**Synthèse Reporting et Analyse**

Cette synthèse présente une vue structurée et détaillée des différentes étapes réalisées dans le cadre du pipeline ETL et du streaming pour répondre aux exigences du projet.

**1. Configuration de l’Environnement**

**1.1. Plateformes et outils utilisés**

* **Kafka** : Utilisé pour collecter et diffuser des messages en temps réel depuis un topic nommé ibtissam.
* **Apache Spark** : Framework utilisé pour les traitements des données, en mode streaming (temps réel) et batch (traitement par lots).
* **Python** : Langage principal utilisé pour implémenter les différentes étapes, notamment l’interaction avec Kafka et Spark.

**1.2. Configuration technique**

* **Kafka Consumer** :
  + Protocoles sécurisés (SSL) configurés avec des fichiers de certificat (ca.pem, service.cert, et service.key).
  + Paramètres essentiels : bootstrap.servers, group.id, et auto.offset.reset.
* **Spark** :
  + Spark Streaming configuré pour consommer des messages en temps réel.
  + Utilisation de SparkSession pour le traitement par lots.

**2. Étapes du Pipeline ETL**

**2.1. Ingestion des données**

* **Source des données** :  
  Les messages Kafka contiennent des données structurées sous la forme :

"Prénom Nom, Âge, Filière"

Exemple :

"Ali Bennis, 21, Informatique"

* **Outil utilisé** : Spark Streaming pour ingérer les messages en continu.

**2.2. Transformation des données**

Les transformations suivantes ont été appliquées :

* **Nettoyage** :
  + Filtrage des messages mal formatés.
  + Validation des champs pour assurer leur cohérence (âge comme entier, séparation correcte des données).
* **Standardisation** :
  + Conversion des noms et prénoms en majuscules.  
    Exemple : *Ali Bennis → ALI BENNIS*
  + Transformation des âges en entier.
* **Filtrage** :
  + Sélection des étudiants ayant plus de 20 ans pour réduire le volume et se concentrer sur un segment spécifique.

**2.3. Stockage et export des données**

* Les données transformées ont été sauvegardées dans un fichier CSV :

"etudiants\_transformed.csv"

Ce fichier contient les colonnes suivantes : first\_name, last\_name, age, et field.

**3. Analyse et Statistiques Générées**

**3.1. Calcul de la moyenne d’âge par filière**

Les données ont été regroupées par filière pour calculer la moyenne d’âge des étudiants.  
Exemple de résultats :

* **Informatique** : Âge moyen de **22.5 ans**.
* **Gestion** : Âge moyen de **23.1 ans**.

**3.2. Analyse descriptive**

* Nombre total d’étudiants par filière.
* Distribution des âges par filière : minimum, maximum, et médiane.

**3.3. Visualisation des résultats (optionnelle)**

* Les résultats peuvent être visualisés à l’aide de graphiques (histogrammes ou diagrammes circulaires) pour une meilleure interprétation.

**4. Défis Rencontrés et Solutions Apportées**

| **Défi** | **Solution proposée** |
| --- | --- |
| **Messages mal formatés** : erreurs de validation des données reçues de Kafka. | Ajout d’un contrôle strict avec une gestion d’exception (try/except). |
| **Certificats SSL** : Problèmes liés à la connexion sécurisée avec Kafka. | Vérification des chemins des certificats et de leur validité. |

**5. Démonstration du Pipeline Complet**

**5.1. Étape 1 : Ingestion des données**

* Les messages sont consommés en temps réel depuis Kafka avec Spark Streaming.
* Les messages bruts sont affichés dans la console pour valider l’ingestion.

**5.2. Étape 2 : Transformation**

* Nettoyage, standardisation, et filtrage des données pour produire des résultats exploitables.
* Les données transformées sont affichées dans la console pour valider les étapes.

**5.3. Étape 3 : Analyse**

* Les statistiques générées (moyennes, regroupements par filière) sont affichées en temps réel.

**5.4. Étape 4 : Export des données**

* Les résultats transformés sont sauvegardés dans un fichier CSV.

**6. Reporting Final**

**Étapes récapitulatives**

1. **Configuration de Kafka et Spark** :  
   Mise en place d’une communication sécurisée pour collecter les messages en continu.
2. **Implémentation des transformations** :  
   Nettoyage, enrichissement, et analyses statistiques via Spark.
3. **Export et visualisation** :  
   Fichier CSV produit pour une exploitation ultérieure et présentation des résultats sous forme de tableau/graphique.

**Impact et utilisations futures**

Ce pipeline ETL et streaming est conçu pour être modulaire et extensible, ce qui permet d’intégrer des données supplémentaires ou d’ajouter de nouvelles analyses selon les besoins futurs.