
**MLOps est un mythe! Pipeline E2E de
test, pack et versionner**



Ali Amine Ghazali

2025

?

?

?

?

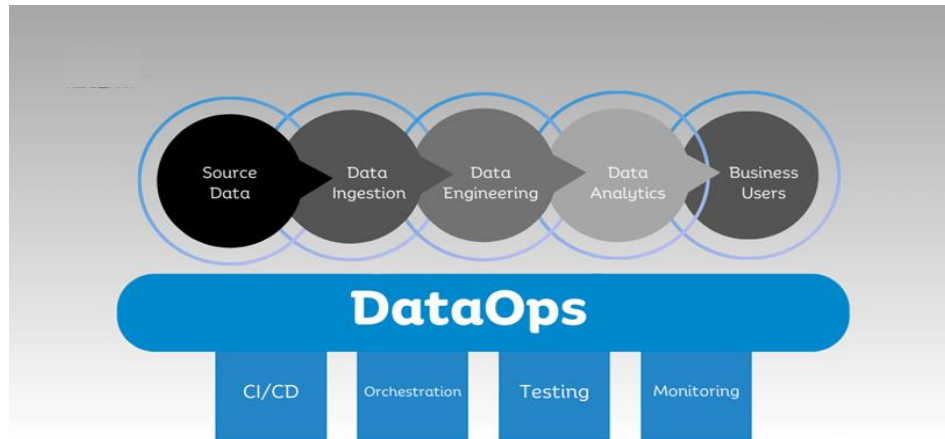
?



?

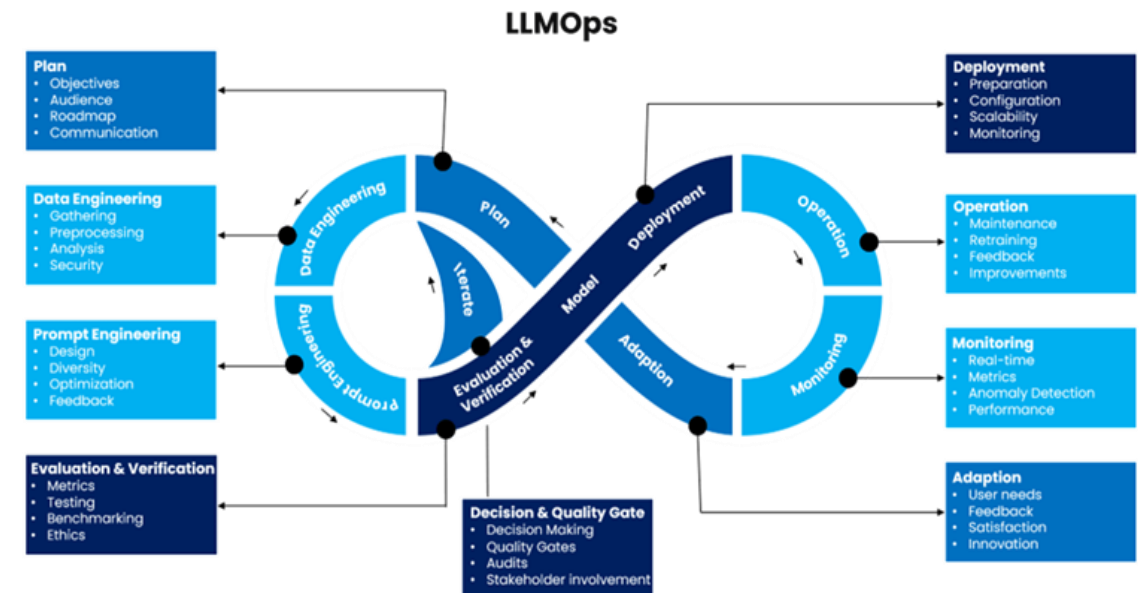
?

CONCEPTS INTERRESSANTS

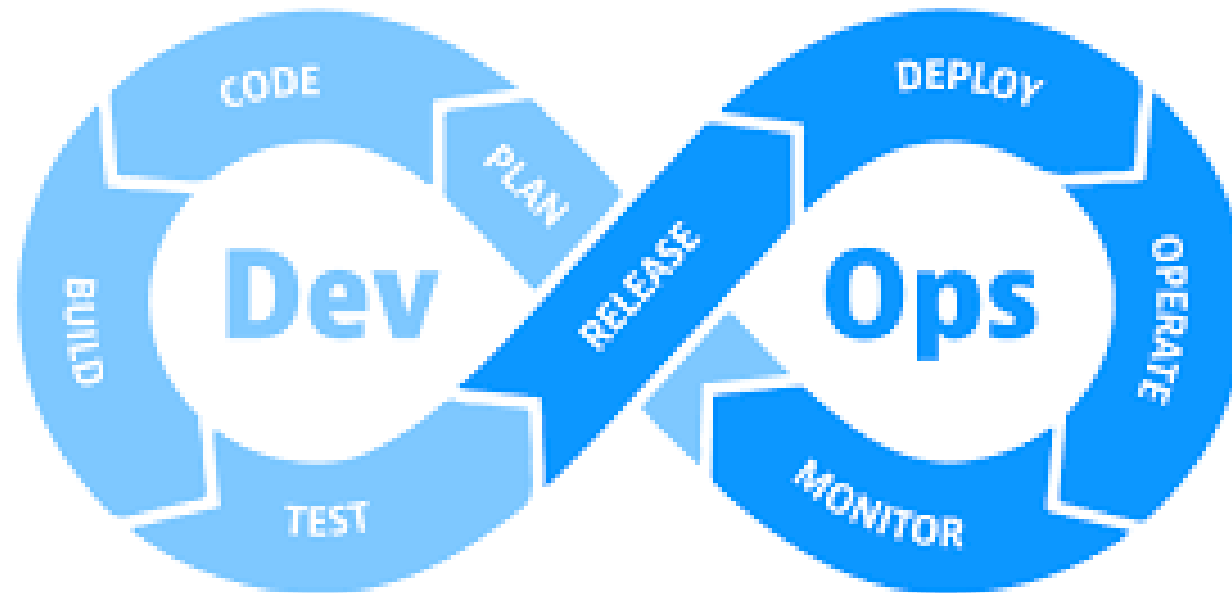


[DataOps: A Comprehensive Guide to Principles, Benefits, and More](#)

