

Contents

Graph Database Helper Lab.....	2
Step 1. VM Networking.....	2
Step 2a. Installing Neo4j.....	4
Step 3. Configuring and Connecting to Neo4j.....	7
Step 4. (Optional) Installing Neo4j with a different version of Java	10
Step 5 (Optional): Common Errors and Fixes	13

Graph Database Helper Lab

The following is a helper lab for the graph database neo4j. The lab assumes you have a recent Ubuntu LTS VM installed from prior class work. As we have discussed, neo4j is a graph-based data that focuses on relationships in contrast to entities or tables. By prioritizing relationships first, it can perform better than traditional transactional databases when the number of joins between tables increases.

Neo4j documentation is well done and the basis for the installation as such is from the neo4j documentation. The credited/referenced tutorial lab is:

<https://neo4j.com/docs/operations-manual/current/installation/linux/debian/#debian-installation>

We also use our textbook for aspects of this lab.

Class Textbook: <https://github.com/PacktPublishing/Big-Data-Architects-Handbook>

Step 1. VM Networking

The first step is to make sure you enable NAT or bridged mode on your Ubuntu VM. It is necessary for the VM to have access to the Internet. If you cannot ping www.google.com from your Ubuntu CLI, you need to change your NIC settings.

You may need to change your Adapter 1 to Bridged mode or NAT mode or you may also be able to have two NICs enabled with one in host-only mode and another in NAT or bridged mode. If two NICs do not work, simply change your Adapter 1 to NAT or bridged mode. See this example:

- General
- System
- Display
- Storage
- Audio
- Network**
- Serial Ports
- USB
- Shared Folders
- User Interface

Network

Adapter 1 Adapter 2 Adapter 3 Adapter 4

☒ Enable Network Adapter

Attached to: NAT

Name:

▶ Advanced

OK

Cancel

Step 2a. Installing Neo4j

Current neo4j documentation suggests the following Java versions:

Neo4j Version JVM compliancy

3.x	Java SE 8 Platform Specification
4.x	Java SE 11 Platform Specification
5.x	Java SE 17 Platform Specification

See: <https://neo4j.com/docs/operations-manual/current/installation/requirements/>

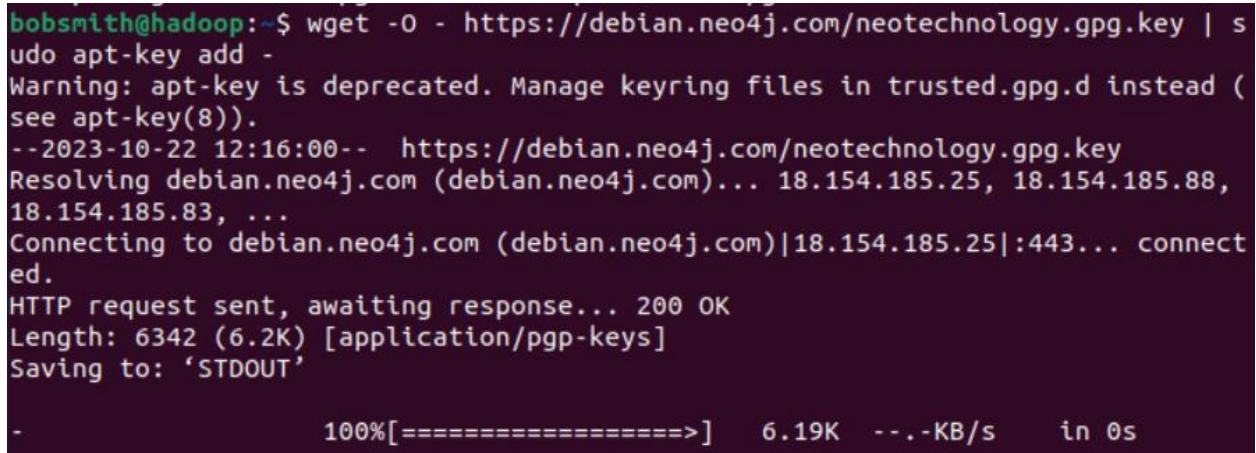
It is encouraged that you take a snapshot of your VM at this time. Anytime you install new software that could impact compatibility, creating a backup is essential!

Next, we are going to use a version of Neo4j that is compatible with our version of Java. In this example, we are using Neo4j 3.x, which is compatible with Java 8. Use the version that is compatible with your current version of Java on the VM it is being installed.

