Fiche 1

Découverte de l'environnement & Construction du Data + Model Pipeline :

- La découverte de l'environnement AWS.
- La configuration de GitHub avec AWS.
- La mise en place d'un Data Pipeline complet.
- La construction d'un Model Pipeline reproductible.

---

1. Découverte d'AWS

1. Connectez-vous à votre console AWS.
2. Repérez les services suivants :
    - ECR (Elastic Container Registry)
    - ECS (Elastic Container Service)
    - S3
    - IAM
    - CloudWatch
    - VPC
3. Tests pratiques : une action par groupe
    - Créez un bucket S3 nommé : `groupeX-data`
    - Créez un repository ECR nommé : `groupeX-mlops`
    - Explorez ECS : clusters, services, tâches
    - Inspectez IAM sans modifier les règles existantes

2. Configuration GitHub ↔ AWS

1. Création d'un utilisateur IAM : déjà fait dans le compte

    - Créer un utilisateur IAM limité pour GitHub Actions :
        - Permissions minimales : ECR, ECS deploy, S3 read/write
        - Créer une clé d'accès

2. Dans GitHub :

- Ajouter les secrets :
  - `AWS_ACCESS_KEY_ID`
  - `AWS_SECRET_ACCESS_KEY`
  - `AWS_REGION`

3. Tester la configuration AWS :par exemple

- Créez le dossier : `.github/workflows/`
- Créez le fichier : `.github/workflows/test-aws.yml`
- Allez sur GitHub → onglet **Actions**
- Sélectionnez le workflow **"Test AWS Connection"**
- Cliquez sur **"Run workflow"** → **"Run workflow"**
- Consultez les logs pour vérifier que tout fonctionne

```yaml
name: Test AWS Connection

on:
  workflow_dispatch

jobs:
  test-aws:
    runs-on: ubuntu-latest

    steps:
      - name: Configure AWS credentials
        uses: aws-actions/configure-aws-credentials@v2
        with:
          aws-access-key-id: ${{ secrets.AWS_ACCESS_KEY_ID }}
          aws-secret-access-key: ${{ secrets.AWS_SECRET_ACCESS_KEY }}
          aws-region: ${{ secrets.AWS_REGION }}

      - name: Test AWS credentials validity
        run: |
          echo "Testing AWS credentials..."
          aws sts get-caller-identity

      - name: List S3 buckets
        run: |
          echo "Listing S3 buckets..."
          aws s3 ls
```

3. Data Pipeline

1. Choix et mise à disposition des données

- Prévoir le modèle et format de stockage initial des données :
  - fichier CSV, JSON, Parquet, base de données SQL légère (SQLite, DuckDB) dans le bucket S3
  - postgres dans rds aura
- Documente le format de données initial

2. Automatisation ETL

- Créer un répertoire : `src/data/` dans le projet
- Exemple scripts à produire :
  - `download_data.py`
  - `clean_transform.py`
  - `load_final.py`
  - `data_pipeline.py` # Orchestrer du data pipeline

3. Stockage des données finales

- Fichiers locaux : CSV, Parquet, JSON, DuckDB, SQLite
- Base SQL locale via Supabase
- Vérifier la reproductibilité de bout en bout

4. Model Pipeline

1. Structure du dossier : exemple

```
src/model/
models/
notebooks/
```

2. Étapes fondamentales

- Chargement données finales
- Split train/test
- Entraînement plusieurs modèles si possible
- Sélection du meilleur modèle
- Sauvegarde du modèle (`joblib`, `pickle`)

3. Automatisation en model pipeline

- Lire les données finales
- Entraîner le modèle
- Générer un rapport de performance
- Exporter le modèle dans `models/`