Projet Final A57 - Folly Tata Ayeboua

Introduction

Ce projet final, réalisé dans le cadre du cours A57, a pour objectif de concevoir, déployer et gérer une application complète de machine learning sur le cloud Azure. Il met en œuvre un pipeline de bout en bout intégrant la gestion du code source, l'automatisation des déploiements, le suivi des expériences, l'exposition d'API, la visualisation des résultats et l'intégration de services d'intelligence artificielle avancés via OpenAI. L'intégration d'OpenAI à MLflow permet de suivre, versionner et comparer les expériences utilisant les modèles d'IA générative, facilitant ainsi l'évaluation et la reproductibilité des résultats. Le projet exploite des technologies modernes telles que MLflow (avec support OpenAI), Streamlit, FastAPI, Postgres et Portainer pour offrir une solution robuste, évolutive et facilement maintenable.

Description du projet

L'application développée permet de :

- Gérer et versionner le code source via GitHub.
- Automatiser les déploiements grâce à des workflows CI/CD avec des runners self-hosted sur Azure.
- Suivre les expériences de machine learning et gérer les modèles avec MLflow.
- Proposer une interface utilisateur interactive pour la visualisation et l'interaction avec les modèles via Streamlit.
- Exposer des API RESTful pour l'inférence en temps réel avec FastAPI.
- Stocker les données, résultats et métadonnées dans une base de données Postgres.
- Superviser les conteneurs et services déployés à l'aide de Portainer.

Objectifs du projet

- Déployer une machine virtuelle (VM) sur Azure pour héberger l'ensemble des services.
- Configurer un réseau virtuel sécurisé et ouvrir les ports nécessaires pour chaque composant.
- Mettre en place une gestion sécurisée des secrets et accès via GitHub.
- Assurer la reproductibilité, la traçabilité et la facilité de déploiement des modèles ML.

Structure du projet

La structure du projet est organisée comme suit :

```
tp-final-a57/
├── .github/          # Workflows CI/CD
├── app/              # Code source de l'application (FastAPI, Streamlit)
├── ml/               # Scripts de machine learning et gestion des
expériences MLflow
├── infra/            # Scripts d'infrastructure (déploiement VM, réseau,
etc.)
├── data/             # Jeux de données et résultats
├── docs/             # Documentation et captures d'écran
├── Dockerfile        # Fichiers Docker pour les différents services
```

```
├── requirements.txt   # Dépendances Python
└── README.md          # Présentation du projet
```

Comment démarrer le projet

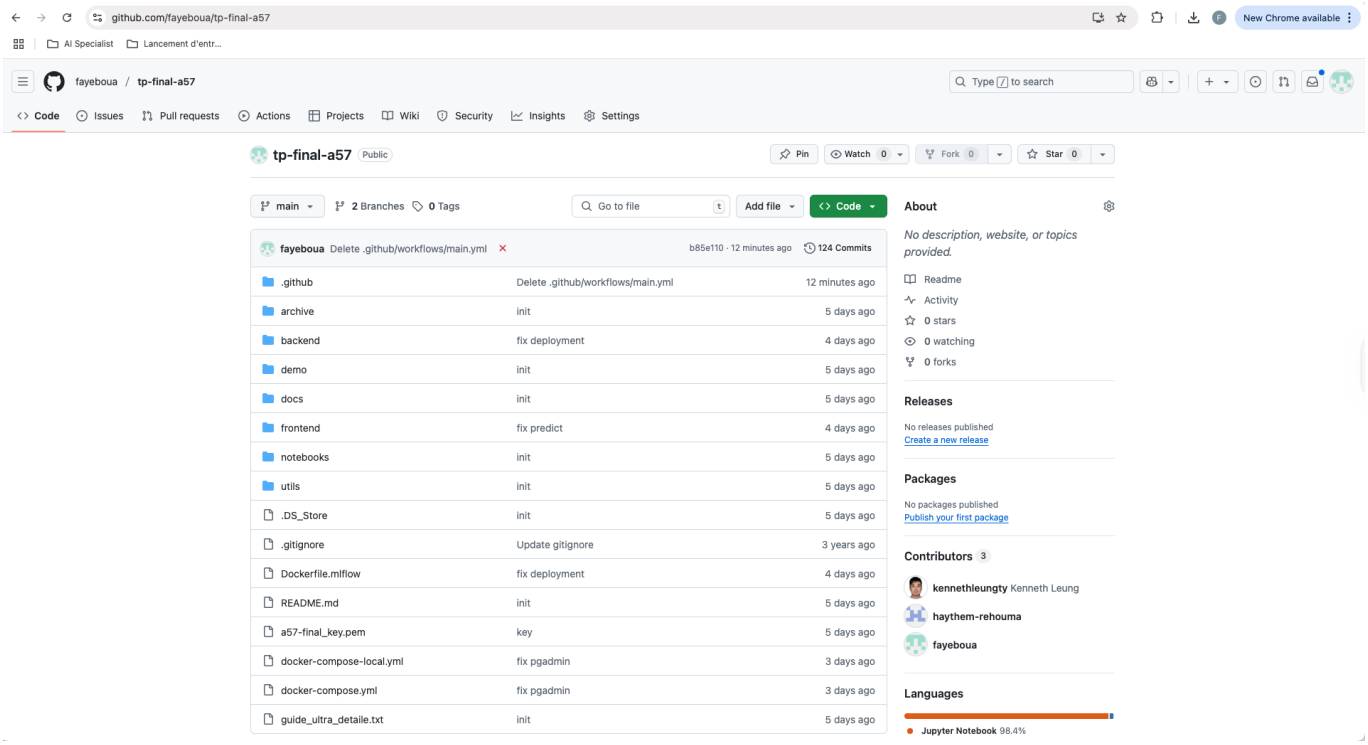
Prérequis

Avant de démarrer le projet, assurez-vous de disposer des éléments suivants :

- **Git installé**
  Téléchargez et installez Git depuis git-scm.com.
- **Compte Azure**
  Créez un compte Azure (azure.microsoft.com) et provisionnez une VM pour héberger les services du projet.
- **Clé API OpenAI**
  Inscrivez-vous sur platform.openai.com pour obtenir une clé API permettant d'utiliser les services d'IA générative.
- **Compte Docker Hub**
  Créez un compte sur hub.docker.com pour stocker et récupérer les images Docker nécessaires au déploiement.
- **Docker et Docker Compose installés**
  Installez Docker et Docker Compose sur votre machine ou sur la VM cible (docs.docker.com/get-docker/).
- **Accès SSH à la VM**
  Générez une paire de clés SSH pour accéder à la VM Azure de façon sécurisée.

Cloner le dépôt

```
git clone https://github.com/fayeboua/tp-final-a57.git
cd tp-final-a57
```



Création de la VM Azure

Déployez une VM dédiée et configurez un réseau virtuel pour isoler et sécuriser l'environnement.

