Database Server Setup Instructions

Installation of Pre-requisites

1. Update Apt Get

**$sudo apt-get update**

1. Install NPM

**$sudo apt-get install npm**

1. Install Build Tools

**$sudo apt-get install -y build-essential**

1. Install Neo4j (<https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-neo4j-on-ubuntu-20-04>)
   1. sudo apt update
   2. sudo apt install apt-transport-https ca-certificates curl software-properties-common
   3. curl -fsSL https://debian.neo4j.com/neotechnology.gpg.key | sudo apt-key add –
   4. sudo add-apt-repository "deb https://debian.neo4j.com stable 4.1"
   5. sudo apt install neo4j
   6. sudo systemctl enable neo4j.service

Set up DB User

By default, Neo4j service runs under the ‘neo4j’ user. There shouldn’t be a need to explicitly create this user, but check to make sure it exists:

**$id neo4j**

Set up the DB Directory

Production:

On Production Environment we’d want to mount an AWS Elastic Block Storage Volume to the server for storing all db related files as well as the log files. Refer: <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-using-volumes.html>

1. List available devices

**$lsblk**

Look for the device that is not already mounted, because that's the EBS volume that we want to mount

1. Check if we need to format or not

**$sudo file -s /dev/<device\_name>** # as identified in step 1 above

If the output is just "data" then we need to format using the following command

1. **sudo mkfs -t ext4 /dev/<device\_name>** #only if the format in 2 was 'data'
2. **sudo mkdir -p /var/www/database/data** #create mount point directory
3. **sudo mount /dev/<device\_name> /var/www/database/data**
4. **sudo mkdir /var/www/database/data/log** #create log directory
5. **sudo mkdir /var/www/database/data/db** #create db directory
6. **sudo chown -R neo4j /var/www/database/data/** #change ownership to neo4j user

Staging:

On Staging environment, we’ll just use the local disk space on the server machine to store all the files. So, simply make sure the /var/www/staging/db folder is created and is owned by the user ‘**neo4j** (should have happened as part of the earlier steps)

1. **$sudo su**
2. **$mkdir –p /var/www/database**
3. **$mkdir /var/www/database/data/**
4. **$mkdir –p /var/www/database/data/db**
5. **$mkdir –p /var/www/database/data/log**
6. **$cd /var/www/database**
7. **$chown -R neo4j /var/www/database/data** #change owner to ‘neo4j’ user

