# Neo4j instance files

Default folders to frequently use for managing the Neo4j instance.

| **Purpose** | **Description** |
| --- | --- |
| Tools | The /usr/bin folder contains the tooling scripts you will typically run to manage the Neo4j instance. |
| Configuration | Neo4j.conf is the primary configuration file for the Neo4j instance and resides in the /etc/neo4j folder. |
| Logging | The /var/log/neo4j folder contains log files that you can monitor. |
| Database(s) | The /var/lib/neo4j/data folder contains the database(s). |

# Post-installation preparation

To control who can manage the Neo4j instance

## Changing the neo4j password

neo4j-admin set-initial-password newPassword

### Managing the Neo4j instance

You can start and stop the instance regardless of whether the neo4j service is enabled.

sudo systemctl start neo4j

sudo systemctl stop neo4j

sudo systemctl restart neo4j

sudo systemctl status neo4j

### Checking the status of the instance

sudo systemctl status neo4j

### Viewing the neo4j log

sudo journalctl -u neo4j # To view the entire neo4j log file.

sudo journalctl -e -u neo4j # To view the end of the neo4j log file.

sudo journalctl -u neo4j -b > neo4j.log # Where you can view neo4j.log in an editor.

### Exercise #1: Managing the Neo4j instance

Exercise steps:

1. Open a terminal on your system.
2. View the status of the Neo4j instance.

sudo systemctl status neo4j

1. Stop the Neo4j instance.

sudo systemctl stop neo4j,

1. View the status of the Neo4j instance.

sudo systemctl status neo4j,

1. Examine the Neo4j log file.

sudo journalctl -e -u neo4j

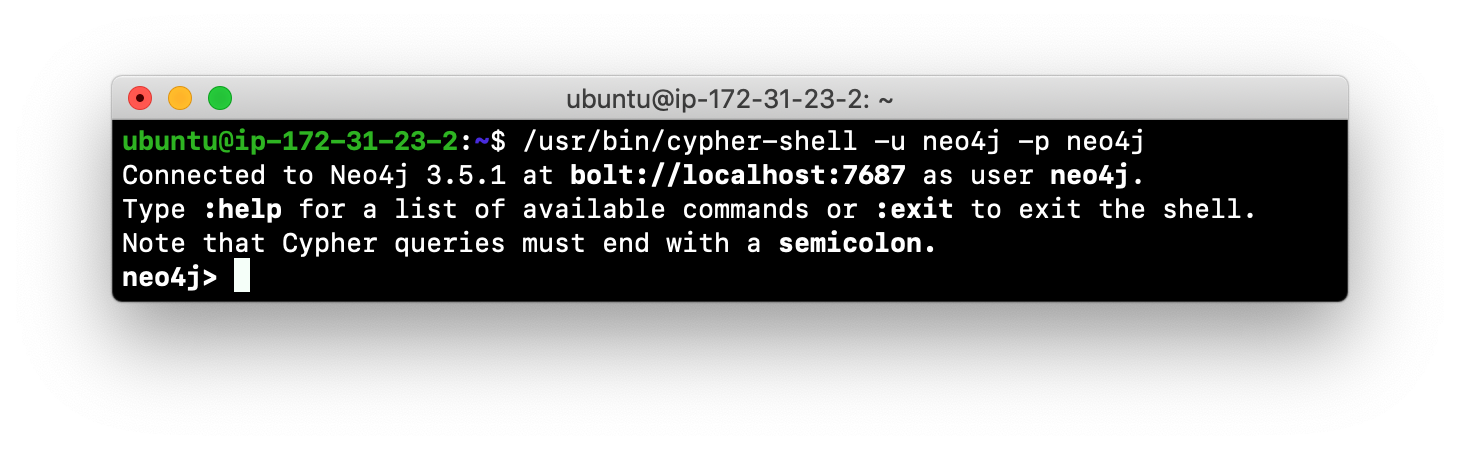
1. Examine the files and folders created for this Neo4j instance.

cd /etc/neo4j

ls -al

1. Using cypher-shell
   1. Enables you to access the Neo4j database from a terminal window.

cypher-shell -u neo4j -p neo4j



* 1. Changing the default password

ALTER CURRENT USER

SET PASSWORD FROM 'secret' TO 'secret123'

**Exercise: Copying a Neo4j database**

* Never copy database from one location in the filesystem to another location.
* Copy a Neo4j database by creating an offline backup.
* To create offline backup:

