

Climate MLOps : Guide Complet

Système de Prédition Météo Autonome (Marrakech)

Architecture, Continuous Training & Déploiement Azure

Sirgianeouiçal

28 Décembre 2025

Contents

1	Introduction et Vision	3
2	Architecture Technique	3
2.1	Composants Principaux	3
2.2	Image Docker Unifiée	3
3	Le Cycle de Continuous Training	3
4	Guide de Validation et Déploiement	4
4.1	Phase 1 : Validation Locale	4
4.2	Phase 2 : Préparation Azure (ACR)	4
4.3	Phase 3 : Déploiement Container Apps	4
5	Maintenance et Nettoyage	5
6	Conclusion	5

1 Introduction et Vision

Le projet **Climate MLOps** n'est pas seulement un modèle de prédiction météo ; c'est un écosystème autonome capable de collecter des données, de s'auto-entraîner et de se déployer dans le Cloud. La vision centrale est le **Continuous Training** : le système s'adapte aux changements climatiques de Marrakech sans intervention humaine.

2 Architecture Technique

Le système repose sur une architecture multi-conteneurs (10 services) orchestrée par Docker.

2.1 Composants Principaux

- **Base de Données** : PostgreSQL pour le stockage des relevés historiques et des métadonnées Airflow.
- **Orchestration** : Apache Airflow gère le pipeline de données et les cycles d'entraînement.
- **ML Tracking** : MLflow enregistre chaque expérience, métrique et modèle produit.
- **Frontend & API** : FastAPI sert les prédictions via un Dashboard interactif.

2.2 Image Docker Unifiée

Pour garantir la cohérence, nous utilisons une seule image ('climate-mlops-weather-api') basée sur `apache/airflow:2.7.3-python3.10`. Cette image contient à la fois le moteur de calcul (Scikit-learn, Optuna), l'API et l'orchestrateur.

3 Le Cycle de Continuous Training

Le pipeline Airflow ('climate_pipeline_dag') est le cœur du système.

1. **Collecte Quotidienne (6h00)** : Récupération des données réelles depuis l'API Open-Meteo.
2. **Cycle de 7 jours** : Tous les 7 jours, le système lance un ré-entraînement complet.
3. **Sélection du "Best Model"** : Trois modèles s'affrontent (Linear Regression, Random Forest, Gradient Boosting). Le gagnant est choisi via un score composite (RMSE, R^2 , temps de calcul).
4. **Quality Gates** : Un modèle n'est promu en production que si son $R^2 \geq 0.7$ et s'il s'améliore par rapport au modèle précédent.

4 Guide de Validation et Déploiement

4.1 Phase 1 : Validation Locale

Avant tout déploiement Cloud, nous validons l'intégrité du système en local :

```
1 # 1. Lancer la stack locale
2 make up
3 # 2. Vérifier la santé du système
4 python3 verify_deployment.py
5 # 3. Exécuter les tests unitaires
6 make test
```

4.2 Phase 2 : Préparation Azure (ACR)

```
1 # Connexion et définition de la souscription
2 az login --use-device-code
3 az account set --subscription "1815cb03-0ab6-4382-9f78-d03c507c84e4"
4
5 # Création du registre
6 az acr create -g rg-projet -n climate-mlops-reg --sku Basic --admin-
    enabled true
7
8 # Push de l'image validée
9 az acr login --name climate-mlops-reg
10 export ACR_URL=$(az acr show -n climate-mlops-reg --query loginServer -o
    tsv)
11 docker tag climate-mlops-weather-api:latest $ACR_URL/climate-mlops:
    latest
12 docker push $ACR_URL/climate-mlops:latest
```

Bénéfices du Cloud

- **Scalabilité Automatique** : Gère les pics de charge sans intervention manuelle.
- **Infrastructure Gérée** : Haute disponibilité garantie par Azure (Switzerland North).
- **Modèle Serverless** : Paiement à l'usage réel, idéal pour l'optimisation des coûts.

```
1 az containerapp create \
2   --name weather-api \
3   --resource-group rg-projet \
4   --environment climate-mlops-env \
5   --image climate-mlops-reg.azurecr.io/climate-mlops:latest \
6   --target-port 8000 \
7   --ingress external \
8   --cpu 1.0 --memory 2.0Gi \
9   --env-vars PYTHONPATH="/workspace"
```

5 Maintenance et Nettoyage

Pour arrêter tous les services et libérer les ressources Azure après usage :

```
1 # Supprimer toutes les ressources Azure du projet  
2 az group delete --name rg-projet --yes --no-wait
```

6 Conclusion

Ce guide définit un workflow MLOps complet. Grâce à la conteneurisation et à l'automatisation d'Airflow, le système de Marrakech est prêt pour la production, offrant une solution robuste et scalable pour la prévision climatique.

Temporary page!

\LaTeX was unable to guess the total number of pages correctly. As there was some unprocessed data that should have been added to the final page this extra page has been added to receive it.

If you rerun the document (without altering it) this surplus page will go away, because \LaTeX now knows how many pages to expect for this document.