Construction d'un Pipeline de Données avec Kafka, Logstash et ElasticStack, intégration avec Hadoop ou Spark (au choix)

Objectif Général

Créer un pipeline de données complet intégrant les étapes suivantes :

1. **Récupération des données via une API publique.**
2. **Transmission des données via Kafka.**
3. **Transformation des données avec Logstash et indexation dans Elasticsearch.**
4. **Analyse et visualisation des données avec Kibana.**
5. **Traitement des données avec au choix : Hadoop ou Spark.**

Projet à réaliser en **binôme** à rendre au plus tard **le 20 février**

Travail Demandé

Partie 1 : Collecte des Données via une API Publique

1. Choisir une API
2. Configurer une étape pour extraire les données régulièrement et les transmettre au pipeline Kafka.

Partie 2 : Transmission des Données avec Kafka

1. Créer un **topic Kafka** pour transporter les données.
2. Configurer un producteur Kafka pour envoyer les données collectées.
3. Configurer un consommateur Kafka pour transférer les données vers Logstash.

Partie 3 : Transformation et Indexation des Données

1. Utiliser **Logstash** pour :
    - Lire les données depuis Kafka.
    - Appliquer des transformations ou filtres si nécessaire.
    - Indexer les données dans Elasticsearch.
2. Construire un **mapping Elasticsearch** adapté :
    - Définir des analyzers et des filtres spécifiques.

Partie 4. Requêtes

1. Réaliser des requêtes dans Elasticsearch :
    - 1 requête textuelle
    - 1 requêtes comprenant une aggrégation
    - 1 requête N-gram.
    - 1 requêtes floues (fuzzy).
    - 1 série temporelle.

Partie 4 : Analyse et Visualisation avec Kibana

2. Créer des visualisations pertinentes (histogrammes, courbes de temps, etc.) à partir des requêtes.

Partie 5 : Traitement des Données avec Hadoop ou Spark (au choix)

- Choisir entre **Hadoop** ou **Spark** pour effectuer un traitement avancé :
    1. **Avec Hadoop :**
        - Stocker les données sur HDFS.
        - Réaliser un traitement MapReduce pour, par exemple, calculer des statistiques globales ou des agrégats.
    2. **Avec Spark :**
        - Charger les données depuis Elasticsearch ou HDFS.
        - Appliquer des transformations ou des calculs distribués (e.g., moyenne, somme, etc.).

Contraintes et Livrables

1. **Pipeline Fonctionnel :** démontrer le pipeline collecte, transmet, transforme, stocke et visualise les données correctement.
2. **Documentation Complète :**
    - Décrire chaque étape et justifier les choix techniques.
    - Comparer les performances et les capacités entre Hadoop et Spark (si possible).
3. **Résultats sur Kibana :**
    - Visualisations.
    - Requêtes Elasticsearch pertinentes.
4. **Code, Scripts et Configuration :**
    - Fournir tous les fichiers nécessaires (Kafka, Logstash, Elasticsearch, Hadoop/Spark).

Barème et Évaluation

Partie	Livrable attendu	Points
1. Collecte des Données	Script/configuration API et exemple de données extraites.	10
2. Kafka	Configuration des producteurs et consommateurs Kafka, captures d'écran des topics.	15
3. Logstash & Elasticsearch	Fichier de configuration Logstash, mapping Elasticsearch, captures d'écran des données indexées et les 5 requêtes	25
4. Kibana	Captures d'écran des visualisations et résultats des requêtes avancées Elasticsearch.	20

Partie	Livrable attendu	Points
5. Hadoop ou Spark (au choix)	Script/configuration de traitement, résultats du calcul en JSON ou CSV, et explication technique des choix réalisés.	20
Documentation & Organisation	Présentation claire du projet : structure des fichiers, commentaires dans le code, organisation du dépôt de code ou dossier.	10

Barème Total : 100 Points

Note Importante :

- Les étudiants doivent **livrer uniquement des fichiers de configuration, des scripts et des résultats (JSON/CSV)**.
- **Aucune démonstration en direct ne sera demandée.**