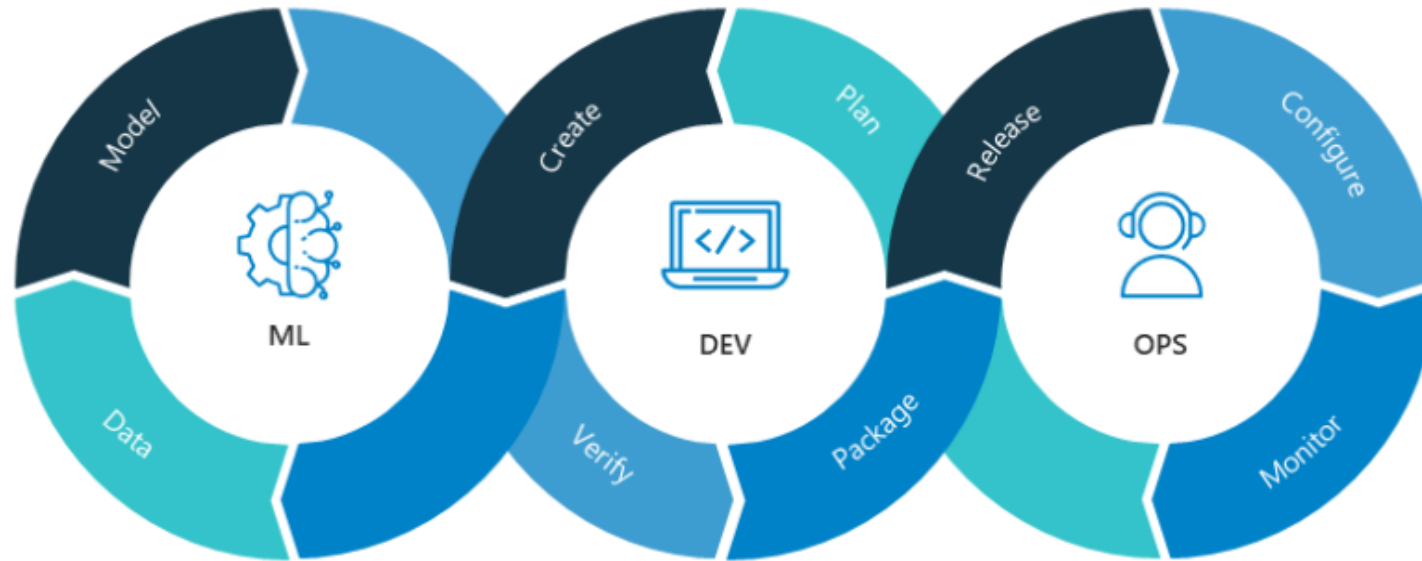
[LLMOps Explained: Boost Your LLM Implementation](#)



