MLOps & Orchestration

Du notebook chaotique au pipeline industriel automatisé

KNOEPFFLER Leo-Paul · MBOUP Modou · ZEHRAOUI Mazilda

Contexte et Enjeux

Pourquoi passer du Notebook à l'Industrialisation ?

LE CHAOS DES PIPELINES

Le Problème

✗ Scripts manuels, zéro reproductibilité

✗ model_final_v2_VRAIMENT_FINAL.pkl

✗ Impossible de retrouver les paramètres d'une expérience passée

✗ Les modèles ne voient jamais la production

✗ Aucune alerte si le pipeline plante à 3h du matin

La Solution : MLOps

✓ Automatisation du tracking des expériences

✓ Orchestration des tâches avec dépendances

✓ Monitoring continu des performances

✓ Réessais automatiques en cas de panne

✓ Planification sans intervention humaine

✓ Versioning et registre des modèles

L'orchestrateur gère le COMMENT et MLflow gère le QUOI

MLFLOW : LE HUB ML

MLflow est la plateforme standard open-source pour gérer le cycle de vie du Machine Learning.

Tracking

Log automatique des paramètres, métriques et artefacts. Chaque run est traçable et comparable via l'interface web.

Models

Packaging standardisé, agnostique du framework. Même format pour sklearn, PyTorch, TensorFlow...

Registry

Gestion des versions et des stages : None → Staging → Production → Archived. Audit complet.

Serving

Déploiement en une commande : REST API locale, inférence batch, intégration cloud (SageMaker, Azure ML).

DÉVELOPPER → SUIVRE → ENREGISTRER → SERVIR → AUTOMATISER

POURQUOI MLFLOW ?

Agnostique

Fonctionne avec Scikit-learn, PyTorch, TensorFlow, Spark MLlib et bien d'autres. Un seul outil pour tous vos frameworks.

Reproductibilité

Retrouvez exactement le code, les données et l'environnement d'une expérience passée.

Déploiement simplifié

Passage du modèle vers une API REST ou le Cloud en une commande. Compatible SageMaker, Azure ML, Vertex AI.

Collaboration

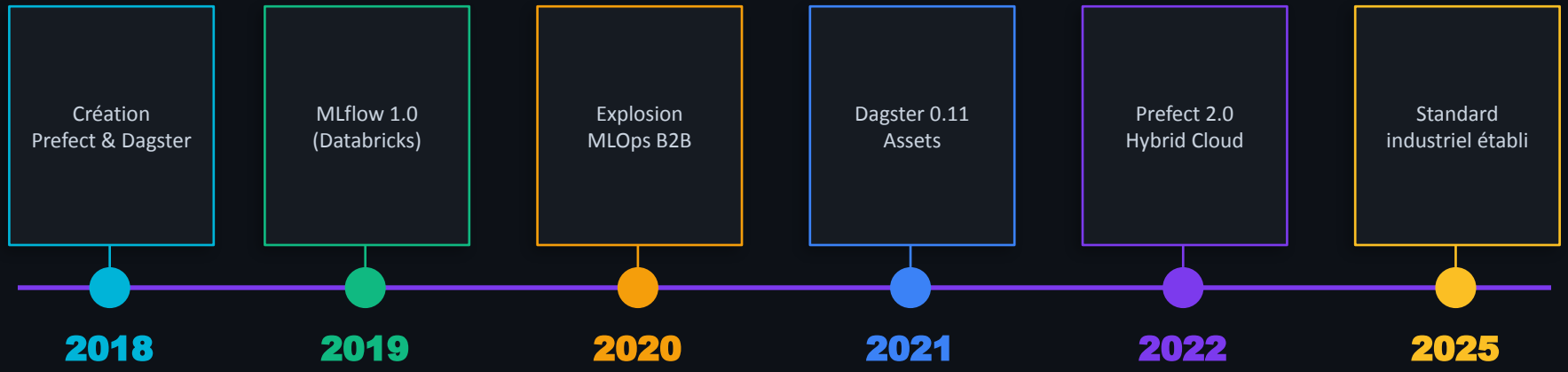
Partagez vos résultats avec toute l'équipe instantanément via l'interface web. Comparez les runs de différents membres.

