**Rapport Technique : Pipeline ETL et Streaming avec PySpark & Kafka**

**Table des Matières**

1. Introduction
2. Architecture technique
3. Installation et configuration
4. Pipeline ETL
5. Streaming de données
6. Optimisations
7. Documentation technique

**1. Introduction**

**Objectif du Projet**

Ce projet implémente un pipeline ETL (Extract, Transform, Load) utilisant PySpark et intègre un système de streaming de données avec simulation Kafka. Il permet de :

* Traiter des données étudiantes
* Transformer et analyser en temps réel
* Générer des statistiques automatisées

**Technologies Utilisées**

* Apache Spark 3.4.1
* PySpark
* Kafka (simulé)
* Python 3.7+
* Java 8

**2. Architecture Technique**

**Composants Principaux**

1.  graph TD

A[Source de Données] --> B[PySpark ETL]

B --> C[Transformations]

C --> D[Export CSV]

E[Kafka Producer] --> F[Spark Streaming]

F --> G[Analyses Temps Réel]

### Flux de Données

1. Ingestion données brutes (CSV)
2. Transformations PySpark
3. Export données traitées
4. Streaming temps réel

## 3. Installation et Configuration

### Prérequis Système

1.  # Java 8+

sudo apt-get install openjdk-8-jdk

# Python 3.7+

python3 -m pip install --upgrade pip

Installation Spark

1. wget https://dlcdn.apache.org/spark/spark-3.5.3/spark-3.5.3-bin-hadoop3.tgz

tar xf spark-3.4.1-bin-hadoop3.tgz

pip install findspark

Configuration Environnement

1.  import os

os.environ["JAVA\_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"

os.environ["SPARK\_HOME"] = "/content/spark-3.4.1-bin-hadoop3"

import findspark

findspark.init()

## 4. Pipeline ETL

### Phase d'Extraction

1.  from pyspark.sql import SparkSession

spark = SparkSession.builder \

.master("local[\*]") \

.appName("ETL\_Pipeline") \

.getOrCreate()

df = spark.read.csv(file\_path,

header=True,

inferSchema=True)

Phase de Transformation

1. from pyspark.sql.functions import col, upper, avg

# Transformation des noms

df\_transformed = df.withColumn("Nom", upper(col("Nom"))) \

.withColumn("Prénom", upper(col("Prénom")))

# Filtrage par âge

df\_filtered = df\_transformed.filter(col("Age") > 20)

# Statistiques par filière

df\_statistics = df\_filtered.groupBy("Filière") \

.agg(avg("Age").alias("Moyenne\_Age"))

Phase de Chargement

1.  # Export données transformées

df\_filtered.write.csv(

"/content/etudiants\_transformes.csv",

header=True,

mode="overwrite"

)

# Export statistiques

df\_statistics.write.csv(

"/content/statistiques\_filiere.csv",

header=True,

mode="overwrite"

)

## 5. Streaming de Données

### Simulation Kafka Producer

1.  import json

import time

def simulate\_kafka\_producer(df\_csv, output\_path):

with open(output\_path, "w") as f:

for row in df\_csv.collect():

record = {

"Nom": row["Nom"],

"Prénom": row["Prénom"],

"Age": row["Age"],

"Filière": row["Filière"]

}

json.dump(record, f)

f.write("\n")

time.sleep(2)

Configuration Streaming

1.  # Lecture stream

df = spark.readStream.text(output\_path)

# Transformations streaming

df\_transformed = df.withColumn("Nom", upper(col("value")))

# Configuration output

query = df\_transformed.writeStream \

.outputMode("append") \

.format("console") \

.start()

## 6. Optimisations

### Performance

1. Configuration Spark
   * Mémoire executor
   * Nombre de partitions
   * Cache données fréquentes
2. Gestion Ressources

1.  spark.conf.set("spark.executor.memory", "4g")

spark.conf.set("spark.driver.memory", "4g")

### Monitoring

1. Métriques Clés
   * Temps traitement
   * Utilisation mémoire
   * Latence streaming
2. Logging

1.  from pyspark.sql import SparkSession

spark = SparkSession.builder \

.config("spark.eventLog.enabled", "true") \

.config("spark.eventLog.dir", "./spark-logs") \

.getOrCreate()

## 7. Documentation Technique

### Structure du Code

1.  project/

├── src/

│ ├── etl/

│ │ ├── extract.py

│ │ ├── transform.py

│ │ └── load.py

│ └── streaming/

│ ├── producer.py

│ └── consumer.py

├── config/

│ └── spark-config.py

└── tests/

└── test\_etl.py

### Maintenance

1. Tests Unitaires
2. Gestion versions
3. Monitoring performance

### Sécurité

* Validation données
* Gestion erreurs
* Logs sécurité

## Contact et Support

Pour toute question ou support technique : J.baumbilia@suprh.com