Docker Compose for Machine Learning & Data Science

WEEK 2 — Multi-Container ML Workflows

Goal: Connect multiple containers using Docker Compose to build a basic ML pipeline.

Concepts You’ll Learn:

- Multi-Service Architecture – Running multiple containers in one project.
- Networking – Automatic internal DNS between services.
- Environment Variables – Securely passing credentials/configs.
- Data Persistence – Using named volumes to save database state.
- Dependency Control – Using depends_on and healthcheck.

Project Structure:

```
docker-ml-pipeline/ ■ ■■■ Dockerfile ■■■ docker-compose.yml ■■■ notebooks/ ■■■
data/
```

Dockerfile:

```
FROM python:3.11-slim WORKDIR /app RUN pip install --no-cache-dir numpy pandas
scikit-learn psycopg2-binary SQLAlchemy 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: ml_lab ports: -
"8888:8888" volumes: - ./notebooks:/app/notebooks environment: -
JUPYTER_ENABLE_LAB=yes - DB_HOST=db - DB_PORT=5432 - DB_NAME=ml_data -
DB_USER=ml_user - DB_PASSWORD=ml_pass depends_on: - db restart: unless-stopped db:
image: postgres:15 container_name: ml_db restart: unless-stopped environment: -
POSTGRES_DB=ml_data - POSTGRES_USER=ml_user - POSTGRES_PASSWORD=ml_pass volumes: -
pgdata:/var/lib/postgresql/data healthcheck: test: ["CMD-SHELL", "pg_isready -U
ml_user -d ml_data"] interval: 5s retries: 5 volumes: pgdata:
```

Test Database Connection (Notebook Code):

```
import os, psycopg2, pandas as pd conn = psycopg2.connect(
host=os.getenv("DB_HOST", "db"), database=os.getenv("DB_NAME", "ml_data"),
user=os.getenv("DB_USER", "ml_user"), password=os.getenv("DB_PASSWORD", "ml_pass")
) cur = conn.cursor() cur.execute("CREATE TABLE IF NOT EXISTS experiments(id SERIAL
PRIMARY KEY, name TEXT, accuracy FLOAT);") cur.execute("INSERT INTO
experiments(name, accuracy) VALUES ('model_v1', 0.93);") conn.commit() df =
pd.read_sql("SELECT * FROM experiments;", conn) print(df) cur.close();
conn.close()
```

Mini Project: Build a two-container ML pipeline with JupyterLab and PostgreSQL. Persist data across restarts.

Skill	You Should Be Able To…	Verified
Define multiple services in Compose	Run Jupyter + DB together	■
Connect containers via network	Access DB using hostname 'db'	■
Persist data using volumes	Data survives restarts	■
Use environment variables	Securely store DB credentials	■
Run health checks and depends_on	Ensure correct startup order	■