Configuração do Hive 3.0 e Cloudera HUE em um ecossistema Hadoop!

Criado em Agosto/2018
Autor: Vinícius José Leodido Correia
linkedin: https://www.linkedin.com/in/vinileodido/

Primeiramente iremos instalar o Hive 3.0 e em seguida o HUE, um Self Service BI mantido pela Cloudera sobre licença Apache com uma UX bastante significativa.

Porém, antes de instalarmos o Hive alguns requisitos são necessários.

- Postgresql

Iremos instalar os clients do Postgresql e o Postgresql Server.

```
[hduser@hdpmaster ~]$ sudo yum list postgresql postgresql-server
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirror.globo.com
 * epel: mirror.globo.com
 * extras: mirror.globo.com
 * updates: mirror.globo.com
Available Packages
postgresql.i686                                  9.2.23-3.el7_4                                  base
postgresql.x86_64                                9.2.23-3.el7_4                                  base
postgresql-server.x86_64                         9.2.23-3.el7_4                                  base
[hduser@hdpmaster ~]$
```

Instalando o Postgres

```
[hduser@hdpmaster ~]$ sudo yum install postgresql postgresql-server -y
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirror.globo.com
 * epel: mirror.globo.com
 * extras: mirror.globo.com
 * updates: mirror.globo.com
Resolving Dependencies
--> Running transaction check
---> Package postgresql.x86_64 0:9.2.23-3.el7_4 will be installed
--> Processing Dependency: postgresql-libs(x86-64) = 9.2.23-3.el7_4 for package: postgresql-9.2.23-3.el7_4.x86_64
--> Processing Dependency: libpq.so.5()(64bit) for package: postgresql-9.2.23-3.el7_4.x86_64
---> Package postgresql-server.x86_64 0:9.2.23-3.el7_4 will be installed
--> Running transaction check
---> Package postgresql-libs.x86_64 0:9.2.23-3.el7_4 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================================================
 Package                          Arch                      Version                          Repository                  Size
================================================================================================================================
Installing:
 postgresql                       x86_64                    9.2.23-3.el7_4                   base                       3.0 M
 postgresql-server                x86_64                    9.2.23-3.el7_4                   base                       3.8 M
Installing for dependencies:
 postgresql-libs                  x86_64                    9.2.23-3.el7_4                   base                       234 k

Transaction Summary
================================================================================================================================
Install  2 Packages (+1 Dependent package)

Total download size: 7.0 M
Installed size: 33 M
Downloading packages:
(1/3): postgresql-libs-9.2.23-3.el7_4.x86_64.rpm                                                | 234 kB  00:00:00
(2/3): postgresql-9.2.23-3.el7_4.x86_64.rpm                                                     | 3.0 MB  00:00:00
(3/3): postgresql-server-9.2.23-3.el7_4.x86_64.rpm                                              | 3.8 MB  00:00:00
--------------------------------------------------------------------------------------------------------------------------------
Total                                                                   6.0 MB/s | 7.0 MB  00:00:01
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : postgresql-libs-9.2.23-3.el7_4.x86_64                                             1/3
  Installing : postgresql-9.2.23-3.el7_4.x86_64                                                  2/3
  Installing : postgresql-server-9.2.23-3.el7_4.x86_64                                           3/3
  Verifying  : postgresql-9.2.23-3.el7_4.x86_64                                                  1/3
  Verifying  : postgresql-server-9.2.23-3.el7_4.x86_64                                           2/3
  Verifying  : postgresql-libs-9.2.23-3.el7_4.x86_64                                             3/3
```

```
Installed:
  postgresql.x86_64 0:9.2.23-3.el7_4                    postgresql-server.x86_64 0:9.2.23-3.el7_4

Dependency Installed:
  postgresql-libs.x86_64 0:9.2.23-3.el7_4

Complete!
[hduser@hdpmaster ~]$
```

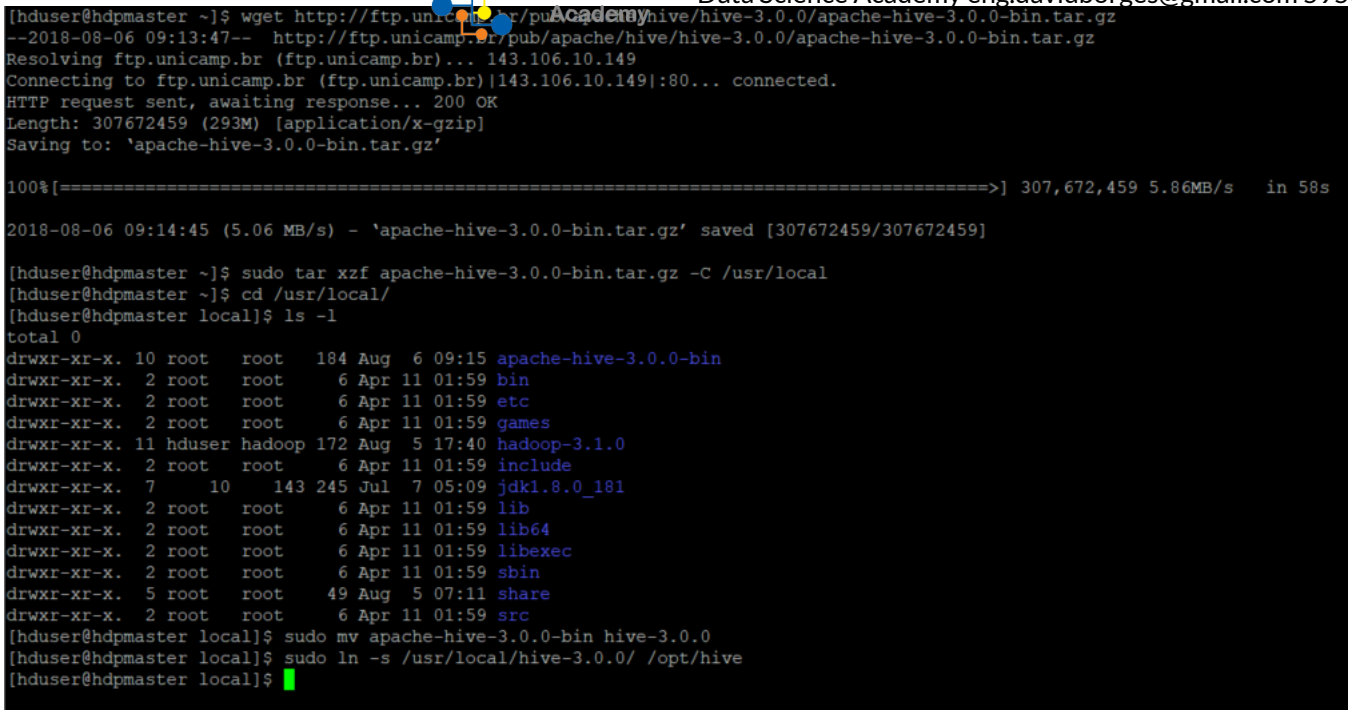
Inicialmente, está desabilitado no sistema:

```
[hduser@hdpmaster ~]$ sudo systemctl status postgresql
● postgresql.service - PostgreSQL database server
   Loaded: loaded (/usr/lib/systemd/system/postgresql.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
```

