Installing Neo4j Server on Linux Server

Author: David Vadnais, Bder Khan, and Frankie Torres

**Prerequisites**

* Linux server with public ip

**Information**

Lines in green are notes and lines in black are commands. I used vi/centos7 to produce this guide. The three dots known as an ellipsis within opened file examples stand for additional commercial off the shelf code that you do not need to edit.

**Derivations**

These are alternative solutions you may want to consider.

* It is possible to load encryption directly into neo4j
  + Pros
    - easier first-time setup (in theory)
    - Light weight
  + Cons
    - Less generic ssl solution for future server uses.
    - Less containerizeable to Docker/Kubernetes
    - Less mature product so less support is available
  + How
    - <https://stackoverflow.com/questions/29735738/how-to-use-ssl-certificates-in-neo4j-instead-of-self-signed-certificates-or-sna>
* Using no firewall or using ufw
  + Do not do this on a public vm
* Use bolt+s instead of neo4j+s for graphical user interface (GUI) to server connection
* Containerization
  + If you already have Docker or Kubernetes running use <https://hub.docker.com/_/neo4j> instead of all this work

**This guide follows these steps**

1. Check your Linux distro
2. Install Java OpenJDK8 or OpenJDK11
3. Add neo4j repo
4. Install neo 4j
5. Setup neo4j conf (Minimum viable product)
6. Let's encrypt/nginx
7. Firewalls
8. Neo4j and systemd
9. Uploading a CSV
10. Debug

# 1 Check your Linux distro

**# To get your linux distro run:**

cat /etc/\*-release

**# I used a fedora/red hat enterprise linux/centos-based distro so my installer of choice is yum (dnf for 8+)**

**# for ubuntu swap yum with apt**

**# for alpine swap yum with apk**

**# Best practice**

yum -y update

# 2 Install Java OpenJDK8 or OpenJDK11

**# Install OpenJDK 11. Use OpenJDK unless you have a license for another JDK. Use java 11 because that is what is required. Some version of neo4j require java 8 so check if you want a version that needs this.**

**# Open JDK11**

**# centos:** [**https://computingforgeeks.com/how-to-install-java-11-openjdk-11-on-rhel-8/**](https://computingforgeeks.com/how-to-install-java-11-openjdk-11-on-rhel-8/)

**# Debian :** [**https://www.ubuntu18.com/ubuntu-install-openjdk-11/**](https://www.ubuntu18.com/ubuntu-install-openjdk-11/)

yum install java-11-openjdk-devel

**# Open JDK 8**

**# More info for centos/rhel/fedora:** [**https://phoenixnap.com/kb/how-to-install-java-centos-8**](https://phoenixnap.com/kb/how-to-install-java-centos-8)

**# More info for ubuntu:** [**https://linoxide.com/ubuntu-how-to/install-java-ubuntu-20-04/**](https://linoxide.com/ubuntu-how-to/install-java-ubuntu-20-04/)

java -version **#Check if desired java version is already installed**

yum install java-1.8.0-openjdk

# 3 Add neo4j repo

**# more info centos:** [**https://medium.com/@prabhin.mp/how-to-install-neo4j-on-centos-d5c1be242471**](https://medium.com/@prabhin.mp/how-to-install-neo4j-on-centos-d5c1be242471)

**# more info ubuntu :** [**https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-neo4j-on-ubuntu-20-04**](https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-neo4j-on-ubuntu-20-04)

**# Add your GPG key**

wget <http://debian.neo4j.org/neotechnology.gpg.keyrpm> --import neotechnology.gpg.key

**# Add your repository**

cat <<EOF> /etc/yum.repos.d/neo4j.repo

[neo4j]

name=Neo4j Yum Repo

baseurl=http://yum.neo4j.org

enabled=1

gpgcheck=1

EOF

# 4 Install neo4j

yum -y install neo4j

**# I had to do this (below), but the official docs say it is not needed** [**https://neo4j.com/docs/operations-manual/current/installation/linux/rpm/#linux-rpm-install-standard**](https://neo4j.com/docs/operations-manual/current/installation/linux/rpm/#linux-rpm-install-standard)

yum –y install <https://dist.neo4j.org/neo4j-java11-adapter.noarch.rpm>

exec bash

# 5 Setup neo4j conf

**# Edit /etc/neo4j/neo4j.conf**

vim /etc/neo4j/neo4j.conf

|  |
| --- |
| ...  #org.neo4j.server.webserver.address=0.0.0.0 **# remove # from**  **front and set 0.0.0.0 to your public ip address**  **...** |