1. Stop the Neo4j instance.

stop database neo4j # Put this command in Cypher in browser

OR

sudo systemctl stop neo4j

1. Ensure that the folder where you will dump the database exists.

ls -la /var/lib/neo4j/data/databases

sudo mkdir /usr/local/dumps

sudo chown -R neo4j:neo4j /usr/local/dumps

ls -al /usr/local/dumps

sudo rm -rf /usr/local/dumps/\*

1. Use the dump command of the neo4j-admin tool to create the dump file.

neo4j-admin dump --database=neo4j --to=/usr/local/dumps/neo4j-graph-dump

start database neo4j # Put this command in Cypher in browser

OR

sudo systemctl start neo4j

1. You can now copy the dump file between systems.

* Then, if you want to create a database from any offline backup file to use for a Neo4j instance, you must:

1. Determine what you will call the new database and adjust neo4j.conf to use this database as the active database.
2. Use the load command of the neo4j-admin tool to create the database from the dump file using the same name you specify in the neo4j.conf file.

neo4j-admin load --from=/usr/local/dumps/neo4j-graph-dump --database=neo4j3db

ls -la /var/lib/neo4j/data/databases

1. Create database in Cypher-Shell or Cypher Browser

:use system

Create database neo4j3db

:use neo4j3db

match(n) return count(n);

**Note:** Dumping and loading a database is done when the Neo4j database instance is stopped.

* Offline backup is typically done for initial setup and development purposes
* Online backup and restore is done in a production environment.

### Exercise: Deleting a Neo4j database

To delete a Neo4j database used by a Neo4j instance you must:

* Stop the Neo4j instance or Database

:USE SYSTEM

STOP Database neo4j3db

OR

sudo systemctl stop neo4j

* Remove the folder for the active database

sudo rm -rf /var/lib/neo4j/data/databases/neo4j3db

* Also need to remove database from Cypher

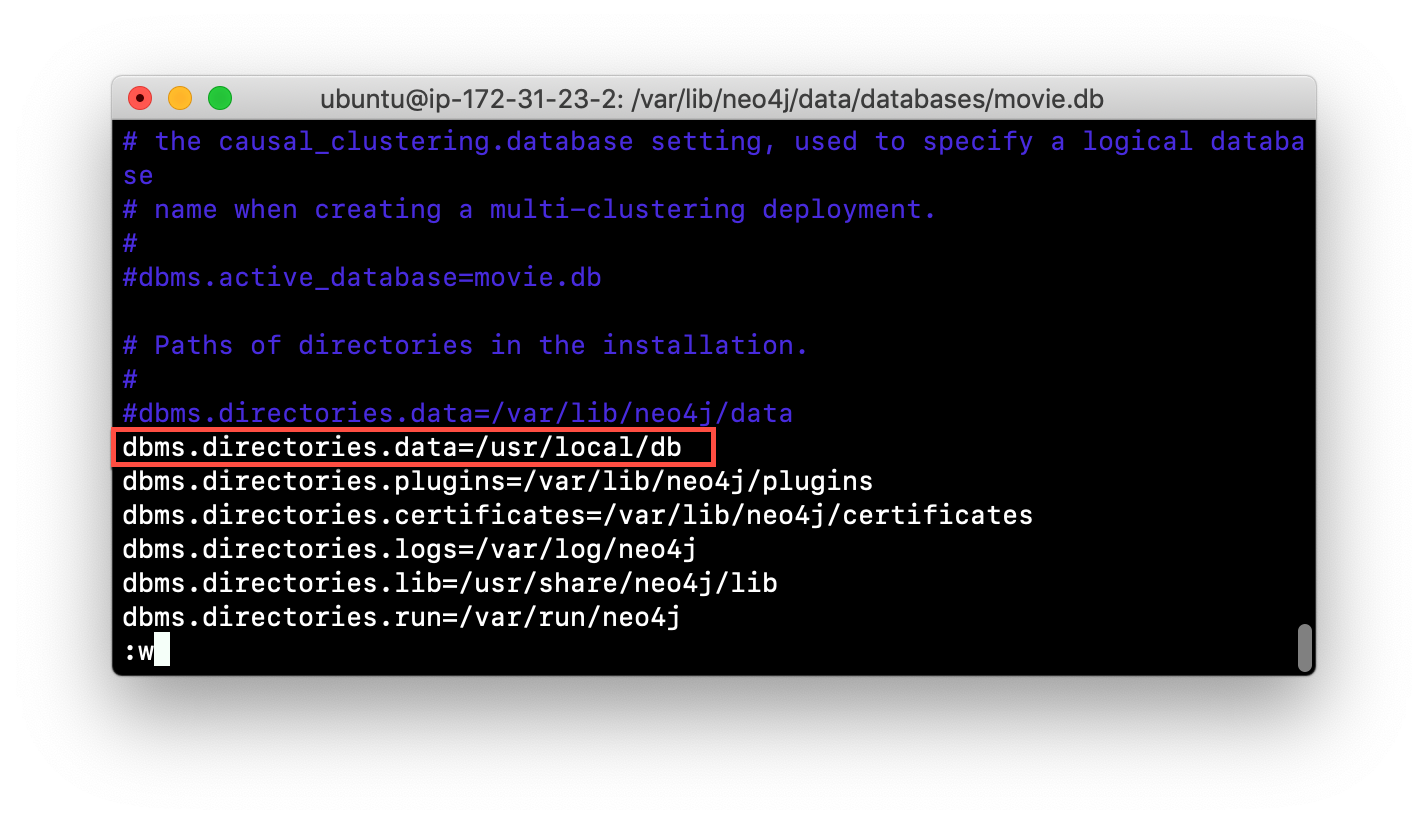
:USE SYSTEM

DROP DATABASE neo4j3db

### Modifying the location of the database

sudo systemctl stop neo4j

sudo vim /etc/neo4j/neo4j.conf

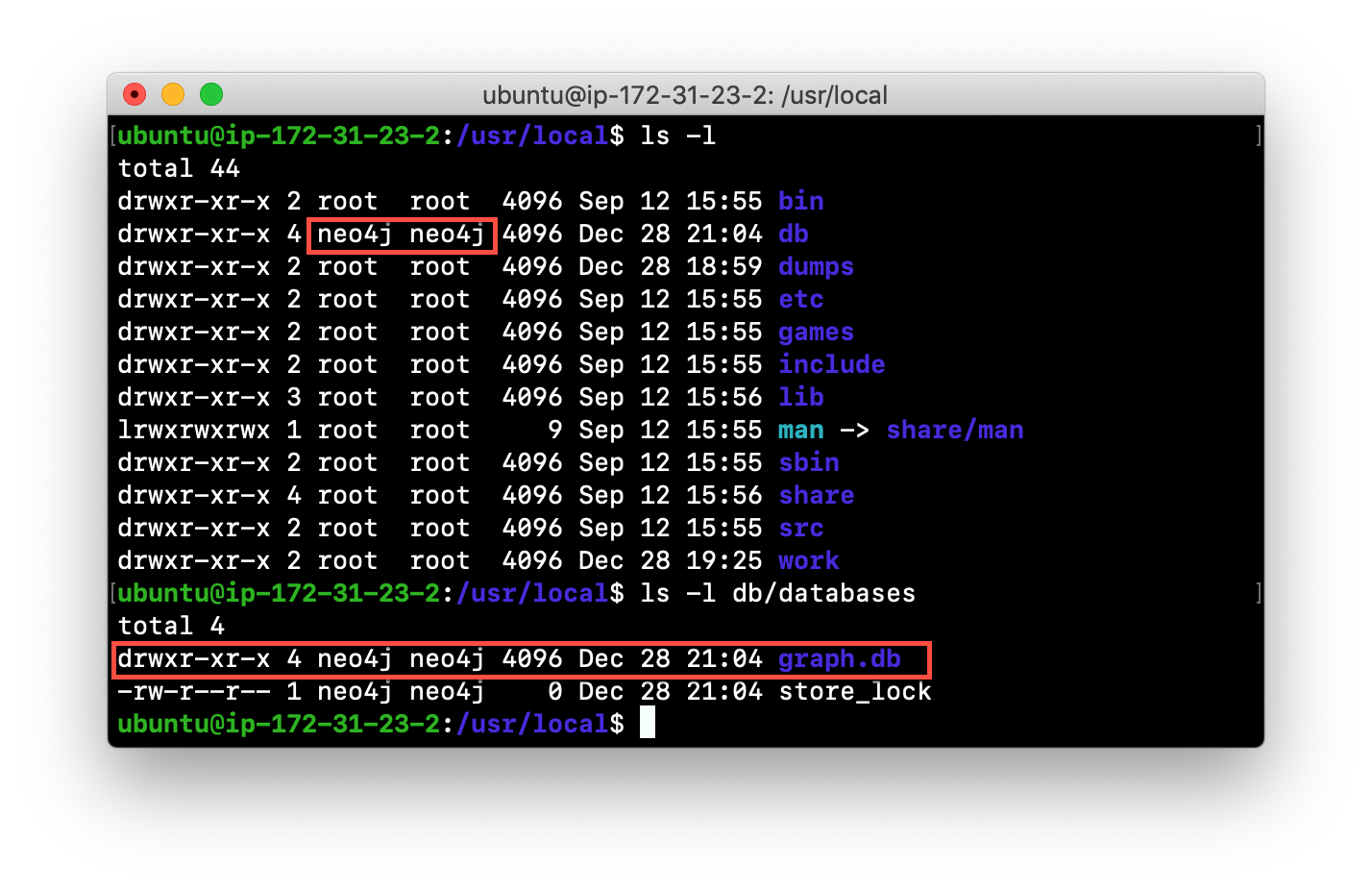


mkdir -p /usr/local/db

### Starting Neo4j instance with a new location

sudo systemctl start neo4j

* Ensure new location exist
* A new database named <active\_database> will be created
* We are using the default database name in the configuration file.
* To move existing database must dump and load the database to safely copy it to the new location



### Checking the consistency of a database

If a specific database has been corrupted, you can perform a consistency check on it.

#Modify the Neo4j configuration to use a database named graph2db, rather than graph.db.

sudo systemctl stop neo4j

cd /var/lib/neo4j/data/databases/

sudo neo4j-admin check-consistency --database=neo4j3db --report-dir=/usr/local/reports

**Note:** No report will be written to the reports folder if the consistency check passed.

Corrupt the database by modifying the file

**neo4j3db/neostore.nodestore.db** by adding some text to the file.

#Run consistency check

sudo neo4j-admin check-consistency --database=neo4j3db --report-dir=/usr/local/reports

### Examples: Scripting with cypher-shell: Adding constraints

Create 3 file:

vim AddNodes.cypher

CREATE (m:Movie:Action {title: 'Batman Begins'})

RETURN m.title;

vim AddNodes.sh

cat ./ AddNodes.cypher | /usr/bin/cypher-shell -u neo4j -p neo4j --format verbose

#PrepareDB.sh that initializes the log file, PrepareDB.log, and calls the script to add the constraints:

vim PrepareDB.sh

rm -rf ./PrepareDB.log

./AddNodes.sh 2>&1 >> ./PrepareDB.log

