**Deployment Instruction on a Linux operating system**

This ETL tool with its scheduler pulls data from the GTAS database regularly and loads it into the Neo4j database by creating nodes and relationships.

N**ote:** Although the following directories can be created at any path on your disc, in order to shorten the deployment/configuration steps and follow the default settings, it is recommended to use the paths and file locations stated on this document.

1. **Prerequisite**

* Java 8 is required to install and run this project.
* The GTAS database should be up and running.

1. **Required components**
2. Download the *Pentaho Kettle-Neo4j Remix* zip file :

* Page: <http://www.kettle.be/>
* Direct link: <https://s3.amazonaws.com/kettle-neo4j/kettle-neo4j-remix-8.2.0.3-519-REMIX.zip>

1. Download the *Neo4j Community Edition*:

* Page: [https://neo4j.com/download-center/](%20https:/neo4j.com/download-center/)
* Direct link: [https://go.neo4j.com/download-thanks.html?edition=community&release=3.5.3&flavour=unix&\_ga=2.43964837.1214999148.1553022176-274628146.1546638944&\_gac=1.162495054.1551465140.Cj0KCQiAzePjBRCRARIsAGkrSm4i3YqNIWXuUxRUPHnUmQtkKO5\_b3dXKYGqVOhJLtp4PdRPhl8dwm0aAoeKEALw\_wcB](https://go.neo4j.com/download-thanks.html?edition=community&release=3.5.3&flavour=unix&_ga=2.43964837.1214999148.1553022176-)

1. Download the *gtas-neo4j-etl* project: <https://github.com/US-CBP/GTAS/tree/dev>
2. Download and build the *Spring-Boot ETL scheduler* from the dev branch:

<https://github.com/US-CBP/GTAS/tree/dev>

1. **Pentaho Data Integration Tool Installation**
2. Go to the root directory: (*cd /*)
3. Create the pentaho directory under opt: (*sudo mkdir /opt/pentaho)*
4. Assign the ownership of this directory and its subdirectories to a user:

(*sudo chown -R <username> /opt/pentaho*)

1. Allow the user and the group of the user to have the right access to the directories

(*sudo chmod -R 755 /opt/pentaho)*

1. Go to the directory where you downloaded the *kettle-neo4j-remix-8.2.0.3-519-REMIX.zip*
2. Unzip the Pentaho Kettle-Neo4j Remix tool that you downloaded from step II into the */opt/pentaho* folder so that the /opt/pentaho/data-integration/…folder can be created: (unzip kettle-neo4j-remix-8.2.0.3-519-REMIX.zip -d /opt/pentaho/)
3. Copy the Maria-DB JDBC driver from the project you downloaded in section II step 3 (*gtas-neo4j-etl/drivers/mariadb-java-client-2.2.1*) to /opt/pentaho/data-integration/lib folder.
4. Reassign the ownership of this directory and its subdirectories to a user:

(*sudo chown -R <username> /opt/pentaho*)

1. Allow the user and the group of the user to have the right access to the copied files

(*sudo chmod -R 755 /opt/pentaho)*

1. *Important:* Copy the *.pentaho* folder with its contents fromthe *pdi-conf* folder of the project you downloaded in section II step 3 to the user home directory on your local or remote server. Without this metadata store, you cannot connect to Neo4j.
2. **Neo4j Community Edition Installation and Configuration**
3. **Installation**
4. Copy the downloaded neo4j archive file (*neo4j-community-3.5.3-unix.tar.gz* ) to /opt folder
5. Change your current directory to /opt: (*cd /opt*)
6. Unzip the neo4j-community-3.5.3-unix.tar.gz as follows:

(*sudo tar -xzf neo4j-community-3.5.3-unix.tar.gz*)

\*After successfully unzipping the archive file, you can remove the *neo4j-community-3.5.3-unix.tar.gz* to save a space on your disk

1. Assign the ownership of this directory and its subdirectories to a user:

(*sudo chown -R <username> /opt/neo4j-community-3.5.3*)

1. Allow the user and the group of the user to have the right access to the directories

(*sudo chmod -R 755 /opt/neo4j-community-3.5.3)*

1. **Configuration:**

Edit the configuration file as follows:

1. *sudo vi /opt/neo4j-community-3.5.3/conf/neo4j.conf*
2. Uncomment the following properties and change values of the properties when required:
   * + - dbms.active\_database=gtas.db
       - dbms.security.auth\_enabled=true
       - dbms.connectors.default\_listen\_address=0.0.0.0 (*for remote server only*)
       - dbms.connectors.default\_advertised\_address= *<server-ip-address>* for remote server, *localhost* for localhost deployment
       - dbms.connectors.default\_advertised\_address=localhost (*for localhost server only*)
       - dbms.connector.bolt.listen\_address=:7687
       - dbms.connector.http.listen\_address=:7474
3. Save the changes to the neo4j.conf file (:wq)
4. **Test your Installation and Configuration:**
5. To start Neo4j: /opt/neo4j-community-3.5.3/bin/neo4j start
6. For remote server installation: <http://server-ip-address:7474>
7. For localhost installation: <http://localhost:7474>
8. Default username=*neo4j*
9. Default password=*neo4j*
10. When prompted, change your password to *admin* for this deployment and sign in.
11. To Stop Neo4j: /opt/neo4j-community-3.5.3/bin/neo4j stop
12. **Running Neo4j as a service on Cent OS/RHEL/Amazon Linux** *(for server configuration only):*
13. Copy the *neo4j.service* file from the project that you downloaded in section II step 3 (*gtas-neo4j-etl/systemd/neo4j.service*) to /etc/systemd/system on the server.
14. Execute the following commands:

* *sudo touch /etc/systemd/system/neo4j.service*
* *sudo chmod 664 /etc/systemd/system/neo4j.service*
* *sudo systemctl enable /etc/systemd/system/neo4j.service*
* *sudo systemctl daemon-reload*

1. To test the service configuration:

* To start the service: *sudo systemctl start neo4j.service*
* Check the status of the service*: sudo systemctl status neo4j.service*
* To stop the service: *sudo systemctl stop neo4j.service*

The service should start and stop without errors

1. **Job, Transformation and Configuration files**

Create the following directories:

1. /gtas-neo4j-etl (*sudo mkdir /gtas-neo4j-etl*)
2. /gtas-neo4j-etl/config (*sudo mkdir /gtas-neo4j-etl/config* )
3. /gtas-neo4j-etl/job (*sudo mkdir /gtas-neo4j-etl/job* )
4. /gtas-neo4j-etl/job/temp (*sudo mkdir /gtas-neo4j-etl/job/temp*)
5. /gtas-neo4j-etl/log (*sudo mkdir /gtas-neo4j-etl/log*)
6. Assign the ownership of this directory and its subdirectories to a user:

(*sudo chown -R <username> /gtas-neo4j-etl/* )

1. Allow the user and the group of the user to have the right access to the directories

(*sudo chmod -R 755 /gtas-neo4j-etl/)*

1. Copy all the PDI jobs *(.kjb* files) and transformations (*.ktr* files) from the *job* folder of the project that you downloaded in section II step 3 to */gtas-neo4j-etl/job* folder.
2. Copy the *application.propertie*s file and *gtas-neo4j-config.properties* files from the *config* folder of the project that you downloaded in section II step 3 to /gtas-neo4j/config folder.
3. Assign ownership of the newly downloaded files and directories to the user:

(*sudo chown -R <username> /gtas-neo4j-etl*)

1. Allow the user and the group of the user to have the right access to the installed tools: (*sudo chmod -R 755 /gtas-neo4j-etl)*
2. If you followed the recommended default directory structure, and would like to use the default log-level (which is Minimal) and the default job execution interval (which is 60 seconds), there is nothing to change on the *application.properties* file. Otherwise edit the file using (*sudo vi /gtas-neo4j-etl/config/application.properties )and save (:wq)*
3. Edit the *gtas-neo4j-config.properties* file. (*sudo vi /gtas-neo4j-etl/config/gtas-neo4j-config.properties)*

