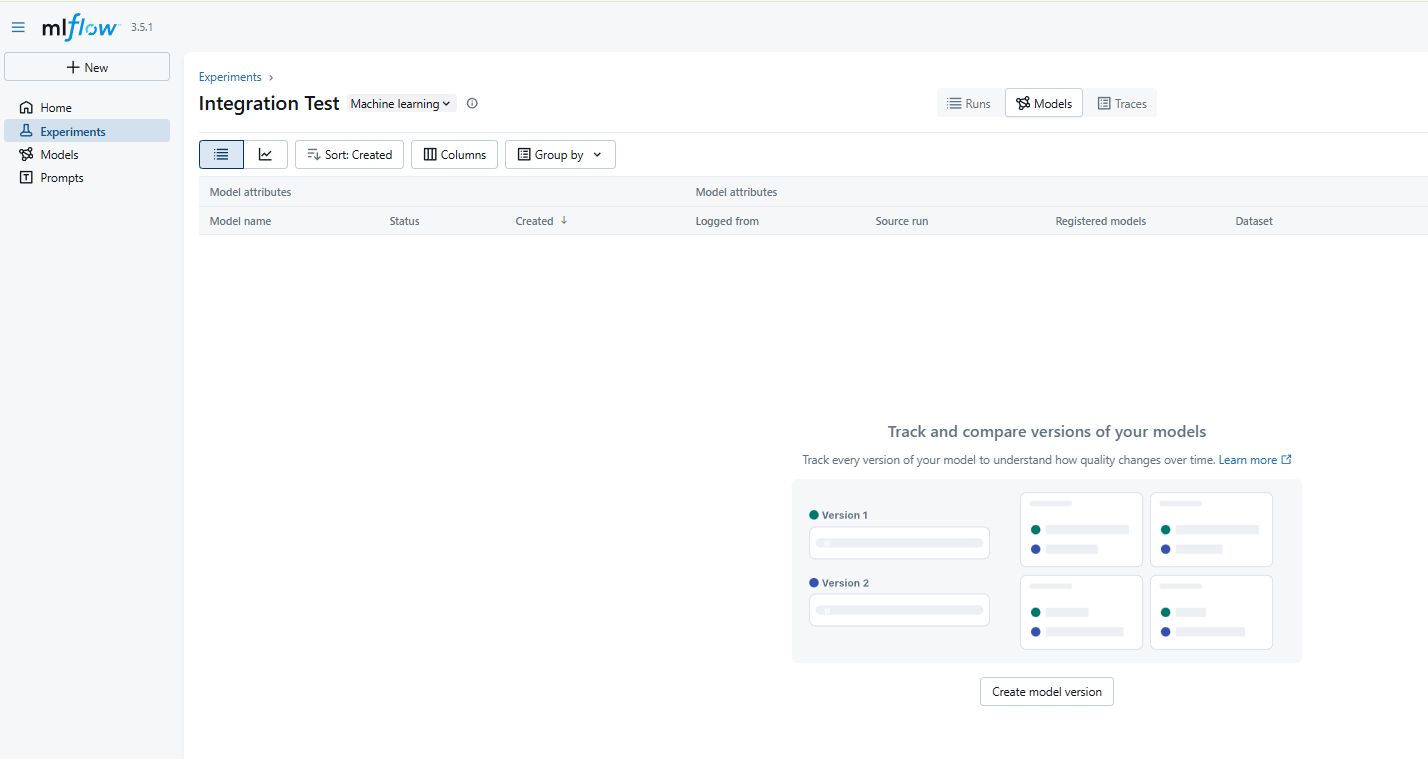
Các services được xây dựng như trong repo. Ý tưởng:

Airflow sẽ làm nhiệm vụ điều khiển hệ thống hoạt động tự động. Nhận vào DAG file là 1 file python mục đích training model. Airflow cũng có các task trong DAG, sẽ có task để đánh giá mô hình, quyết định nó là mô hình tốt mới chuyển sang MLFlow để lưu trữ (local, S3, ..)

Sau khi chạy sẽ tạo ra file model và chuyển đến MLFlow, MLFlow có nhiệm vụ lưu trữ và giữ nhật ký để quản lý version. Bên cạnh đó cũng lưu xuống postgeSQL.