CONCEPT DEVOPS

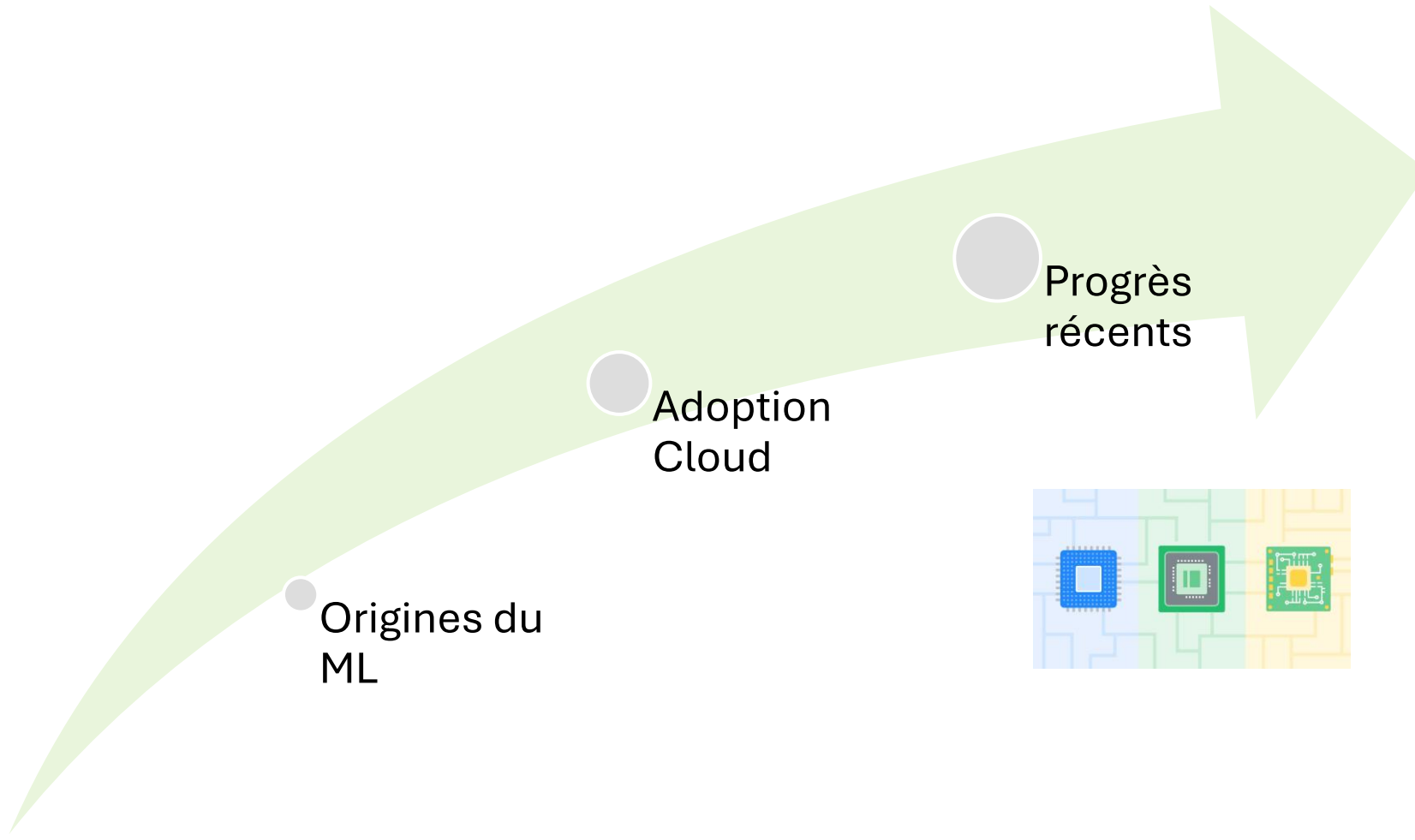


MLOPS

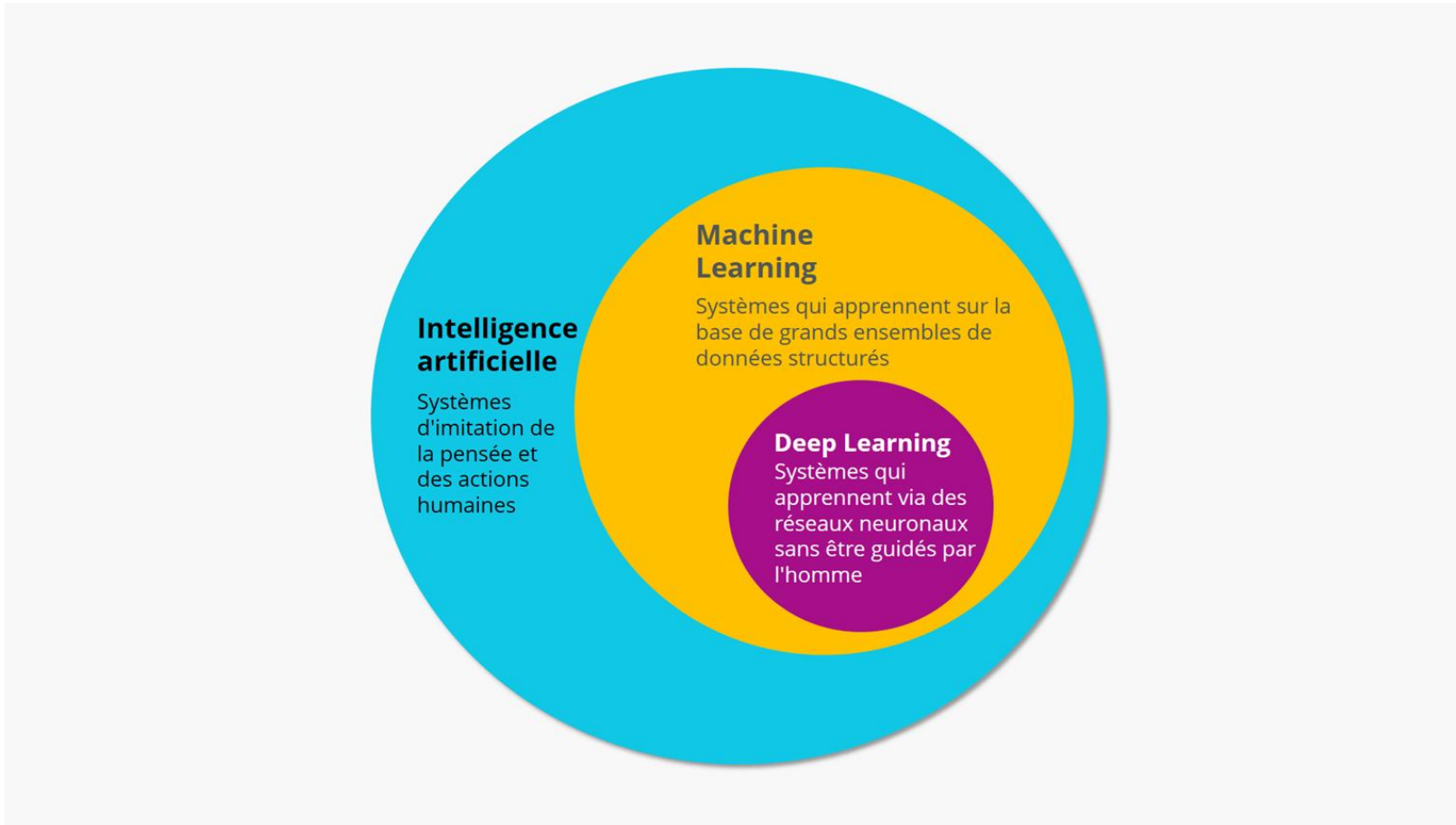


« La combinaison des principes DevOps avec les spécificités du Machine Learning »

