Atelier Pipeline As A Code

Lab : Conception d'un pipeline de données modulaire avec traitement en temps réel et en batch

Objectif

Vous êtes consultant data pour une entreprise à New York. Votre équipe est chargée de construire un pipeline de données modulaire et évolutif capable d'ingérer, stocker, transformer et modéliser des données de trajets en taxi ainsi que des conditions météorologiques en temps réel à NYC. Le résultat final alimentera un tableau de bord pour mieux comprendre la mobilité et l'impact de la météo.

Vous devez construire un pipeline de bout en bout incluant :

- Le traitement batch des données historiques de trajets en taxi
- Le traitement en streaming (simulé) des données météorologiques
- Le stockage dans un data lake (MinIO) et un entrepôt DWH (PostgreSQL)
- L'orchestration de tout le flux avec Airflow
- La modélisation et la transformation des données avec dbt

Jeux de données

1. **Données Yellow Taxi NYC (Batch)**
    - Format : Parquet
    - Source : TLC Trip Record Data - TLC
2. **API OpenWeatherMap (Temps réel)**
    - Format : JSON
    - Source : Current weather and forecast - OpenWeatherMap

Stack tech :

- Python
- PySpark – Traitement batch
- PyFlink – Traitement en streaming

- MinIO – Data lake
- PostgreSQL – Entrepôt de données
- dbt – Modélisation des données
- Airflow – Orchestration
- (Optionnel) Pandas

Partie 1 – Ingestion des données

A. Données Yellow Taxi (Batch)

1. Un script Python qui :
    - Télécharge les données de janvier 2023 des trajets en taxi
    - Les stocke dans MinIO
2. Concevoir un DAG Airflow pour :
    - Appeler le script
    - Déclencher une transformation PySpark
    - Stocker les données transformées dans PostgreSQL

B. Données Météo (Streaming)

1. Un script Python qui :
    - Récupère les données météo pour NYC toutes les heures
    - Enregistre chaque réponse sous forme de fichier JSON horodaté dans MinIO
2. Concevoir un DAG Airflow pour :
    - Appeler le script
    - Lancer un job Flink qui traite les nouveaux fichiers JSON dans MinIO
    - Enregistrer les relevés météo dans PostgreSQL

Partie 2 – Transformation des données

A. PySpark : Transformation Taxi

1. Lire les données brutes depuis MinIO

2. Nettoyer et transformer pour extraire :
    - Durée du trajet (dropoff - pickup)
    - Tranches de distance : 0–2 km, 2–5 km, >5 km
    - Type de paiement (via table de correspondance)
    - Pourcentage de pourboire (tip_amount / fare_amount)
    - Heure de prise en charge, jour de la semaine
    - Informations de zone (via pickup_location_id et table des zones si dispo)
3. Sauvegarder le résultat dans PostgreSQL sous le nom fact_taxi_trips

B. PyFlink : Transformation météo

1. Surveiller le dossier météo de MinIO pour des nouveaux fichiers JSON (par heure)
2. Extraire les champs pertinents :
    - Température, humidité, vitesse du vent, condition météo
    - Horodatage
3. Ajouter des champs supplémentaires :
    - Catégorie météo (Clair, Pluvieux, Orageux)
    - Fonction temporelle (heure, jour de la semaine)
4. Sauvegarder dans PostgreSQL sous le nom dim_weather

Partie 3 – Modélisation des données avec dbt

Vos modèles dbt seront stockés dans le dossier dbt_project/.

Modèles sources :

- source_fact_taxi_trips (table PostgreSQL)
- source_dim_weather (table PostgreSQL)

Modèles intermédiaires et finaux :

1. **trip_enriched** (table de faits)
    - Jointure entre fact_taxi_trips et dim_weather selon l'heure de prise en charge
    - Inclure :

        - Détails du trajet
        - Catégorie météo
        - Pourcentage de pourboire
        - Type de paiement
2. **trip_summary_per_hour**
    - Agréger par heure et catégorie météo :
        - Nombre de trajets
        - Durée moyenne des trajets
        - Pourboire moyen
3. **high_value_customers** (table de dimensions)
    - Identifier les passagers (si le champ passenger_count est pertinent) qui :
        - Ont effectué plus de 10 trajets
        - Ont dépensé plus de 300 $ au total
        - Ont donné en moyenne plus de 15 % de pourboire

Questions

Spark

- Quelle est la distribution des durées de trajets ?
- Les longs trajets reçoivent-ils plus de pourboires ?
- Quelles sont les heures de prise en charge les plus chargées ?
- Existe-t-il une corrélation entre la distance du trajet et le pourcentage de pourboire ?

Flink

- Quelle est la température moyenne lors des pics de trajets ?
- Quel est l'impact du vent ou de la pluie sur le nombre de trajets ?

dbt / Analyse

- Quels comportements de trajets observe-t-on selon les types de météo ?
- À quelle heure observe-t-on le plus de clients à haute valeur ?

- La météo influence-t-elle le comportement en matière de pourboires ?

Livrables attendus

- Scripts Python modulaires (un par source ou processeur)
- DAGs Airflow
- Scripts de transformation Spark et Flink
- Schéma PostgreSQL (DDL si nécessaire)
- Dossier du projet dbt avec models/, sources.yml, dbt_project.yml
- Un court notebook ou fichier Markdown répondant aux questions analytiques principales