

# Ingénieur Déploiement IA

## Étape 1 : Configuration initiale du projet Flask

### 1.1. Installation des dépendances

#### 1. Environnement virtuel :

- Créez un environnement virtuel pour isoler les dépendances de votre projet.

#### 2. Bibliothèques à installer :

- Installez Flask pour le développement web.
- Installez Flask-CORS pour gérer les requêtes inter-domaines.
- Installez des bibliothèques pour le traitement d'images et l'intelligence artificielle, comme TensorFlow, PyTorch, ONNX, et Rasterio pour le traitement géospatial.

### 1.2. Structure du projet

Organisation du projet avec une structure de répertoires claire :

- **/static/** : Contient les fichiers statiques tels que CSS, JavaScript et images.
- **/templates/** : Contient les fichiers HTML pour le rendu des pages web.
- **/models/** : Contient les modèles d'intelligence artificielle.
- **/data/** : Contient les images satellites et autres données géospatiales.
- **app.py** : Fichier principal de l'application Flask.
- **requirements.txt** : Fichier de gestion des dépendances.

### 1.3. Création de l'application Flask

- **Configuration de base** : Définissez les routes pour les différentes fonctionnalités, comme le chargement d'images, l'exécution de modèles d'IA et l'affichage des résultats.
- **Gestion des erreurs** : Préparez des réponses appropriées pour les erreurs, comme des messages d'erreur clairs si un modèle n'est pas pris en charge ou si des données sont manquantes.

## Étape 2 : Implémentation des fonctionnalités principales

### 2.1. Chargement d'images satellites

- **Fonctionnalité** : Permettre aux utilisateurs de charger des images satellites
- **Interactivité** : Fournir une interface pour que les utilisateurs sélectionnent une région sur une carte et envoient les coordonnées correspondantes à l'application.

### 2.2. Exécution de modèles d'IA

- **Fonctionnalité** : Exécuter des modèles d'intelligence artificielle (comme TensorFlow et ONNX) sur les images chargées.
- **Types de modèles** : Permettre aux utilisateurs de choisir le modèle à exécuter et retourner les résultats sous forme de prédictions.
- **Analyse des résultats** : Intégrer des indicateurs de performance clés (KPI) pour évaluer la performance des modèles (précision, rappel, etc.).

### 2.3. Affichage des résultats

- **Visualisation** : Générer des cartes d'occupation, des visualisations des résultats des modèles, et des indicateurs de performance.
- **Interface utilisateur** : Créer une interface intuitive pour afficher les résultats sous forme graphique, facilitant la compréhension des données par les utilisateurs.

## Étape 3 : Technologies et outils

### 3.1. Technologies de développement

- **Flask** : Pour la création de l'API et le développement du backend.
- **GDAL et Rasterio** : Pour le traitement et l'analyse des images géospatiales.

### 3.2. Outils supplémentaires

- **Docker** : Pour la conteneurisation de l'application, facilitant le déploiement.
- **Git** : Pour la gestion de version et le suivi des modifications.
- **PostgreSQL avec PostGIS** : Pour la gestion des données géospatiales si nécessaire.

## Étape 4 : Tests et mise en production

### 4.1. Tests

- **Tests unitaires** : Rédiger des tests unitaires pour chaque fonctionnalité de l'application afin de s'assurer qu'elles fonctionnent comme prévu.
- **Tests d'intégration** : Vérifier que le frontend et le backend interagissent correctement.

### 4.2. Déploiement

- **Mise en production** : Déployer l'application sur un serveur cloud (AWS, Google Cloud, etc.) en utilisant des conteneurs Docker.
- **Surveillance** : Implémenter une solution de surveillance pour suivre les performances de l'application et résoudre les problèmes en temps réel.

## Étape 5 : Documentation et support

### 5.1. Documentation

- **API** : Rédiger une documentation complète de l'API, incluant les points de terminaison, les paramètres requis, et les exemples de requêtes/réponses.
- **Guide utilisateur** : Créer un guide utilisateur pour aider les utilisateurs à naviguer sur la plateforme.

### 5.2. Support

- **Canal de support** : Mettre en place un canal de support (comme un forum ou une adresse email) pour que les utilisateurs puissent poser des questions et signaler des problèmes.