First, add the key for neo4j:

```
wget -O - https://debian.neo4j.com/neotechnology.gpg.key | sudo apt-key add -
```

It should download a GPG key to ensure an authentic download, here is a screenshot:



```
bobsmith@hadoop:~$ wget -O - https://debian.neo4j.com/neotechnology.gpg.key | s
udo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (
see apt-key(8)).
--2023-10-22 12:16:00-- https://debian.neo4j.com/neotechnology.gpg.key
Resolving debian.neo4j.com (debian.neo4j.com)... 18.154.185.25, 18.154.185.88,
18.154.185.83, ...
Connecting to debian.neo4j.com (debian.neo4j.com)|18.154.185.25|:443... connect
ed.
HTTP request sent, awaiting response... 200 OK
Length: 6342 (6.2K) [application/pgp-keys]
Saving to: 'STDOUT'

-                  100%[=====>]   6.19K  --.-KB/s   in 0s
```

Next, add the correct version of Neo4j to your apt repository and apt sources list:

```
echo 'deb https://debian.neo4j.com stable 3.5' | sudo tee /etc/apt/sources.list.d/neo4j.list
```

Check your sources list to ensure the right version will install:

```
sudo tee /etc/apt/sources.list.d/neo4j.list
```

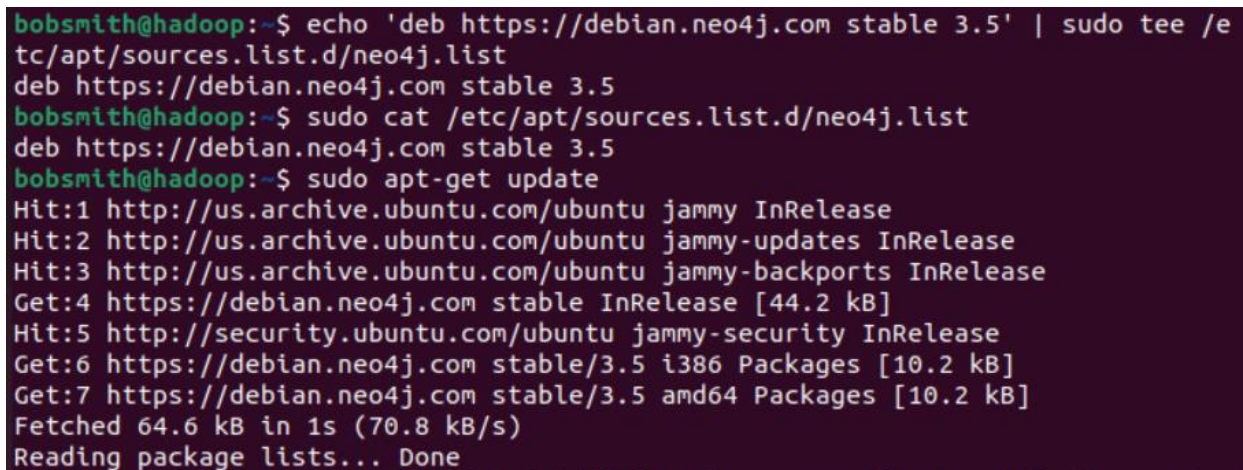
In Ubuntu, neo4j requires the universe rep, run:

```
sudo add-apt-repository universe
```

Then, update your package repository when the version is correct!

```
sudo apt-get update
```

Here is an example screenshot:



```
bobsmith@hadoop:~$ echo 'deb https://debian.neo4j.com stable 3.5' | sudo tee /e
tc/apt/sources.list.d/neo4j.list
deb https://debian.neo4j.com stable 3.5
bobsmith@hadoop:~$ sudo cat /etc/apt/sources.list.d/neo4j.list
deb https://debian.neo4j.com stable 3.5
bobsmith@hadoop:~$ sudo apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 https://debian.neo4j.com stable InRelease [44.2 kB]
Hit:5 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:6 https://debian.neo4j.com stable/3.5 i386 Packages [10.2 kB]
Get:7 https://debian.neo4j.com stable/3.5 amd64 Packages [10.2 kB]
Fetched 64.6 kB in 1s (70.8 kB/s)
Reading package lists... Done
```

Next, find the most recent version of Neo4j to install:

```
apt list -a neo4j | more
```

Use the most recent version, as an example of version 3.5:

```
sudo apt-get install neo4j=1:3.5.35
```

