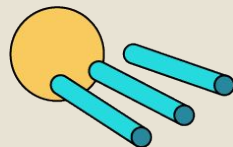


KICK-OFF PÉDAGOGIQUE

MLOps



SOMMAIRE

1

Métier et cursus

2

Présentation de l'accompagnement

3

Le parcours pédagogique

4

Déroulement des projets

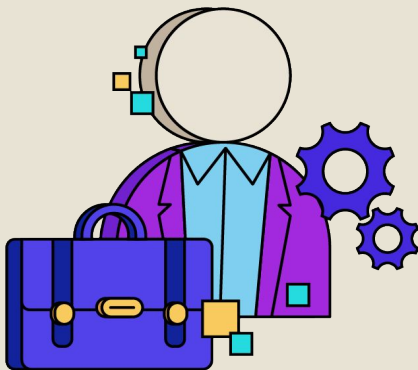


Métier



QU'EST-CE QU'UN ML ENGINEER ?

Le ML Engineer est chargé de :



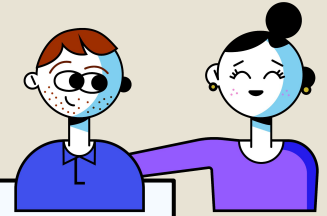
Intégrer des **process de DevOps** dans une architecture Data

Mettre en place et maintenir une approche **CI/CD**

Déployer, automatiser, mettre en **production** les modèles de ML

Gérer les **architectures** et les **flux** de données

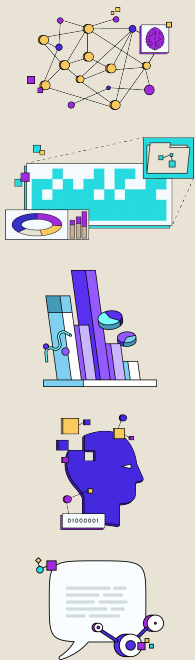
L'ACCOMPAGNEMENT PÉDAGOGIQUE



LE CHEF DE COHORTE



Rova
Chef de la cohorte



Référent de promotion

Valide le **parcours pédagogique**

Accompagne sur l'ensemble de la
formation

A votre **écoute**

Avec un délai de réponse est de **24h**
ouvrées



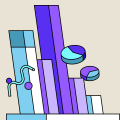
PROGRAM MANAGER



réfèrent non-pédagogique



disponible pour échanger



suit l'ensemble de la **formation**



est à votre **écoute**



le délai de réponse est de **4H ouvrées**



Gaelle
Program Manager de la
cohorte

LE PARCOURS PÉDAGOGIQUE



PRÉSENTATION DU PARCOURS PÉDAGOGIQUE



Environment setup

Linux & Bash, Docker, Unit Testing, APIs & Security, LLM Development Environment



Experiment tracking and versioning

MLflow, Weights & Biases, DVC & DagsHub, NGINX, LangChain & LLM Experimentation



Orchestration, deployment, and LLMOps

Airflow, Prefect, BentoML, RAG & LLMOps



Monitoring and Agentic AI

Prometheus & Grafana, Kafka, Agentic Design Patterns



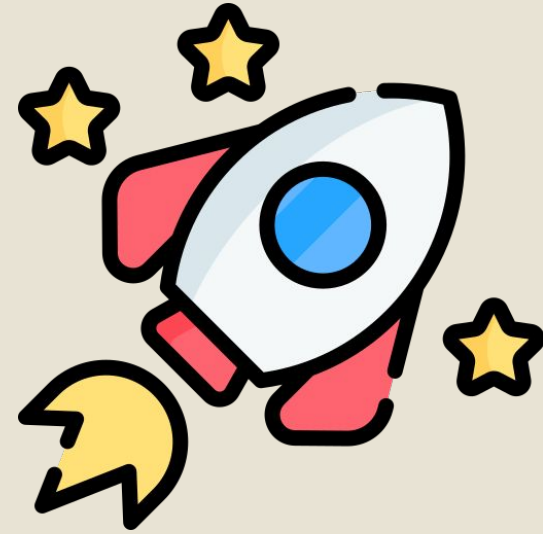
Scaling and MLOps Platform

Kubernetes, ZenML, AWS Cloud Practitioner

- Parcours divisé en **5 sprints** composés de :
 - Modules obligatoires
 - Modules optionnels
 - Masterclass
 - Projets avec livrables intermédiaires
- Total de 4 masterclass obligatoires
- Parcours pédagogique partagé à l'issue de la séance : pensez à le consulter régulièrement