Orchestration de Workflows

De Airflow à Prefect : choisir son chef d'orchestre

TENDANCES MLOPS (2018–2025)

L'évolution vers des architectures découplées et légères



Adoption exponentielle des architectures découplées Prefect/Dagster + MLflow

Les entreprises migrent d'Airflow monolithique vers des stacks légères centrées données

MLflow devient le standard universel de tracking, indépendamment de l'orchestrateur choisi

APACHE AIRFLOW

Le standard du marché — créé par Airbnb (2014)

Utilisé par Uber, Slack, Twitter, Adobe et des milliers d'entreprises

Scheduling puissant

Cron, intervalles, triggers événementiels, dépendances entre DAGs

Configuration as Code Python

Pipeline entier en Python pur — versionnable, testable, CI/CD ready

Écosystème immense

1000+ opérateurs : Spark, Kubernetes, BigQuery, S3, dbt...

Points de friction ML

- XCom limité à 48 KB — impossible de passer un DataFrame
- I/O disque obligatoire entre chaque tâche ML
- 30+ lignes de boilerplate avant la logique métier
- Dépendances déclarées séparément du code (task1 >> task2)
- Infrastructure lourde : webserver, scheduler, metadata DB

PREFECT : ORCHESTRATION 2.0

Pas de DAG rigide

Boucles, conditions dynamiques, sous-flows. Le pipeline s'adapte aux données à l'exécution.

Hybrid Cloud

Données en local, orchestration dans le cloud. Compatible Docker, Kubernetes, Prefect Cloud.

Résilience native

Retries avec backoff exponentiel, cache intelligent, parallélisme — zéro configuration.

Intégration MLflow

Chaque flow peut ouvrir un run MLflow. Paramètres et métriques loggés automatiquement.

Atelier Prefect

1. Tasks & Flows
2. Résilience & Retries
3. Cache & Parallélisme
4. Paramètres & Sous-flows
5. Pipeline ML + MLflow
6. Déploiement & Scheduling
7. Notifications Discord/Slack

DAGSTER : APPROCHE ASSET

On n’orchestre plus des tâches, on produit des données (Assets).

Data Assets

Chaque étape définit un asset nommé. Dagster infère automatiquement les dépendances — pas besoin de les déclarer.

Testabilité

Conçu pour être testé localement. Matérialisation partielle : re-calculez uniquement les assets obsolètes.

Observabilité

Lineage natif : comprenez d'où vient chaque ligne. Graphe d'assets interactif, historique, métadonnées.

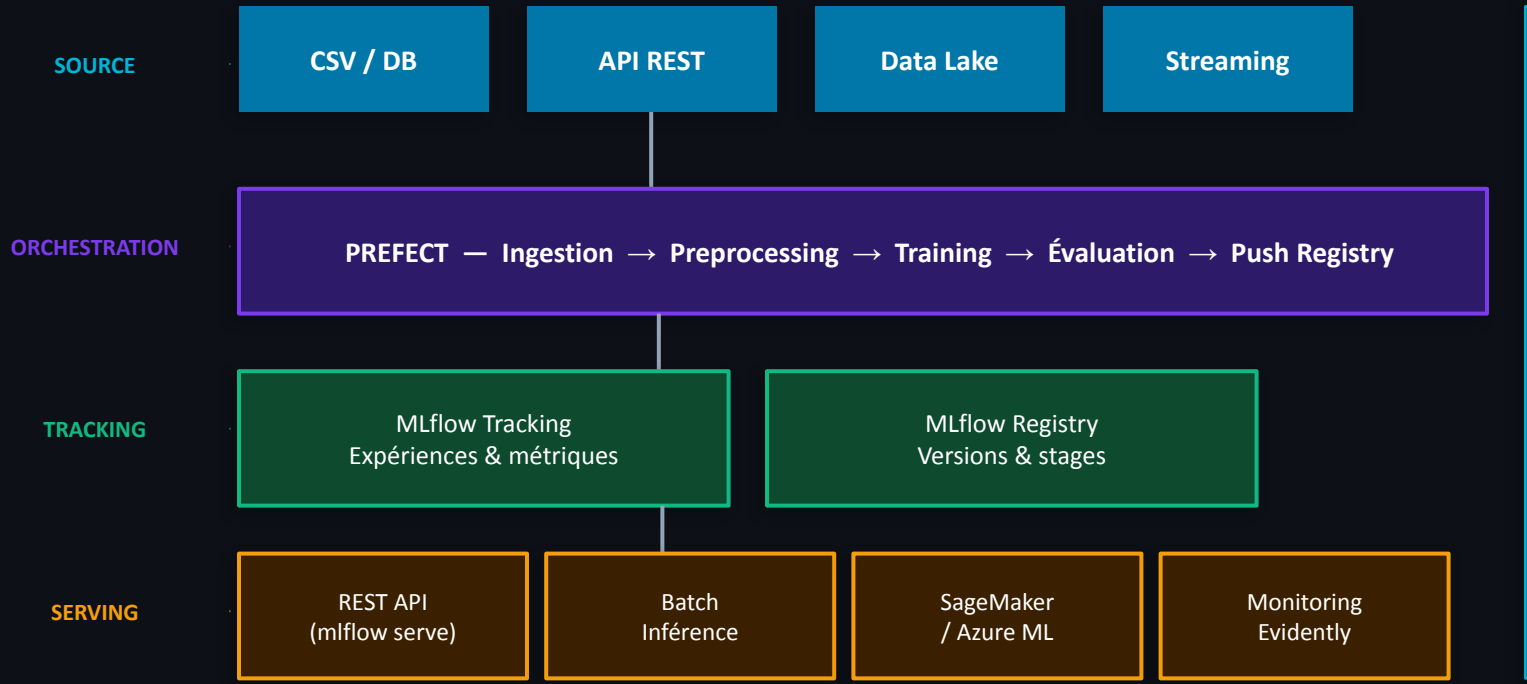
Pourquoi on choisit Prefect pour ce projet ?