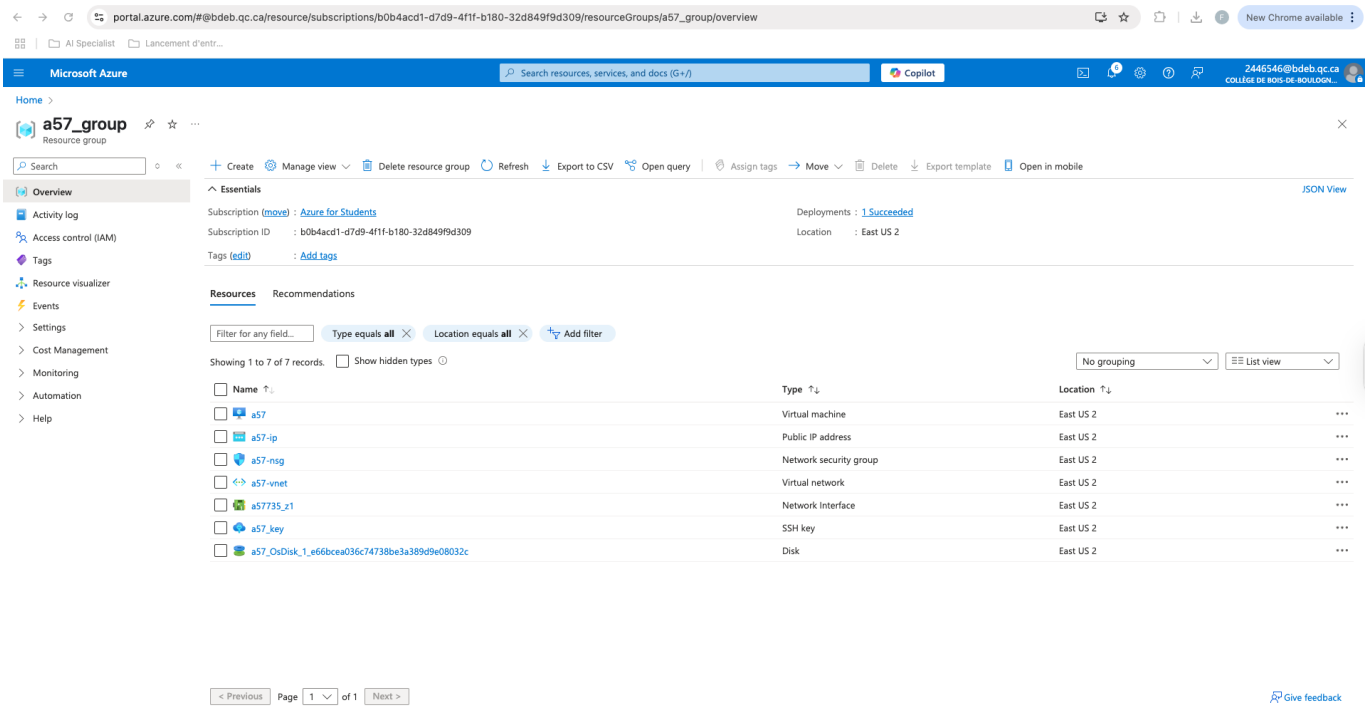
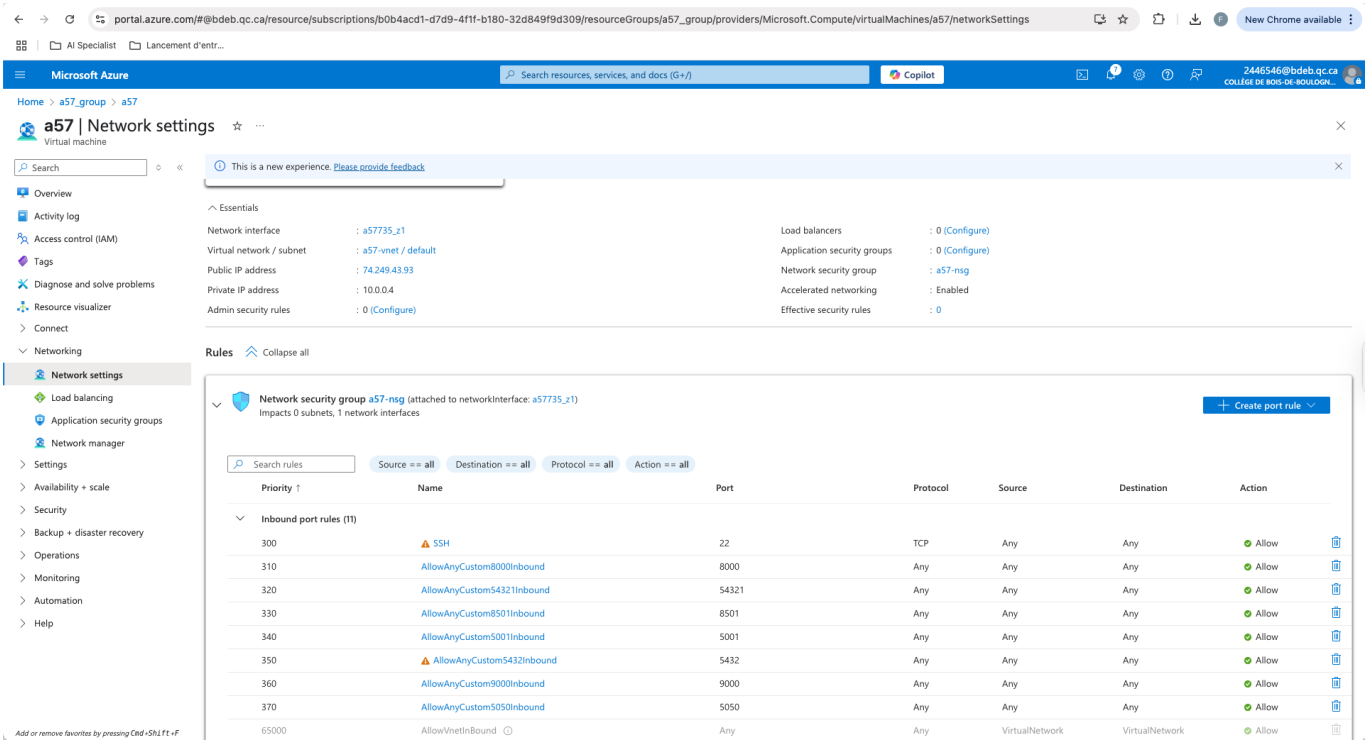


Tableau des ports utilisés

Service	Port par défaut	Description
Streamlit	8501	Interface web pour la visualisation
MLflow	50001	Suivi et gestion des expériences ML
FastAPI	8000	API RESTful pour l'inférence

Service	Port par défaut	Description
Postgres	5432	Base de données relationnelle
Portainer	9000	Supervision et gestion des conteneurs Docker

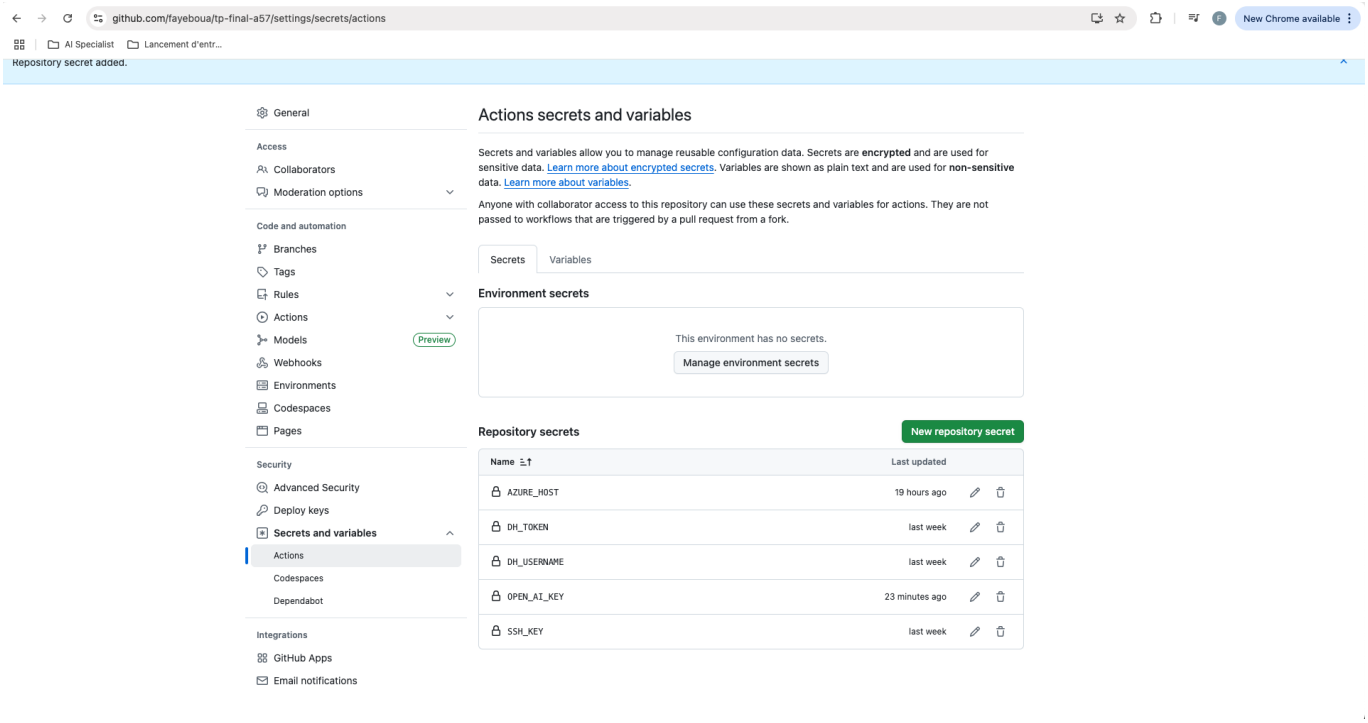
> **Remarque :** Assurez-vous d'ouvrir ces ports dans la configuration réseau de la VM Azure pour permettre l'accès aux différents services.