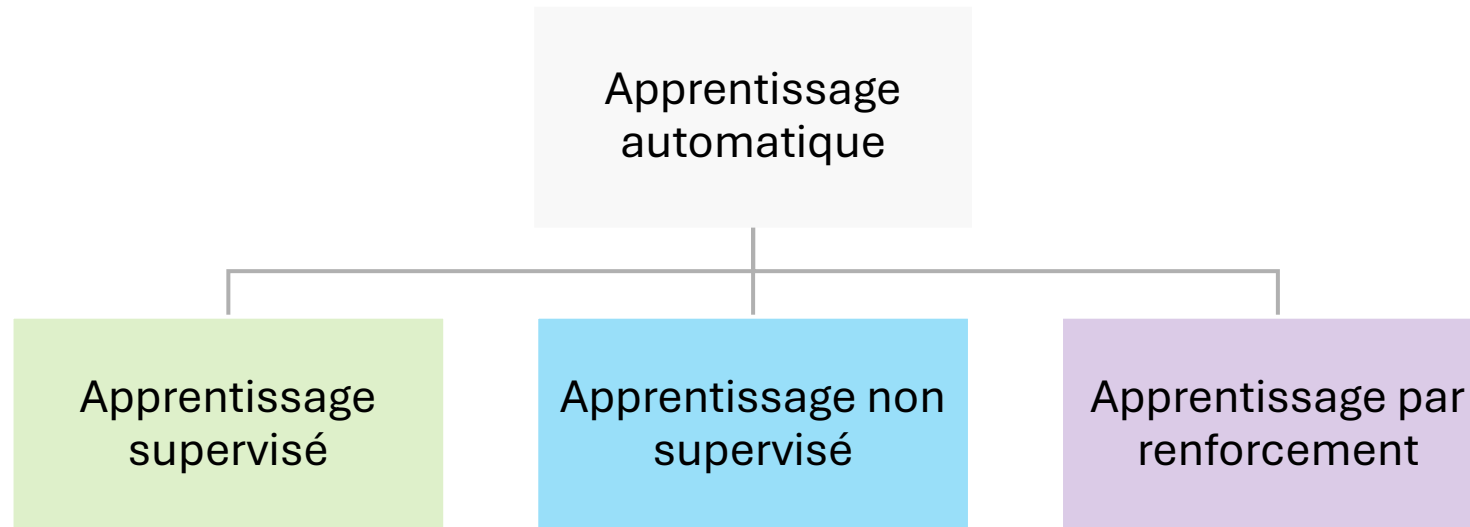
CONTEXTE & ÉVOLUTION DU MACHINE LEARNING



DISTINCTION DES NOTIONS - IA - ML - DL



TYPES D'APPRENTISSAGE AUTOMATIQUE



un programme classique et déterministe

Code produit final est un binaire

Résulte d'un entraînement sur des données

Produit final est un fichier contenant les poids entraînés

<https://github.com/confooca/2025>



NOS EXPERTS AU PROGRAMME

ConFoo.CA
DEVELOPER CONFERENCE

Q LEARNING... DE LA
THÉORIE À LA PRATIQUE

26 février @ 15 h

MLOPS EST UN MYTHE! PIPELINE
E2E DE TEST, PACK ET
VERSIONNER

28 février @ 11 h

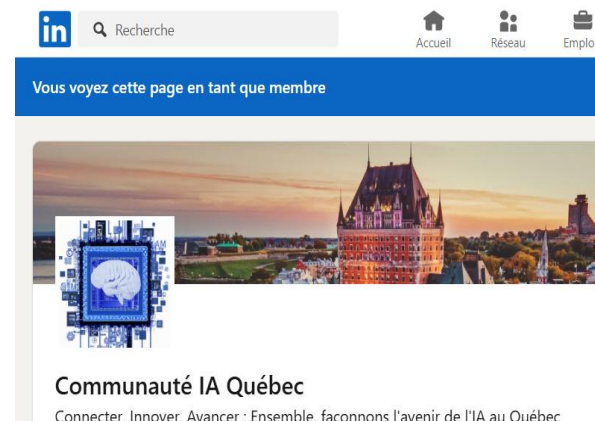
COFOMO



Ali Amine Ghazali

Directeur, Centre
d'excellence en intelligence
artificielle

<https://www.linkedin.com/in/aliamine-ghazali/>



Membre du conseil canadien des normes – comité ISO/IEC – 42001 Technologies de l'information — Intelligence artificielle — Système de management

DÉPLOIEMENT D'UNE APPLICATION ML

SYNTHÈSE FONCTIONNELLE



Préparation des données



**Entraînement et évaluation du
modèle**



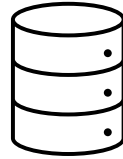
Sérialisation et inférence



Assurance qualité et déploiement

DÉPLOIEMENT D'UNE APPLICATION ML

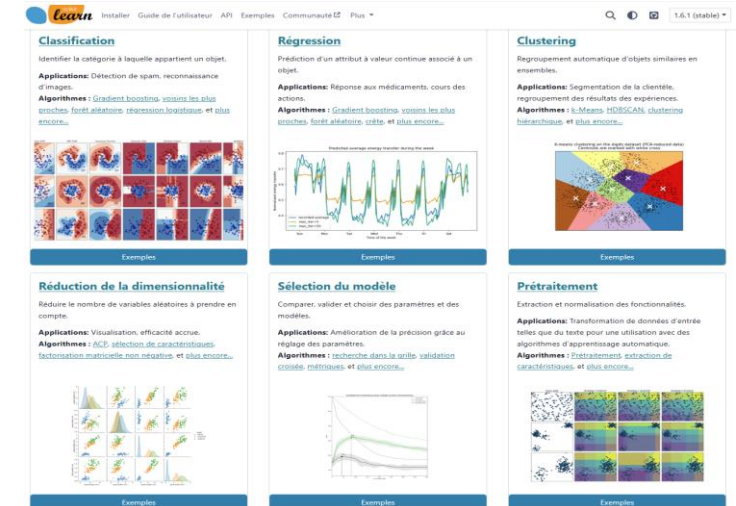
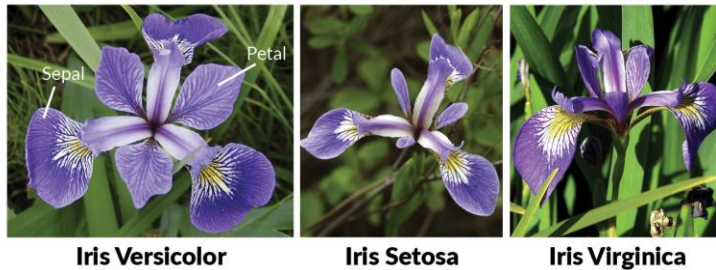
TECHNOLOGIES



Dataset Iris

Ses principales caractéristiques sont :

- 150 échantillons répartis en trois espèces d'iris (50 par espèce).
- Quatre mesures pour chaque fleur : longueur et largeur des sépales, longueur et largeur des pétales.
- Chaque observation est associée à l'une des trois espèces : *Iris setosa*, *Iris versicolor* et *Iris virginica*.

