

TP Hadoop

Si besoin, pour télécharger la VM : https://www.cloudera.com/downloads/quickstart_vms/5-12.html

Documentation HDFS : <https://hadoop.apache.org/docs/r2.7.1/hadoop-project-dist/hadoop-hdfs/HDFSCommands.html>

Généralités

Accéder à la VM

1. Lancer la VM cloudera depuis les postes de travail :

Menu des applications > VMcatalog > cloudera-5.12

Le lancement de la VM met environ 3 minutes.

2. La configuration clavier de la VM est par défaut anglais. Pour le basculer en français il faut ouvrir un invite de commande et taper :

```
setxkbmap fr
```

Configuration pour un clavier mac:

```
setxkbmap -layout fr -variant mac
```

3. Accéder à Hue

Hue est disponible sur le port 8888. Vous pouvez y accéder avec les credentials :

Identifiant : cloudera

Mot de passe : cloudera

Interaction avec HDFS

1. Tester HDFS avec HUE via la création d'un répertoire. Aller dans :

File Browser > New > Directory. Nom : raw_data

Ce répertoire sera un répertoire d'arrivée pour nos données.

2. Tester HDFS en ligne de commande en insérant des données dans le répertoire nouvellement créé.

Pour retrouver son répertoire : `hadoop fs -ls`

Pour mettre un fichier : `hadoop fs -put`

Pour quitter le safemode

```
sudo -u hdfs hdfs dfsadmin -safemode leave
```

Le dataset provient de <https://www.data.gouv.fr/fr/datasets/les-elus-municipaux-version-enrichie/>. Vous pouvez récupérer le fichier à déplacer avec `wget`. Dézipper le fichier.nouvellement créé

```
unzip https://s3.eu-central-1.amazonaws.com/telecom-hadoop/elus_mun2014.zip
```

Réponse :

```
wget https://s3.eu-central-1.amazonaws.com/telecom-hadoop/elus_mun2014.zip
unzip elus_mun2014.zip
```

```
hadoop fs -put elus_mun2014.csv /user/cloudera/raw_data/  
hadoop fs -ls /user/cloudera/raw-data  
hadoop fs -put elus_mun2014.csv raw_data  
hadoop fs -put elus_mun2014.csv raw_data/autre_nom.csv
```

Automatisation des interactions avec HDFS via script bash

Le shell va nous permettre d'automatiser certaines opérations avant l'insertion dans HDFS. Il peut par exemple s'agir d'un renommage avant l'insertion dans HDFS.

Le but de notre shell sera de prendre en entrée le fichier, de le renommer en `tp_hadoop_[timestamp]_csv` puis de l'insérer dans notre dossier "raw_data".

Créer le fichier sur votre poste client

```
touch loaddata.sh
```

Mettre les droits en exécution sur le fichier avec la commande "chmod"

```
chmod u+x
```

Utiliser un éditeur pour écrire dans le fichier (nano, vim, vi, etc.)

```
vim loaddata.sh
```

Pour insérer du text : i

Pour sortir du mode "insert" : echap

Pour sauver + quitter : :wq

Tester votre poste de travail avec un Hello World.

La première ligne doit comporter l'instruction `#!/bin/bash`. Le code du programme sera sur les suivantes.

Afficher un hello world se fait avec `echo`.

Réponse pour Hello World :

```
touch loaddata.sh
chmod u+x loaddata.sh
vim loaddata.sh
script :
    #!/bin/bash
    echo "Hello, world!"
./loaddata.sh
```

Via le script, prendre le fichier `elus_mun`, renommer en `tp_hadoop_[timestamp]_csv` puis l'insérer dans le dossier HDFS "raw_data".

Utiliser la fonction `date` pour récupérer le timestamp epoch, l'ajouter au nom du fichier, puis envoyer le fichier dans HDFS.

Réponse:

```
#!/bin/bash
hadoop fs -put elus_mun2014.csv raw_data/tp_hadoop_$(date +%d-%m-%y)_csv

#!/bin/bash

now=$(date +"%m-%d-%Y")

now='tp_hadoop_'$now'.csv'
mv elus_mun2014.csv $now
```

```
hadoop fs -put $now raw_data
```

3. Pour aller plus loin dans HDFS

Envoyer un fichier avec la date-heure.