**# Turn on neo 4j (Note: Go to section 8 to learn the best way to do this)**

service neo4j restart

service neo4j enable  
  
**# Log into neo4j and by entering your web browser the url and port 7474 (Note: this is by default http) (Note2: unless your web browser is old this will fail and you will need to set up encryption: section 6)**



# 6 Let's encrypt/nginx

**# Unfortunately, most modern web browsers will not allow back end traffic to move unencrypted even after a self-signed certificate is accepted, or an insecure connection is allowed. The solution to this problem I used is nginx and let's encrypt although alternatives do exist and using apache webserver would be the most lightweight solution.**

**# More info about lets encrypt install :** [**https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-centos-7**](https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-centos-7)

**# Compare security from different certs :** [**https://www.digitalocean.com/community/tutorials/a-comparison-of-let-s-encrypt-commercial-and-private-certificate-authorities-and-self-signed-ssl-certificates**](https://www.digitalocean.com/community/tutorials/a-comparison-of-let-s-encrypt-commercial-and-private-certificate-authorities-and-self-signed-ssl-certificates)

yum install epel-release

yum install certbot-nginx

yum install nginx

systemctl start nginx

vi /etc/nginx/nginx.conf

server\_name \_; **#Replace underscore with yourServer.com and www.yourServer.com. Save your edits with esc :wq**

nginx –t **# Checks config**

systemctl daemon-reload

systemctl reload nginx

systemctl status nginx

**# Open the firewall ports desired. see section 7**

**# Create cert**

certbot --nginx -d example.com -d [www.example.com](http://www.example.com)

**# Updating Diffie-Hellman Parameters**

openssl dhparam -out /etc/ssl/certs/dhparam.pem 2048

**# Add to /etc/nginx/nginx.conf**

Vi /etc/nginx/nginx.conf

|  |
| --- |
| ... ssl\_dhparam /etc/ssl/certs/dhparam.pem; ... |

nginx –t

systemctl reload nginx

**# use cron to enable auto renewal of the certificate**

crontab –e

|  |
| --- |
| …  15 3 \* \* \* /usr/bin/certbot renew –quiet |

**# At this point you have https to a port through the reverse proxy nginx now we must set the two server ports to be covered by nginx lets convert to 7474 -->2828 and 7687-->2829 below is an example for 2828**

Vi /etc/nginx/nginx.conf

|  |
| --- |
| …  # Settings for a TLS enabled server.  server {  listen 2828 ssl http2 default\_server;  listen [::]:2828 ssl http2 default\_server;  server\_name \_;  root /usr/share/nginx/html;    ssl\_certificate "/etc/letsencrypt/live/example.com/cert.pem";  ssl\_certificate\_key "/etc/letsencrypt/live/example.com/privkey.pem";  ssl\_session\_cache shared:SSL:1m;  #ssl\_session\_timeout 10m;  #ssl\_ciphers HIGH:!aNULL:!MD5;  #ssl\_prefer\_server\_ciphers on;    # Load configuration files for the default server block.  include /etc/nginx/default.d/\*.conf;    location / {  proxy\_pass http://davidvadnais.com:7474/;  }    error\_page 404 /404.html;  location = /404.html {  }    error\_page 500 502 503 504 /50x.html;  location = /50x.html {  }  }  …. |