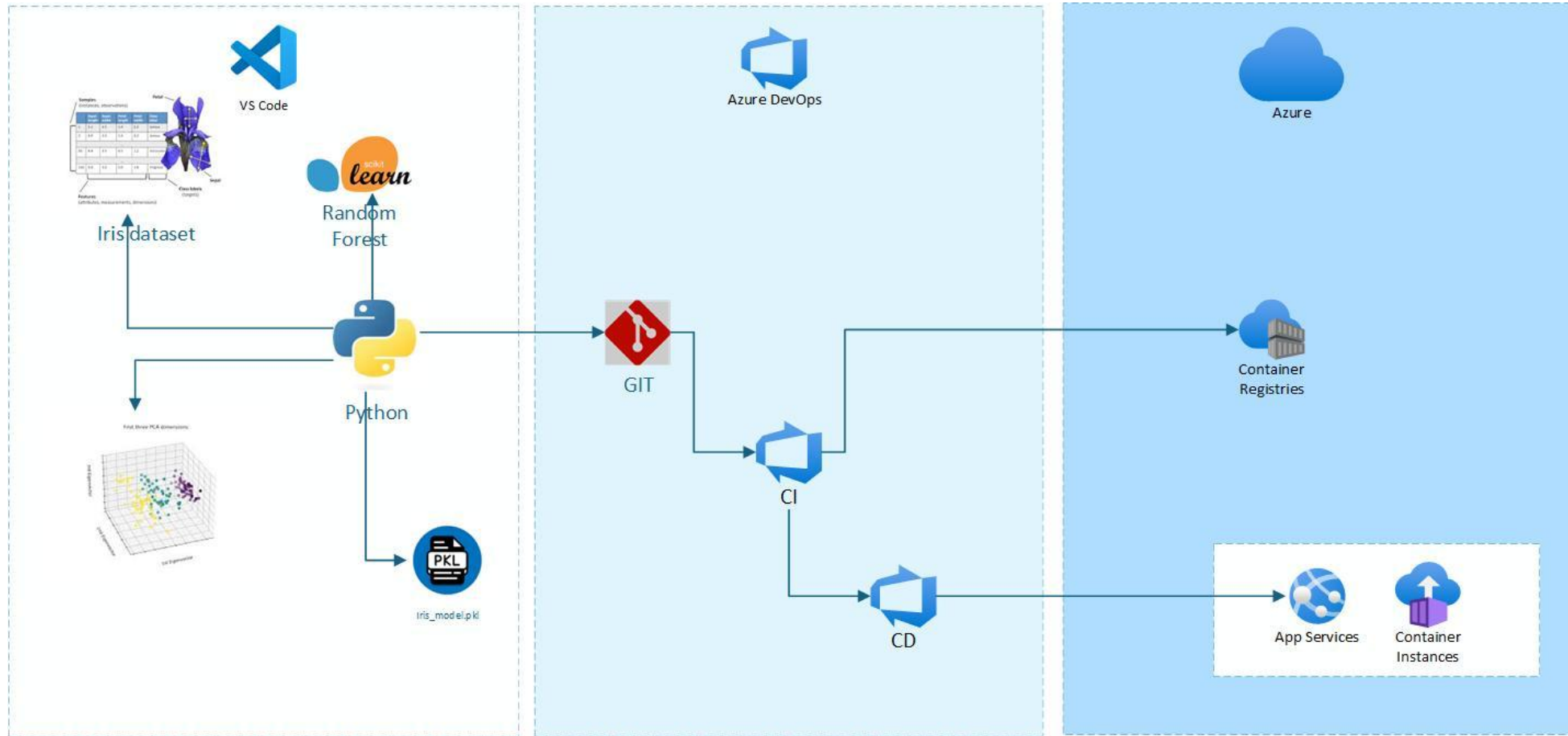


RandomForest



DÉPLOIEMENT D'UNE APPLICATION ML

ARCHITECTURE



DÉPLOIEMENT D'UNE APPLICATION ML

EXÉCUTION DE L'APPLICATION

1. Charge le jeu de données Iris.
2. Lancer l'algorithme d'entraînement.
3. Évalue le résultat.
4. Sauvegarde le modèle.
5. Tester l'application

DÉPLOIEMENT D'UNE APPLICATION ML

CHARGEMENT ET PRÉPARATION DES DONNÉES IRIS

- Source de données :

L'application utilise le jeu de données Iris, qui peut être chargé directement via la bibliothèque *scikit-learn*.

- Préparation :

Les données sont extraites sous forme de caractéristiques (features) et d'étiquettes (labels).

DÉPLOIEMENT D'UNE APPLICATION ML

ENTRAÎNEMENT DU MODÈLE

MLflow :

1. Séparation des données : Les données sont divisées en ensembles d'entraînement et de test pour évaluer la performance.
2. Apprentissage : Le modèle est ajusté sur l'ensemble d'entraînement avec **RandomForestClassifier**.
3. Évaluation : Une métrique est calculée sur l'ensemble de test pour vérifier la qualité de l'entraînement.
4. Sauvegarde : Une fois entraîné, le modèle est sérialisé (via pickle) et sauvegardé dans un fichier (**iris_model.pkl**), ce qui permet de le réutiliser ultérieurement sans devoir le réentraîner.

 `iris_model.pkl`

DÉPLOIEMENT D'UNE APPLICATION ML

COMMENT FONCTIONNE LA PRÉDICTION

- Chargement du modèle :

Une fonction dédiée permet de charger le modèle sauvegardé.

- Prédiction d'un échantillon :

Une fonction reçoit un vecteur de caractéristiques (par exemple, [5.1, 3.5, 1.4, 0.2]), utilise le modèle chargé pour effectuer une prédiction, et retourne la classe prédite.