Vérifier avant l'envoi dans HDFS que le même nom de fichier n'existe pas dans le répertoire avant l'insertion. Sinon ajouter un “_” au nom du fichier avant l'insertion du fichier dans HDFS

Explication : HDFS gère mal le cas où un fichier du même nom est déjà présent dans HDFS

```
#!/bin/bash
now=$(date +"%m-%d-%Y")
now='tp_hadoop_'$now'.csv'
echo $now
mv elus_mun2014.csv $now
if [ $(hadoop fs -f "$now") ]
then
    echo "$now found."
else
    echo "$now not found."
    hadoop fs -put $now /user/cloudera
echo "put done"
fi
```

Transformations avec Pig

Dans un premier temps, nous allons charger le fichier dans PIG via la fonction `LOAD` et afficher 10 lignes du fichier avec les fonctions `DUMP` et `LIMIT`. (si vous affichez tout le contenu du fichier, le navigateur risque d'être surchargé). Ajoutez une structure aux données au moment du `LOAD` en utilisant `PigStorage` (qui permet de spécifier une structure et des formats au moment du chargement). La structure attendue est la suivante : "liste, nom, prenom, naissance, profession, nationalite", en utilisant le format `CHARARRAY` pour chacune. Le caractère de séparation est ",".

```
# Pour décrire la structure du fichier
grunt> A = LOAD « raw_data/elus_mun2014.csv » USING PigStorage(',') AS
(list:chararray, nom:chararray, prenom:chararray, naissance:chararray,
profession:chararray, nationalite:chararray);
grunt> B = LIMIT A 10;
grunt> DUMP B;

grunt> DESCRIBE A;

# Pour compter le nombre de lignes dans A
grunt> A_group_all = GROUP A ALL;
grunt> A_count = FOREACH A_group_ALL GENERATE COUNT(A) AS COUNT;
grunt> DUMP A_count;
Successfully read 525177 records (37720912 bytes) from "hdfs://
quickstart.cloudera:8020/user/cloudera/raw_data/elus_mun2014.csv"
```

Pour filtrer sur un champ

```
A = LOAD 'nom_fichier' USING PigStorage(',') AS
(liste:CHARARRAY,nom:CHARARRAY,prenom:CHARARRAY,sexe:CHARARRAY,naissance:CHARARR
AY,profession:CHARARRAY,nationalite:CHARARRAY);
A_unique = DISTINCT A;
A_clean = FILTER A_unique BY sexe is not null;
B = GROUP A_clean ALL;
C = FOREACH B GENERATE COUNT(A_clean.nom);
DUMP C;
```

Donner les noms retirés

```
C = FILTER A BY sexe IS NULL;
DUMP C;
```

Pour tester l'écriture dans HDFS, sauvegarder votre fichier dans HDFS avec `STORE` en utilisant `PigStorage`.

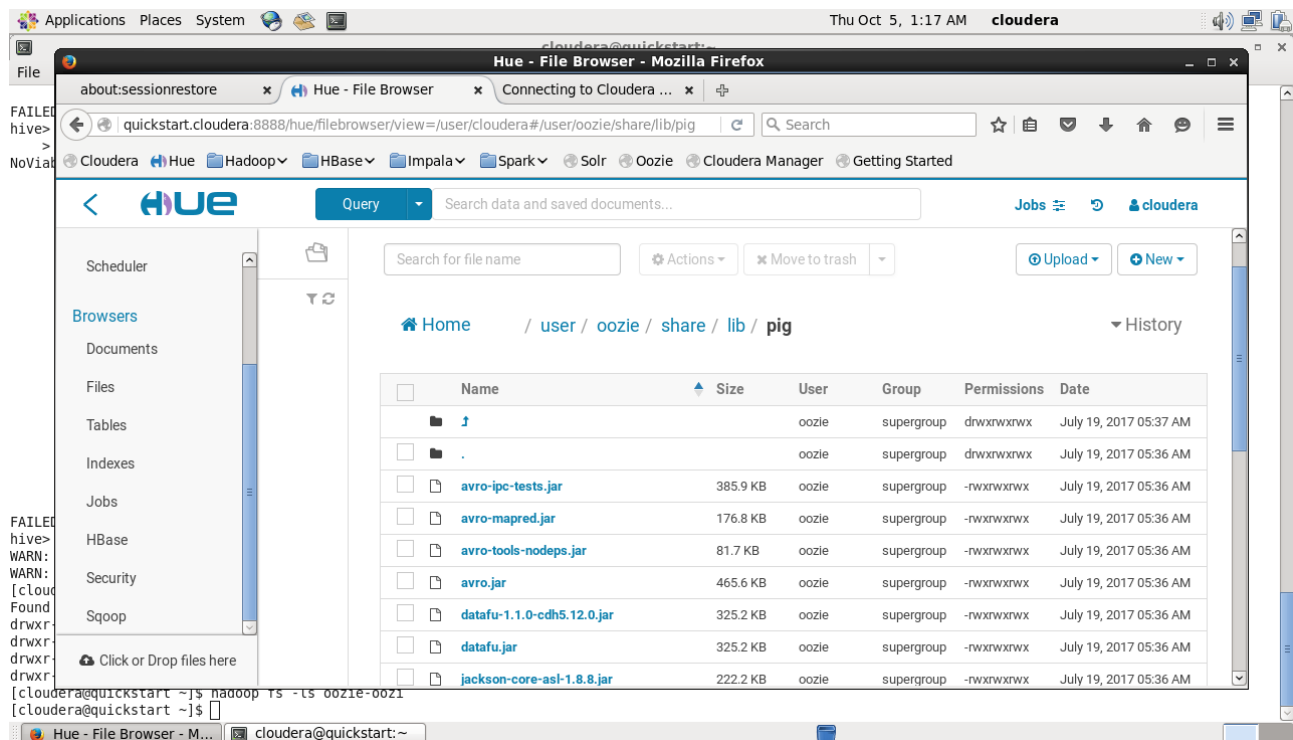
```
STORE A_cleaned INTO 'dossier_de_destination_hdfs' USING PigStorage(',');
```