Iniciando o postgres:

```
hduser@hdpmaster:~
[hduser@hdpmaster ~]$ sudo systemctl enable postgresql
Created symlink from /etc/systemd/system/multi-user.target.wants/postgresql.service to /usr/lib/systemd/system/postgresql.service
.
[hduser@hdpmaster ~]$ sudo postgresql-setup initdb
Initializing database ... /var/lib/pgsql/data is not writeable by postgres
[hduser@hdpmaster ~]$ sudo chown -R postgres:postgres /var/lib/pgsql/data
[hduser@hdpmaster ~]$
[hduser@hdpmaster ~]$
[hduser@hdpmaster ~]$ sudo postgresql-setup initdb
Initializing database ... OK

[hduser@hdpmaster ~]$ sudo systemctl start postgresql
[hduser@hdpmaster ~]$ sudo systemctl status postgresql
● postgresql.service - PostgreSQL database server
   Loaded: loaded (/usr/lib/systemd/system/postgresql.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2018-08-06 09:02:00 -03; 13s ago
  Process: 1746 ExecStart=/usr/bin/pg_ctl start -D ${PGDATA} -s -o -p ${PGPORT} -w -t 300 (code=exited, status=0/SUCCESS)
  Process: 1741 ExecStartPre=/usr/bin/postgresql-check-db-dir ${PGDATA} (code=exited, status=0/SUCCESS)
 Main PID: 1749 (postgres)
   CGroup: /system.slice/postgresql.service
           ├─1749 /usr/bin/postgres -D /var/lib/pgsql/data -p 5432
           ├─1750 postgres: logger process
           ├─1752 postgres: checkpointer process
           ├─1753 postgres: writer process
           ├─1754 postgres: wal writer process
           ├─1755 postgres: autovacuum launcher process
           └─1756 postgres: stats collector process

Aug 06 09:01:59 hdpmaster systemd[1]: Starting PostgreSQL database server...
Aug 06 09:02:00 hdpmaster systemd[1]: Started PostgreSQL database server.
[hduser@hdpmaster ~]$
```

Efetue o download do Hive (Version 3.0.0) através de um dos mirrors do endereço https://hive.apache.org/downloads.html e descompacte na pasta /usr/local; em seguida iremos criar o symlink para o diretorio /opt/hive:



Edite as variaveis de ambiente para o Hive:

```
$ vi ~/.bash_profile

#Adicione as seguintes variaveis abaixo
#HIVE_HOME
export HIVE_HOME=/opt/hive
export PATH=$PATH:$HIVE_HOME/bin
export CLASSPATH=$CLASSPATH:$HADOOP_HOME/lib/*:.
export CLASSPATH=$HIVE_HOME/lib/*:.
```

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/.local/bin:$HOME/bin

export PATH


#JAVA_HOME
export JAVA_HOME=/opt/jdk
export PATH=$PATH:$JAVA_HOME/bin

#HADOOP_HOME
export HADOOP_HOME=/opt/hadoop
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin

#HIVE_HOME
export HIVE_HOME=/opt/hive
export PATH=$PATH:$HIVE_HOME/bin
export CLASSPATH=$CLASSPATH:$HADOOP_HOME/lib/*:.
export CLASSPATH=$HIVE_HOME/lib/*:.
```

Para verificar se as configurações foram aprovisionadas corretamente no sistema verifique com o seguinte comando:

```
$ hive --version
```

```
[hduser@hdpmaster ~]$ source ~/.bash_profile
[hduser@hdpmaster ~]$
[hduser@hdpmaster ~]$
[hduser@hdpmaster ~]$ hive --version
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hive-3.0.0/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-3.1.0/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/Static
LoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive 3.0.0
Git git://vgargwork.local/Users/vgarg/repos/hive.apache.master.latest -r ce61711a5fa54ab34fc74d86d521ecaeea6b072a
Compiled by vgarg on Fri May 18 11:38:33 PDT 2018
From source with checksum 81fcb93b608965ed7ac968bae1187fab
[hduser@hdpmaster ~]$
```

Antes de iniciarmos os serviços do Hive, precisaremos utilizar o script de criação do Metastore (Banco de dados que armazenará os metadados do Hive) no Postgresql;
Por que estamos fazendo este procedimento do Metastore para o Postgresql? O Metastore vem preparado para utilização através do Derby, um pequeno sistema de banco de dados local que é utilizado pelo Hive, porém, o mesmo só permitirá que uma sessão ativa seja instanciada para uso do serviço. Quando efetuamos a migração do Derby para um RBDMS como Oracle, MySql e Postgres, conseguimos utilizar mais de uma sessão simultaneamente.

Faça a instalação do driver jdbc Postgresql para a pasta de bibliotecas do Hive:

```
[hduser@hdpmaster ~]$ wget https://jdbc.postgresql.org/download/postgresql-42.2.4.jar
--2018-08-06 10:12:46--  https://jdbc.postgresql.org/download/postgresql-42.2.4.jar
Resolving jdbc.postgresql.org (jdbc.postgresql.org)... 174.143.35.228, 2001:4800:1501:1::228
Connecting to jdbc.postgresql.org (jdbc.postgresql.org)|174.143.35.228|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 814992 (796K) [application/java-archive]
Saving to: 'postgresql-42.2.4.jar'

100%[=======================================================================================>] 814,992      738KB/s   in 1.1s

2018-08-06 10:12:48 (738 KB/s) - 'postgresql-42.2.4.jar' saved [814992/814992]

[hduser@hdpmaster ~]$ sudo cp postgresql-42.2.4.jar /opt/hive/lib/
[hduser@hdpmaster ~]$
```

Localize o diretório de scripts Sql (Schema versão 3.0.0)do Metastore para o Postgres no diretório de instalação do Hive:

```
[hduser@hdpmaster ~]$ ls -l /opt/hive/scripts/metastore/upgrade/postgres/hive*schema*3.0.0*
-rw-r--r--. 1 root root 52500 May 15 18:42 /opt/hive/scripts/metastore/upgrade/postgres/hive-schema-3.0.0.postgres.sql
[hduser@hdpmaster ~]$
```

Crie o Metastore Database e a conta de usuario no Postgres:

```
$ sudo -u postgres psql

postgres=# CREATE USER hiveuser WITH PASSWORD 'mypassword';
postgres=# CREATE DATABASE metastore;
postgres=# GRANT ALL PRIVILEGES ON DATABASE metastore to hiveuser;
postgres=# \q
```

```
[hduser@hdpmaster ~]$ sudo -u postgres psql
could not change directory to "/home/hduser"
psql (9.2.23)
Type "help" for help.

postgres=# CREATE USER hiveuser WITH PASSWORD 'h1v3u$3r';
CREATE ROLE
postgres=# CREATE DATABASE metastore;
CREATE DATABASE
postgres=# \du
                             List of roles
 Role name |                   Attributes                   | Member of
-----------+------------------------------------------------+-----------
 hiveuser  |                                                | {}
 postgres  | Superuser, Create role, Create DB, Replication | {}

postgres=# GRANT ALL PRIVILEGES ON DATABASE metastore to hiveuser;
GRANT
postgres=# \q
[hduser@hdpmaster ~]$
```

Vamos editar o arquivo pg_hba.conf para que possamos acessar o host com os usuários que criaremos daqui em diante:

```
[hduser@hdpmaster ~]$ sudo ls -l /var/lib/pgsql/data/pg_hba.conf
[sudo] password for hduser:
-rw-------. 1 postgres postgres 4232 Aug  6 09:01 /var/lib/pgsql/data/pg_hba.conf
[hduser@hdpmaster ~]$ sudo vi /var/lib/pgsql/data/pg_hba.conf
```

```
#
# This file is read on server startup and when the postmaster receives
# a SIGHUP signal.  If you edit the file on a running system, you have
# to SIGHUP the postmaster for the changes to take effect.  You can
# use "pg_ctl reload" to do that.