Configurer les variables d'environnement et secrets GitHub

Ajoutez les secrets (clés API, identifiants, etc.) dans la section "Secrets" de GitHub pour sécuriser les workflows CI/CD.

Nom du secret	Description
AZURE_HOST	Adresse de la machine virtuelle Azure
DH_TOKEN	Jeton d'accès Docker Hub
DH_USERNAME	Nom d'utilisateur Docker Hub
SSH_KEY	Clé SSH privée pour l'accès à la VM
OPEN_AI_KEY	Clé API pour accéder aux services OpenAI



Configuration des runners self-hosted

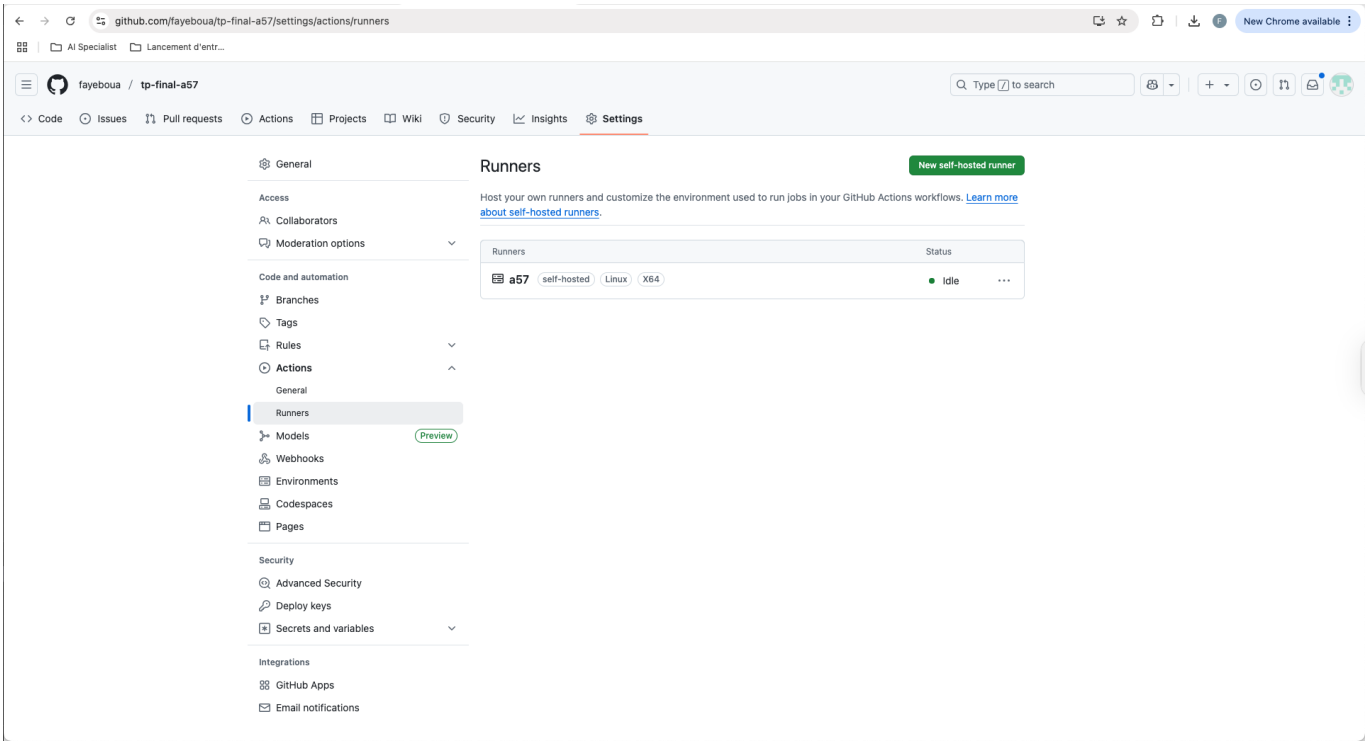
Déployez des runners GitHub auto-hébergés sur la VM Azure pour exécuter les pipelines d’intégration et de déploiement continus.

```
# Créez un dossier
mkdir actions-runner && cd actions-runner

# Téléchargez le dernier package runner
curl -o actions-runner-linux-x64-2.325.0.tar.gz -L
https://github.com/actions/runner/releases/download/v2.325.0/actions-
runner-linux-x64-2.325.0.tar.gz

# (Optionnel) Validez le hash
echo "5020da7139d85c776059f351e0de8fdec753affc9c558e892472d43ebeb518f4
actions-runner-linux-x64-2.325.0.tar.gz" | shasum -a 256 -c

# Extrayez l’installateur
tar xzf ./actions-runner-linux-x64-2.325.0.tar.gz
```

```
New release '24.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Fri Jun  6 22:59:49 2025 from 142.127.148.76
azureuser@a57:~$ ls -al ./actions-runner
total 117328
drwxr-xr-x 6 azureuser docker         4096 Jun  6 22:32 .
drwxr-x--- 7 azureuser azureuser      4096 Jun  7 02:26 ..
-rw-rw-r-- 1 azureuser docker          268 Jun  6 22:32 .credentials
-rw------- 1 azureuser docker         1667 Jun  6 22:32 .credentials_rsaparams
-rw-rw-r-- 1 azureuser docker           13 Jun  6 22:32 .env
-rw-rw-r-- 1 azureuser docker           99 Jun  6 22:32 .path
-rw-rw-r-- 1 azureuser docker          382 Jun  6 22:32 .runner
drwxrwxr-x 4 azureuser docker         4096 Jun  7 00:00 _diag
drwxrwxr-x 7 azureuser docker         4096 Jun  7 00:00 _work
-rw-rw-r-- 1 azureuser docker    120045291 Jun  6 22:31 actions-runner-linux-x64-2.325.0.tar.gz
drwxr-xr-x 4 azureuser docker        16384 Jun  2 18:43 bin
-rwxr-xr-x 1 azureuser docker         2458 Jun  2 18:42 config.sh
-rwxr-xr-x 1 azureuser docker          646 Jun  2 18:42 env.sh
drwxr-xr-x 4 azureuser docker         4096 Jun  2 18:43 externals
-rw-r--r-- 1 azureuser docker         1619 Jun  2 18:42 run-helper.cmd.template
-rwxr-xr-x 1 azureuser docker         2663 Jun  6 22:32 run-helper.sh
-rwxr-xr-x 1 azureuser docker         2663 Jun  2 18:42 run-helper.sh.template
-rwxr-xr-x 1 azureuser docker         2535 Jun  2 18:42 run.sh
-rwxr-xr-x 1 azureuser docker           65 Jun  2 18:42 safe_sleep.sh
-rwxr-xr-x 1 azureuser docker         5216 Jun  6 22:32 svc.sh
```

