Proposition de projet — Cluster Hadoop (Docker) & Analyse des vidéos YouTube sur la guerre à Gaza

Contexte général

Ce projet comporte **deux parties complémentaires** :

1. Une **partie système** pour déployer et configurer un mini-cluster Hadoop (Docker) — objectif pédagogique : comprendre l'infrastructure Big Data distribuée.
2. Une **partie analytique** qui utilise ce cluster et des outils Big Data pour collecter, traiter et analyser les **vidéos et commentaires YouTube** liés à la guerre à Gaza afin d'étudier les tendances et opinions des internautes.

Partie 1 — Infrastructure : cluster Hadoop sous Docker

Objectif

Installer et documenter un cluster Hadoop simulé avec Docker comportant :

- 1 **NameNode** (maître),
- 1 **Secondary NameNode** (sauvegarde / checkpoint),
- 4 **DataNodes** (stockage distribué).

Livrables techniques

a. docker-compose.yml et/ou Dockerfiles pour chaque service (namenode, secondary-namenode, datanode×4).

b. Scripts d'initialisation et d'auto-configuration (création d'users HDFS, dossiers, permissions).

c. Manuel d'installation pas à pas (prérequis, commandes, vérifications via web UI, dépannage).

d. Tests fonctionnels : exécution d'un job simple (ex : WordCount) et vérification du stockage/réplication HDFS.

Exigences matérielles & recommandations

- Machine hôte : Docker Desktop ou Docker Engine (8–16 GB RAM recommandé pour le cluster local).
- Ports exposés pour interfaces Web (NameNode UI, ResourceManager UI).

Partie 2 — Analyse de données YouTube sur la guerre à Gaza

Objectif

Collecter et analyser les métadonnées, titres et **commentaires** de vidéos YouTube liées au conflit pour identifier :

- Tendances thématiques (mots-clés),
- Sentiments et émotions dominantes dans les commentaires,
- Évolutions temporelles et corrélations avec événements majeurs,
- Principales chaînes/vidéos générant engagement.

Étapes méthodologiques

1. **Définition de la collecte**
    - Mots-clés de recherche : ex. Gaza, genocide Gaza, Palestine, Israel Gaza, en plusieurs langues.
    - Périmètre temporel (ex. : 6 derniers mois / depuis date X).
    - Champs collectés : videoId, title, description, tags, publishedAt, channelTitle, viewCount, likeCount, commentCount, et tous les **commentaires**.
2. **Collecte**
    - Utiliser **YouTube Data API v3** (clé API) et pagination pour récupérer métadonnées et commentaires.
    - Si volume élevé, stocker d'abord en HDFS via ton cluster Docker (fichiers JSON/CSV) ou en stockage local puis importer dans HDFS.
3. **Prétraitement**
    - Nettoyage : suppression des URLs, mentions, balises HTML, emojis selon besoin.
    - Détection de langue et normalisation (tokenisation, lemmatisation).
    - Traduction optionnelle (vers une langue pivot) si tu veux analyses comparables multi-lingues.
4. **Analyses descriptives**
    - Comptage de mots-clés et tags (top K).
    - Fréquences par chaîne, par langue, par pays si disponible.
    - Top vidéos par engagement (views, likes, commentaires).

- Visualisations : barplots, séries temporelles, nuages de mots.

5. **Outils et technologies proposées**
   - ✓ **Collecte** : googleapiclient (YouTube Data API v3), scripts Python.
   - ✓ **Stockage & traitement** : HDFS sur cluster Hadoop (partie 1) + **PySpark** pour traitement distribué.
   - ✓ **Visualisation** : Matplotlib, Seaborn, Plotly (optionnel), notebooks Jupyter.

Livrables analytiques

- Script(s) de collecte et d'ingestion dans HDFS.
- Pipeline de traitement PySpark (prétraitement, extraction de features, calculs distribués).
- Dashboard de visualisation et rapport d'analyse (interprétation des résultats).