# Put your actual configuration here
# ----------------------------------
#
# If you want to allow non-local connections, you need to add more
# "host" records.  In that case you will also need to make PostgreSQL
# listen on a non-local interface via the listen_addresses
# configuration parameter, or via the -i or -h command line switches.



# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     peer
# IPv4 local connections:
host    all             all             127.0.0.1/32            ident
host    all             all             0.0.0.0/0               md5
# IPv6 local connections:
host    all             all             ::1/128                 ident
# Allow replication connections from localhost, by a user with the
# replication privilege.
#local   replication     postgres                                peer
#host    replication     postgres        127.0.0.1/32            ident
#host    replication     postgres        ::1/128                 ident
~
~
~
~
```

Em seguida iremos editar o arquivo postgresql.conf :

```
#------------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------------------

# - Connection Settings -

listen_addresses = '*'
#listen_addresses = 'localhost'         # what IP address(es) to listen on;
                                        # comma-separated list of addresses;
                                        # defaults to 'localhost'; use '*' for all
                                        # (change requires restart)
#port = 5432                            # (change requires restart)
```

Reinicie o postgresql:

```
$ sudo systemctl restart postgresql
```

Entre novamente no postgresql com o usuario postgres e altere para que o database a ser utilizado seja o metastore; Em seguida, execute o script de criação do hive-schema-3.0.0.postgres.sql conforme comando abaixo:

```
$ sudo -u postgres psql

postgres=# \c metastore
metastore=#\i  /usr/local/hive-3.0.0/scripts/metastore/upgrade/postgres/hive-schema-3.0.0.postgres.sql
```

```
[hduser@hdpmaster ~]$ sudo -u postgres psql
could not change directory to "/home/hduser"
psql (9.2.23)
Type "help" for help.

postgres=# \c metastore
You are now connected to database "metastore" as user "postgres".
metastore=# \i /usr/local/hive-3.0.0/scripts/metastore/upgrade/postgres/hive-schema-3.0.0.postgres.sql
```

Agora precisamos conceder permissão para todas os objetos do metastore para o usuário que utilizaremos "hiveuser":

```
$ sudo -u postgres psql

postgres=# \c metastore
metastore=#  \pset tuples_only on
metastore=# \o /tmp/grant-privs
metastore=#    SELECT 'GRANT SELECT, INSERT, UPDATE, DELETE ON "' || schemaname || '". "' || tablename ||'" TO hiveuser ;'
metastore=#    FROM pg_tables
metastore=#    WHERE tableowner = CURRENT_USER and schemaname = 'public';
metastore=# \o
metastore=# \pset tuples_only off
metastore=# \i /tmp/grant-privs
```

```
[root@hdpmaster ~]# sudo -u postgres psql
could not change directory to "/root"
psql (9.2.23)
Type "help" for help.

postgres=# \c metastore
You are now connected to database "metastore" as user "postgres".
metastore=# \pset tuples_only on
Showing only tuples.
metastore=# \o /tmp/grant-privs
metastore=#     SELECT 'GRANT SELECT, INSERT, UPDATE, DELETE ON "' || schemaname || '". "' || tablename ||'" TO hiveuser ;'
metastore-#     FROM pg_tables
metastore-#     WHERE tableowner = CURRENT_USER and schemaname = 'public';
metastore=# \o
metastore=# \pset tuples_only off
Tuples only is off.
metastore=# \i /tmp/grant-privs
```

Desconecte do postgresql:

```
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
GRANT
metastore=# \q
[root@hdpmaster ~]#
```

Vamos editar o arquivo hive-site.xml :

```
[hduser@hdpmaster ~]$ cd /opt/hive/conf/
[hduser@hdpmaster conf]$ ls -l
total 328
-rw-r--r--. 1 root root   1596 May 15 18:42 beeline-log4j2.properties.template
-rw-r--r--. 1 root root 298702 May 18 15:38 hive-default.xml.template
-rw-r--r--. 1 root root   2365 May 15 18:42 hive-env.sh.template
-rw-r--r--. 1 root root   2274 May 15 18:42 hive-exec-log4j2.properties.template
-rw-r--r--. 1 root root   3086 May 15 18:42 hive-log4j2.properties.template
-rw-r--r--. 1 root root   2060 May 15 18:42 ivysettings.xml
-rw-r--r--. 1 root root   3558 May 15 18:42 llap-cli-log4j2.properties.template
-rw-r--r--. 1 root root   7163 May 15 18:42 llap-daemon-log4j2.properties.template
-rw-r--r--. 1 root root   2662 May 15 18:42 parquet-logging.properties
[hduser@hdpmaster conf]$ vi hive-site.xml
```

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?><!--
  Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License.  You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->

<configuration>
  <property>
    <name>javax.jdo.option.ConnectionURL</name>
    <value>jdbc:postgresql://192.168.1.200:5432/metastore</value>
  </property>

  <property>
    <name>javax.jdo.option.ConnectionDriverName</name>
    <value>org.postgresql.Driver</value>
  </property>

  <property>
```

```
  <name>javax.jdo.option.ConnectionUserName</name>
  <value>hiveuser</value>
 </property>

 <property>
  <name>javax.jdo.option.ConnectionPassword</name>
  <value>h1v3m3t4$t0r3</value>
 </property>

 <property>
  <name>hive.server2.thrift.port</name>
  <value>10000</value>
  <description>HiveServer2 thrift interface - TCP port number to listen on, default 10000</description>
 </property>

 <property>
  <name>hive.server2.thrift.bind.host</name>
  <value>localhost</value>
  <description>HiveServer2 bind host</description>
 </property>

 <property>
  <name>datanucleus.autoCreateSchema</name>
  <value>false</value>
 </property>

 <property>
  <name>hive.metastore.schema.verification</name>
  <value>true</value>
 </property>
</configuration>
```

Verificando se o Hive está funcionando:

```
[hduser@hdpmaster ~]$ hive --version
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hive-3.0.0/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-3.1.0/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive 3.0.0
Git git://vgargwork.local/Users/vgarg/repos/hive.apache.master.latest -r ce61711a5fa54ab34fc74d86d521ecaeea6b072a
Compiled by vgarg on Fri May 18 11:38:33 PDT 2018
From source with checksum 81fcb93b608965ed7ac968bae1187fab
[hduser@hdpmaster ~]$
```

Iniciando o schematool do Hive (Atenção! Como executamos o sql de criação do Hive manualmente, alguns Warnings poderão ser exibidos em tela!):

```
[hduser@hdpmaster ~]$ ls -l $HIVE_HOME/bin
total 44
-rwxr-xr-x. 1 root root   881 May 15 18:42 beeline
drwxr-xr-x. 3 root root  4096 Aug  6 09:15 ext
-rwxr-xr-x. 1 root root 10026 May 15 18:42 hive
-rwxr-xr-x. 1 root root  1900 May 15 18:42 hive-config.sh
-rwxr-xr-x. 1 root root   885 May 15 18:42 hiveserver2
-rwxr-xr-x. 1 root root   880 May 15 18:42 hplsql
-rwxr-xr-x. 1 root root  3064 May 15 18:42 init-hive-dfs.sh
-rwxr-xr-x. 1 root root   832 May 15 18:42 metatool
-rwxr-xr-x. 1 root root   884 May 15 18:42 schematool
[hduser@hdpmaster ~]$
[hduser@hdpmaster ~]$
[hduser@hdpmaster ~]$ schematool -dbType postgres -initSchema
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hive-3.0.0/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-3.1.0/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Metastore connection URL:        jdbc:postgresql://192.168.1.200:5432/metastore
Metastore Connection Driver :    org.postgresql.Driver
Metastore connection User:       hiveuser
Starting metastore schema initialization to 3.0.0
Initialization script hive-schema-3.0.0.postgres.sql









Initialization script completed
schemaTool completed
[hduser@hdpmaster ~]$
```

