logs système pour détecter en temps réel les erreurs, warnings et événements critiques.

2. Architecture du pipeline

Le pipeline se compose des étapes suivantes :

Étape	Technologie	Rôle
Génération de logs	Kafka Producer	Simuler l'arrivée continue de logs
Ingestion	Kafka Topic	Stockage temporaire des messages
Traitement en temps réel	Spark Streaming (Scala)	Transformation, filtrage, comptage
Stockage distribué	HDFS	Stockage des logs filtrés et résultats
Analyse / Visualisation	Spark SQL / DataFrame	Requêtes et agrégations

Diagramme du pipeline :



3. Technologies utilisées

3.1 Apache Kafka

- Rôle : Gestion de messages en temps réel (logs)
- Concepts : Producer, Consumer, Topic

3.2 Apache Spark

- Modules utilisés : RDD, DataFrame, Spark SQL, Spark Streaming
- Rôle : Lecture, transformation et traitement des flux de données

3.3 HDFS

- Rôle : Stockage distribué des logs
- Avantages : Scalabilité et tolérance aux pannes

3.4 Scala

- Rôle : Langage principal pour coder Spark
- Avantage : Intégration directe avec Spark

4. Implémentation

4.1 Kafka – Création du topic et Producer

```
C:\Windows\system32>cd C:\kafka  
  
C:\kafka>.\bin\windows\kafka-topics.bat --create --topic logs-topic --bootstrap-server localhost:9092 --partitions 1 --replication-factor 1  
Created topic logs-topic.  
  
C:\kafka>
```

```
C:\kafka>.\bin\windows\kafka-console-producer.bat --topic logs-topic --bootstrap-server localhost:9092  
>ERROR Server down  
>INFO User login  
>WARN Disk almost full  
>Terminer le programme de commandes (O/N) ? o
```

Console Kafka Producer avec logs envoyés

```
C:\kafka>.\bin\windows\kafka-console-consumer.bat --topic logs-topic --bootstrap-server localhost:9092 --from-beginning  
ERROR Server down  
INFO User login  
WARN Disk almost full
```

4.2 Spark Streaming (Scala) – Lecture et traitement des logs

5. Architecture Technique

5.1 Langage et Outils

- Langage : Java
- IDE : Android Studio
- SDK minimum : Android 5.0 (API 21)

5.2 Organisation du Projet

Le projet est organisé en plusieurs packages afin d'assurer une bonne lisibilité du code :

- activities : contient les différentes activités
- models : contient la classe Note
- adapters : contient l'adapter personnalisé

6. Adapter Personnalisé

Un **Adapter personnalisé** (NoteAdapter) a été développé en héritant de la classe BaseAdapter.

Rôle de l'Adapter :

- Faire le lien entre la liste de notes (ArrayList<Note>) et la ListView
- Inflater un layout personnalisé (item_note.xml) pour chaque ligne
- Afficher dynamiquement les données de chaque note

Gestion des couleurs :

- Priorité **Haute** → Rouge
- Priorité **Moyenne** → Orange
- Priorité **Basse** → Vert

Cela permet à l'utilisateur d'identifier rapidement l'importance de chaque note.

7. Navigation entre les Activités

La navigation entre les écrans est réalisée à l'aide **d'Intents explicites** :

- Passage de l'écran liste vers l'écran ajout
- Retour à la liste après l'enregistrement grâce à finish()
- Passage des données d'une note vers l'écran de détails via putExtra()

Cette approche respecte le cycle de vie des activités Android.

8. Résultats Obtenus

L'application fonctionne correctement et permet :

- L'ajout de nouvelles notes
- L'affichage dynamique de la liste
- La consultation des détails d'une note
- Une interface simple et intuitive
- Une navigation fluide entre les écrans

9. Conclusion

Ce projet nous a permis de mettre en pratique les concepts fondamentaux du développement Android, notamment :

- La création d'activités
- La gestion des Intents
- L'utilisation des ListView
- Le développement d'un Adapter personnalisé
- L'organisation d'un projet Android professionnel

L'application **MyNotes** répond entièrement aux exigences du cahier des charges et constitue une base solide pour des améliorations futures comme la persistance des données ou l'ajout de fonctionnalités multimédia.