Service jupyter dung để thử nghiệm cách hoạt động ghi lên MLFlow (sẽ thay thế bởi luồng tự động từ airflow)

Model hiện đang chia sẻ nhau qua cách mount vào thư mục chung trên host.

Sau khi có model mới thì API inference sẽ sử dụng nó.

Sau khi các services hoạt động.

Thử test kết nối chúng lại với nhau:

**1. Xác Nhận Luồng Dữ liệu (Data Flow)**

Mục tiêu là chứng minh mô hình có thể được huấn luyện trong JupyterLab và được lưu trữ thành công trong MLflow.

**Bước 1.1: Kết nối MLflow từ JupyterLab**

1. **Truy cập JupyterLab UI** tại http://localhost:8888.
2. Tạo một notebook mới và chạy đoạn code Python sau để xác nhận JupyterLab có thể thấy MLflow server:

**Hiện tại chỉ work manually với IP:** [**http://IP:5000**](http://IP:5000)**. Cần tìm hiểu và cải tiến**

Python

import mlflow

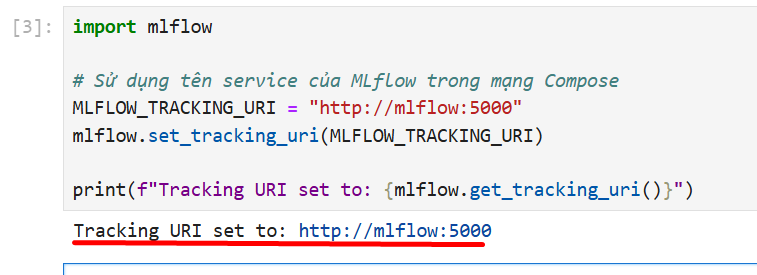
# Sử dụng tên service của MLflow trong mạng Compose

MLFLOW\_TRACKING\_URI = "http://mlflow:5000"

mlflow.set\_tracking\_uri(MLFLOW\_TRACKING\_URI)

print(f"Tracking URI set to: {mlflow.get\_tracking\_uri()}")

Look like:



**Bước 1.2: Huấn luyện & Lưu mô hình Giả lập (Mock Training)**

Trong cùng một notebook, viết code để mô phỏng quá trình huấn luyện và lưu một artifact:

Python

import mlflow

import numpy as np

import pickle

# 1. Tạo mô hình giả (chỉ là một đối tượng Python đơn giản)

class MockModel:

def predict(self, x):

return x \* 2

mock\_model = MockModel()

# 2. Bắt đầu MLflow Run và Log

with mlflow.start\_run(run\_name="mock\_training\_run") as run:

# Log metrics & params

mlflow.log\_param("test\_size", 0.2)

mlflow.log\_metric("mock\_accuracy", 0.99)

# LƯU MÔ HÌNH (artifact)

# Vì đây là mô hình giả, ta dùng pickle, sau đó log nó như một artifact

model\_path = "mock\_model.pkl"

with open(model\_path, "wb") as f:

pickle.dump(mock\_model, f)

mlflow.log\_artifact(model\_path, artifact\_path="model")

print(f"Run ID: {run.info.run\_id}")

print("Model artifact logged successfully.")

**Bước 1.3: Kiểm tra Kết quả trên MLflow UI**

1. Truy cập **MLflow UI** tại http://localhost:5000.
2. Bạn phải thấy một thí nghiệm (Experiment) mới và một Run tên là **mock\_training\_run**.
3. **Quan trọng:** Nhấp vào Run đó, đi đến tab **Artifacts**, và xác nhận có một thư mục model chứa tệp mock\_model.pkl (hoặc tệp mô hình của bạn).

Nếu 3 bước này hoạt động, bạn đã xác nhận được **kết nối giữa JupyterLab -> MLflow -> Postgres/Volumes** là hoàn hảo.

Debug command:

Check quyền folder:

docker exec -it mlflow\_server ls -ld /opt/mlflow /opt/mlflow/artifacts

cap full quyên:

docker exec -it mlflow\_server chmod 777 /opt/mlflow

check network:

podman network inspect mlops-stack\_mlops\_net

access container:  
docker exec -it airflow\_webserver /bin/bash