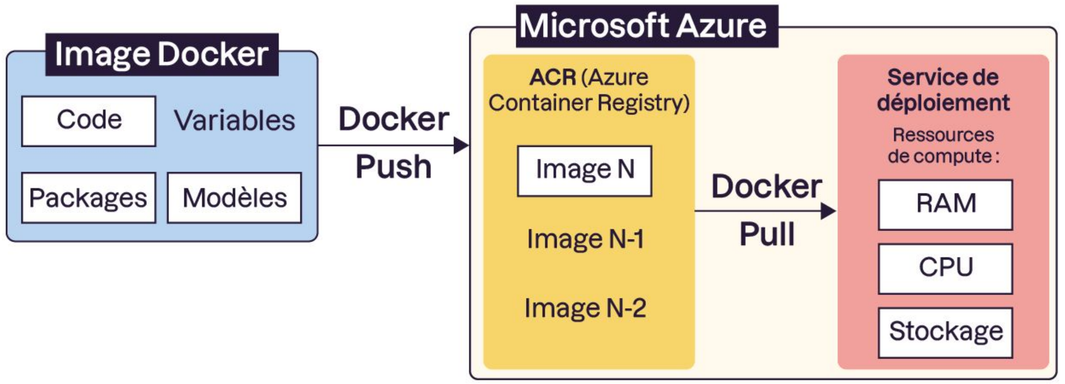
Fiche résumé : Initiez vous au ML Engineering

1. Comprendre les enjeux du ML Engineering dans les projets ML
2. Apprendre à exposer un modèle via une API avec FastAPI
3. Conteneuriser le projet avec Docker pour assurer sa portabilité
4. Automatiser l'intégration et le déploiement avec CI/CD
5. Gérer les modèles ML avec MLflow et un tracking collaboratif



Bonnes pratiques 👍

- ✅ Utiliser FastAPI pour exposer les modèles via une API claire.
- ✅ Conteneuriser le projet avec Docker pour garantir la portabilité.
- ✅ Créer une API simplifiée pour les utilisateurs non techniques.
- ✅ Séparer les inputs utilisateurs des features techniques.
- ✅ Implémenter des tests unitaires sur chaque fonction critique.
- ✅ Suivre les standards de CI/CD pour industrialiser les livraisons.
- ✅ Utiliser MLflow pour tracer et comparer les modèles ML.
- ✅ Stocker les artefacts sur un bucket dédié (ex : S3).

Erreurs classiques 👎

- ❌ Déployer un modèle sans tests unitaires ni validation.
- ❌ Forcer les utilisateurs à manipuler des features encodées.
- ❌ Ignorer la reproductibilité des environnements Python.
- ❌ Copier le dossiermlrunsdans Git au lieu de le tracker.
- ❌ Construire un Dockerfile sans optimiser le layering.
- ❌ Oublier les metrics de santé dans le suivi de l'infra.
- ❌ Lancer des tâches sans orchestrateur fiable comme Airflow.
- ❌ Remplacer un modèle sans évaluer ses impacts précis.

Définitions 🔎

ML Engineering
Discipline qui vise à industrialiser les modèles ML en production grâce à des outils d'ingénierie logicielle.

FastAPI
Framework Python léger pour créer rapidement des APIs web performantes, utilisé pour exposer les modèles ML.

Docker
Outil permettant d'empaqueter une application et ses dépendances dans un conteneur portable et reproductible.

MLflow
Plateforme de gestion du cycle de vie des modèles ML, avec tracking des expérimentations et registry collaboratif.