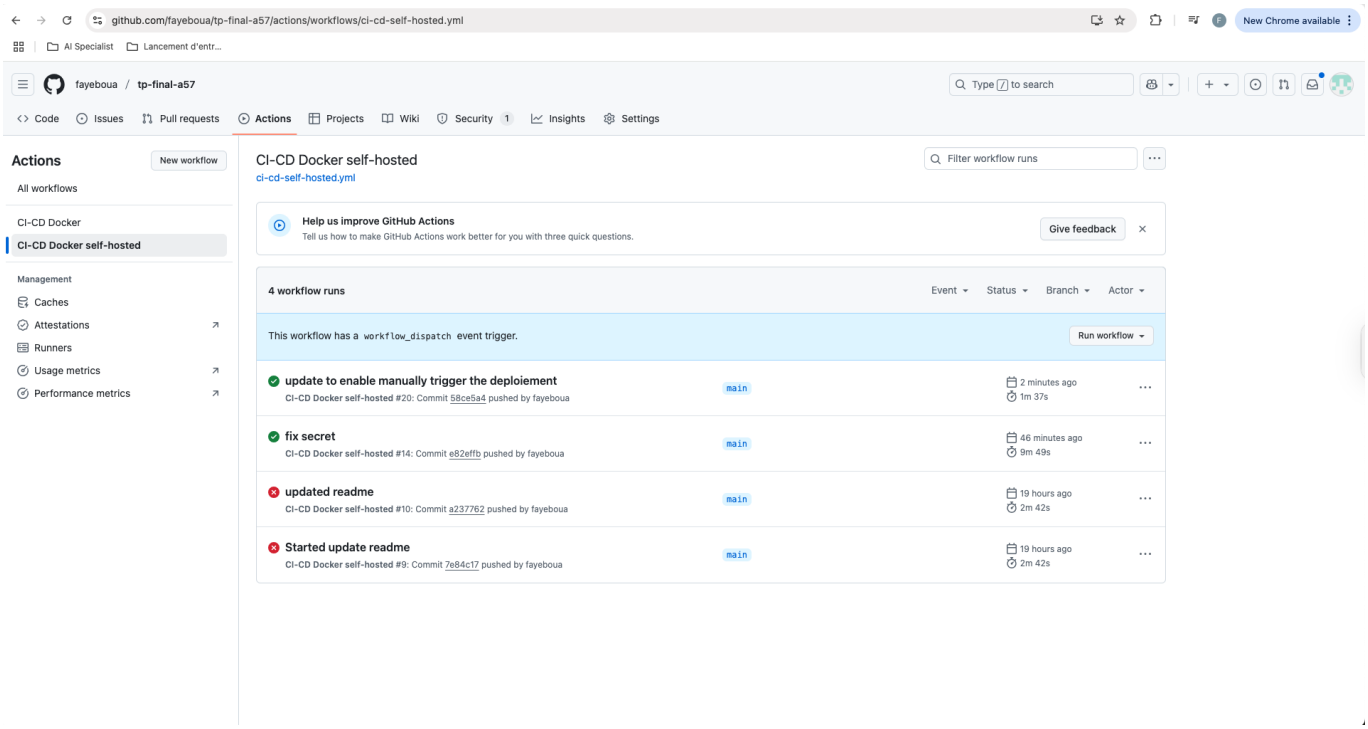
```
# Créez le runner et démarrez la configuration
./config.sh --url https://github.com/fayeboua/tp-final-a57 --token
<VOTRE_TOKEN_ICI>

# Dernière étape, lancez-le !
./run.sh
```

```
drwxr-xr-x 4 azureuser docker         4096 Jun  2 18:43 externals
-rw-r--r-- 1 azureuser docker         1619 Jun  2 18:42 run-helper.cmd.template
-rwxr-xr-x 1 azureuser docker         2663 Jun  6 22:32 run-helper.sh
-rwxr-xr-x 1 azureuser docker         2663 Jun  2 18:42 run-helper.sh.template
-rwxr-xr-x 1 azureuser docker         2535 Jun  2 18:42 run.sh
-rwxr-xr-x 1 azureuser docker           65 Jun  2 18:42 safe_sleep.sh
-rwxr-xr-x 1 azureuser docker         5216 Jun  6 22:32 svc.sh
azureuser@a57:~$ cd  ./actions-runner
azureuser@a57:~/actions-runner$ ./run.sh

√ Connected to GitHub

Current runner version: '2.325.0'
2025-06-07 18:10:37Z: Listening for Jobs
2025-06-07 18:10:38Z: Running job: build-and-push
```



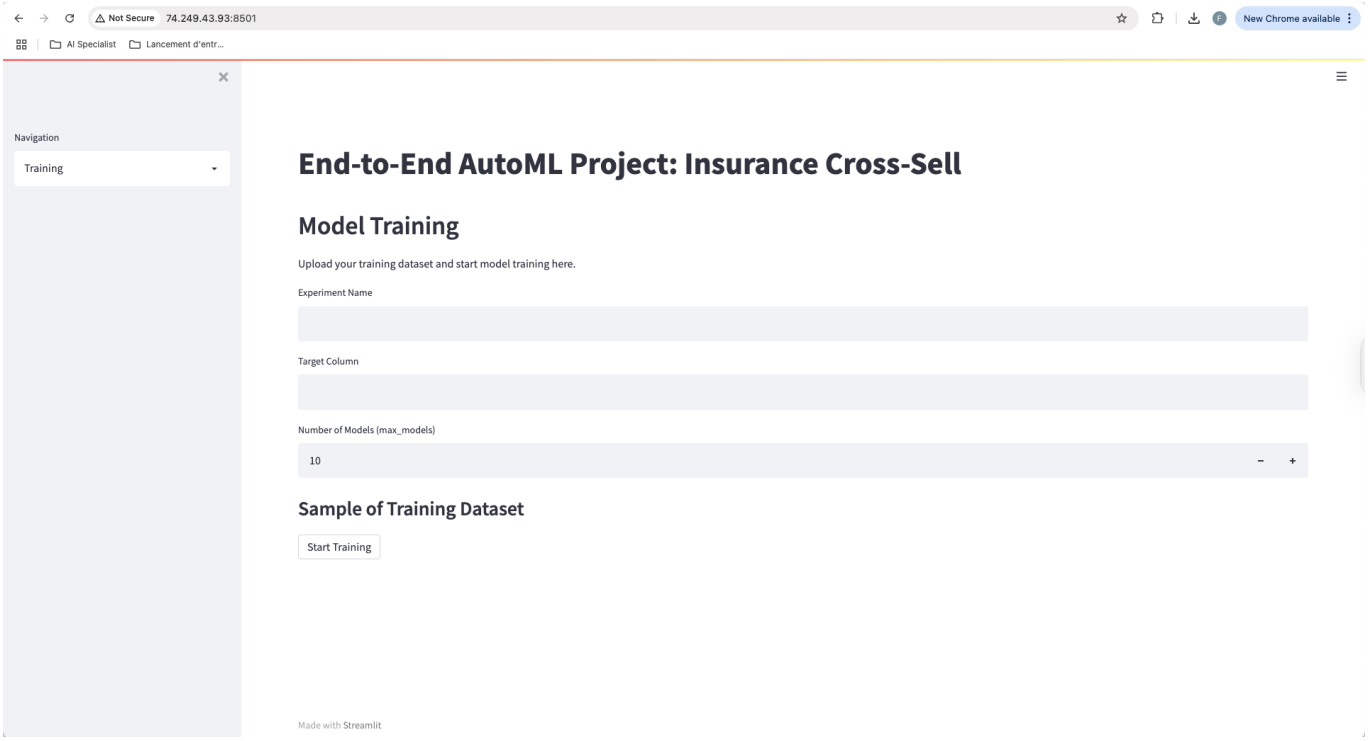
Déploiement automatique de l'application

Le déploiement s’effectue automatiquement après un push. Il peut également être déclenché manuellement en cliquant sur le bouton 'Run workflow'.



