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 Java (for Neo4j) (<http://neo4j.com/docs/operations-manual/current/installation/linux/debian/>)
2. **$sudo add-apt-repository ppa:webupd8team/java**
3. **$sudo apt-get update**
4. **$sudo apt-get install oracle-java8-installer**
5. **$update-java-alternatives --list** #list java versions on machine
6. **$sudo update-java-alternatives --jre --set java-1.8.0-openjdk-amd64** #replace last argument with the Java 8 version name
7. Install Neo4j
8. **$sudo wget --no-check-certificate -O - https://debian.neo4j.org/neotechnology.gpg.key | apt-key add -**
9. **$echo 'deb http://debian.neo4j.org/repo stable/' | sudo tee /etc/apt/sources.list.d/neo4j.list** # add -a for append (>>)
10. **$sudo apt-get update**
11. **$sudo apt-get --assume-yes --allow-unauthenticated install neo4j=3.1.1** #don't prompt, and allow unauthenticated

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 /var/www/production/db** #create mount point directory
3. **sudo mount /dev/<device\_name> /var/www/production/db**
4. **sudo mkdir /var/www/production/db/log** #create log directory
5. **sudo chown –R neo4j /var/www/production/db** #change ownership to node 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/staging**
3. **$mkdir /var/www/staging/db**
4. **$mkdir /var/www/staging/db/log**
5. **$cd /var/www/staging**
6. **$chown -R neo4j /var/www/staging/db** #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 (depending on production or staging):

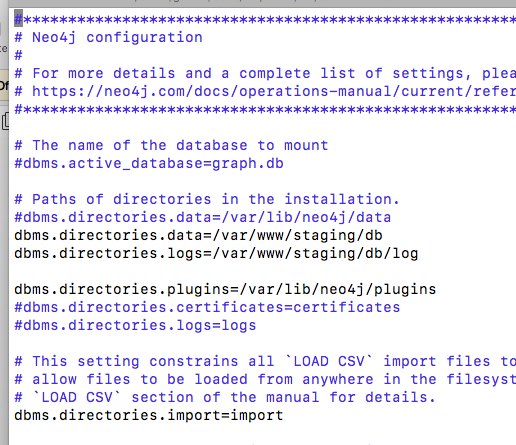
**dbms.directories.data=/var/www/production/db**

**dbms.directories.logs=/var/www/production/db/log**

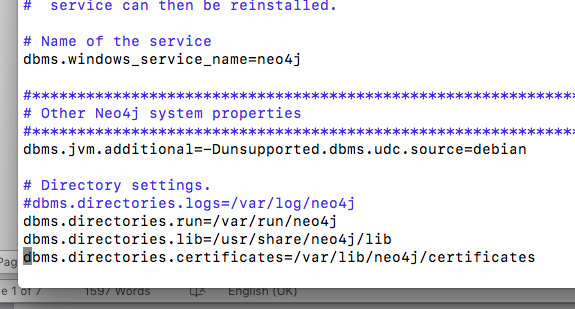
or

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

**dbms.directories.logs=/var/www/staging/db/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**

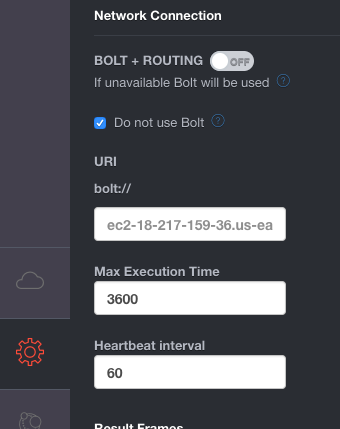
1. **$sudo service neo4j restart** #Restart the neo4j service
2. You should now be able to access the web admin using the public url, something like:

[**http://ec2-18-221-25-182.us-east-2.compute.amazonaws.com:7474/**](http://ec2-18-221-25-182.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!

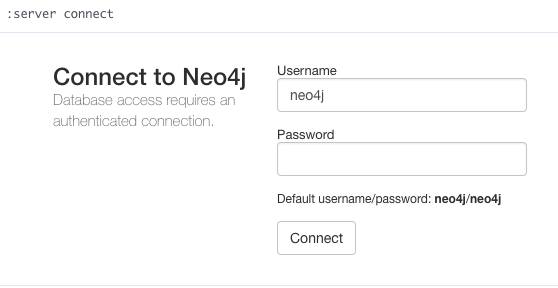
**In the Web Admin Portal:**

1. Disable Bolt in the web admin interface. Go to **Settings > BOLT + ROUTING > Check "Do not use bolt"**

****

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**

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)