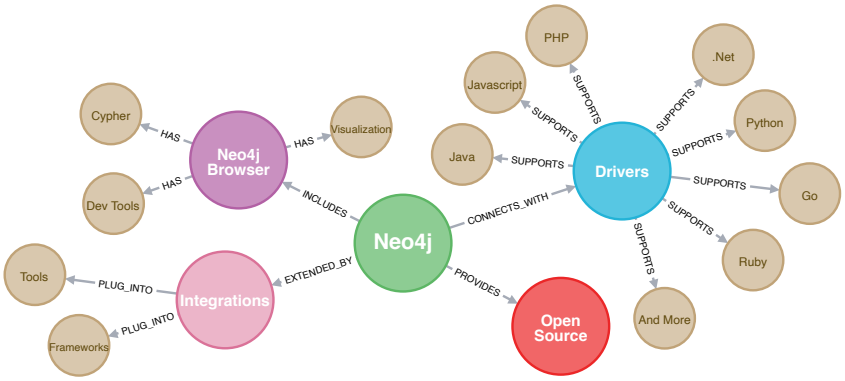


Neo4j Setup Day!

Set up Neo4j on your local or EC2 machine

A. Install the software
B. Set up a configuration/data/logs directory tree
C. Edit the neo4j.conf file
D. Start neo4j, the Neo4j server
E. Interact with it using the Neo4j Browser

A. Install Neo4j Server

- The website makes Neo4j Desktop the most prominent download but we prefer to work with the server directly, so what we want for the class is Neo4j Server:

https://neo4j.com/download-center/#community

(Neo4j Desktop or their sandbox option works well for learning/experimentation, so you aren't disallowed from downloading or using these—it's just that running the server is the exposure that the course hopes to provide)

- Many package managers also have this available (e.g., apt, brew) so that option may apply to you

- Installing an an AWS EC2 host is nearly the same process—just all done remotely, and you would also need to open up certain ports for access from outside (specifically, ports 7474, 7473, and 7687—and possibly port 22 for ssh if not open already)
- Specific Neo4j tips are given here:

https://neo4j.com/docs/operations-manual/current/cloud-deployments/neo4j-aws/

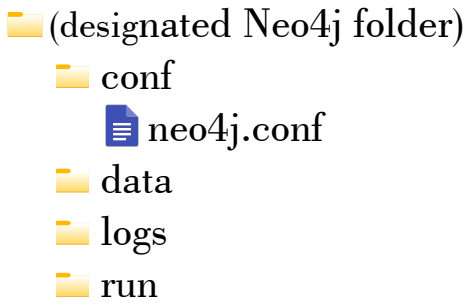
- Further, Neo4j provides a specific AMI (Amazon Machine Image) that already has it installed:

https://aws.amazon.com/marketplace/pp/B071P26C9D

B. Create...

- Neo4j typically installs with defaults all ready to go, but they tend to be meant for system-level use (you can see these defaults by running `neo4j console`)
- In-keeping with our theme of being more aware of where things are, we'll forego these defaults—it takes some extra work but it's worth having the extra control
- Learning how to do this will make it easier to wipe things out and just start over, if needed (in addition to knowing clearly where Neo4j is reading/writing things)

1. Decide where you want Neo4j's files to live and create a directory/folder at this location
2. Inside that directory/folder, create more folders: conf, data, logs, run
3. There should be a file called neo4j.conf somewhere in your installation—find it and copy it into conf
4. After these steps, your structure should look like this:



C. ...and Configure

1. Open your copy of neo4j.conf and edit these settings—put these at a bottom so that they override earlier values, if any:

```
dbms.directories.data=<absolute path to data>
dbms.directories.logs=<absolute path to logs>
dbms.directories.run=<absolute path to run>
```

2. Note that these settings must be absolute paths: start with "/" on macOS and Linux; start with the drive letter on Windows

3. The data, log, and run directories will be initialized by Neo4j as needed, so they can start out empty—if you want to completely start over, you can empty out their contents as well

4. Being comfortable with creating these directories and editing neo4j.conf comes in handy when using the high-performance import tool (more on that in the case study)—one of this utility's requirements is that it must run on a previously unused database (exact words from the documentation!)…in other words, a new one

5. Knowing how to do this also allows you to create multiple storage locations, each independent of the other, in case you want to experiment or explore

D. Run Neo4j, Run

- With a non-default configuration like this, neo4j will need to be told where to look when it runs—lo and behold, we tell it via environment variable:

```
NEO4J_CONF=<absolute path to conf> neo4j console
```

- Here, `conf` is the directory, not the neo4j.conf file
- Once more, remember to use an absolute path
- You should be good to go if the last message you see on the console is `Started`

- To see more detailed debug messages, run this in another window (this path can be relative):

```
tail -f <path to logs>/debug.log
```

- To stop the server, hit `Control-C`
- If you want to start over in terms of data but don't need a whole new directory structure, you can just stop the Neo4j server and wipe out everything in the data folder
  - The next server start will re-initialize the database
  - …including the `neo4j` user's password
- Bulk import is best done while the server is off—and remember again, it requires a new data folder

E. Neo4j Browser Time

- Neo4j comes with both text (`cypher-shell`) and visual (Neo4j Browser) clients out of the box—due to the nature of graph databases, the visual client generally works better unless your queries return humongous graphs (more on resource usage later)

- Hit up Neo4j Browser at `http://localhost:7474`
- For a brand-new, just-created database, the initial username and password are: `neo4j`/`neo4j`—you will be asked to change this upon first login

- Once you're in, Neo4j Browser has a command-line like feel: you enter your next command at the top, then run it by clicking the ▶️ button on the right—the results are then displayed and you are prompted for your next command; rinse and repeat
- Like clients for other systems, you can issue a mix of Cypher queries (`MATCH`, `CREATE`, etc.) or system/utility commands—see the side menu for examples and help
- The Neo4j website's "Getting Started with Cypher" provides a good introductory walkthrough:

https://neo4j.com/developer/cypher/intro-cypher/

If You Have Room...

- You will notice—especially with large amounts of data and graphs—that Neo4j is a bit…demanding with respect to computing resources
- By default, Neo4j starts at 512 megabytes of memory—it can grow this dynamically but generally does better if you give it more than that from the get-go
- Neo4j is written in Java, so its memory management behavior and settings correspond to Java's

- Appending the following settings to your neo4j.conf file to whatever amount your machine can afford may help with performance:

```
dbms.memory.heap.initial_size=<custom amount>
dbms.memory.heap.max_size=<custom amount>
```

- Ideally, set these to the same amount—whatever you can afford to spare above 512 megabytes
- The amounts should be formatted as numbers followed by units (i.e., `m` for megabytes, `g` for gigabytes, etc.):

```
dbms.memory.heap.initial_size=4g
dbms.memory.heap.max_size=4g
```