Projet : Construire un Pipeline de Données (Master 2)

Objectifs du projet :

- Comprendre les principes fondamentaux de la construction d'un pipeline de données.
- Maîtriser les outils et technologies nécessaires pour créer, déployer et gérer des pipelines de données.
- Appliquer des concepts avancés pour l'optimisation, la sécurisation et la surveillance des pipelines de données.
- Développer un pipeline de données complet, de l'ingestion à l'analyse et à la visualisation.

Étape 1 : Préparation de l'environnement

1. Choix de la plateforme Big Data

L'objectif ici est de sélectionner un environnement Big Data, comme **Hadoop** ou **Spark**. Si vous optez pour un environnement plus léger, pour des questions de coûts, vous n'êtes pas obligés de réaliser cette infrastructure sur Cloud. Je vous conseille une installation locale d'Apache Spark.

2. Choix de la plateforme BI ou Big Data

Sélectionnez un outil de BI pour la visualisation des données, comme **Power BI**, **Grafana** ou **DataDog**.

3. Installation des outils

Installez les outils nécessaires sur votre environnement de développement. Vous pouvez utiliser **Apache Spark en local** et **Power BI Desktop** *(gratuit)*, **Tableau Public** *(gratuit)* ou une **version locale de Grafana** par exemple.

4. Téléchargement des données

Sélectionnez et téléchargez un dataset public pour vos tests. Ils sont disponibles sur des plateformes comme **Kaggle**, ou des données ouvertes comme sur **data.gouv.fr** (*open data*).

Étape 2 : Extraction des données (Extract)

1. Chargement des données dans HDFS ou un Data Lake :

Si vous utilisez **Hadoop**, chargez les données dans **HDFS** (Hadoop Distributed File System). En cas d'utilisation d'un Data Lake, téléchargez les fichiers de données dans le répertoire correspondant.

2. Connexion à la source de données :

Connectez le dataset avec votre source de données via Hadoop ou Sparks par exemple (fichier CSV, JSON, base de données relationnelle).

- Exemple avec un fichier CSV :

```
python
from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("DataPipelineTP").getOrCreate()
df = spark.read.csv("path_to_your_file.csv", header=True, inferSchema=True)
```

Étape 3 : Transformation des données (Transform)

1. Nettoyage des données :

Appliquez des transformations simples comme le nettoyage des valeurs nulles, la conversion des types de données, et l'élimination des doublons.

- Exemple :

```
python
df_cleaned = df.dropna().dropDuplicates()
df_cleaned = df_cleaned.withColumn("new_column",df_cleaned["existing_column"].cast("Integer"))
```

2. Transformation des données :

Si des transformations sont nécessaires, vous devez les effectuer pendant la phase "Transform" comme la normalisation des données, la création de colonnes dérivées, ou l'agrégation.

3. Stockage des données transformées :

Enregistrez les données transformées dans un format adapté pour le BI, comme Parquet ou directement dans une base de données SQL.

Étape 4 : Chargement des données dans un outil BI (Load)

1. Connexion à la source de données depuis l'outil BI :

Connectez-vous à la source de données transformée. Si vous avez stocké les données dans un fichier Parquet, connectez-vous à ce fichier ; si c'est dans une base de données SQL, connectez-vous à cette base de données.

2. Création des tableaux de bord :

Utilisez l'outil BI pour créer des visualisations basées sur les données transformées. Par exemple, créez des graphiques montrant les totaux par catégorie, les tendances au fil du temps, ou des analyses comparatives.

3. Interaction avec les données :

Ajoutez des filtres, des sélecteurs de date et des indicateurs pour rendre le tableau de bord interactif et testez différentes visualisations pour explorer les insights cachés dans les données.

Étape 5 : Validation et optimisation

1. Validation des données :

Vérifiez que les données affichées dans le tableau de bord correspondent aux résultats attendus après les transformations, comparez les totaux et les moyennes avec ceux des données brutes pour vous assurer de la précision. Ces résultats seront attendus pour la présentation.

2. Optimisation :

Si nécessaire, revenez aux étapes précédentes pour optimiser les transformations, par exemple en améliorant les performances du traitement Spark ou en affinant les agrégations.

3. Documentation :

Documentez chaque étape du processus, expliquant les choix faits pour l'extraction, la transformation, et la visualisation des données.

Étape 6 : Présentation du projet

1. Préparation de la présentation :

Préparez une présentation montrant les étapes clés de votre pipeline de données, les défis rencontrés, et les insights obtenus à partir des visualisations.

2. Démonstration en direct :

Faites une démonstration en direct du pipeline, en montrant comment les données brutes sont transformées et visualisées dans l'outil BI.

3. Discussion des résultats :

Interprétez les résultats obtenus à partir des visualisations BI, en expliquant comment ces résultats peuvent être utilisés pour la prise de décision dans un contexte commercial.

Notation :

La notation s'effectuera selon cette grille :

Présentation : 10 points

- Présentation du groupe
- Présentation de l'objectif du projet
- Présentation de l'architecture utilisée
- Présentation des outils utilisés pour construire le(s) pipeline(s)
- Présentation du Workflow de la donnée (son parcours, son état)
- Tableau de répartition des tâches
- Exemple de code expliqué dans le pipeline (hors SQL)
- Résultats obtenus et discussion sur les résultats

Démonstration du Pipeline : 10 points

- Présentation de la base de données choisie (Open Data)
- Phase "Extract"
    - Démonstration de l'outil d'ingestion (batch ou stream)
    - Présentation de l'outil servant de Data Lake ou de Data Warehouse
- Phase "Transform"
    - Démonstration de l'automatisation de transformation de données (Normalisation, ajustement..)
- Phase "Load"
    - Démonstration du stockage des données transformées dans un Data Lake ou un Data Warehouse
- Phase "Data Visualisation"
    - Démonstration par un outil de BI ou de Big Data de la modélisation des données (Graphiques, Tableaux etc..)

Toute prise de risque concernant les outils ou la complexité du projet sera vue comme un potentiel bonus à la note.