Chargement, requête et visualisation avec Hive

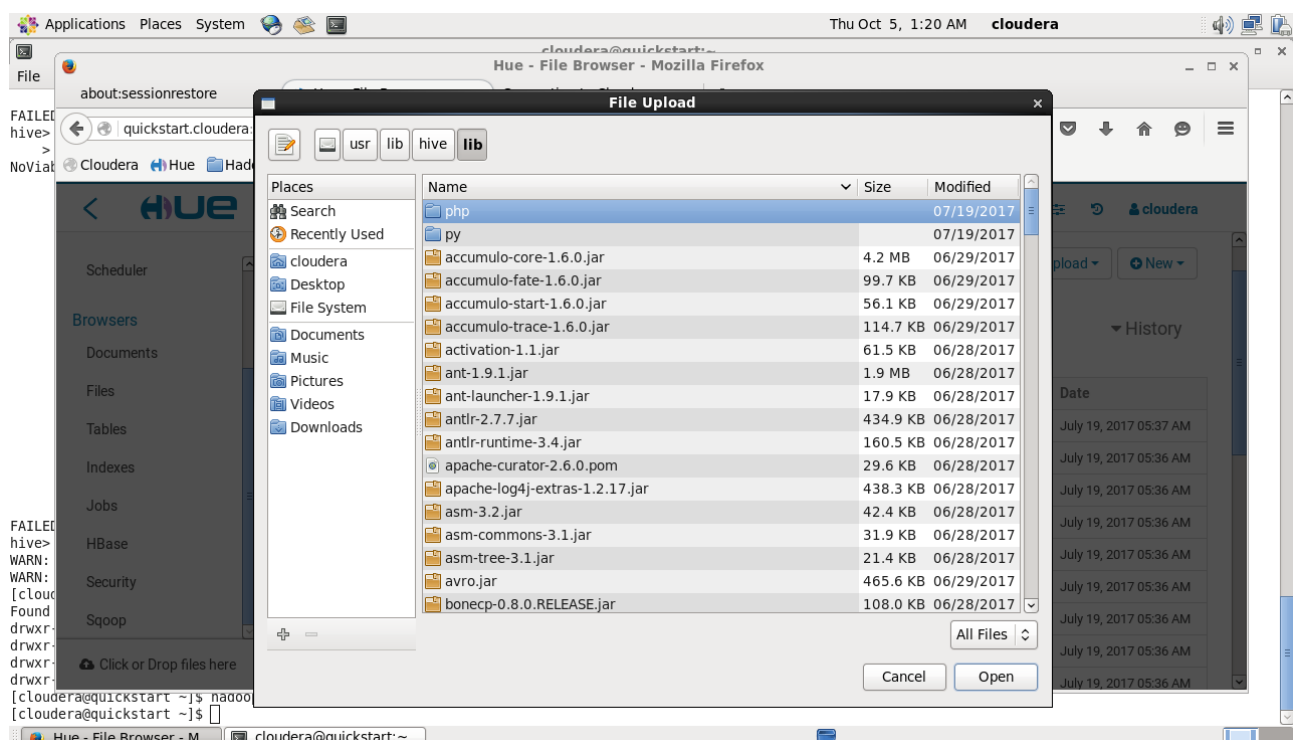
En cas d'erreur : **Error "Could not connect to quickstart.cloudera:10000"**

Sortir du safemode : `hadoop dfsadmin -safemode leave`

Dans Hue :



