Master : Intelligence Artificielle Embarquée (MIAE)
Année : 2025-2026
Module : DataOps & MLOps
Prof. : Yasmine LAMARI - y.lamari@uiz.ac.ma

Projet Intégrateur Master DataOps/MLOps

Proposition de sujet : Plateforme de Détection d'Anomalies en Temps Réel pour la Qualité de l'Air
(`https://explore.openaq.org/locations/3847229`)

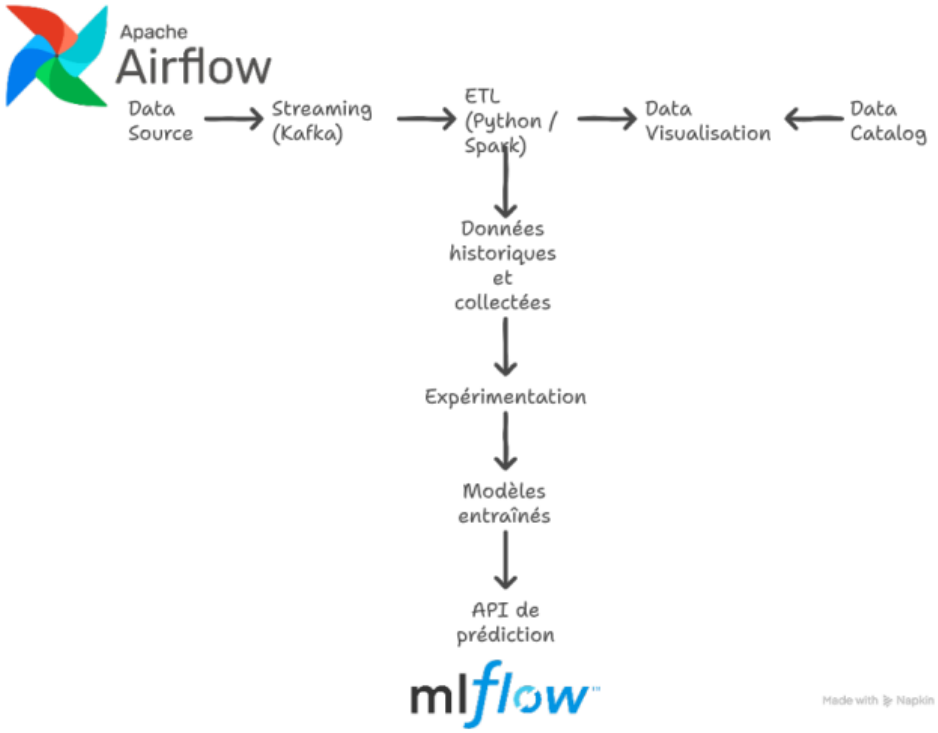
1. Contexte du Projet

Problématique : Développer une plateforme complète de surveillance de la qualité de l'air qui traite des données de capteurs IoT en streaming, détecte des anomalies, entraîne des modèles prédictifs, et rend les données accessibles via une interface de visualisation et un catalogue.

Objectifs :

- Intégrer DataOps et MLOps dans un scénario réel
- Maîtriser l'orchestration avec Apache Airflow
- Gérer le cycle de vie des modèles avec MLflow
- Traiter des données streaming
- Implémenter un data catalog (optionnel) et des visualisations (Dashboard)

2. Architecture Technique



3. Étapes du Projet

Phase 1 : Ingestion et ETL Streaming (2 semaines)

- **Outils :** Kafka, Spark Structured Streaming/Python, PostgreSQL, MongoDb, ...
- **Tâches :**
  1. Simuler des capteurs IoT (générateur de données Python, ou via API).
  2. Configurer un cluster Kafka local (3 topics : raw_data, processed_data, anomalies).
  3. Développer un consumer Spark/Python pour le nettoyage et calcul de métriques.
  4. Stocker dans PostgreSQL (temps réel + batch quotidien).

Phase 2 : Orchestration avec Airflow (2 semaines)

- **Outils :** Apache Airflow, Docker
- **Tâches :**
  1. Créer un DAG pour le pipeline quotidien (Extract → Validate → Train → Evaluate).
  2. Implémenter des capteurs (*sensors*) d'Airflow.
  3. Créer un DAG pour le *backfill* (réexécution historique).

Phase 3 : Gestion du Cycle de Vie ML avec MLflow (2 semaines)

- Outils : MLflow, Scikit-learn/XGBoost
- Tâches :

- Expérimenter avec différents modèles (régression, forêts aléatoires).
- Suivi des paramètres et métriques avec **MLflow Tracking**.
- Mise en place de **MLflow Registry** (Staging → Production).
- Créer une API de prédiction avec **FastAPI**.

4. Données et Modélisation

Exemple de Jeu de Données Simulé :

```
{
 "sensor_id": "CAP_001",
 "timestamp": "2024-01-15T14:30:00Z",
 "location": {"lat": 48.8566, "lon": 2.3522},
 "measurements": {
    "pm2_5": 12.5,
    "pm10": 25.3,
    "no2": 40.2
 }
}
```

5. Livrables Attendus

Livrable 1 : Code et Infrastructure (GitHub, Docker-compose, DAGs).
Livrable 2 : Rapport Technique (Architecture, choix justifiés, métriques).
Livrable 3 : Présentation (Soutenance de 20 minutes + Démo).

Bon courage !