*EXT\_VAR\_DATA\_LOADER\_NAME = GTAS-NEO4J DATA LOADER*

*EXT\_VAR\_GTAS\_DB\_HOST\_NAME = (localhost* or *hostname of remote database server*)

*EXT\_VAR\_GTAS\_DB\_NAME = gtas*

*EXT\_VAR\_GTAS\_DB\_PORT\_NUMBER = 3306*

*EXT\_VAR\_GTAS\_DB\_USER\_NAME = GTAS database user name*

*EXT\_VAR\_GTAS\_DB\_PASSWORD = GTAS database password*

*EXT\_VAR\_NEO4J\_DB\_BOLT\_PORT=7687*

*EXT\_VAR\_NEO4J\_DB\_BROWSER\_PORT=7474*

*EXT\_VAR\_NEO4J\_DB\_HOST\_NAME = localhost* for localhost or *hostname of remote database server*

*EXT\_VAR\_NEO4J\_DB\_USER\_NAME = neo4j*

*EXT\_VAR\_NEO4J\_DB\_PASSWORD = admin (unless you set a different password in Section IV)*

**See the Configuration file descriptions section for more configuration options.**

1. **Views**

Make sure the following views exist in your database. If they don’t exist in the GTAS database, use the sql scripts (included in the *sql* folder of the project that you downloaded in section II step 3) to create them.

1. **Views**
2. *neo4j\_vw*
3. *neo4j\_hit\_vw*
4. **Job Scheduler**
5. **Deploying Neo4j Job Scheduler as a service on Cent OS/RHEL/Amazon Linux** *(for remote server configuration only):*
6. Copy the *gtas-neo4j-job-scheduler-1.jar* file from the target folder of the project that you built on Section II step 4 to */gtas-neo4j-etl*
7. Adjust the access privileges: *sudo chmod 755 /gtas-neo4j-etl/gtas-neo4j-job-scheduler-1.jar*
8. Copy the *neo4j-scheduler.service* file from your project to /etc/systemd/system on the server.
9. Execute the following commands:

* *touch /etc/systemd/system/neo4j-scheduler.service*
* *chmod 664 /etc/systemd/system/neo4j-scheduler.service*
* *sudo systemctl enable /etc/systemd/system/neo4j-scheduler.service*
* *sudo systemctl daemon-reload*

1. To test the service configuration:

* Make sure your GTAS database is up and running
* Start your Neo4j database: *sudo systemctl start neo4j.service*
* To start the service: *sudo systemctl start neo4j-scheduler.service*
* Check the status of the service*: sudo systemctl status neo4j-scheduler.service*
* To stop the service: *sudo systemctl stop neo4-schedulerj.service*

