Docker Compose for Machine Learning & Data Science

WEEK 3 — Experiment Tracking & Model Training

Goal: Set up MLflow and PostgreSQL with Docker Compose for experiment tracking and model logging.

Concepts You’ll Learn:

- Experiment Tracking – Centralized logging of model parameters, metrics, and artifacts.
- MLflow Architecture – Tracking server + backend store + artifact store.
- Service Integration – Jupyter, MLflow, and PostgreSQL communicating via Compose network.
- Artifact Persistence – Storing trained models and experiment results.
- Environment Variables & Ports – Configure service URIs and connection details.

Project Structure:

```
docker-mlflow-lab/ ■ ■■■ Dockerfile ■■■ docker-compose.yml ■■■ notebooks/ ■■■
artifacts/
```

Dockerfile:

```
FROM python:3.11-slim WORKDIR /app RUN pip install --no-cache-dir numpy pandas
scikit-learn mlflow psycopg2-binary matplotlib seaborn jupyterlab EXPOSE 8888 CMD
["jupyter", "lab", "--ip=0.0.0.0", "--allow-root", "--no-browser"]
```

docker-compose.yml:

```
version: "3.9" services: jupyterlab: build: . container_name: mlflow_lab ports: -
"8888:8888" volumes: - ./notebooks:/app/notebooks environment: -
MLFLOW_TRACKING_URI=http://mlflow:5000 - DB_HOST=db - DB_PORT=5432 -
DB_NAME=mlflowdb - DB_USER=ml_user - DB_PASSWORD=ml_pass depends_on: - mlflow
restart: unless-stopped mlflow: image: ghcr.io/mlflow/mlflow container_name:
mlflow_server command: > mlflow server --backend-store-uri
postgresql://ml_user:ml_pass@db/mlflowdb --default-artifact-root /artifacts --host
0.0.0.0 --port 5000 ports: - "5000:5000" volumes: - ./artifacts:/artifacts
depends_on: - db restart: unless-stopped db: image: postgres:15 container_name:
mlflow_db environment: - POSTGRES_DB=mlflowdb - POSTGRES_USER=ml_user -
POSTGRES_PASSWORD=ml_pass volumes: - pgdata:/var/lib/postgresql/data healthcheck:
test: ["CMD-SHELL", "pg_isready -U ml_user -d mlflowdb"] interval: 5s retries: 5
restart: unless-stopped volumes: pgdata:
```

Test MLflow Logging (Notebook Code):

```
import mlflow, mlflow.sklearn from sklearn.datasets import load_diabetes from
sklearn.linear_model import LinearRegression from sklearn.metrics import
mean_squared_error import os
mlflow.set_tracking_uri(os.getenv("MLFLOW_TRACKING_URI"))
mlflow.set_experiment("docker_compose_experiment") X, y =
load_diabetes(return_X_y=True) model = LinearRegression() model.fit(X, y) preds =
model.predict(X) mse = mean_squared_error(y, preds) with mlflow.start_run():
mlflow.log_param("model_type", "LinearRegression") mlflow.log_metric("mse", mse)
mlflow.sklearn.log_model(model, "model") print("Run logged successfully!")
```

Mini Project: Run MLflow with PostgreSQL and JupyterLab using Docker Compose, log experiments and models, and verify persistence.

Skill	You Should Be Able To…	Verified
Run 3-service Compose stack	Jupyter + MLflow + DB running	■
Access MLflow UI	http://localhost:5000 shows dashboard	■
Log experiment from Jupyter	Run appears in MLflow UI	■
Persist artifacts & metadata	Data survives restart	■
Modify MLflow URI/DB config	Use environment variables effectively	■