Iremos instanciar o serviço do HiveServer2:

Importante averiguar nos arquivos de configuração do Hive, se as propriedades do Thrift foram adicionadas ao arquivo hive-site.xml:

```
<property>
   <name>hive.server2.thrift.port</name>
   <value>10000</value>
   <description>TCP port number to listen on, default 10000</description>
</property>
```

```
[hduser@hdpmaster ~]$ ls -l $HIVE_HOME/bin
total 44
-rwxr-xr-x. 1 root root   881 May 15 18:42 beeline
drwxr-xr-x. 3 root root  4096 Aug  6 09:15 ext
-rwxr-xr-x. 1 root root 10026 May 15 18:42 hive
-rwxr-xr-x. 1 root root  1900 May 15 18:42 hive-config.sh
-rwxr-xr-x. 1 root root   885 May 15 18:42 hiveserver2
-rwxr-xr-x. 1 root root   880 May 15 18:42 hplsql
-rwxr-xr-x. 1 root root  3064 May 15 18:42 init-hive-dfs.sh
-rwxr-xr-x. 1 root root   832 May 15 18:42 metatool
-rwxr-xr-x. 1 root root   884 May 15 18:42 schematool
[hduser@hdpmaster ~]$
[hduser@hdpmaster ~]$
[hduser@hdpmaster ~]$
[hduser@hdpmaster ~]$ nohup hive --service hiveserver2 &
[1] 4092
[hduser@hdpmaster ~]$ nohup: ignoring input and appending output to 'nohup.out'

[hduser@hdpmaster ~]$
```

Vamos verificar se o hiveserver2 está em execução no background do sistema:

```
[hduser@hdpmaster ~]$ jps
4247 Jps
1928 SecondaryNameNode
1708 NameNode
4092 RunJar
2142 ResourceManager
[hduser@hdpmaster ~]$ ps -aef | grep 4092
hduser    4092  3665 28 16:10 pts/0    00:00:12 /opt/jdk/bin/java -Dproc_jar -Dproc_hiveserver2 -Dlog4j.config
line.UnsupportedTerminal -Dyarn.log.dir=/opt/hadoop/logs -Dyarn.log.file=hadoop.log -Dyarn.home.dir=/opt/hadoo
p.log.file=hadoop.log -Dhadoop.home.dir=/opt/hadoop -Dhadoop.id.str=hduser -Dhadoop.root.logger=INFO,console -
ve-service-3.0.0.jar org.apache.hive.service.server.HiveServer2
hduser    4259  3665  0 16:11 pts/0    00:00:00 grep --color=auto 4092
[hduser@hdpmaster ~]$
```

Antes de testarmos o beeline, será necessário a alteração de algumas propriedades no arquivo core-site.xml nos arquivos de configuração do hadoop:

```
$ vi $HADOOP_CONF_DIR/core-site.xml

  <property>
    <name>hadoop.proxyuser.<user_connected_to_hdfs>.hosts</name>
    <value>*</value>
  </property>
  <property>
    <name>hadoop.proxyuser.<user_connected_to_hdfs>.groups</name>
    <value>*</value>
  </property>


<property>
     <name>hadoop.proxyuser.hduser.hosts</name>
     <value>*</value>
</property>
<property>
     <name>hadoop.proxyuser.hduser.groups</name>
     <value>*</value>
</property>


#Interrompa os serviços do Hadoop
$ stop-dfs.sh; stop-yarn.sh
```

```
#Reinicie novamente os serviços
$ start-dfs.sh; start-yarn.sh
```

Vamos testar o beeline:

```
$ beeline

beeline> !connect jdbc:hive2://hdpmaster:10000 scott tiger org.apache.hive.jdbc.HiveDriver
```

```
[hduser@hdpmaster ~]$ beeline
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hive-3.0.0/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-3.1.0/share/hadoop/common/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Beeline version 3.0.0 by Apache Hive
beeline> !connect jdbc:hive2://hdpmaster:10000 scott tiger org.apache.hive.jdbc.HiveDriver
Connecting to jdbc:hive2://hdpmaster:10000
Connected to: Apache Hive (version 3.0.0)
Driver: Hive JDBC (version 3.0.0)
Transaction isolation: TRANSACTION_REPEATABLE_READ
0: jdbc:hive2://hdpmaster:10000> show tables;
INFO  : Compiling command(queryId=hduser_20180806162937_975f84c3-50e4-4052-8a09-a74b70fdccec): show tables
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Semantic Analysis Completed
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:tab_name, type:string, comment:from deserializer)], properties:null)
INFO  : Completed compiling command(queryId=hduser_20180806162937_975f84c3-50e4-4052-8a09-a74b70fdccec); Time taken: 0.8 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hduser_20180806162937_975f84c3-50e4-4052-8a09-a74b70fdccec): show tables
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hduser_20180806162937_975f84c3-50e4-4052-8a09-a74b70fdccec); Time taken: 0.038 seconds
INFO  : OK
INFO  : Concurrency mode is disabled, not creating a lock manager
+-----------+
| tab_name  |
+-----------+
+-----------+
No rows selected (1.154 seconds)
0: jdbc:hive2://hdpmaster:10000> !quit
Closing: 0: jdbc:hive2://hdpmaster:10000
[hduser@hdpmaster ~]$
```

Instalando as dependencias do HUE

como usuário root, instale os seguintes pacotes

```
$ yum install ant
$  yum install gcc g++
$  yum install make
$ yum install python-devel.x86_64
$ yum groupinstall "Development Tools"
$ yum install krb5-devel
$ yum install libxslt-devel libxml2-devel
$ yum install mysql-devel.x86_64

$ yum install ncurses-devel zlib-devel texinfo gtk+-devel gtk2-devel
$ yum install qt-devel tcl-devel tk-devel kernel-headers kernel-devel
```

```
$ yum install yum install gmp-devel.x86_64
$ yum install sqlite-devel.x86_64
$ yum install cyrus-sasl.x86_64

$ yum install postfix system-switch-mail cyrus-imapd cyrus-plain cyrus-md5 cyrus-utilsyum install postfix system-switch-mail cyrus-imapd cyrus-plain cyrus-md5 cyrus-utils

$ yum install memcached-devel.x86_64
$ yum install libevent libevent-devel
$ yum install postfix
$ yum install cyrus-sasl
$ yum install cyrus-imapd
$ yum install openldap-devel
```