cat ./PrepareDB.log

Visit - http://<IPAddress>:7474/browser/

Query:

MATCH(n) return n

### Managing plugins

Some applications can use Neo4j out-of-the-box

But many applications require additional functionality that could be:

1. A library supported by Neo4j such as GraphQL or GRAPH ALGORITHMS.
2. A community-supported library, such as APOC.
3. Custom functionality that has been written by the developers of your application.

### Example: Installing the APOC plugin

APOC contains over 450 user-defined procedures that make accessing a graph incredibly efficient and much easier than writing your own Cypher statements to do the same thing.

Open - http://<IP Address>:7474/browser/

CALL dbms.procedures()

# Note that APOC procedures not available

Install APOC: Obtain the plugin from - <https://github.com/neo4j-contrib/neo4j-apoc-procedures/tags>

cd /var/lib/neo4j/plugins

wget <https://github.com/neo4j-contrib/neo4j-apoc-procedures/releases/download/4.1.0.2/apoc-4.1.0.2-all.jar>

sudo chmod +x \*.jar

ls -al

sudo systemctl restart neo4j

Open - http://<IP Address>:7474/browser/

:USE <db\_name>

CALL dbms.procedures()

CALL apoc.help("apoc")

WITH "https://api.stackexchange.com/2.2/questions?pagesize=100&order=desc&sort=creation&tagged=neo4j&site=stackoverflow&filter=!5-i6Zw8Y)4W7vpy91PMYsKM-k9yzEsSC1\_Uxlf" AS url

CALL apoc.load.json(url) YIELD value

UNWIND value.items AS item

RETURN item.title, item.owner, item.creation\_date, keys(item)

### Configuring connector ports for the Neo4j instance

The Neo4j instance uses default port numbers that may conflict with other processes on your system

The ports frequently used are the connector ports:

| **Name** | **Port #** | **Description** |
| --- | --- | --- |
| HTTP | 7474 | Used by Neo4j Browser and REST API |
| HTTPS | 7473 | Used by REST API. Requires additional SSL configuration. |
| Bolt | 7687 | Bolt connection used by Neo4j Browser, cypher-shell, and client applications. |

