Airline Project

Modèle de prédiction de retards de vols

Équipe de travail

Datascientest - Parcours Data engineer

Mehdi FEKIH

Sirine DHOUIB

Ayoub RABEH

Maxime ROUX

CONTENU

Introduction & contexte

Contexte:

Projet “fil rouge”: Prédiction du retard de vols

cursus Datascientest “Data Engineer”

Bootcamp 3 mois

Environnement et collaboration:

- Github - https://github.com/maxroux/mai24_bde_airlines/
- Points d’étapes hebdomadaires, mentor DS

Réalisations:

- Import des données
- Implémentation de modèles ML
- Déploiement et Monitoring

Récolte & Stockage des données

01 Choix des données: ***Lufthansa API***

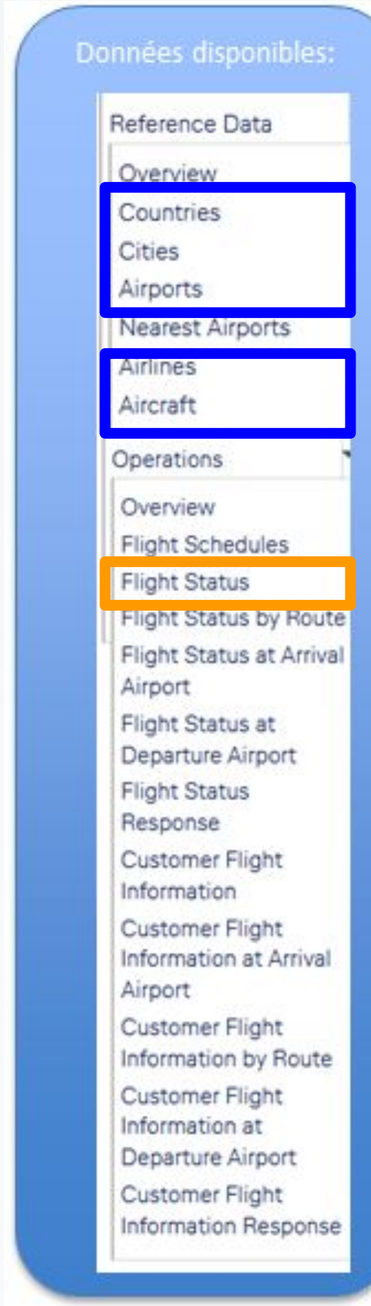
- Données de référence (statiques): Countries, Cities, Airports, Airlines, Aircraft
  ⇒ **Plusieurs sources “Reference data”**
- Données de vols (dynamiques): aéroport départ/arrivée, horaires, statut, etc.
  ⇒ **1 seule source “Flight Status”**