DÉPLOIEMENT D'UNE APPLICATION ML

LANCER LES TESTS

Tests unitaires permettant la vérification des fonctionnalités :

- Test 1 = Existence du fichier de modèle, vérifie que le modèle a été correctement sauvegardé après l'entraînement.
- Test 2 = Chargement du modèle, s'assure que le modèle peut être chargé et est bien du type attendu (RandomForestClassifier).
- Test 3 = Prédiction teste la fonction de prédiction sur un échantillon connu pour s'assurer que le résultat appartient aux classes définies.

```
(env) PS C:\Users\ali\Documents\ConFoo\ConfooMLOps> pytest.exe
===== test session starts =====
platform win32 -- Python 3.11.5, pytest-8.3.4, pluggy-1.5.0
rootdir: C:\Users\ali\Documents\ConFoo\ConfooMLOps
collected 3 items

tests\test_app.py ... [100%]

===== 3 passed in 2.33s =====
(env) PS C:\Users\ali\Documents\ConFoo\ConfooMLOps>
```

DÉPLOIEMENT D'UNE APPLICATION ML

RÉSULTAT DE LA CLASSIFICATION

Si l'on note N le nombre total d'échantillons dans l'ensemble de test et N_{correct} le nombre d'échantillons correctement classifiés par le modèle, alors l'accuracy se calcule comme suit :

$$\text{Accuracy} = N_{\text{correct}} / N$$

```
(env) PS C:\Users\aliam\Documents\ConFoo\ConfooMLOps> pytest.exe
===== test session starts =====
===== test session starts =====
platform win32 -- Python 3.11.5, pytest-8.3.4, pluggy-1.5.0
platform win32 -- Python 3.11.5, pytest-8.3.4, pluggy-1.5.0
rootdir: C:\Users\aliam\Documents\ConFoo\ConfooMLOps
collected 3 items
collected 3 items

tests\test_app.py ... [100%]

===== 3 passed in 2.00s =====
(env) PS C:\Users\aliam\Documents\ConFoo\ConfooMLOps> .\env\Scripts\python.exe .\app.py
Accuracy sur le jeu de test : 1.00
(env) PS C:\Users\aliam\Documents\ConFoo\ConfooMLOps> 
```

DÉPLOIEMENT D'UNE APPLICATION ML

AJOUT D'UNE ÉTAPE DE PACKAGING

Une image Docker est construite à partir d'un Dockerfile, qui contient l'ensemble des instructions pour reproduire l'environnement de l'application.

```
Dockerfile > ...
1  # Dockerfile
2  FROM python:3.10-slim
3
4  # Définir le répertoire de travail
5  WORKDIR /app
6
7  # Copier le fichier de dépendances et l'installer
8  COPY requirements.txt .
9  RUN pip install --no-cache-dir -r requirements.txt
10  RUN pip freeze > requirements.txt
11
12
13  # Copier le reste du code
14  COPY . .
15
16  # Exposer le port si vous déployez une application web (adapté selon votre besoin)
17  EXPOSE 8000
18
19  # Commande par défaut (ici, par exemple, lancer l'entraînement du modèle)
20  CMD ["python", "app.py"]
21
```

DÉPLOIEMENT D'UNE APPLICATION ML

CRÉATION DU PIPELINE AZURE DEVOPS (CI)

Avec les instructions suivantes :

- **Trigger**
- **Pool d'agents**
- **Étape "UsePythonVersion«**
- **Installation des dépendances**

Exécution des tests

La commande pytest est exécutée avec l'option --junitxml=results.xml pour générer un rapport au format JUnit, nécessaire à la tâche suivante. L'option --maxfail=1 arrête l'exécution dès le premier échec, ce qui est utile pour une remontée rapide des erreurs.

- **Publication des résultats de test**

La tâche PublishTestResults prend en charge la publication du rapport de test pour visualiser les résultats directement dans l'interface Azure DevOps.

```
stages:
- stage: Build_Test
  displayName: 'Étape de Build et Tests'
  jobs:
  - job: Build_and_Test
    displayName: 'Installation des dépendances et exécution des tests'
    steps:
    - task: UsePythonVersion@0
      inputs:
        versionSpec: '3.x'
        displayName: 'Utiliser Python 3.x'

    - script: |
        python -m pip install --upgrade pip
        pip install -r requirements.txt
        displayName: 'Installer les dépendances'

    - script: |
        pytest --maxfail=1 --disable-warnings -q --junitxml=results.xml
        displayName: 'Exécuter les tests unitaires'

    - task: PublishTestResults@2
      inputs:
        testResultsFormat: 'JUnit'
        testResultsFiles: '**/results.xml'
        failTaskOnFailedTests: true
        displayName: 'Publier les résultats des tests'
```

DÉPLOIEMENT D'UNE APPLICATION ML

CRÉATION DU PIPELINE AZURE DEVOPS (CI)

Un stage nommé Docker_Deploy

- Étape 1 qui construit l'image Docker
- Étape 2 qui publie l'image

```
- stage: Docker_Deploy
  jobs:
  - job: Build_and_Push_Docker
    steps:
    - task: AzureCLI@2
      displayName: 'Se connecter à Azure et pousser l'image'
      inputs:
        azureSubscription: 'ACR Confoo service connection' # de type azurearm
        scriptType: 'bash'
        scriptLocation: 'inlineScript'
        inlineScript: |
          # Connexion au registre ACR
          az acr login --name testmlregistry11

          # Construction de l'image Docker
          docker build -t testmlregistry11.azurecr.io/$(dockerRegistry):$(Build.BuildId) .

          # Push vers ACR
          docker push testmlregistry11.azurecr.io/$(dockerRegistry):$(Build.BuildId)
```

DÉPLOIEMENT D'UNE APPLICATION ML

CRÉATION DU PIPELINE RELEASE(CD)

Cofomo-Azure / IA-AO / Pipelines / Releases / New release pipeline / Release-1 Search

+ New release pipeline > Release-1

Pipeline Variables History | + Deploy Cancel Refresh Edit ...