Atenção:
Como o pacote ant instala o openJdk e realiza um "bypass" na instalação que você efetuou no sistema e aprovisionou nas variaveis de ambiente, liste as variaveis simbolicas que estão no diretório /etc/alternatives/jav* em um arquivo de bkp e em seguida, as remova do diretorio;

```
[hduser@hdpmaster ~]$ sudo ls -l /etc/alternatives/jav* >> alternatives_java.txt
[hduser@hdpmaster ~]$ cat alternatives_java.txt
lrwxrwxrwx. 1 root root 73 Aug  6 17:13 /etc/alternatives/java -> /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.181-3.b13.el7_5.x86_64/jre/bin/java
lrwxrwxrwx. 1 root root 77 Aug  6 17:13 /etc/alternatives/java.1.gz -> /usr/share/man/man1/java-java-1.8.0-openjdk-1.8.0.181-3.b13.el7_5.x86_64.1.gz
lrwxrwxrwx. 1 root root 70 Aug  6 17:13 /etc/alternatives/javac -> /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.181-3.b13.el7_5.x86_64/bin/javac
lrwxrwxrwx. 1 root root 78 Aug  6 17:13 /etc/alternatives/javac.1.gz -> /usr/share/man/man1/javac-java-1.8.0-openjdk-1.8.0.181-3.b13.el7_5.x86_64.1.gz
lrwxrwxrwx. 1 root root 72 Aug  6 17:13 /etc/alternatives/javadoc -> /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.181-3.b13.el7_5.x86_64/bin/javadoc
lrwxrwxrwx. 1 root root 80 Aug  6 17:13 /etc/alternatives/javadoc.1.gz -> /usr/share/man/man1/javadoc-java-1.8.0-openjdk-1.8.0.181-3.b13.el7_5.x86_64.1.gz
lrwxrwxrwx. 1 root root 70 Aug  6 17:13 /etc/alternatives/javah -> /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.181-3.b13.el7_5.x86_64/bin/javah
lrwxrwxrwx. 1 root root 78 Aug  6 17:13 /etc/alternatives/javah.1.gz -> /usr/share/man/man1/javah-java-1.8.0-openjdk-1.8.0.181-3.b13.el7_5.x86_64.1.gz
lrwxrwxrwx. 1 root root 70 Aug  6 17:13 /etc/alternatives/javap -> /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.181-3.b13.el7_5.x86_64/bin/javap
lrwxrwxrwx. 1 root root 78 Aug  6 17:13 /etc/alternatives/javap.1.gz -> /usr/share/man/man1/javap-java-1.8.0-openjdk-1.8.0.181-3.b13.el7_5.x86_64.1.gz
lrwxrwxrwx. 1 root root 60 Aug  6 17:13 /etc/alternatives/java_sdk -> /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.181-3.b13.el7_5.x86_64
lrwxrwxrwx. 1 root root 60 Aug  6 17:13 /etc/alternatives/java_sdk_1.8.0 -> /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.181-3.b13.el7_5.x86_64
lrwxrwxrwx. 1 root root 68 Aug  6 17:13 /etc/alternatives/java_sdk_1.8.0_exports -> /usr/lib/jvm-exports/java-1.8.0-openjdk-1.8.0.181-3.b13.el7_5.x86_64
lrwxrwxrwx. 1 root root 60 Aug  6 17:13 /etc/alternatives/java_sdk_1.8.0_openjdk -> /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.181-3.b13.el7_5.x86_64
lrwxrwxrwx. 1 root root 68 Aug  6 17:13 /etc/alternatives/java_sdk_1.8.0_openjdk_exports -> /usr/lib/jvm-exports/java-1.8.0-openjdk-1.8.0.181-3.b13.el7_5.x86_64
lrwxrwxrwx. 1 root root 68 Aug  6 17:13 /etc/alternatives/java_sdk_exports -> /usr/lib/jvm-exports/java-1.8.0-openjdk-1.8.0.181-3.b13.el7_5.x86_64
lrwxrwxrwx. 1 root root 60 Aug  6 17:13 /etc/alternatives/java_sdk_openjdk -> /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.181-3.b13.el7_5.x86_64
lrwxrwxrwx. 1 root root 68 Aug  6 17:13 /etc/alternatives/java_sdk_openjdk_exports -> /usr/lib/jvm-exports/java-1.8.0-openjdk-1.8.0.181-3.b13.el7_5.x86_64
[hduser@hdpmaster ~]$ sudo rm /etc/alternatives/jav*
[hduser@hdpmaster ~]$ sudo ls -l /etc/alternatives/jav*
ls: cannot access /etc/alternatives/jav*: No such file or directory
[hduser@hdpmaster ~]$
```

Execute novamente um source em suas variáveis de ambiente (Para que o sistema confirma novamente qual o Jdk está aprovisionado):

```
$ source ~/.bash_profile
```

Por fim, execute o comando java -version a fim de verificar qual o Jdk realmente seu sistema estará utilizando:

```
$ java -version
```

```
[hduser@hdpmaster ~]$ java -version
java version "1.8.0_181"
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)
[hduser@hdpmaster ~]$
```

Em seguida, realize a instalação do Maven em seu sistema:

https://maven.apache.org/download.cgi

```
$ wget http://mirror.nbtelecom.com.br/apache/maven/maven-3/3.5.4/binaries/apache-maven-3.5.4-bin.tar.gz
```

```
$ sudo tar xzf apache-maven-3.5.4-bin.tar.gz -C /usr/local

$ sudo mv  apache-maven-3.5.4-bin maven-3.5.4
$ sudo ln -s /usr/local/maven-3.5.4 /opt/maven


$ vi ~/.bash_profile

#MAVEN_HOME
export MAVEN_HOME=/opt/maven
export PATH=$PATH:$MAVEN_HOME/bin
```

Para averiguar se a instalação ocorreu com sucesso:

```
$ mvn -version
```

```
[hduser@hdpmaster ~]$ mvn -version
Apache Maven 3.5.4 (1edded0938998edf8bf061f1ceb3cfdeccf443fe; 2018-06-17T15:33:14-03:00)
Maven home: /opt/maven
Java version: 1.8.0_181, vendor: Oracle Corporation, runtime: /usr/local/jdk1.8.0_181/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "3.10.0-862.9.1.el7.x86_64", arch: "amd64", family: "unix"
[hduser@hdpmaster ~]$
```

Continuaremos instalando mais alguns pacotes necessários:

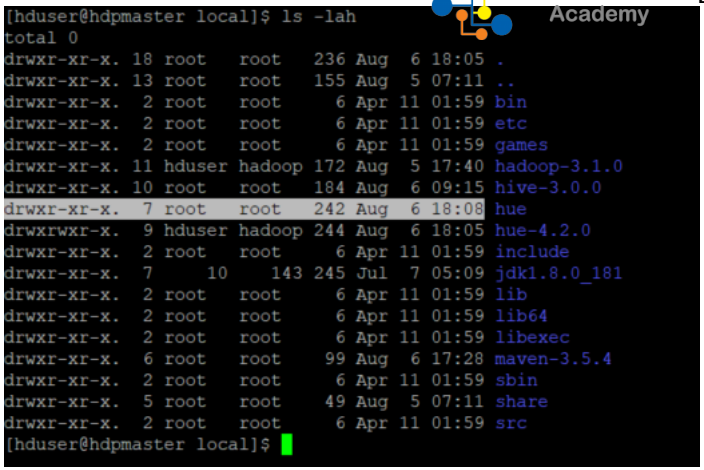
```
$ sudo yum install asciidoc cyrus-sasl-devel cyrus-sasl-gssapi cyrus-sasl-plain
$ sudo yum install krb5-devel libffi-devel libxml2-devel libxslt-devel openldap-devel gmp-devel sqlite-devel
$ sudo yum install python-devel
$ sudo yum install  mysql mysql-devel


$ sudo yum install libtidy
$ sudo yum install openssl-devel
```