When it asks if you want to continue enter “Y” and enter

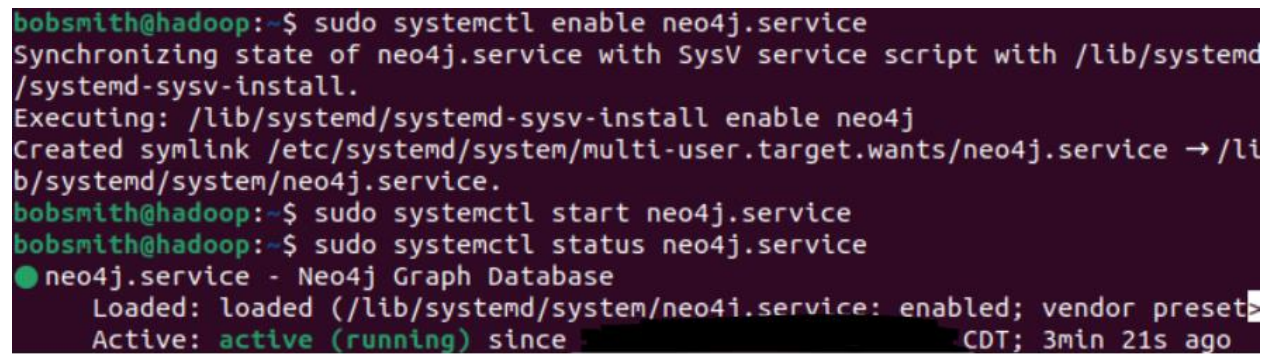
Once it is installed, enable its service and try to start it with the commands:

```
sudo systemctl enable neo4j.service
```

```
sudo systemctl start neo4j.service
```

```
sudo systemctl status neo4j.service
```

Here is an example screenshot:



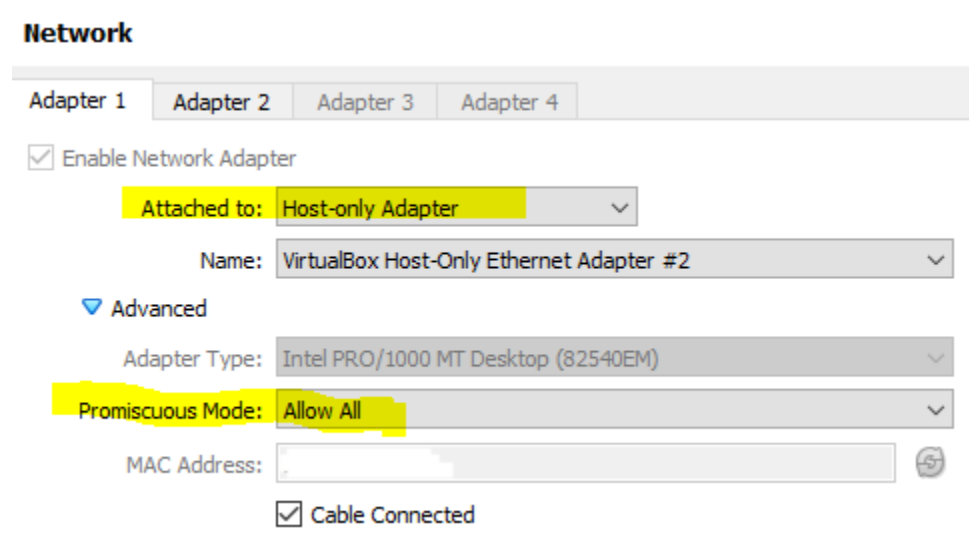
```
bobsmith@hadoop:~$ sudo systemctl enable neo4j.service
Synchronizing state of neo4j.service with SysV service script with /lib/systemd
/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable neo4j
Created symlink /etc/systemd/system/multi-user.target.wants/neo4j.service → /li
b/systemd/system/neo4j.service.
bobsmith@hadoop:~$ sudo systemctl start neo4j.service
bobsmith@hadoop:~$ sudo systemctl status neo4j.service
● neo4j.service - Neo4j Graph Database
   Loaded: loaded (/lib/systemd/system/neo4i.service: enabled; vendor preset>
   Active: active (running) since      CDT; 3min 21s ago
```

Next, update your locate search database so that you can find the latest Neo4j files:
sudo updatedb

Locate your neo4j.conf file:
locate neo4j.conf

Step 3. Configuring and Connecting to Neo4j

Before you can connect to Neo4j, it is important that your VM have the proper networking settings. In Oracle VirtualBox, you can change back to host only mode. Here is an example screenshot:



Once you have neo4j running, you can use the web browser or commander for additional functionality beyond cypher shell. Note, your website ports may be different as in the three installations I have they vary. Also, note the newest version can use bolt and you may not need to install a separate web service. To see the port or change it edit your neo4j.conf file:
`sudo nano /etc/neo4j/neo4j.conf`