Release

Manually triggered
by Ali Ghazali
2025-02-27 19 h 07

Artifacts

_ConfooMLOps
20250214.4
 main

Stages

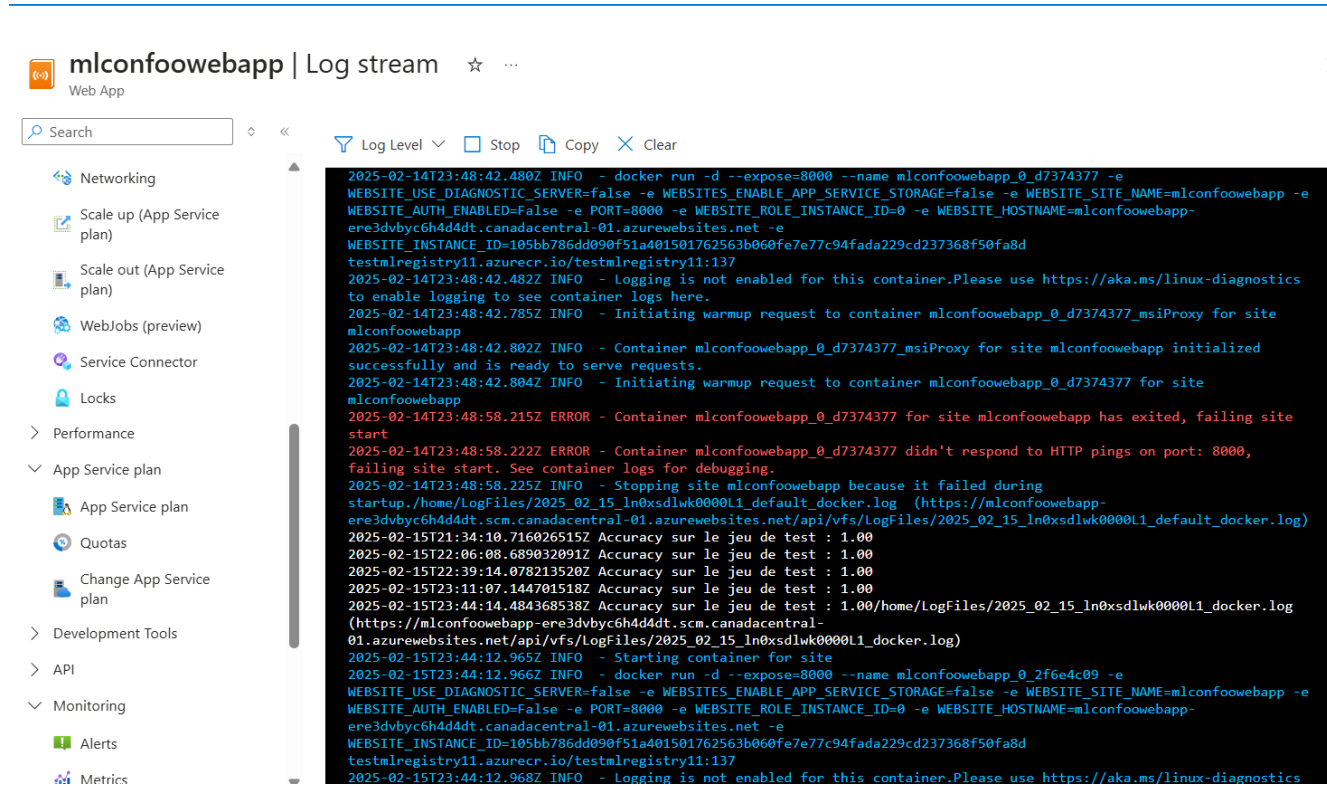
Dev
✓ Succeeded
on 2025-02-27 19 h 09

Production
✓ Succeeded
on 2025-02-27 19 h 19

roups

DÉPLOIEMENT D'UNE APPLICATION ML

DEPLOIEMENT WEB APP BASÉE SUR UN CONTENEUR



The screenshot displays the Azure portal interface for a web application named "mlconfoowebapp". The left sidebar shows various management options, with "Monitoring" and "Metrics" selected. The main area shows the "Log stream" for the application. The logs are displayed in a dark-themed window with a search bar and filters. The logs show the following sequence of events:

- 2025-02-14T23:48:42.480Z INFO - docker run -d --expose=8000 --name mlconfoowebapp_0_d7374377 -e WEBSITE_USE_DIAGNOSTIC_SERVER=false -e WEBSITES_ENABLE_APP_SERVICE_STORAGE=false -e WEBSITE_SITE_NAME=mlconfoowebapp -e WEBSITE_AUTH_ENABLED=false -e PORT=8000 -e WEBSITE_ROLE_INSTANCE_ID=0 -e WEBSITE_HOSTNAME=mlconfoowebapp-ere3dvbyc6h4d4dt.canadacentral-01.azurewebsites.net -e WEBSITE_INSTANCE_ID=105bb786dd090f51a401501762563b060fe7e77c94fada229cd237368f50fa8d testmlregistry11.azurecr.io/testmlregistry11:137
- 2025-02-14T23:48:42.482Z INFO - Logging is not enabled for this container. Please use <https://aka.ms/linux-diagnostics> to enable logging to see container logs here.
- 2025-02-14T23:48:42.785Z INFO - Initiating warmup request to container mlconfoowebapp_0_d7374377_msiProxy for site mlconfoowebapp
- 2025-02-14T23:48:42.802Z INFO - Container mlconfoowebapp_0_d7374377_msiProxy for site mlconfoowebapp initialized successfully and is ready to serve requests.
- 2025-02-14T23:48:42.804Z INFO - Initiating warmup request to container mlconfoowebapp_0_d7374377 for site mlconfoowebapp
- 2025-02-14T23:48:58.215Z ERROR - Container mlconfoowebapp_0_d7374377 for site mlconfoowebapp has exited, failing site start
- 2025-02-14T23:48:58.222Z ERROR - Container mlconfoowebapp_0_d7374377 didn't respond to HTTP pings on port: 8000, failing site start. See container logs for debugging.
- 2025-02-14T23:48:58.225Z INFO - Stopping site mlconfoowebapp because it failed during startup. /home/LogFiles/2025_02_15_ln0xsdllwk0000L1_default_docker.log (https://mlconfoowebapp-ere3dvbyc6h4d4dt.scm.canadacentral-01.azurewebsites.net/api/vfs/LogFiles/2025_02_15_ln0xsdllwk0000L1_default_docker.log)
- 2025-02-15T21:34:10.716026515Z Accuracy sur le jeu de test : 1.00
- 2025-02-15T22:06:08.689032091Z Accuracy sur le jeu de test : 1.00
- 2025-02-15T22:39:14.078213520Z Accuracy sur le jeu de test : 1.00
- 2025-02-15T23:11:07.144701518Z Accuracy sur le jeu de test : 1.00
- 2025-02-15T23:44:14.484368538Z Accuracy sur le jeu de test : 1.00/home/LogFiles/2025_02_15_ln0xsdllwk0000L1_docker.log (https://mlconfoowebapp-ere3dvbyc6h4d4dt.scm.canadacentral-01.azurewebsites.net/api/vfs/LogFiles/2025_02_15_ln0xsdllwk0000L1_docker.log)
- 2025-02-15T23:44:12.965Z INFO - Starting container for site
- 2025-02-15T23:44:12.966Z INFO - docker run -d --expose=8000 --name mlconfoowebapp_0_2f6e4c09 -e WEBSITE_USE_DIAGNOSTIC_SERVER=false -e WEBSITES_ENABLE_APP_SERVICE_STORAGE=false -e WEBSITE_SITE_NAME=mlconfoowebapp -e WEBSITE_AUTH_ENABLED=false -e PORT=8000 -e WEBSITE_ROLE_INSTANCE_ID=0 -e WEBSITE_HOSTNAME=mlconfoowebapp-ere3dvbyc6h4d4dt.canadacentral-01.azurewebsites.net -e WEBSITE_INSTANCE_ID=105bb786dd090f51a401501762563b060fe7e77c94fada229cd237368f50fa8d testmlregistry11.azurecr.io/testmlregistry11:137
- 2025-02-15T23:44:12.968Z INFO - Logging is not enabled for this container. Please use <https://aka.ms/linux-diagnostics>

DÉPLOIEMENT D'UNE APPLICATION ML

OPTION 2 DÉPLOIEMENT DE LA SOLUTION COMME ACI

The screenshot displays the Microsoft Azure portal interface for a container instance named 'aciconfoo'. The left sidebar shows the navigation menu with 'Containers' selected. The main content area shows the 'aciconfoo' container instance details. The instance is in a 'Waiting' state. The logs section shows the output of the container, which is 'Accuracy sur le jeu de test : 1.00'.

Container instances
CofomoAzureLab06 (CofomoAzureLab06.onmicro...)

+ Create Manage view ...

Filter for any field...

Name ↑↓

aciconfoo ...

aciconfoo Containers ☆ ...

Search Refresh Give feedback

1 container and 0 init containers

Name	Image	State	Previous state	Start time	Restart count
aciconfoo	testmlregistry11.azurec...	Waiting	-		0

Events Properties **Logs** Connect

To troubleshoot your container further, please [navigate to Logs](#). [Learn more](#)

Accuracy sur le jeu de test : 1.00

The background is a vibrant blue field filled with a complex network of glowing lines and dots. The lines are thin and vary in brightness, some appearing as sharp streaks while others are more diffuse. Small, bright blue dots are scattered throughout, often at the intersections of the lines, creating a sense of a digital or molecular structure. The overall effect is one of dynamic energy and technological sophistication.

DÉMONSTRATION

BONNES PRATIQUES MLOPS

ML , MLflow

- Séparer le chargement des données des fonctions de ML (ex: prédiction).
- Modéliation
- Versionner le modèle.
- Tester les fonctions de ML y compris celle de chargement des données.

DevOps

- Utiliser des outils de couverture de code.
- Produire une documentation complète de la solution exemple **Readme**.
- Publier les tests
- Rendre le plus possible l'application modulaire (ex: conteneur).
- Scripter le plus d'opérations possibles (CI/CD/Pack /Version/publication de tests..)
- Utiliser les notions d'approbation

<https://github.com/confooca/2025>



NOS EXPERTS AU PROGRAMME

ConFoo.CA
DEVELOPER CONFERENCE

Q LEARNING... DE LA
THÉORIE À LA PRATIQUE

26 février @ 15 h

MLOPS EST UN MYTHE! PIPELINE
E2E DE TEST, PACK ET
VERSIONNER

28 février @ 11 h

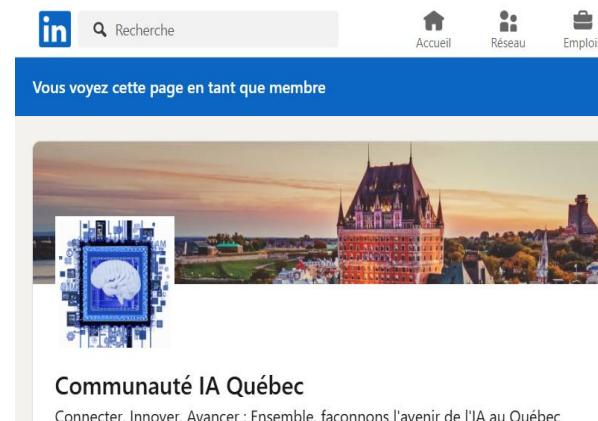
COFOMO



Ali Amine Ghazali

Directeur, Centre
d'excellence en intelligence
artificielle

<https://www.linkedin.com/in/aliamine-ghazali/>



Membre du conseil canadien des normes – comité ISO/IEC – 42001 Technologies de l'information — Intelligence artificielle — Système de management