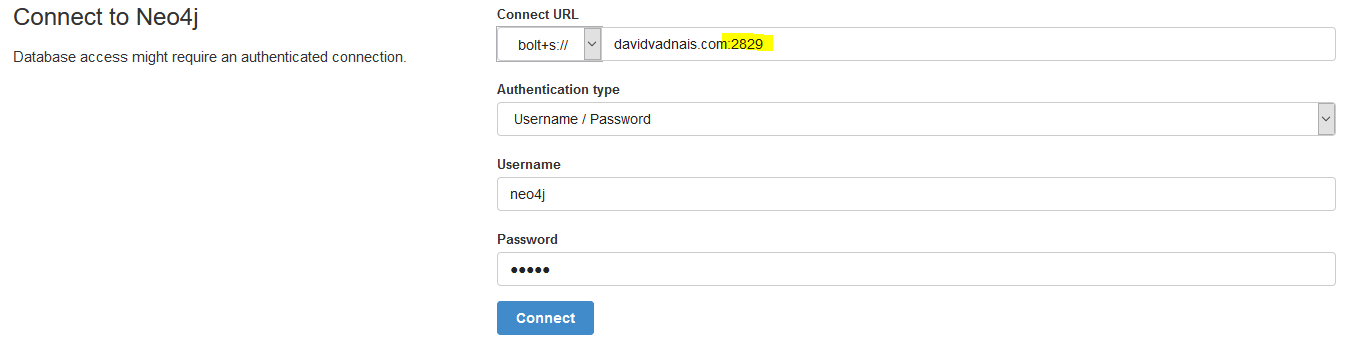
Vi /etc/neo4j/neo4j.conf

|  |
| --- |
| …  # The address at which this server can be reached by its clients. This may be the server's IP address or DNS name, or  # it may be the address of a reverse proxy which sits in front of the server. This setting may be overridden for  # individual connectors below.  dbms.default\_advertised\_address=example.com  … # Bolt connector  dbms.connector.bolt.enabled=true  #dbms.connector.bolt.tls\_level=OPTIONAL  #dbms.connector.bolt.listen\_address=davidvadnais.com:7687  #dbms.connector.bolt.advertised\_address=davidvadnais.com:7687    # HTTP Connector. There can be zero or one HTTP connectors.  dbms.connector.http.enabled=true  dbms.connector.http.listen\_address=davidvadnais.com:7474  #dbms.connector.http.advertised\_address=:7474 |

At this point you should be able to access your server at example.com:2828. Access your server through a bolt connection with server connect shown below:



Do not forget to change your end point to the encrypted endpoint as shown below



# 7 Firewalls

**# Most linux distros use a combination of iptables, selinux, and firewalld. Here is how you add ports to each of these**

**# Our ports are**

* **7474** 
  + **2828 IF USING REVERSE PROXY NGINX**
* **7687**
  + **2829 IF USING REVERSE PROXY NGINX**

**# firewalld**

**More info:**

<https://firewalld.org/documentation/howto/open-a-port-or-service.html>

firewall-cmd --permanent --zone=public --add-port=7474/tcp

firewall-cmd --reload

firewall-cmd --get-zones

**# Iptables**

iptables -I INPUT -p tcp -m tcp --dport 7474 -j ACCEPT

**# Selinux**

semanage port -a -t http\_port\_t -p tcp 7474

# 8 Neo4j and systemd

**# It is now considered best practice to use systemd for services. Here is an example systemd drop-in for neo4j**

|  |
| --- |
| [Unit]  Description=Neo4j Graph Database  After=network-online.target  Wants=network-online.target    [Service]  ExecStart=/usr/share/neo4j/bin/neo4j console  Restart=on-failure  User=neo4j  Group=neo4j  Environment="NEO4J\_CONF=/etc/neo4j" "NEO4J\_HOME=/var/lib/neo4j"  LimitNOFILE=60000  TimeoutSec=120    [Install]  WantedBy=multi-user.target |

systemctl start neo4j

systemctl status neo4j

# 9 Uploading a CSV

For security purposes neo4j will only open csv’s stored in /var/lib/neo4j/import/\*

# 10 Adding Bloom to neo4j server

More information : <https://neo4j.com/docs/bloom-user-guide/current/bloom-installation/>

Download jar at : <https://neo4j.com/download-center/#bloom>

Scp 4.x or 3.x jar to your server or follow instructions for standalone server (good for microservices)

Place Jar in

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| File type | Description | Linux / macOS / Docker | Windows | Debian / RPM | Neo4j Desktop |
| Plugins | Custom code that extends Neo4j, for example, user-defined procedures, functions, and security plugins. | *<neo4j-home>/plugins* | *<neo4j-home>\plugins* | */var/lib/neo4j/plugins* | From the *Open* dropdown menu of your Neo4j instance, select *Terminal*, and navigate to *<installation-version>/plugins*. |

# 11 Debugging

Unfortunately, things will not go as smoothly as I would like when following this guide since I made it after completing the install instead of during. Here are some debugging tips

* Browser
  + From your browser hit F12.
    - This will bring up the web page debugger
  + Go to network and redo whatever traffic failed. This should allow you to monitor your traffic
* Linux CLI
  + tcpdump
    - Watch traffic coming over a port
    - More info: <https://www.tcpdump.org/manpages/tcpdump.1.html>
  + Netstat
    - Shows active ports
    - I use options “netstat -tulpn”
    - More info: <https://linux.die.net/man/8/netstat>
  + Route
    - show / manipulate the IP routing table
    - More info : <https://linux.die.net/man/8/route>
  + Journalctl -xe
    - For when systemctl status SERVICE just does not cut it