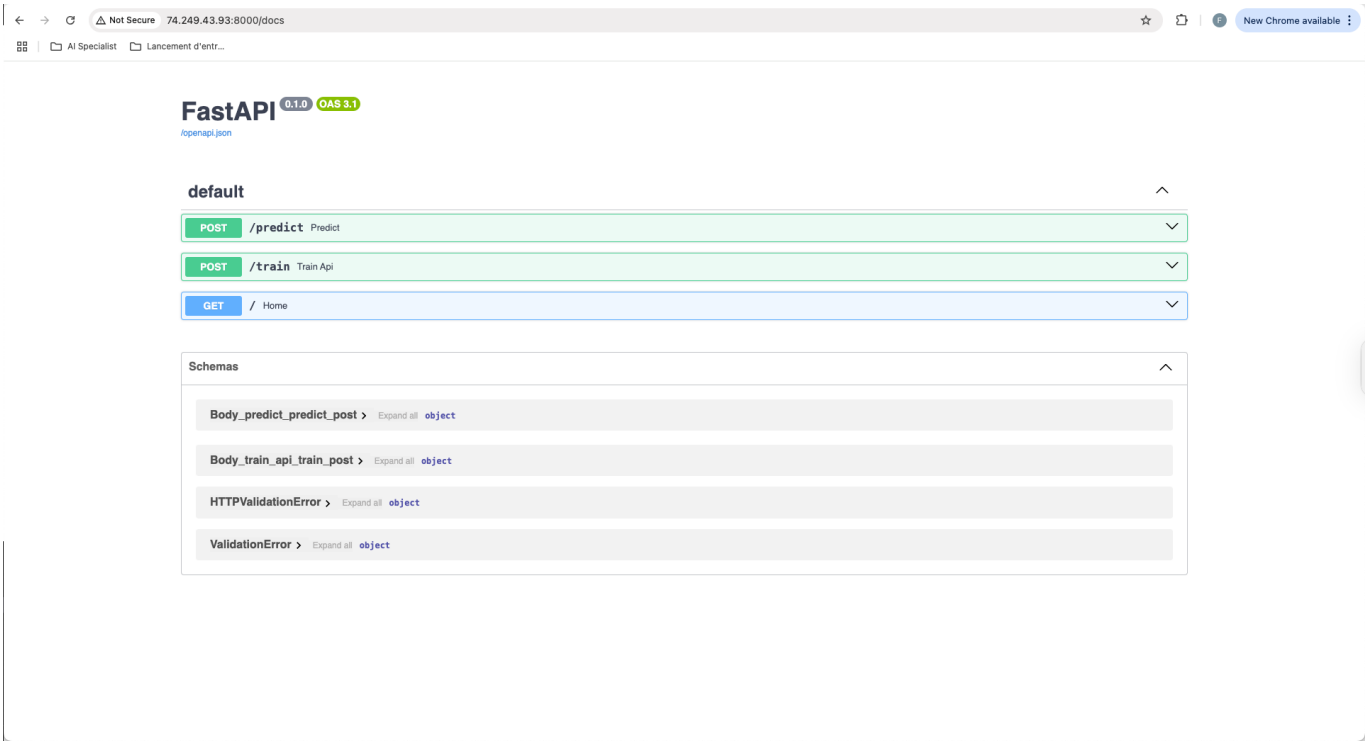
Interface utilisateur avec Streamlit

Développement d’une interface web interactive avec Streamlit pour visualiser les résultats, explorer les données et interagir avec les modèles déployés.



API d’inférence avec FastAPI

FastAPI expose des endpoints RESTful permettant d'effectuer des inférences à partir d'applications externes ou de scripts automatisés.



Connectioin à la VM Azure

```
sudo ssh -i a57_key.pem azureuser@<VOTRE_ADRESSE_IP>
```

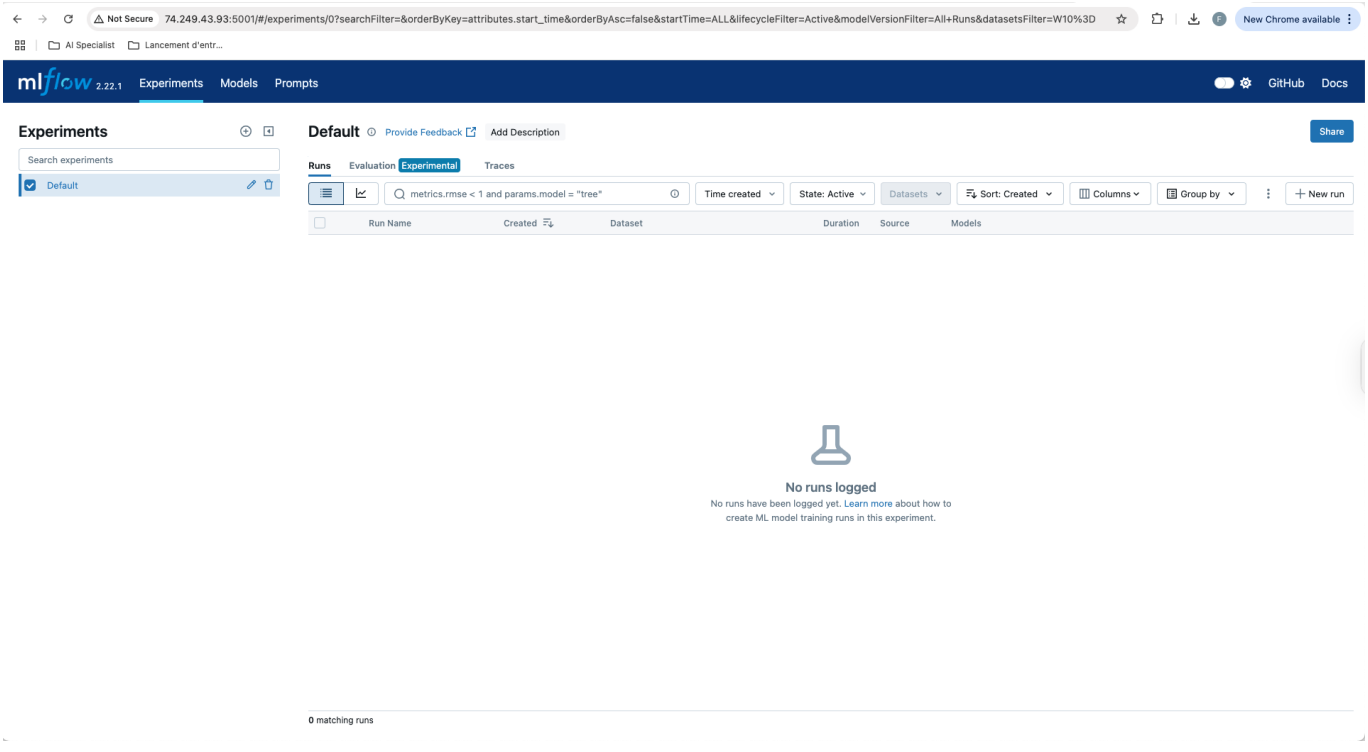
Remplacez `<VOTRE_ADRESSE_IP>` par l'adresse IP publique de votre VM Azure.