The service should start and stop without errors

1. **Deploying Neo4j Job Scheduler** *(on localhost)*
2. Copy the *gtas-neo4j-job-scheduler-1.jar* file from the target folder of the project that you built on Section II step 4 to */gtas-neo4j-etl*
3. Adjust the access privileges: *sudo chmod 755 /gtas-neo4j-etl/gtas-neo4j-job-scheduler-1.jar*
4. Make sure your GTAS database is up and running
5. Start your Neo4j database: */opt/neo4j-community-3.5.3/bin/neo4j start*
6. Go to the gtas-neo4j-etl folder: *cd /gtas-neo4j-etl*
7. Start the scheduler: *java –jar gtas-neo4j-job-scheduler-1.jar*
8. **Configuration file descriptions**
9. **gtas-neo4j-config.properties**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Value** | **Description** |
| EXT\_VAR\_DATA\_LOADER\_NAME | GTAS-NEO4J DATA LOADER | Default name of the PDI ETL Job |
| EXT\_VAR\_GTAS\_DB\_HOST\_NAME | localhost | Default hostname of the server hosting the ETL job |
| EXT\_VAR\_GTAS\_DB\_NAME | gtas | The name of the GTAS MariaDB database |
| EXT\_VAR\_GTAS\_DB\_PORT\_NUMBER | 3306 | The default port of the GTAS MariaDB database |
| EXT\_VAR\_GTAS\_DB\_USER\_NAME | root | The default username of the GTAS MariaDB database |
| EXT\_VAR\_GTAS\_DB\_PASSWORD | admin | The default password of the GTAS MariaDB database |
| EXT\_VAR\_NEO4J\_DB\_HOST\_NAME | jdbc:neo4j:bolt://localhost:7687 | The database connection string of the Neo4j database |
| EXT\_VAR\_NEO4J\_DB\_USER\_NAME | neo4j | The default username of the Neo4j database |
| EXT\_VAR\_NEO4J\_DB\_PASSWORD | admin | The password for the Neo4j database |
| EXT\_VAR\_TEMP\_GTAS\_PASSENGER\_FULL\_FILE\_NAME | gtas-passenger-temp.txt | The temporary filename that the ETL job uses to process passenger data. |
| EXT\_VAR\_TEMP\_GTAS\_HITS\_FULL\_FILE\_NAME | gtas-hit-temp.txt | The temporary filename that the ETL job uses to process passenger hit data. |
| EXT\_VAR\_TEMP\_GTAS\_MESSAGE\_ID\_FILE | gtas-message-id-temp.txt | The temporary filename that the ETL job uses to process message\_ids. |
| EXT\_VAR\_GTAS\_MESSAGE\_STATUS\_TABLE | message\_status | The name of the table (in GTAS) where message status is updated updated |
| EXT\_VAR\_DOCUMENT\_BATCH\_SIZE | 5000 | Default batch size for merging the document node and the passenger-document relationship |
| EXT\_VAR\_PASSENGER\_BATCH\_SIZE | 5000 | Default batch size for merging passenger and flight nodes. |
| EXT\_VAR\_ADDRESS\_BATCH\_SIZE | 5000 | Default batch size for merging the Address node and the passenger-address relationship |
| EXT\_VAR\_CREDITCARD\_BATCH\_SIZE | 5000 | Default batch size for merging the CreditCard node and the passenger-CreditCard relationship |
| EXT\_VAR\_EMAIL\_BATCH\_SIZE | 5000 | Default batch size for merging the Email node and the passenger-email relationship |
| EXT\_VAR\_PHONE\_BATCH\_SIZE | 5000 | Default batch size for merging the Phone node and the passenger-phone relationship |
| EXT\_VAR\_PASSENGER\_HIT\_BATCH\_SIZE | 5000 | Default batch size for merging the Hit node and the passenger-hit relationship |
| EXT\_VAR\_FLIGHT\_HIT\_BATCH\_SIZE | 5000 | Default batch size for merging the flight-hit relationship. |
| EXT\_VAR\_GTAS\_MSG\_STATUS\_UPDATE\_BATCH\_SIZE | 5000 | Default batch size for updating the message\_status table in GTAS |
| EXT\_VAR\_GTAS\_MSG\_STATUS | NEO\_LOADED | The value used to update the message\_status table (ms\_status column) after the job finishes execution. |
| EXT\_VAR\_NEO4J\_MAX\_POOL\_SIZE | 10 | The maximum size of the connection pool for the Neo4j database |

1. **application.properties**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Value** | **Description** |
| opSystem | linux | The type of the operating system on which the Pentaho Data Integration tool runs |
| pdiDir | /opt/pentaho/data-integration/kitchen.sh | The location of the Pentaho Data Integration (kitchen) launcher file |
| jobDir | /gtas-neo4j-etl/job/gtas-to-neo-job.kjb | The director of the main job that starts the ETL process |
| logLevel | Minimal | The log level of PDI.  Valid Options:  **Nothing**: Don't show any output  **Error:** Only show errors.  **Minimal:** Only use minimal logging  **Basic:** This is the default basic logging level  **Detailed:** Give detailed logging output  **Debug:** For debugging purposes, very detailed output.  **Row level:** Logging at a row level, this can generate a lot of data. |
| logDir | /gtas-neo4j-etl/log/gtas-to-neo-job | The name of the directory and log file that PDI uses |
| configFilePropertyName | EXT\_ETL\_CONFIG\_FILE | The name of the parameter that the PDI uses to locate the gtas-neo4j-config.properties |
| configFile | /gtas-neo4j-etl/config/gtas-neo4j-config.properties | The location of the gtas-neo4j-config.properties |