Kick-off Projet MLOPS



Sessions de mentorat



Organisation des sessions

- Un point d'1h **maximum** sur **zoom / teams** fréquemment (Toutes les semaines)
- L'objectif sera de passer en revue vos derniers travaux et de répondre à vos questions
- A la fin de chaque point, le mentor pourra vous fixer des objectifs (en termes de réalisation ou de date) collectifs ou individuels (pour équilibrer la charge de travail)
- L'équipe en entier sera responsable du backlog



Les points à présenter

Les points à présenter par les membres de l'équipe lors de chaque point :

1. Les tâches réalisées par les différents membres de l'équipe
2. Les succès, challenges et éventuels blocages
3. Ce qu'il vous reste pour atteindre les objectifs (le backlog)



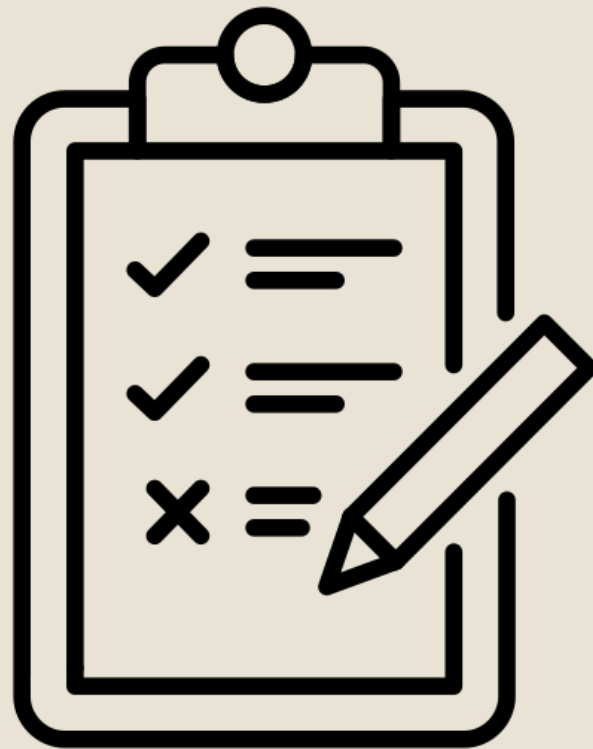
Méthodologie projet MLOPS



Vue d'ensemble

Phases Principales

1. Foundations
2. Microservices & versioning
3. Orchestration & Déploiement
4. Monitoring & Mise à l'échelle



Phase 1 : Fondations

Objectifs

- Cadrer votre projet MLOPS
- Mettre en place l'environnement de développement reproductible
- Implémenter des pipelines basiques
- Démarrer les pratiques de versioning et de tests automatisés reproductibles
- Implémenter une API basique

Composants Clés à Implémenter

1. Versioning du code (Git)
2. Environnements reproductibles
3. Pipelines de traitement des données et d'entraînement ML basique
4. Documentation initiale

Défis Attendus

- Cohérence des environnements
- Organisation du code
- Gestion des données
- Bases de la collaboration



Phase 2 : Microservices, Suivi des Expériences & Versioning

Objectifs

- Suivi efficace des expériences d'entraînement
- Versioning des données et modèles
- Décomposition en microservices interdépendants
- Début d'orchestration simple

Composants Clés à Implémenter

1. Système de suivi des expériences
2. Versioning des données
3. Isolation des services
4. Versioning to bout en bout

Point d'attention

- Organisation des expériences
- Traçabilité des données
- Reproductibilité des résultats
- Automatisation des pipelines



Phase 3 : Orchestration & Déploiement

Objectifs

- Finalisation de l'orchestration de bout en bout et implémentation du réentraînement
- Mise en place d'une pipeline d'intégration continu (CI)
- Optimisation et sécurisation de l'API (avec autorisation et authentification) de prédiction
- Mise en place d'une scalabilité légère

Composants Clés à Implémenter

1. APIs sécurisés
2. CI basique (CD optionnel)
3. Processus de déploiement et de réentraînement automatisé

Point d'attention

- Bonne pratique de sécurisation
- Stratégie de test et de réentraînement
- Automatisation du déploiement



Phase 4 : Surveillance & Maintenance

Objectifs

- Surveiller la performance des modèles et la qualité des données
- Mettre en place un dashboard de monitoring automatisé
- Finaliser la logique de maintenance et de mise à jour des composants / modèles
- Nice to have : front-end simplifié (streamlit, fastHTML etc.)