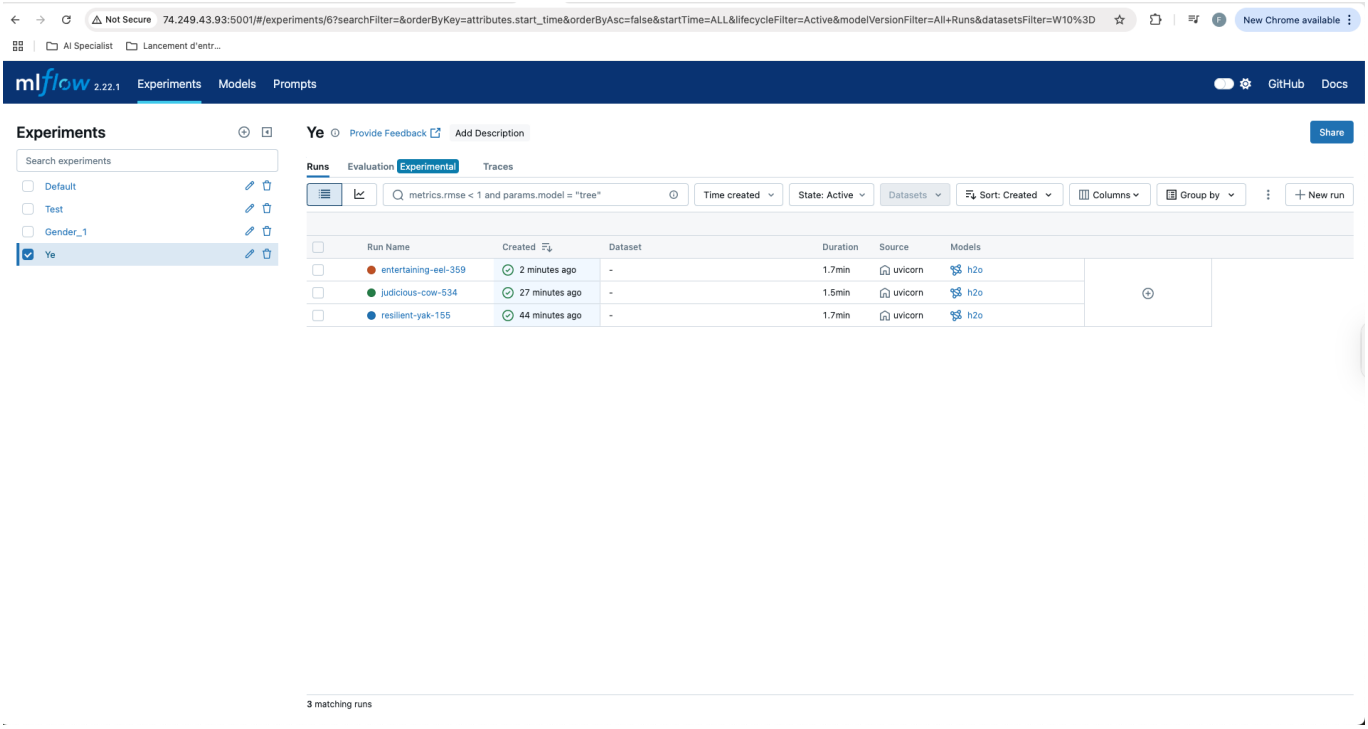
Liste de services

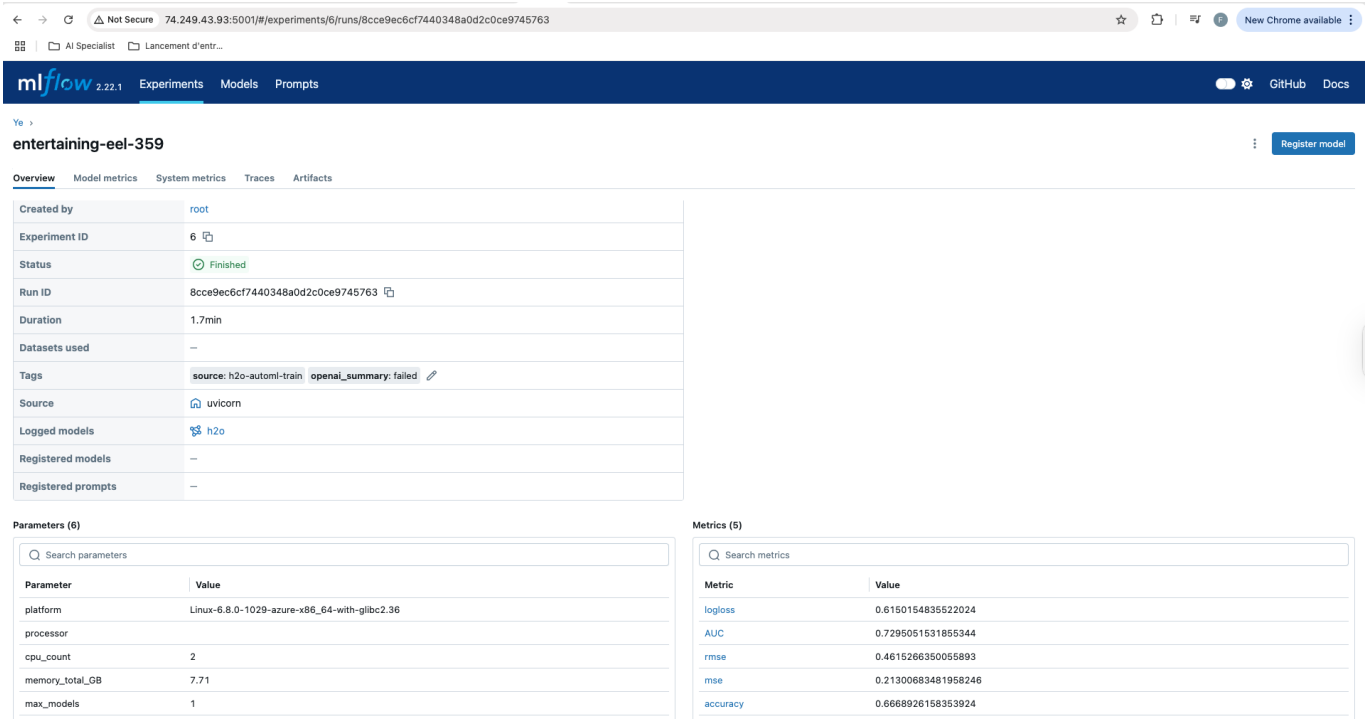
```
docker ps
```

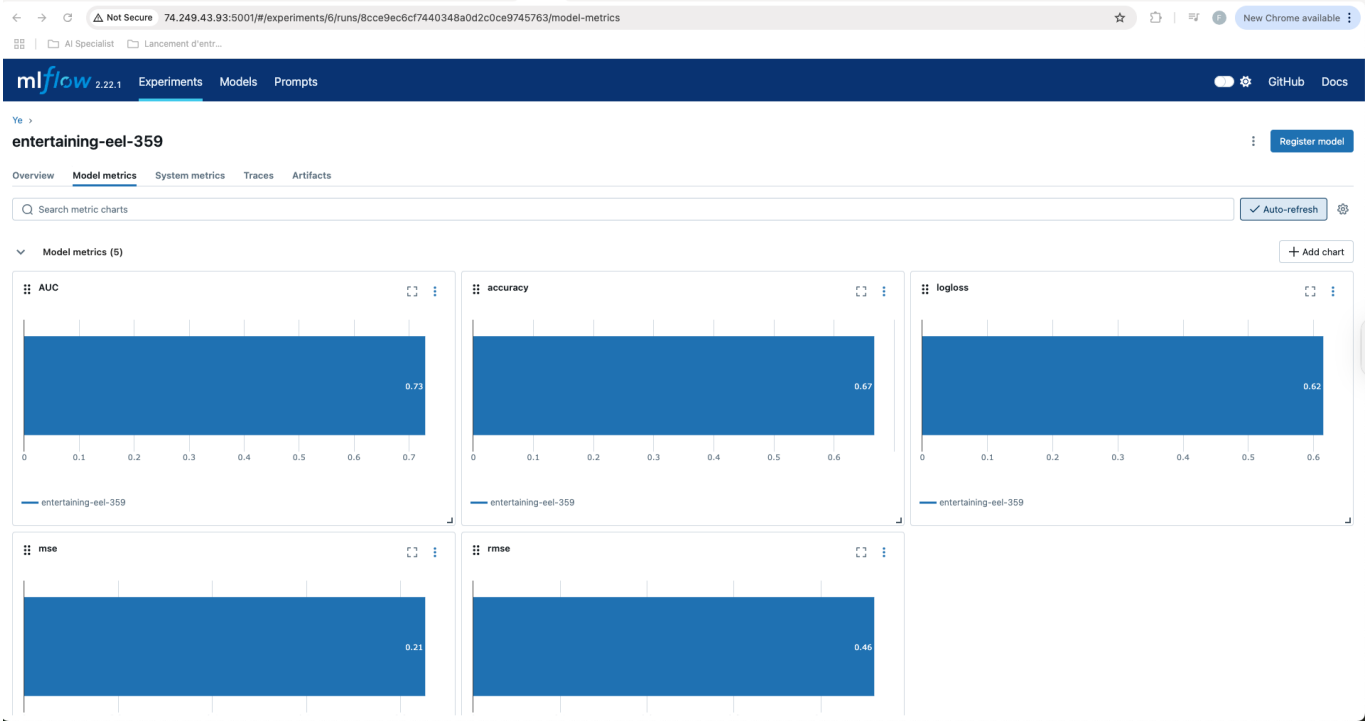
Suivi des expériences avec MLflow

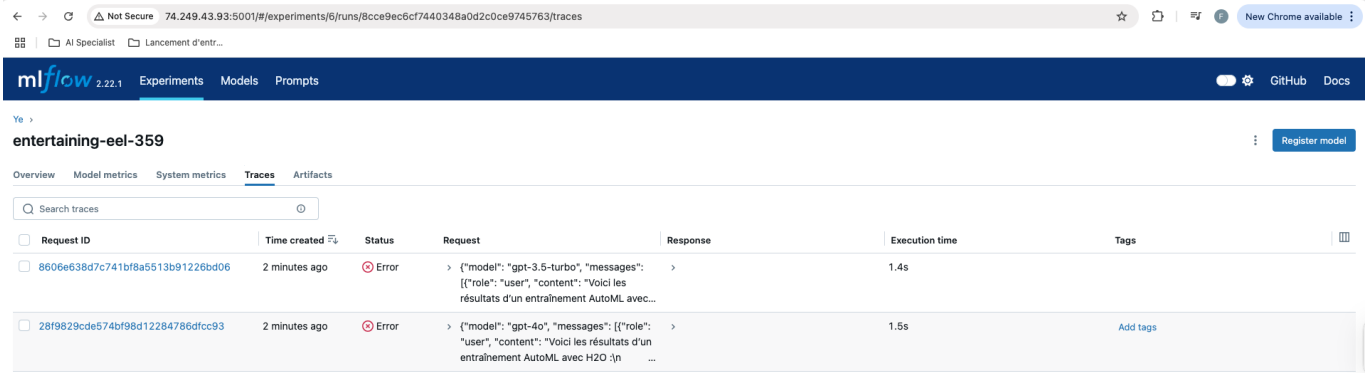
MLflow est utilisé pour enregistrer, comparer et gérer les différentes expériences de machine learning, assurant la reproductibilité et la traçabilité des modèles.

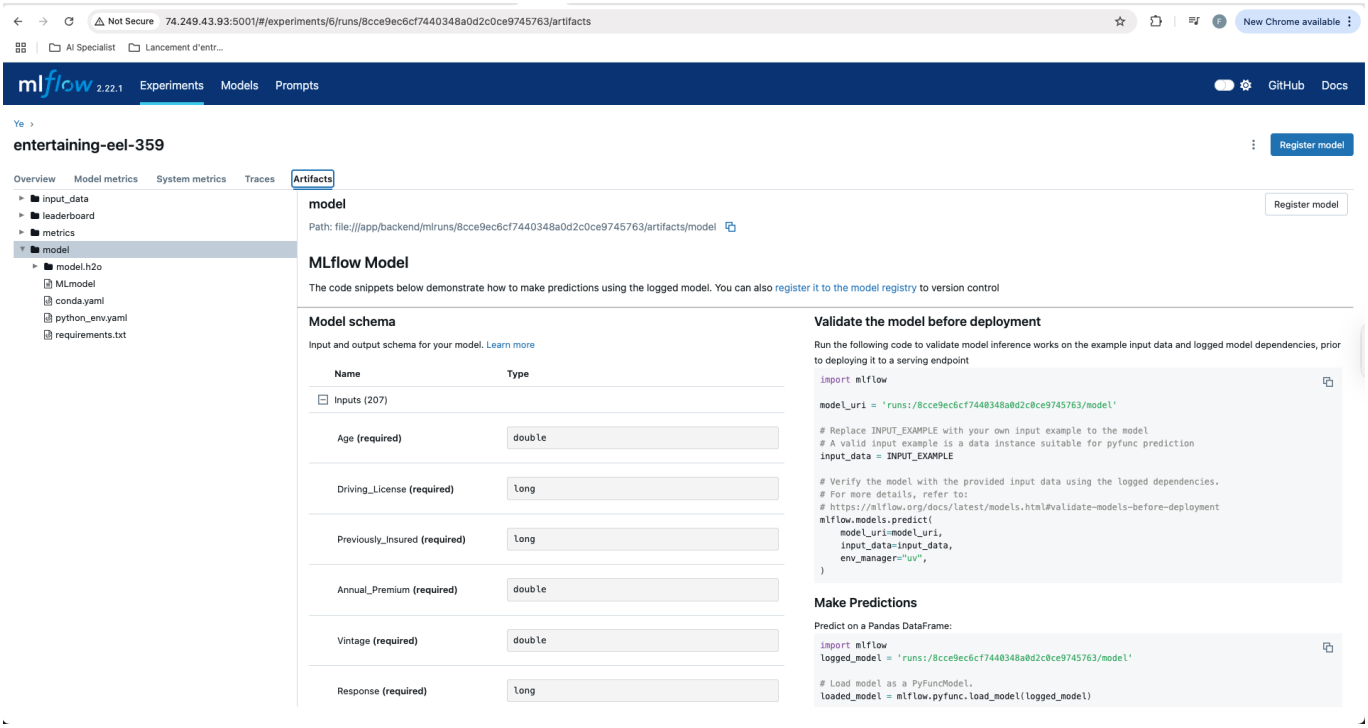

Not Secure 74.249.43.93:5001/#/experiments/0?searchFilter=&orderByKey=attributes.start_time&orderByAsc=false&startTime=ALL&lifecycleFilter=Active&modelVersionFilter=All+Runs&datasetsFilter=W10%3D
New Chrome available
AI Specialist
Lancement d'entr...
mlflow 2.22.1
Experiments
Models
Prompts
GitHub
Docs
Experiments
Search experiments
Default
Default
Provide Feedback
Add Description
Share
Runs
Evaluation
Experimental
Traces
metrics.rmse < 1 and params.model = "tree"
Time created
State: Active
Datasets
Sort: Created
Columns
Group by
New run
Run Name
Created
Dataset
Duration
Source
Models
No runs logged
No runs have been logged yet. Learn more about how to create ML model training runs in this experiment.
0 matching runs


Not Secure 74.249.43.93:5001/#/experiments/6?searchFilter=&orderByKey=attributes.start_time&orderByAsc=false&startTime=ALL&lifecycleFilter=Active&modelVersionFilter=All+Runs&datasetsFilter=W10%3D
New Chrome available
