02/20/25 - Docker Compose and Neo4j

Neo4j

Query Language and Plugins

Cypher

Neo4j's graph query language created in 2011

Goal: SQL-equivalent language for graph databases

Provides a visual way of matching patterns and relationships

`(nodes)-[:CONNECT_TO]→(otherNodes)`

APOC Plugin

Awesome Procedures on Cypher

Add-on library that provides hundreds of procedures and functions

Graph Data Science Plugin

Provides efficient implementations of common graph algorithms

Docker Compose

Supports multi-container management

Set-up is declarative—using YAML docker-compose.yaml file

- Services

- Volumes

- Networks, etc.

One command can be used to start, stop, or scale a number of services at one time

Provides a consistent method for producing an identical environment

Interaction is mostly via command line

docker-compose.yaml

```yaml
services:
  neo4j:
    container_name: neo4j
    image: neo4j:latest
    ports:
      - 7474:7474
      - 7687:7687
    environment:
      - NEO4J_AUTH=neo4j/${NEO4J_PASSWORD}
      - NEO4J_apoc_export_file_enabled=true
      - NEO4J_apoc_import_file_enabled=true
      - NEO4J_apoc_import_file_use__neo4j__config=true
      - NEO4J_PLUGINS=["apoc", "graph-data-science"]
    volumes:
      - ./neo4j_db/data:/data
      - ./neo4j_db/logs:/logs
      - ./neo4j_db/import:/var/lib/neo4j/import
      - ./neo4j_db/plugins:/plugins
```

Never put "secrets" in a docker compose file. Use .env files.

Service: piece of software that runs on a computer at all times

- Runs in the background

Ports

- Number before the colon: host port (on this machine)

- Number after the colon: port in the container

- Largest (standard) port number: 65,535

  - Power of two

- Ports reserved for systems: 0 to 1023

Volumes

- Before the colon: local folders on your laptop that are created by Docker (if they don't already exist)

.env Files

Put in gitignore

Stores a collection of environment variables

Good way to keep environment variables for different platforms separate

- .env.local

- .env.dev

- .env.prod

Docker Compose Commands

To test if you have Docker CLI properly installed, run: `docker --version`

Major Docker Commands

- `docker compose up`
    - Must be executed in the same folder as the .yaml file
    - Spins everything up (containers don't exist yet)

- `docker compose up -d`
    - Detached mode, not attached to lifespan of the stuff

- `docker compose down`
    - Turn it off and delete

- `docker compose start`
    - Turn it back on (containers exist already)

- `docker compose stop`
    - Stop

- `docker compose build`

- `docker compose build --no-cache`