Look for these settings:

```
# Bolt connector
dbms.connector.bolt.enabled=true
#dbms.connector.bolt.tls_level=DISABLED
#dbms.connector.bolt.listen_address=:7687
#dbms.connector.bolt.advertised_address=:7687
# HTTP Connector. There can be zero or one HTTP connectors.
dbms.connector.http.enabled=true
#dbms.connector.http.listen_address=:7474
#dbms.connector.http.advertised_address=:7474
```

In the most recent version ports 7687 and 7474 are used for management interfaces. You may want to change the default configuration for the default listening address to allow remote connections. Find and change the settings here:

```
# With default configuration Neo4j only accepts local connections.
# To accept non-local connections, uncomment this line:
server.default_listen_address=0.0.0.0
```

Note, in other versions the setting may be different. For example, in version 3.5 it is:
dbms.connectors.default_listen_address=0.0.0.0

Here is an example screenshot in version 3.5 under the network settings part of the neo4j.conf file:

```
bobsmith@hadoop: /etc/neo4j
GNU nano 6.2 /etc/neo4j/neo4j.conf *
#*****
# Network connector configuration
#*****

# With default configuration Neo4j only accepts local connections.
# To accept non-local connections, uncomment this line:
dbms.connectors.default_listen_address=0.0.0.0
```

If necessary, change port 7474 to whatever port your install is listening, save and exit the configuration file.

If/when you change configuration settings please remember to restart the associated service:

```
sudo systemctl restart neo4j.service
```

```
sudo systemctl status neo4j.service
```

You can now login and manage Neo4j using your web browser with the ports configured in your neo4j configuration file above:

<http://localhost:7474>

Change “localhost” with your hostname or IP address to remotely connect to Neo4j using your favorite web browser.

From the terminal, you can use cypher-shell:

```
cypher-shell -u neo4j -p neo4j
```

It may ask you to change the default neo4j password upon login.

To see neo4j options enter:

```
neo4j@neo4j> :help
```

To exit neo4j use the :exit command:

```
neo4j@neo4j> :exit
```


Here is an example screenshot:

```
bobsmith@hadoop:/etc/neo4j$ cypher-shell -u neo4j -p neo4j
Connected to Neo4j 3.5.35 at bolt://localhost:7687 as user neo4j.
Type :help for a list of available commands or :exit to exit the shell.
Note that Cypher queries must end with a semicolon.
neo4j> :help

Available commands:
:begin      Open a transaction
:commit     Commit the currently open transaction
:exit       Exit the logger
:help       Show this help message
:history    Print a list of the last commands executed
:param      Set the value of a query parameter
:params     Prints all currently set query parameters and their values
:rollback   Rollback the currently open transaction
```

Congratulations, you have your graph database installed and running!

One of the first things you may want to change in your password:

```
neo4j> CALL dbms.changePassword('put new password here');
```

To exit the CLI at any time use a colon followed by the exit keyword:

```
neo4j> :exit
```

Step 4. (Optional) Installing Neo4j with a different version of Java

Oftentimes, our big data system environment runs on different servers than our database servers. If you plan to install Neo4j on a DataNode in your big data system, it is encouraged you make backup copies of your VMs and/or take snapshots prior to making any changes to Java.

Please take a snapshot of your VM at this time!

If the incorrect version of Java is being referenced, cypher-shell may not run and/or produce an error. To resolve this here are a few steps.

Run:

```
sudo update-java-alternatives --list
```

Update to the correct Java version:

```
sudo update-java-alternatives --jre --set <java17name>
```

For Java 17 an example is:

```
sudo update-java-alternatives --jre --set java-1.17.0-openjdk-amd64
```

You may also need to update .bashrc in your home directory.

```
nano .bashrc
```

Add:

```
# Java 17 for neo4j
```

```
export JAVA_HOME=/usr/lib/jvm/java-1.17.0-openjdk-amd64
```

Remember to source your bashrc file:

```
source .bashrc
```

Once updated, try cypher-shell again.

```
cypher-shell -u neo4j -p neo4j
```

To see neo4j options enter:

```
neo4j@neo4j> :help
```

To exit neo4j use the :exit command:

```
neo4j@neo4j> :exit
```

Following installation, it is recommended you practice using neo4j using the textbook examples.