Por fim, iremos efetuar a instalação do HUE;

Acesse a página da distribuição em http://gethue.com

```
$ wget https://www.dropbox.com/s/0rhrlnjmyw6bnfc/hue-4.2.0.tgz

$ sudo tar xzf hue-4.2.0.tgz -C /usr/local

$ cd /usr/local
$ pushd hue-4.2.0/
$ sudo make install

# Após a instalação
$ popd
```

Iremos alterar as permissões de acesso para a pasta que foi criada:




```
$ sudo useradd hue
$ sudo chown -R hue:hue /usr/local/hue
```

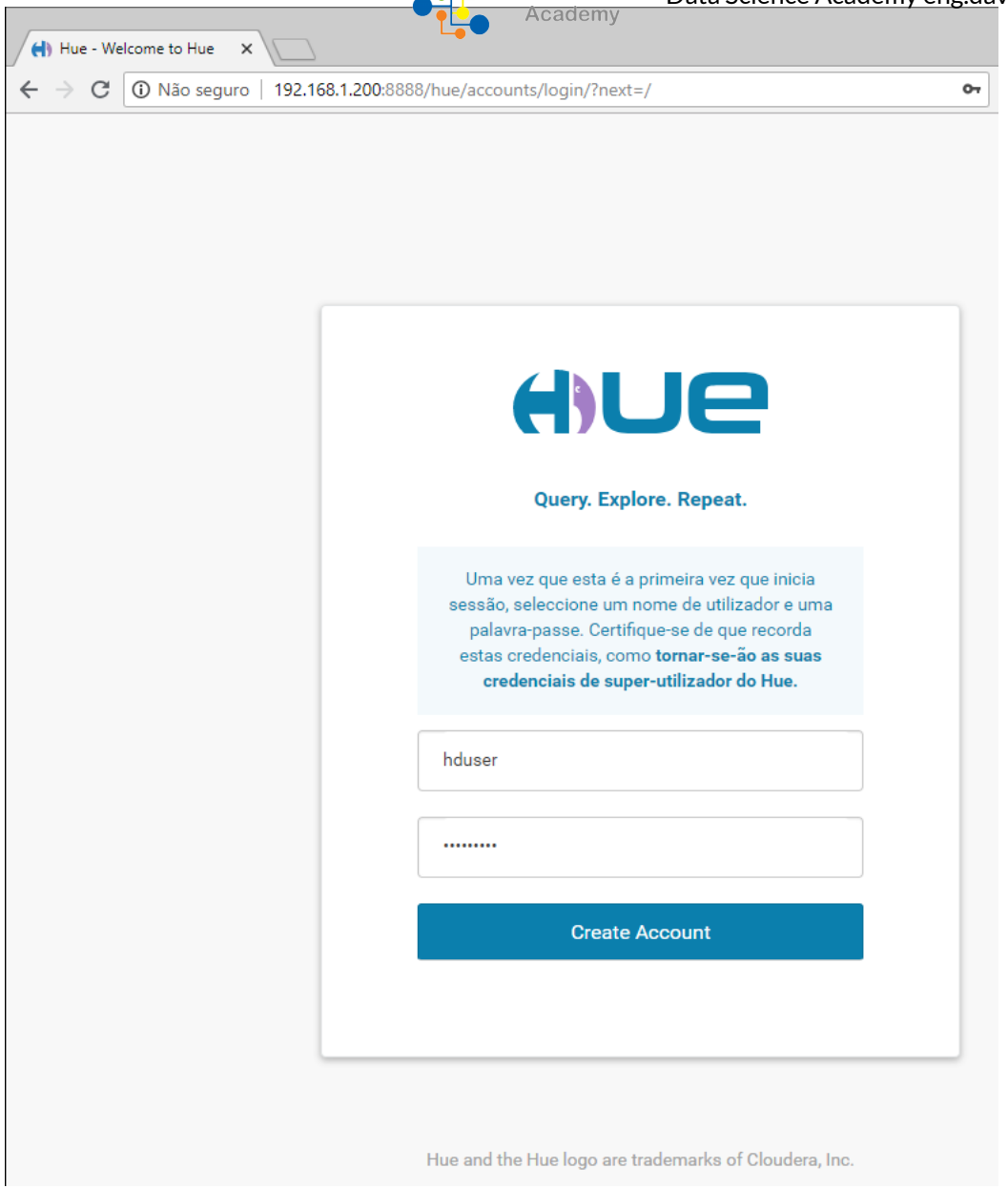
Remova a pasta que utilizamos para realizar o build do HUE:

```
$ sudo rm -r /usr/local/hue-4.2.0

$ sudo ln -s /usr/local/hue /opt/hue

$ vi ~/.bash_profile
```

Iremos inicializar o HUE:

```
$ sudo $HUE_HOME/build/env/bin/supervisor
```

Acesse o browser de sua preferência, com o endereço do Host onde está instalado o HUE utilizando a porta :8888



Efetue Logout do HUE, e vamos para o próximo passo onde iremos migrar do Sqlite para um RBDMS.

Interrompa o serviço do HUE, com o seguinte comando:

```
$ ps -f -u hue
```

```
#Localize o PID do processo runcherrypyserver e realize um kill com usuário root
#Repita por pelo menos 4 vezes.
```

Importante!
O Hue é inicializado através do comando supervisor [sudo $HUE_HOME/build/env/bin/supervisor], o qual desempenha a função de monitorar se o processo runcherrypyserver está em execução e caso o mesmo não esteja localizado na pilha de processos do sistema, o Supervisor instancia um novo processo do runcherrypyserver;
Porém, se o processo runcherrypyserver for identificado "abort" ou "kill" pelo supervisor em um curto intervalo de tempo, por sucedidas vezes (x4) o mesmo irá interromper as tentativas de inicialização do processo runcherrypyserver e o HUE terá seus serviços paralisados, até que seja executado novamente o processo do Supervisor ou sejam corrigidos os Logs de Erros (caso seja o cenário).

```
[hduser@hdpmaster ~]$ ps -f -u hue
UID        PID  PPID  C STIME TTY          TIME CMD
hue      31811 31539 12 19:32 pts/0    00:00:01 /usr/local/hue/build/env/bin/python2.7 /usr/local/hue/build/env/bin/hue runcherrypyserver
[hduser@hdpmaster ~]$ sudo kill 31811
[hduser@hdpmaster ~]$ ps -f -u hue
UID        PID  PPID  C STIME TTY          TIME CMD
[1]+  Exit 1                  sudo $HUE_HOME/build/env/bin/supervisor
[hduser@hdpmaster ~]$
```

Configurações do HUE:

```
$HUE_HOME/desktop/conf
```

Criando o acesso a um banco para os metadados do HUE:

Assim como vimos anteriormente (Hive utilizando o Derby como BD local para seus metadados), o Hue armazena seus metadados também em um repositório local com o Sqlite; Para finalidades de testes, seu uso é recomendado, porém, para uma abordagem em que serão utilizadas diversas conexões simultâneas e com diversos usuários acessando o HUE, efetuar uma migração para um RBDMS como Oracle, MySql ou Postgresql, se faz necessário, conforme recomendação da Cloudera.