AI Specialist
Lancement d'entr...
mlflow 2.22.1
Experiments
Models
Prompts
GitHub
Docs
Experiments
Search experiments
Default
Test
Gender_1
Ye
Ye
Provide Feedback
Add Description
Share
Runs
Evaluation
Experimental
Traces
metrics.rmse < 1 and params.model = "tree"
Time created
State: Active
Datasets
Sort: Created
Columns
Group by
New run
Run Name
Created
Dataset
Duration
Source
Models
entertaining-eel-359
2 minutes ago
-
1.7min
uvicorn
h2o
judicious-cow-534
27 minutes ago
-
1.5min
uvicorn
h2o
resilient-yak-155
44 minutes ago
-
1.7min
uvicorn
h2o
3 matching runs


Not Secure 74.249.43.93:5001/#/experiments/6/runs/8cce9ec6cf7440348a0d2c0ce9745763
New Chrome available
mlflow 2.22.1
Experiments
Models
Prompts
GitHub
Docs
entertaining-eel-359
Register model
Overview
Model metrics
System metrics
Traces
Artifacts
Created by root
Experiment ID 6
Status Finished
Run ID 8cce9ec6cf7440348a0d2c0ce9745763
Duration 1.7min
Datasets used –
Tags source: h2o-automl-train openai_summary: failed
Source uvicorn
Logged models h2o
Registered models –
Registered prompts –
Parameters (6)
Search parameters
Parameter Value
platform Linux-6.8.0-1029-azure-x86_64-with-glibc2.36
processor
cpu_count 2
memory_total_GB 7.71
max_models 1
Metrics (5)
Search metrics
Metric Value
logloss 0.6150154835522024
AUC 0.7295051531855344
rmse 0.4615266350055893
mse 0.21300683481958246
accuracy 0.6668926158353924