Dagster excelle pour les data engineering complexes avec des centaines d'assets. Prefect est plus adapté à notre cas — pipeline ML relativement simple, prise en main rapide, intégration MLflow native, et démarrage Docker local.

Architecture & Mise en Oeuvre

ARCHITECTURE CIBLE COMPLÈTE

Le flux complet : de la donnée brute au modèle en production



PRINCIPES CLÉS

Découplage orchestration / tracking

Immutabilité des artefacts

Versioning à chaque étape

Observabilité end-to-end

PLAN DE MISE EN ŒUVRE

01

Audit

Analyse des besoins et de l'existant

→ Cartographier les workflows actuels

→ Identifier les points de friction

→ Évaluer la maturité de l'équipe

02

Pilot

Setup MLflow & Prefect

→ Déploiement Docker local

→ Premier pipeline instrumenté

→ Formation de l'équipe data

03

Scale

Migration des pipelines critiques

→ Migration progressive des DAGs

→ Mise en place du registre

→ Automatisation du réentraînement

04

Prod

Monitoring & Industrialisation

→ Alertes et SLAs définis

→ Dashboard de monitoring

→ Gouvernance et audit

"L'orchestration ne consiste pas seulement à faire tourner des scripts, mais à garantir la confiance dans la donnée."

📚 **Atelier complet sur GitHub**

`github.com/ajeansquid/Atelier-SISE-MlFlow_orchestrator`

CE QU'IL FAUT RETENIR

Découpler

Prefect = COMMENT · MLflow = QUOI. Deux outils, deux responsabilités distinctes.

Tracer

Chaque paramètre, métrique et artefact est loggé. Fini le 'sais plus comment j'ai eu ce résultat'.

Versionner

Le Registry donne une identité claire à chaque modèle : Staging → Production → Archived.

Automatiser

Scheduling Prefect + Registry MLflow = pipeline qui se réentraîne et se déploie seul.

Questions ?

Merci de votre attention !

SOURCES & RÉFÉRENCES

MLflow Official Docs	*https://mlflow.org/docs/latest/*
MLflow Tracking UI	*https://mlflow.org/docs/latest/assets/images/tracking-metrics-ui-temp.png*
Apache Airflow Official	*https://airflow.apache.org/docs/apache-airflow/stable/_images/dag_overview_runs1.png*
Prefect Documentation	*https://docs.prefect.io/*
Dagster Documentation	*https://docs.dagster.io/*
Prefect UI (Medium)	*https://miro.medium.com/v2/resize:fit:1400/1*d5c-Uu80xyAwv0nhKB9FgA.png*
Atelier SISE — GitHub	*https://github.com/ajeansquid/Atelier-SISE-MlFlow_orchestrator/tree/main*