Point d'attention

- Performance des modèles
- Qualité des données
- Santé du système

Composants Clés à Implémenter

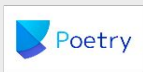
1. Monitoring des performances
2. Détection de dérive des données
3. Système d'alertes
4. Processus de réentraînement



Phase 1

Environment
Setup

CONDA



2. Training

1. Data Management : ETL Pipeline

Data Ingestion



DVC

Data Versioning

Feature Eng.



Storage

MINIO

Training data

(2) Training
Pipeline

Model training

mlflow W&B
Model tracking &
registry

neptune.ai

Model

Deployment
Pipeline

Model API

FastAPI



Phase 2

3. Automation



Production data
monitoring



Phase 4 Monitoring

Model monitoring



Phase 3

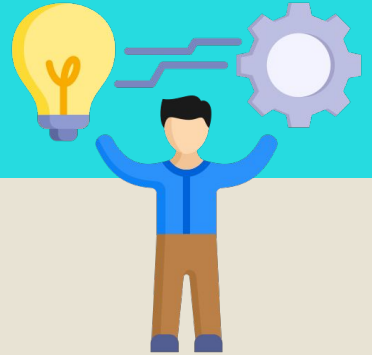
4. Deployment



FastAPI



Bonne pratiques



Bonnes pratiques

Développement 🧑💻

- 🔄 Utiliser le versioning
- 📄 Documenter les décisions
- ✍️ Tester régulièrement
- 👁️ Revoir le code

Gestion des Données 📊

- 📦 Versionner les datasets
- ✅ Valider la qualité des données
- 🔍 Tracer les transformations
- 📅 Documenter les features

Gestion des Modèles 🤖





- 📈 Tracer les expériences
- 💾 Versionner les modèles
- ⚙️ Documenter les paramètres
- 🎯 Valider les résultats

Déploiement 🚀


- ⚡ Automatiser les processus
- ✍️ Tester en profondeur
- 📊 Surveiller les performances
- 📅 Planifier la maintenance

Pour réussir

Implémentation

-  Pipeline ML fonctionnel
-  Processus automatisés
-  Déploiement surveillé
-  Système maintenu

Documentation

-  Doc technique claire
-  Processus à jour
-  Doc API complète
-  Guides maintenance

Progression

-  Compréhension MLOps
-  Bonnes pratiques
-  Collaboration efficace
-  Amélioration continue

* Les critères doivent être évalués régulièrement tout au long du projet



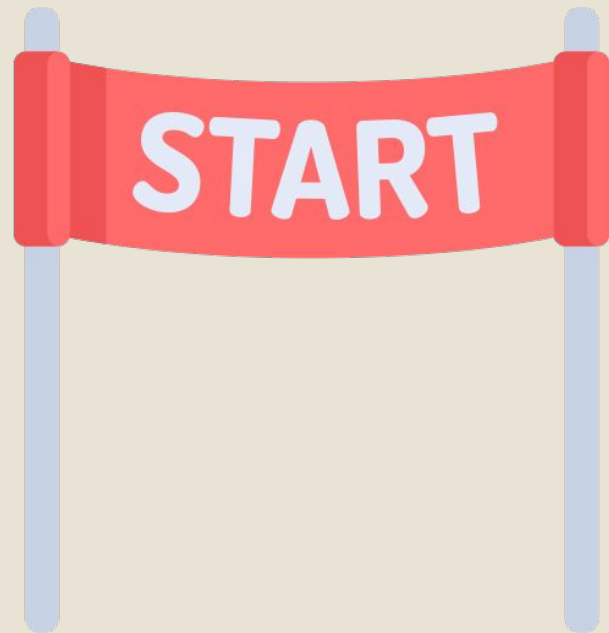
Ressources

Quelques ressources pour vous aider à commencer :

1. https://github.com/ssime-git/mlops_project_template
2. <https://github.com/ssime-git/mlops-boiler-plate>
3. https://github.com/ssime-git/zenml_mlops_template

Quelques exemples de projets d'apprenants :

1. <https://github.com/mchelali/picopix>
2. https://github.com/DataScientest-Studio/jul24_cmlops_reco_film



Question & Feedbacks