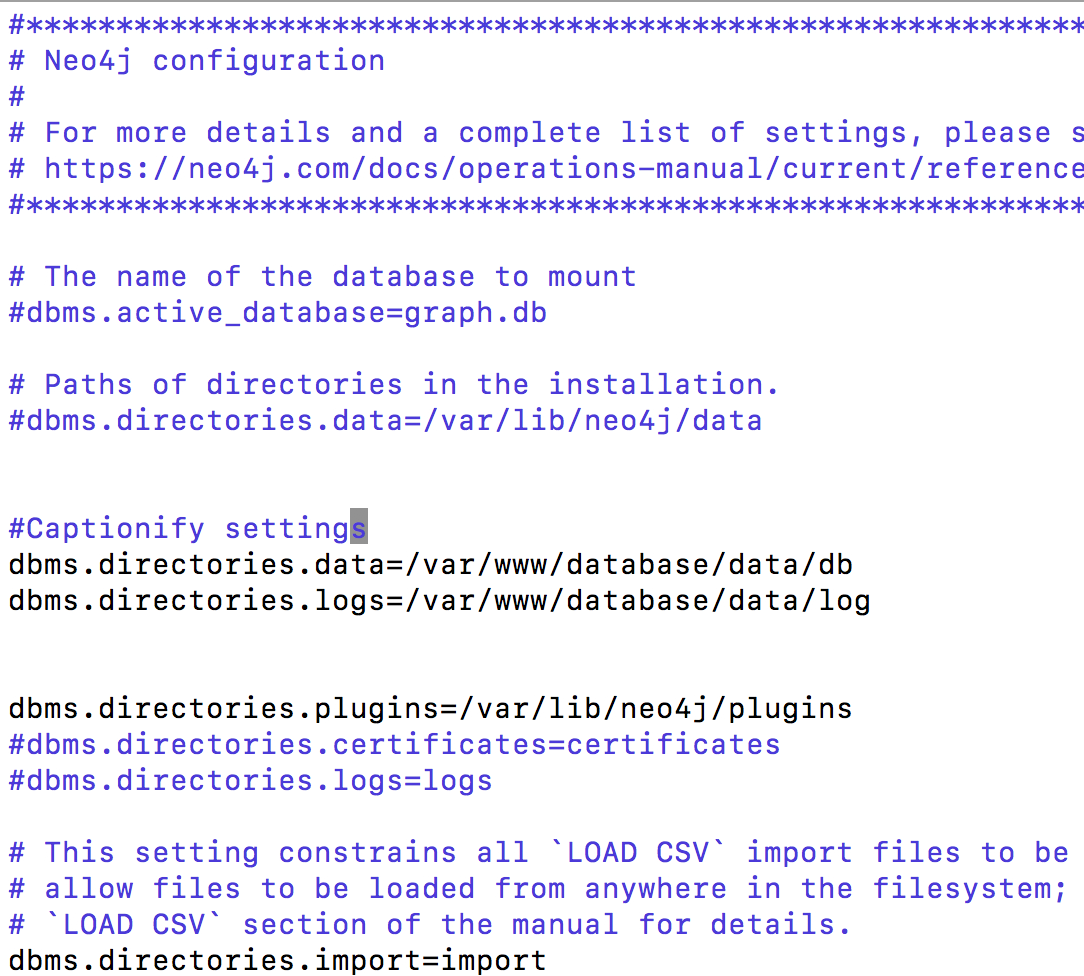
Configure Custom Path for DB and Logs

Configure custom Database and Log Path for Neo4j by editing the configuration file.

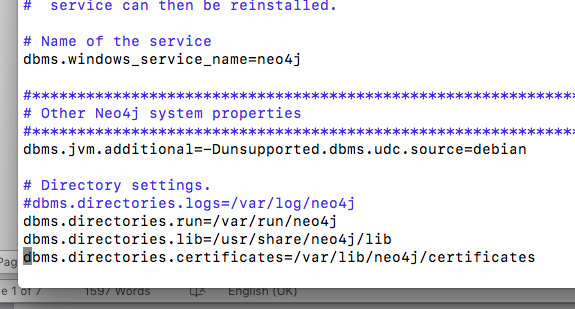
1. **$sudo vi /etc/neo4j/neo4j.conf** #edit the configuration file
2. Find the line below and change the database and log directories as below:

**dbms.directories.data=/var/www/database/data/db**

**dbms.directories.logs=/var/www/database/data/log**



Note: Make sure the same keys are not present elsewhere in the file - comment out any other duplicate key settings by adding a #



Configure Remote Web Admin Access

Configure Neo4j to allow Remote Web Admin Access.

**In the Neo4j.conf configuration file (on the server):**

1. **$sudo vi /etc/neo4j/neo4j.conf** #Edit the /etc/neo4j/neo4j.conf configuration file
2. Find the line under HTTP Connector section and uncomment/edit it to match the following:

**dbms.connector.http.listen\_address=0.0.0.0:7474**

1. **Disable TLS conector for bolt: (uncomment this line in the conf):**

dbms.connector.bolt.tls\_level=DISABLED

1. **Enable non-local connections by uncommeting this line:**

dbms.default\_listen\_address=0.0.0.0

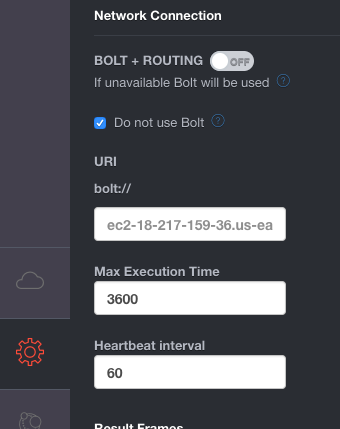
1. **$sudo service neo4j restart** #Restart the neo4j service
2. You should now be able to access the web admin using the public url, something like:

[**http://ec2-52-14-131-115.us-east-2.compute.amazonaws.com:7474/**](http://ec2-52-14-131-115.us-east-2.compute.amazonaws.com:7474/)

Note: Make sure to open up port 7474 on AWS Security Group if the server is in an EC2 instance!

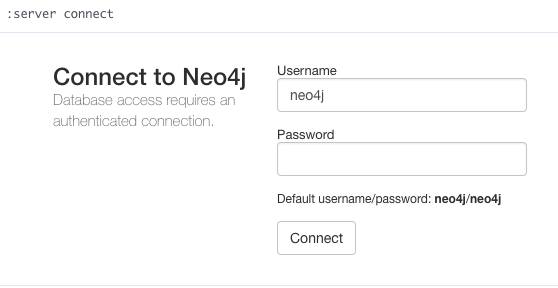
**In the Web Admin Portal:**

1. Disable Bolt in the web admin interface. Go to **Settings > BOLT + ROUTING > Check "Do not use bolt" (not valid for latest version of Neo4j)**

****

1. In the web admin interface**, enter :server** connect, and enter the neo4j login credentials, and it should be connected

Note: default credentials are **neo4j/neo4j**



1. Change the default password to the actual password via the web admin interface (it should prompt to change)

**:server connect**

or

**:server change-password**

**(actually, no need to change if protected by AWS security group and DB server is separate)**

Note: the new password should match whatever you are setting in the .env file in the Node App Server (see App Server instructions for more details)

Security Group Settings on AWS EC2 Console

The Neo4J Database instance needs to be allow inbound traffic through specific ports. In particular, you need to:

* Allow traffic through ports 7474 for the web console
* Allow traffic through port 7687 for the app server (node) to connect to the dB
* Allow traffic through port 22 for you to have SSH access (this can be done temporarily when you need access – up to you)

The final settings should look something like this:

Graphical user interface, application, Teams

Description automatically generated