

## Atelier 7 : Monitoring en MLOps avec MLflow, Elasticsearch et Kibana

### Objectif

L'objectif de ce TP est de **monitorer un modèle de machine learning** en enregistrant les métriques et artefacts avec **MLflow**, en envoyant les logs vers **Elasticsearch**, et en visualisant les données avec **Kibana**.

### Pré-requis

- **WSL** (Windows Subsystem for Linux) installé
- **Ubuntu sous WSL**
- **Docker installé sous WSL**
- **Python (>=3.8) installé**

---

### 1. Installation des outils nécessaires

Installer Docker Compose :

```
sudo apt install -y docker.io docker-compose
```

---

### 2. Déploiement de la stack Elasticsearch & Kibana

Télécharger les deux images docker elasticsearch et kibana. Créez deux conteneurs elasticsearch et kibana qui communiquent ensemble.

**PS :** Vous pouvez faire une recherche pour lancer ces deux conteneurs en utilisant docker-compose :

#### Lancer Elasticsearch et Kibana

docker-compose up -d (si vous travaillez avec docker-compose.yml)

Sinon avec les commandes (docker run .....)

### 3. Adapter le fichier python des fonctions avec Elasticsearch

Réaliser les étapes suivantes au niveau de votre code python :

- Connexion à Elasticsearch

- Envoyer les logs de mlflow vers Elasticsearch

#### 4. Démarrer MLflow

Démarrez le serveur MLflow :

```
mlflow server --backend-store-uri sqlite:///mlflow.db --default-artifact-root  
./mlruns --host 0.0.0.0 --port 5000
```

---

#### 5. Exécuter les étapes grâce à l'utilisation du makefile

---

#### 6. Visualiser les résultats

##### MLflow UI

Accédez à <http://localhost:5000> pour voir les expériences enregistrées.

##### Kibana

1. Accédez à <http://localhost:5601>
2. Allez dans "**Stack Management**" > "**Index Patterns**"
3. Créez un index **mlflow-metrics**
4. Utilisez **Discover** pour explorer les logs envoyés par MLflow.

---

#### 7. Excellence

Vous pouvez proposer le monitoring de n'importe quelle partie du projet. Par exemple, on peut suivre le comportement de notre machine ubuntu (RAM, CPU, Espace disque, etc..), les conteneurs docker, les différents outils utilisés, les données, etc.

#### Conclusion

🎯 Vous avez maintenant un **pipeline de monitoring** en place :

- **MLflow** pour suivre les métriques et sauvegarder le modèle
- **Elasticsearch** pour stocker les logs
- **Kibana** pour visualiser les métriques

🚀 Ce TP vous donne une **base solide** pour intégrer du monitoring en MLOps !