Can change these connector ports by modifying these property values in the neo4j.conf file:

# Bolt connector

dbms.connector.bolt.enabled=true

#dbms.connector.bolt.tls\_level=OPTIONAL

#dbms.connector.bolt.listen\_address=:7687

# HTTP Connector. There can be zero or one HTTP connectors.

dbms.connector.http.enabled=true

#dbms.connector.http.listen\_address=:7474 -> 17474

# HTTPS Connector. There can be zero or one HTTPS connectors.

dbms.connector.https.enabled=true

#dbms.connector.https.listen\_address=:7473

**Make Sure to add the new port number in Inbound Rules in Azure**

#Restart the Neo4j instance after port change

sudo systemctl restart neo4j

Visit: http://<IP Address>:17474/browser/

### Performing online backup and restore

Online backup is used in production where the application cannot tolerate the database being unavailable.

To enable a Neo4j instance to be backed up online, you must add these two properties to your neo4j.conf file:

dbms.backup.enabled=true

dbms.backup.listen\_address=0.0.0.0:6362

sudo systemctl restart neo4j

#Create a folder

sudo mkdir -p /usr/local/backup

sudo mkdir -p /usr/local/work/reports

sudo chown -R neo4j:neo4j /usr/local/backup

sudo chown -R neo4j:neo4j /usr/local/work/reports

#Perform an online backup of the active database

export HEAP\_SIZE=2G

neo4j-admin backup --from=localhost:6362 --backup-dir=/usr/local/backup --database=neo4j --check-consistency=true --report-dir=/usr/local/work/reports --pagecache=4G

### Restore the database from the backup

neo4j-admin restore --from=/usr/local/backup/neo4j --database=neo4j5db

neo4j-admin check-consistency --database=neo4j5db --report-dir=/usr/local/reports

ls -al /var/lib/neo4j/data/databases/

1. Create database in Cypher-Shell or Cypher Browser

:use system

Create database neo4j5db

:use neo4j5db

match(n) return count(n);

### Using the import tool to create a database

* For large datasets, a best practice is to import the data using the import command of the neo4j-admin tool
* This tool creates the database from a set of .csv files.
* The data import creates a database
* You must run the import tool with the Neo4j instance stopped.

sudo mkdir -p /usr/local/import

sudo chown -R neo4j:neo4j /usr/local/import

cd /usr/local/import

vim movies.csv

movieId:ID,title,year:int,:LABEL

tt0133093,"The Matrix",1999,Movie

tt0234215,"The Matrix Reloaded",2003,Movie;Sequel

tt0242653,"The Matrix Revolutions",2003,Movie;Sequel

vim actors.csv

personId:ID,name,:LABEL

keanu,"Keanu Reeves",Actor

laurence,"Laurence Fishburne",Actor

carrieanne,"Carrie-Anne Moss",Actor

vim roles.csv

:START\_ID,role,:END\_ID,:TYPE

keanu,"Neo",tt0133093,ACTED\_IN

keanu,"Neo",tt0234215,ACTED\_IN

keanu,"Neo",tt0242653,ACTED\_IN

laurence,"Morpheus",tt0133093,ACTED\_IN

laurence,"Morpheus",tt0234215,ACTED\_IN

laurence,"Morpheus",tt0242653,ACTED\_IN

carrieanne,"Trinity",tt0133093,ACTED\_IN

carrieanne,"Trinity",tt0234215,ACTED\_IN

carrieanne,"Trinity",tt0242653,ACTED\_IN

neo4j-admin import --database=importdbdb --nodes=movies.csv --nodes=actors.csv --relationships=roles.csv

#Modify the neo4j.conf file to use importdbdb as the active database (Optional)

sudo vim /etc/neo4j/neo4j.conf

dbms.active\_database= importdbdb

sudo systemctl restart neo4j

cypher-shell -u neo4j -p secret

create database importdbdb

:USE importdbdb

MATCH (n) RETURN n

:exit

### Copy a database to another

The copy command of neo4j-admin is used to copy data from an existing database to a new database.

#Use the copy command to take a copy of the database neo4j

STOP DATABASE neo4j

neo4j-admin copy --from-database=neo4j --to-database=neo4jcopy

#A new database with the name copy now exists on the server, but it is not automatically picked up by Neo4j. To start the new database you have to insert it into Neo4j with the following Cypher query:

CREATE DATABASE neo4jcopy

START DATABASE neo4j