O passo a passo poderá ser encontrado nos seguintes endereços (mas deixei as etapas que realizei descritas em sequência):
http://cloudera.github.io/hue/latest/admin-manual/manual.html#configuring-to-access-another-database
https://community.hortonworks.com/questions/399/how-to-move-hue-database-from-default-sqlite-datab.html

```
$ sudo -u postgres psql

postgres=# CREATE USER hueuser WITH password 'hu3u$3r';
postgres=# CREATE DATABASE hue;
postgres=# GRANT ALL PRIVILEGES ON DATABASE hue TO hueuser;
```

Crie um diretorio de backup; Efetue o dump dos dados do HUE para este diretório através do seguinte comando:

```
$ sudo $HUE_HOME/build/env/bin/hue dumpdata > ~/bkp_conf/hue_db_dump.json
```

Efetue também, um backup do arquivo de configurações do HUE antes de efetuarmos modificações no mesmo:

```
$ sudo cp $HUE_HOME/desktop/conf/hue.ini  ~/bkp_conf/
```

Abra o arquivo hue.ini no diretorio de configurações do HUE [$HUE_HOME/desktop/conf/hue.ini] e altere as seguintes propriedades:

```
$ sudo vi $HUE_HOME/desktop/conf/hue.ini

#Localize os parametros de configurações [[database]] e altere informando as respectivas credenciais de seu usuário

   engine=postgresql_psycopg2
   host=192.168.1.200 #Host do Postgresql
   port=5432
   user=hueuser
   password=hu3u$3r
   name=hue
```

```
  schema=public
```

Por fim, as alterações estarão conforme abaixo

```
 [[database]]
   # Database engine is typically one of:
   # postgresql_psycopg2, mysql, sqlite3 or oracle.
   #
   # Note that for sqlite3, 'name', below is a path to the filename. For other backends, it is the database name
   # Note for Oracle, options={"threaded":true} must be set in order to avoid crashes.
   # Note for Oracle, you can use the Oracle Service Name by setting "host=" and "port=" and then "name=<host>:<port>/<service_name>".
   # Note for MariaDB use the 'mysql' engine.
   ## engine=sqlite3
   engine=postgresql_psycopg2
   host=192.168.1.200
   port=5432
   user=hueuser
   password=hu3u$3r
   # conn_max_age option to make database connection persistent value in seconds
   #  https://docs.djangoproject.com/en/1.9/ref/databases/#persistent-connections
   ## conn_max_age=0
   # Execute this script to produce the database password. This will be used when 'password' is not set.
   ## password_script=/path/script
   ## name=desktop/desktop.db
   name=hue
   ## options={}
   # Database schema, to be used only when public schema is revoked in postgres
   ## schema=public

 # Configuration options for specifying the Desktop session.
 # For more info, see  https://docs.djangoproject.com/en/1.4/topics/http/sessions/
```

Importante! Se realmente for utilizar o Postgresql como repositório para os metadados do HUE, a instalação da biblioteca postgresql_psycopg2 do Python se faz necessário pois a base do HUE é um servidor web em Django!

```
$ sudo pip install psycopg2
$ sudo pip install psycopg2-binary
```

```
[hduser@hdpmaster ~]$ sudo pip install psycopg2
Collecting psycopg2
  Downloading https://files.pythonhosted.org/packages/7c/e6/d5161798a5e8900f24216cb730f2c2be5e4758a80d35c8588306831c0c99/psycopg2-2.7.5-cp27-cp27mu-manylinux1_x86_64.whl (2.7MB)
    100% |████████████████████████████████| 2.7MB 1.7MB/s
Installing collected packages: psycopg2
Successfully installed psycopg2-2.7.5
[hduser@hdpmaster ~]$
```

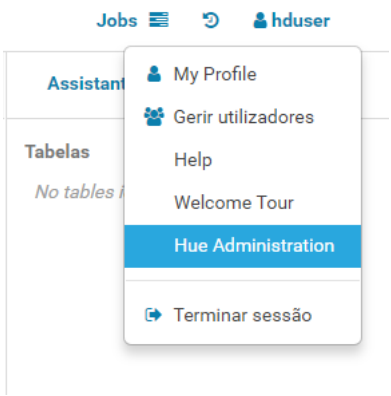
Execute os seguintes comandos abaixo (Importante que o HUE esteja com os serviços interrompidos!)

```
$ sudo $HUE_HOME/build/env/bin/hue syncdb --noinput
$ sudo $HUE_HOME/build/env/bin/hue migrate
$ sudo $HUE_HOME/build/env/bin/hue loaddata ~/bkp_conf/hue_db_dump.json

#Inicie novamente o Hue pelo supervisor
$ sudo nohup $HUE_HOME/build/env/bin/supervisor &
```

Pronto, efetuamos a migração do sqlite para o Postgresql!

Ao efetuar acesso novamente no HUE com o seu usuário cadastrado, exibirão diversas mensagens de erros; Isto ocorre pois o ambiente não está devidamente mapeado de acordo com as configurações que vieram em default no arquivo $HUE_HOME/desktop/conf/hue.ini

Para conferir as mensagens de Log, acesso a opção "HUE Administration" bem próximo do canto superior direito, onde estão as informações do usuário que está logado no momento.



Conforme a tela abaixo, vemos quais respectivas variáveis necessitam de alteração / atualização no arquivo de configuração:

Assistente de início rápido - Hue™ 4.2.0 - Query. Explore. Repeat.

Passo 1: Verificar configuração

Passo 2: Exemplos

Passo 3: Utilizadores

Passo 4: Ir!

A verificar configuração actual

Ficheiros de configuração localizados em `/usr/local/hue/desktop/conf`

Possíveis configurações incorrectas detectadas. Repare e reinicie o Hue.

`hadoop.hdfs_clusters.default.webhdfs_url`	Valor actual: `http://localhost:50070/webhdfs/v1` Não foi possível aceder à raiz do sistema de ficheiros
`Resource Manager`	Failed to contact an active Resource Manager: YARN RM returned a failed response: HTTPConnectionPool(host='localhost', port=8088): Max retries exceeded with url: /ws/v1/cluster/apps?user=hduser&user.name=hue&doAs=hduser (Caused by NewConnectionError(': Failed to establish a new connection: [Errno 111] Connecti refused',))
`Hive`	Failed to access Hive warehouse: /user/hive/warehouse
`HBase Browser`	The application won't work without a running HBase Thrift Server v1.
`Impala`	No available Impalad to send queries to.
`Oozie Editor/Dashboard`	A aplicação não funciona sem um servidor Oozie em execução
`Pig Editor`	A aplicação não funciona sem um servidor Oozie em execução
`Hadoop Security: Sentry Service`	Failed to connect to Sentry API (version 1).
`Hadoop Security: Sentry Service`	Failed to connect to Sentry API (version 2).
`Spark`	The app won't work without a running Livy Spark Server

Throubleshooting nas configurações do HUE

1. Habilitando a restapi webhdfs
   Para corrigirmos o primeiro Warning, necessitaremos atuar diretamente nos arquivos de configuração do Hadoop em seu ecossistema.
   Localize o arquivo hdfs-site.xml em seu diretório de configuração do Hadoop e acrescente a seguinte propriedade:

```
        <property>
           <name>dfs.webhdfs.enabled</name>
           <value>true</value>
        </property>
```

   Reinicie os serviços do seu HDFS;