If the version of Java is incorrect or needs update you can add the OpenJDK versions to your repository:

```
sudo add-apt-repository -y ppa:openjdk-r/ppa
sudo apt-get update
```

Install the correct version given your neo4j version:

```
echo "deb http://httpredir.debian.org/debian stretch-backports main" | sudo tee -a
/etc/apt/sources.list.d/stretch-backports.list
```

Next, update your repository:

```
sudo apt-get update
```

If necessary, install Java:

```
sudo apt install openjdk-17-jre
sudo apt install openjdk-17-jdk
```

To see your Java version, enter the command:

```
java -version
```

If the version is not set properly, set it:

```
sudo update-alternatives --config java
sudo update-alternatives --config javac
```

Note, for the next steps, it is best to get the commands directly from the neo4j documentation. To get the commands go to:

<https://neo4j.com/docs/operations-manual/current/installation/linux/debian/>

Add the neo4j repo:

```
wget -O - https://debian.neo4j.com/neotechnology.gpg.key | sudo apt-key add -
```

You should see key added like this screenshot:

```
bobsmith@hadoop:~$ wget -O - https://debian.neo4j.com/neotechnology.gpg.key | sudo apt-key add -
19:55:37-- https://debian.neo4j.com/neotechnology.gpg.key
Resolving debian.neo4j.com (debian.neo4j.com)... 18.161.156.18, 18.161.156.38, 18.161.156.9, ...
Connecting to debian.neo4j.com (debian.neo4j.com)|18.161.156.18|:443... Warning: apt-key is deprecated. Manage keyrings
g files in trusted.gpg.d instead (see apt-key(8)).
connected.
HTTP request sent, awaiting response... 200 OK
Length: 6342 (6.2K) [application/pgp-keys]
Saving to: 'STDOUT'

-
100%[=====>] 6.19K --.-KB/s in 0.001s
```

Next add neo4j to your sources list:

```
echo 'deb https://debian.neo4j.com stable latest' | sudo tee -a /etc/apt/sources.list.d/neo4j.list
```

```
bobsmith@hadoop:~$ echo 'deb https://debian.neo4j.com stable latest' | sudo tee -a /etc/apt/sources.list.d/neo4j.list
deb https://debian.neo4j.com stable latest
bobsmith@hadoop:~$ sudo add-apt-repository universe
Adding component(s) 'universe' to all repositories.
Press [ENTER] to continue or Ctrl-c to cancel.
Hit:1 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:4 https://debian.neo4j.com stable InRelease [44.2 kB]
```

In Ubuntu, neo4j requires the universe rep, run:

```
sudo add-apt-repository universe
```

You can now see which versions of the package are available with the apt list command:

```
apt list -a neo4j
```

Next, run apt update:

```
sudo apt-get update
```

Note, the enterprise edition of Neo4j can expire and cause issues, thus, it is recommended to install the community edition. Finally, install the **community open source edition** of Neo4j:

To install Neo4j Community Edition:

```
sudo apt-get install neo4j=1:5.5.0
```

Step 5 (Optional): Common Errors and Fixes

One error involves the system nodes that must exist by default. If you receive an error that "graph.db" does not exist or another database, check your directory structure here:

```
/var/lib/neo4j/data/databases$ ls
```

```
neo4j store_lock system
```

neo4j and system are the two directories in the community edition that installed. If graph.db exists or any other with system nodes, make a backup copy and remove it. Once you delete the .db files, attempt to restart neo4j.

Another common error concerns Java. Neo4j operates with Java 8 and 11 but cypher-shell often tries to use 8. If you get an error like this:

```
Error: A JNI error has occurred, please check your installation and try again
Exception in thread "main" java.lang.UnsupportedClassVersionError: org/neo4j/shell/Main has
been compiled by a more recent version of the Java Runtime
```

My recommendation is to install neo4j on an Ubuntu VM only with the current recommended version of Java from the documentation.

If you need to remove neo4j you can run:

```
sudo apt remove neo4j
sudo apt autoclean
sudo apt autoremove
```

If you need to remove Java:

```
sudo apt remove openjdk-17-jre openjdk-17-jdk --purge
```

If you do need to re-install neo4j remove any versions that exist. Examples:

```
sudo apt-get purge neo4j=1:5.5.0
sudo apt-get purge neo4j=1:4.2.3
sudo apt-get purge neo4j=1:4.2.4
sudo apt-get purge neo4j
sudo apt-get autoremove
```

Make sure java 11, 17, or the optimal version is installed:

```
echo "deb http://httpredir.debian.org/debian stretch-backports main" | sudo tee -a
/etc/apt/sources.list.d/stretch-backports.list
```

```
sudo apt-get update
sudo apt install openjdk-17-jre
sudo apt install openjdk-17-jdk
```

Next, check the version of Java:

```
java -version
```

If the version is not correct, set it:

```
sudo update-alternatives --config java
sudo update-alternatives --config javac
sudo add-apt-repository -y ppa:openjdk-r/ppa
```

install the cypher-shell with neo4j:

```
sudo apt-get install -y cypher-shell neo4j
```