Neo4j

Created	@February 20, 2025 9:36 AM
Class	DS 4300
Reviewed	□

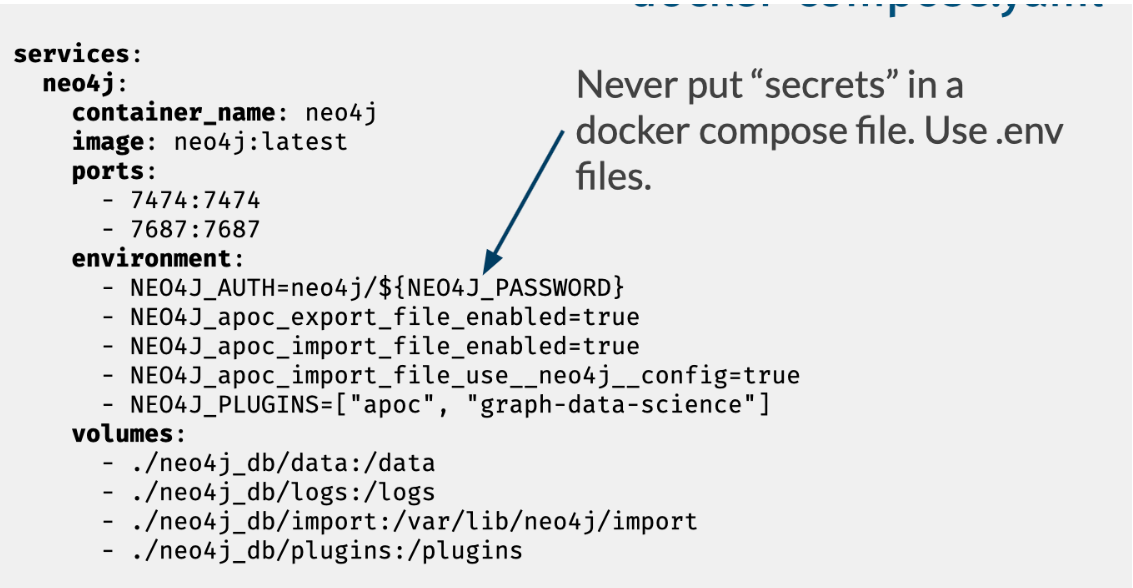
Neo4j

- 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 hundred of procedures and functions
- Graph Data Science Plugin
  - provides efficient implementation of common graph algorithms

Docker Compose & Neo4j

- Docker Compose
  - Supports multi-container management
  - Set up is declarative - using YAML docker-compose.yaml file
    - services
    - volumes
    - networks, etc.

- 1 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
- EX: docker-compose.yaml

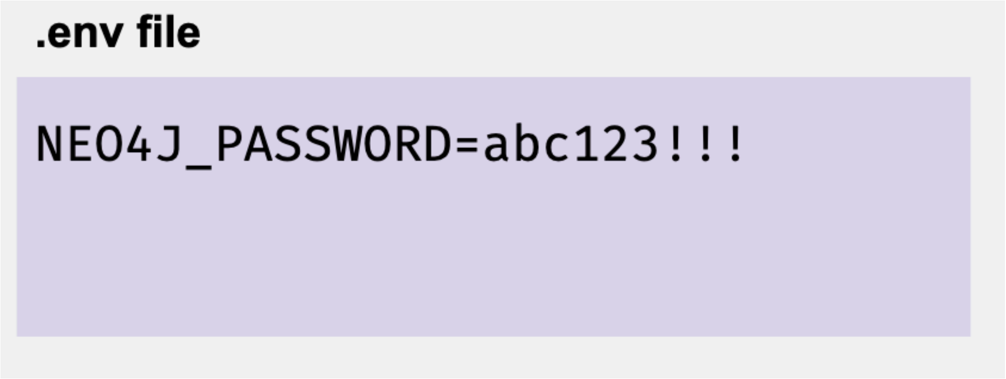


```
services:
  neo4j:
    container_name: neo4j
    image: neo4j:latest
    ports:
      - 7474:7474
      - 7687:7687
    environment:
      - NEO4J_AUTH=neo4j/${NEO4J_PASSWORD}
      - NEO4J_apoc_export_file_enabled=true
      - NEO4J_apoc_import_file_enabled=true
      - NEO4J_apoc_import_file_use__neo4j__config=true
      - 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.

  - Docker Compose sets up multiple containers within a virtual private network that cannot communicate with the outside unless the right port is punched through. The containers within can communicate with each other
  - First neo4j is just name, container_name is just naming, image is the image to use to create the container
  - Ports represent port mapping, number before : is local port on your laptop, after : is the port in the container
    - Max is 65535
    - Reserved ports are ...
  - environment is the packages and other services to install in the container

    - volumes essentially are a storage drive that is stored within the container
    - Service represents a piece of software that runs on computer at all times (basically application that runs constantly on laptop)
  - EX: .env file



.env file

```
NEO4J_PASSWORD=abc123!!!
```

    - Put .env in .gitignore initially otherwise its a pain
    - put password in the .env file
  - Docker commands
    - docker compose up: executed in same folder as yaml
      - spins up all services in compose file
      - first time run use this
    - docker compose up -d:
      - spins up in detached mode (so not just this session)
    - docker compose down:
      - turn off and delete
    - docker compose start
      - start is turning them back on
    - docker compose stop
      - stops the container

- docker compose build
  - builds the container
- docker compose build —no-cache
  - redownloads everything and rebuilds