   Acesse ao arquivo de configuração do HUE [$HUE_HOME/desktop/conf/hue.ini] e localize a seguinte tag [[hdfs_clusters]]
   Altere para a porta correta do seu NameNode (Estou utilizando a versão 3.1.0, portanto a porta está situada na :9870)

```
[[hdfs_clusters]]
  # HA support by using HttpFs

  [[[default]]]
    # Enter the filesystem uri
    fs_defaultfs=hdfs://localhost:8020
```

```
# NameNode logical name.
## logical_name=

# Use WebHdfs/HttpFs as the communication mechanism.
# Domain should be the NameNode or HttpFs host.
# Default port is 14000 for HttpFs.
## webhdfs_url=http://localhost:50070/webhdfs/v1
webhdfs_url=http://localhost:9870/webhdfs/v1
```

2. Alterando para a porta correta do Journal do Yarn [Resource Manager]
   Acesse ao arquivo de configuração do HUE [$HUE_HOME/desktop/conf/hue.ini] e localize a seguinte tag [[yarn_clusters]]
   Altere para a porta correta do seu NameNode (Estou utilizando a versão 3.1.0, portanto a porta está situada na :8088)

```
[[yarn_clusters]]

  [[[default]]]
    # Enter the host on which you are running the ResourceManager
    ## resourcemanager_host=localhost

    # The port where the ResourceManager IPC listens on
    ## resourcemanager_port=8032
    resourcemanager_port=8088
```

3. Alterando o superuser do HDFS nas configurações do HUE
   Acesse ao arquivo de configuração do HUE [$HUE_HOME/desktop/conf/hue.ini] e localize a seguinte tag [desktop]
   Localize a seguinte informação "default_hdfs_superuser=hdfs" e altere para o usuário que está utilizando como superuser HDFS

```
# This should be the hadoop cluster admin
## default_hdfs_superuser=hdfs
default_hdfs_superuser=hduser
```

4. Alterando as Configurações do Hive
   Acesse ao arquivo de configuração do HUE [$HUE_HOME/desktop/conf/hue.ini] e localize a seguinte tag [beeswax]
   Informe qual a localização do arquivo hive-site.xml em seu sistema

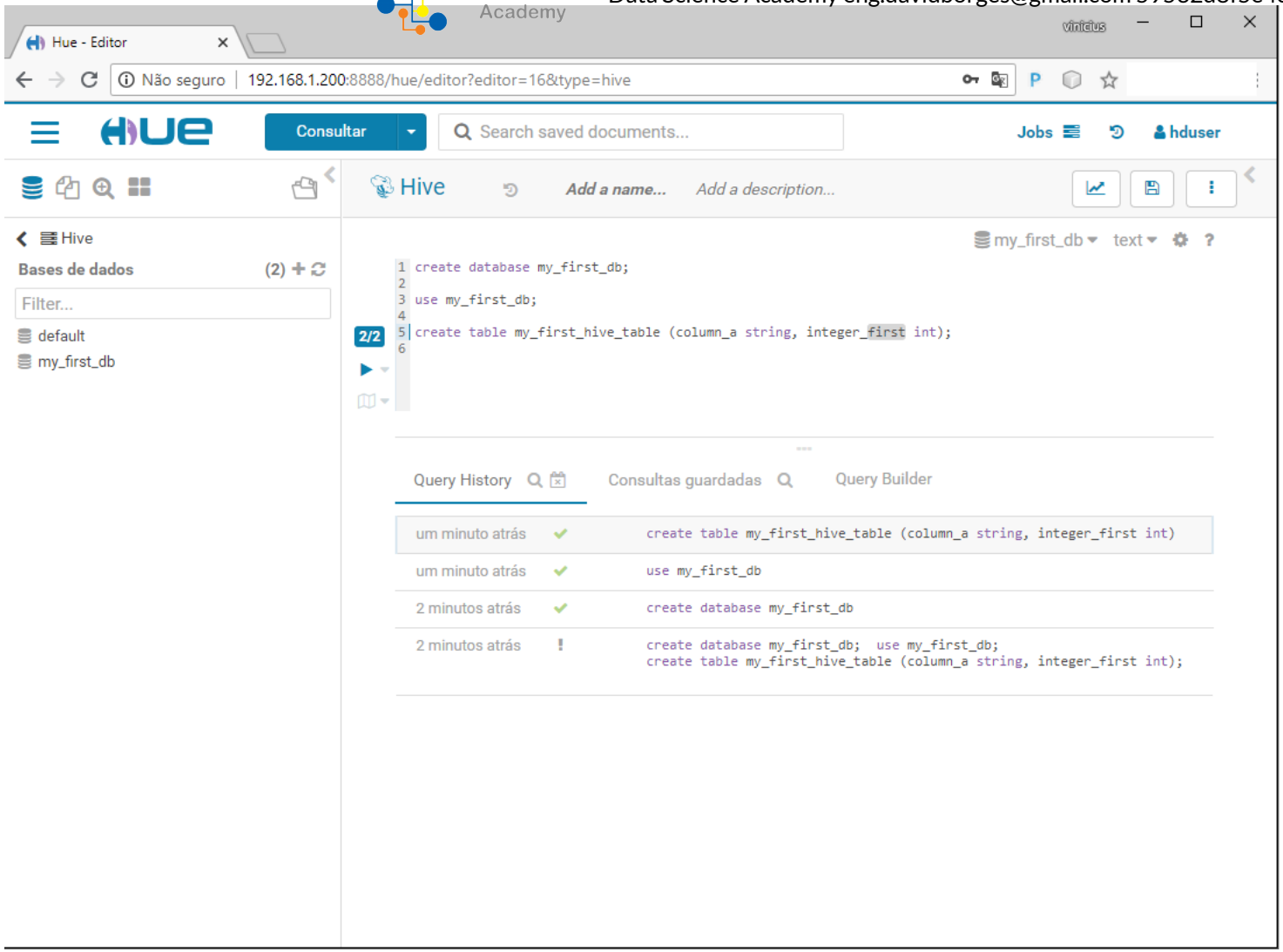
```
[beeswax]

  # Host where HiveServer2 is running.
  # If Kerberos security is enabled, use fully-qualified domain name (FQDN).
  ## hive_server_host=localhost

  # Port where HiveServer2 Thrift server runs on.
  ## hive_server_port=10000

  # Hive configuration directory, where hive-site.xml is located
  ## hive_conf_dir=/etc/hive/conf
  hive_conf_dir=/opt/hive/conf
```

   Certifique-se de que os serviços do hiveserver2 estão em execução; Caso contrário, basta instanciar através do comando:

```
$ nohup hive --service hiveserver2 &
```

Interrompa os serviços do HUE e os reinicie novamente;

Efetue acesso com suas credenciais de acesso, e pronto a interface do HUE com Integração ao Hive e HDFS estão funcionais e operando em conjunto com as demais ferramentas de seu ecossistema.



Vamos consultar se os diretórios com os respectivos nomes das tabelas foram criados no HDFS:

```
[hduser@hdpmaster ~]$ hdfs dfs -ls /user/hive/warehouse
Found 1 items
drwxr-xr-x   - hduser supergroup          0 2018-08-06 20:54 /user/hive/warehouse/my_first_db.db
[hduser@hdpmaster ~]$
[hduser@hdpmaster ~]$
[hduser@hdpmaster ~]$ hdfs dfs -ls /user/hive/warehouse/my_first_db.db
Found 1 items
drwxr-xr-x   - hduser supergroup          0 2018-08-06 20:54 /user/hive/warehouse/my_first_db.db/my_first_hive_table
[hduser@hdpmaster ~]$
```

Agora fica ao seu encargo explorar mais da interface e configurações do HUE, assim como habilitar / desabilitar os serviços que não serão utilizados no momento e realizar throubleshooting quando necessário!