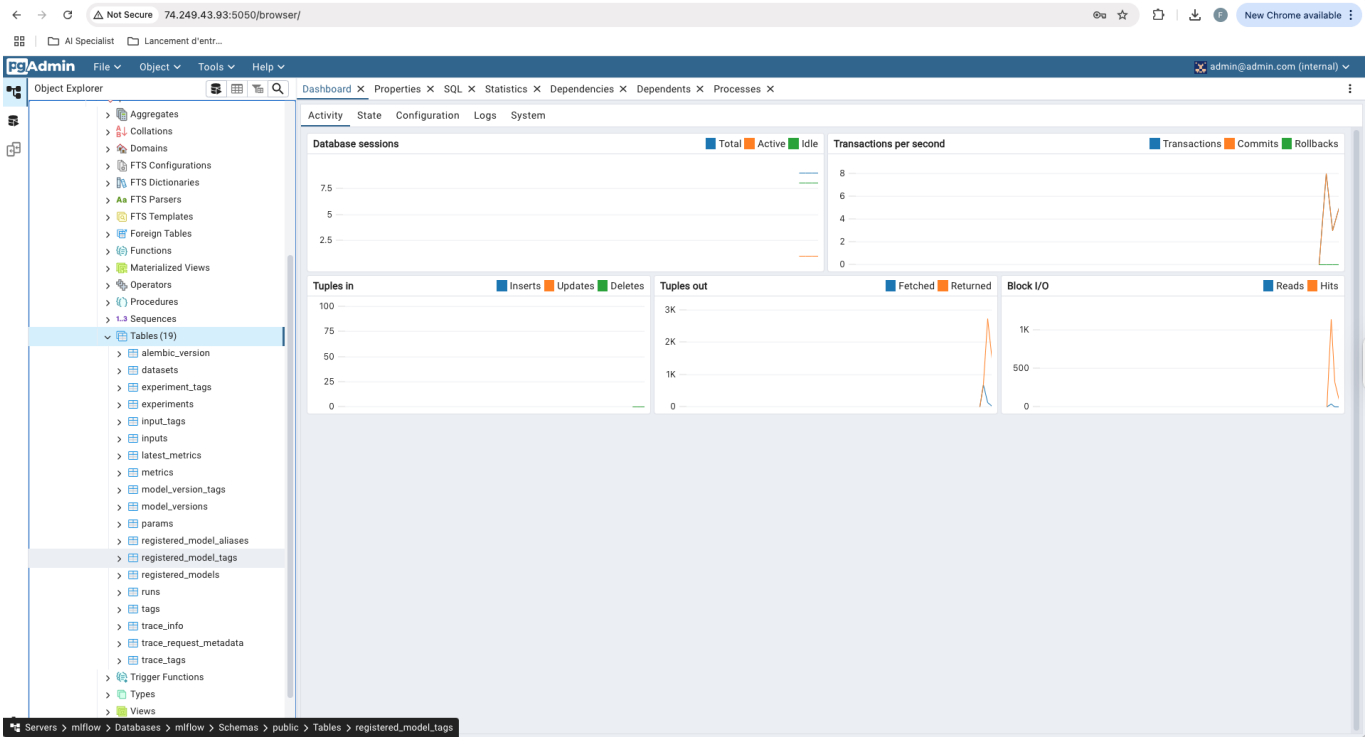

Not Secure 74.249.43.93:5001/#/experiments/6/runs/8cce9ec6cf7440348a0d2c0ce9745763/model-metrics
New Chrome available
mlflow 2.22.1
Experiments
Models
Prompts
GitHub
Docs
entertaining-eel-359
Register model
Overview
Model metrics
System metrics
Traces
Artifacts
Search metric charts
Auto-refresh
Model metrics (5)
Add chart
AUC
0.73
accuracy
0.67
logloss
0.62
mse
0.21
rmse
0.46
entertaining-eel-359





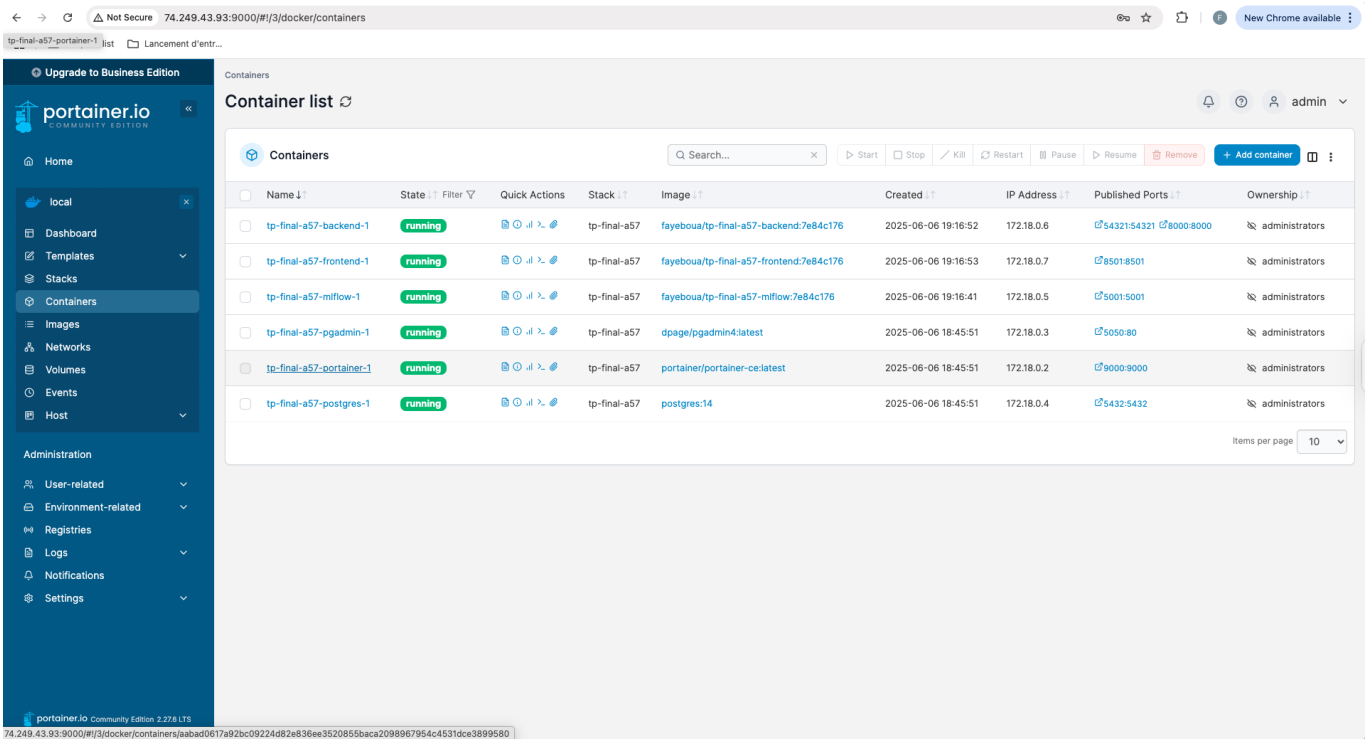
Stockage des données avec Postgres

Postgres sert de base de données centrale pour stocker les données d'entrée, les résultats des prédictions et les métadonnées du projet.



Supervision avec Portainer

Portainer est utilisé pour superviser et gérer les conteneurs Docker déployés sur la VM, facilitant l'administration et la maintenance des services.



Video de présentation

Ci-dessous le lient de la vidéo de présentation. Bon visionnage!!! Video de présentation