02 Récolte des données:

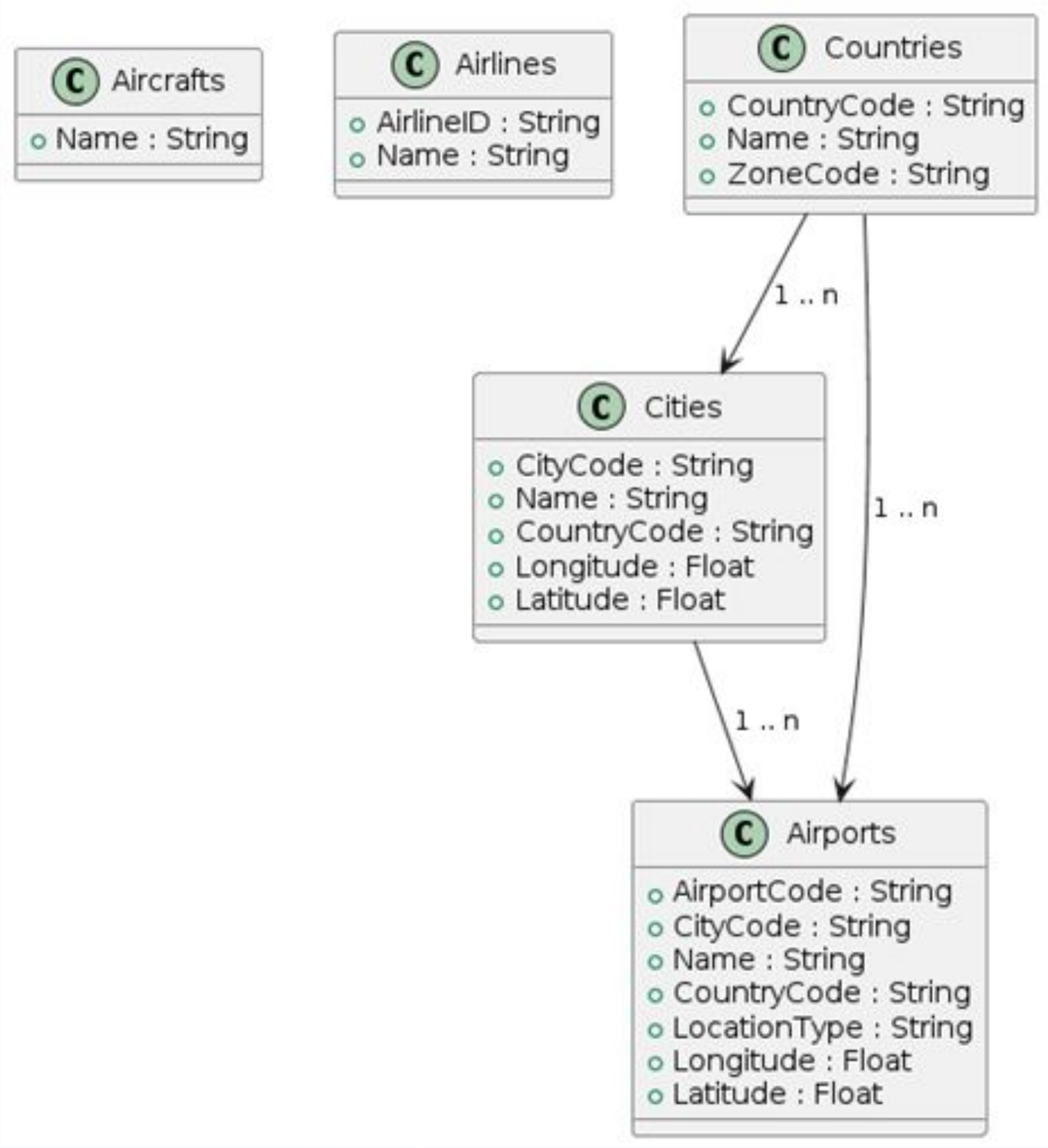
- Connexion à l’API
- Boucle avec limit / offset pour récolter toutes les données
- Planification journalière (manuelle, puis avec Airflow) pour les données de vol

03 Stockage des données:

- Données de référence → PostgreSQL (SGBD relationnelle)
- Données de vols → MongoDB (BDD NoSQL)



Données de référence - UML



Machine Learning

Chargement

Mongodb pour charger les données.

- Horaires des vols départ/arrivé.
- Aéroports.
- Compagnie Aérienne.
- Status du vol.

Prétraitement

- Sélection des variables
- Suppression des duplicates
- Traitement des valeurs manquantes
- Encodages des données :
  - One Hot encoding
  - Frequency encoding
  - Standard Scales

Entrainement

- Forêts aléatoires
- Régression linéaire
- Support Vector Machines
- LGBRM Regressor

Evaluation

- MAE (Erreur Absolue Moyenne).
- MSE (Erreur Quadratique Moyenne).
- RMSE (Racine carré de MSE).
- R^2

Déploiement : stack technique

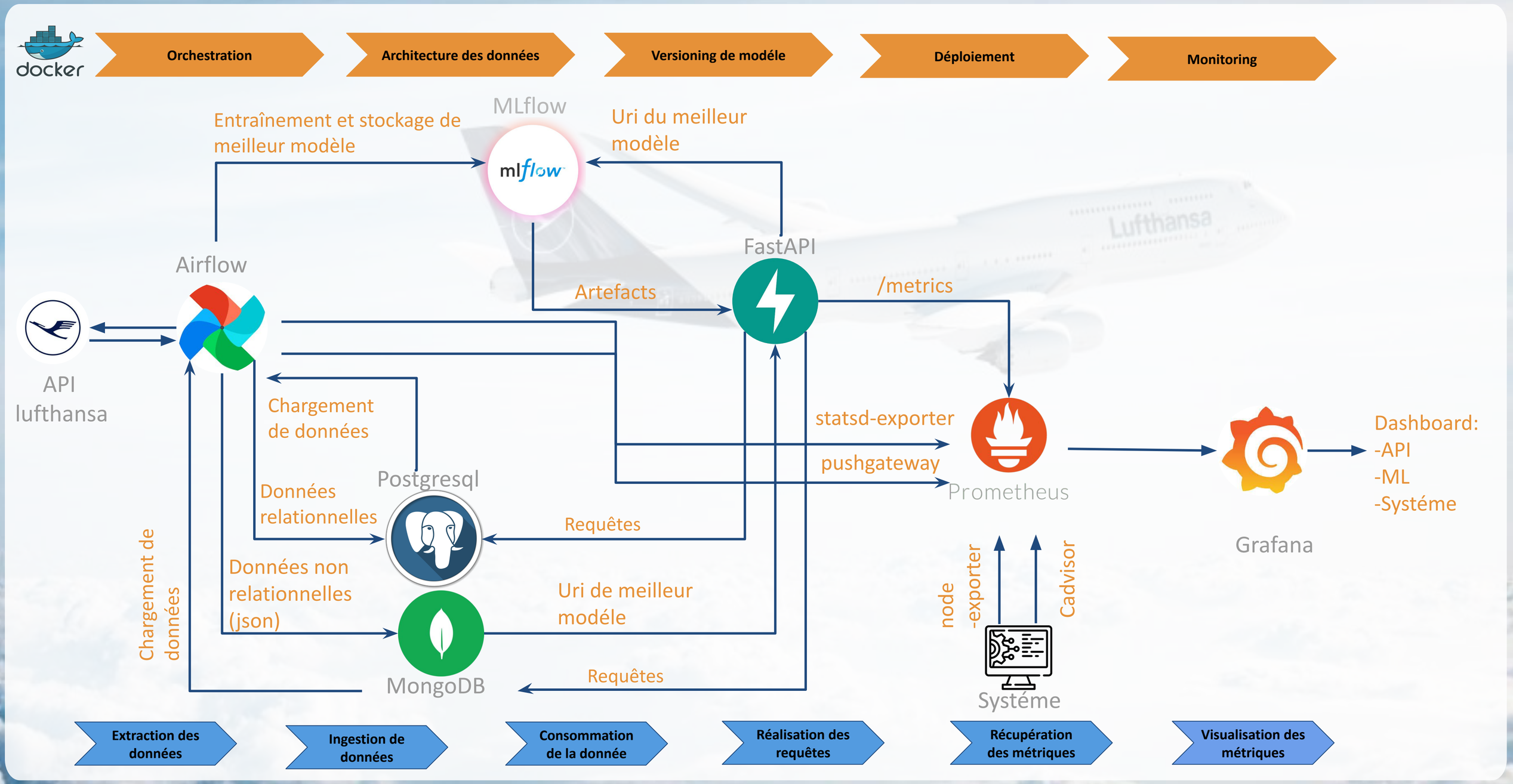
Base de données

Versioning ML

API

Orchestration

Monitoring


docker
Orchestration
Architecture des données
Versioning de modéle
Déploiement
Monitoring
MLflow
Entraînement et stockage de meilleur modèle
Uri du meilleur modèle
mlflow
Airflow
FastAPI
Artefacts
/metrics
API lufthansa
Chargement de données
statsd-exporter
pushgateway
Prometheus
Grafana
Dashboard:
-API
-ML
-Systéme
Postgresql
Données relationnelles
Requêtes
Chargement de données
Données non relationnelles (json)
Uri de meilleur modéle
MongoDB
Requêtes
node -exporter
Cadvisor
Systéme
Extraction des données
Ingestion de données
Consommation de la donnée
Réalisation des requêtes
Récupération des métriques
Visualisation des métriques

Démonstration du déploiement

AIRFLOW : http://airlineproject.duckdns.org:8085/

MLFLOW : http://airlineproject.duckdns.org:5001/

API : http://airlineproject.duckdns.org:8002/

PROMETHEUS : http://airlineproject.duckdns.org:9090/

GRAFANA : http://airlineproject.duckdns.org:3001/

DASHBOARD : http://airlineproject.duckdns.org:8050/

Conclusion

objectif 1

Récolte et consommation de données

objectif 2

Déploiement

Objectif 3

Automatisation et monitoring

Base de données riche

Un accès opérationnel

Une surveillance continue

Merci de votre attention

Améliorations possibles:

- Données en entrée: tests et ajout éventuel de paramètres additionnels (météo, saisonalité, etc.)
- Principal paramètre = heure de départ réel (non disponible par définition pour vols futurs)
  Tests et amélioration du modèle sans heure de départ réelle pour un modèle de prédiction à plus long terme.
- Interface/formulaire API: contrainte pour sélectionner des vols réels (passés = flight status, futurs = flight schedule), en remplacement du formulaire actuel (sélection libre des valeurs)