>> Cliquer sur user et suivre le path oozie/share/lib/pig



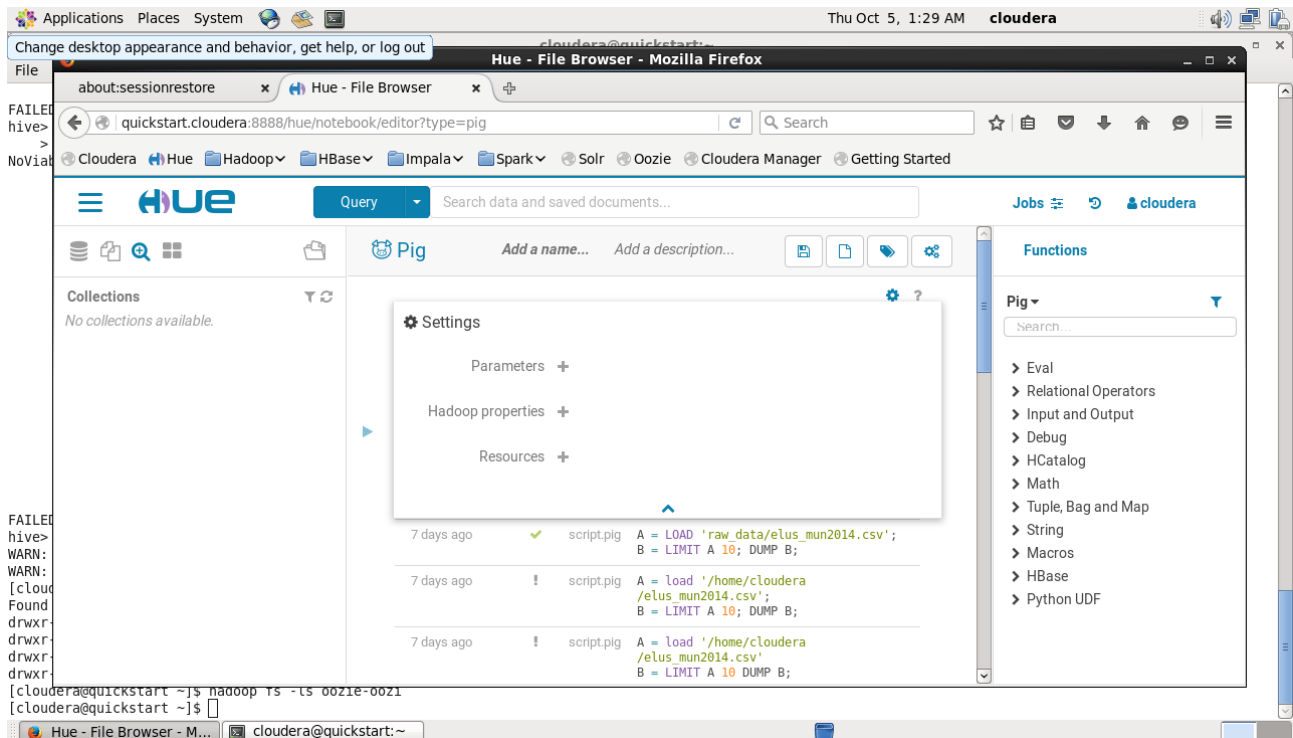
>> Cliquer sur upload > Files > Select files

>> Importer dans ce répertoire HDFS tous les jars situés dans /usr/lib/hive/lib/

>> De même, importer tous les jars situés dans /usr/lib/hive-hcatalog/share/hcatalog/

>> Uploader le fichier hive-site.xml (fichier des configurations de HIVE) situé à /usr/lib/hive/conf/hive-site.xml dans /user/cloudera/

>> Dans Pig, cliquez sur la roue dentée au dessus du Notepad, aller dans Resources, cliquer sur le bouton



Add et ajouter le fichier hive-site.xml (que vous retrouver dans hdfs dans /user/cloudera) : path: /user/cloudera/hive-site.xml

```
# Dans Query Editors > Hive, créer la table elus_municipaux
CREATE TABLE elus_municipaux (liste STRING,nom STRING,prenom STRING,sexe STRING,
naissance STRING,profession STRING,nationalite STRING);
```

```
# Insertion dans Hive
> pig
grunt> A = LOAD 'raw_data/elus_mun2014.csv' USING PigStorage(',') AS
(liste:CHARARRAY,nom:CHARARRAY,prenom:CHARARRAY,sexe:CHARARRAY,naissance:CHARARR
AY,profession:CHARARRAY,nationalite:CHARARRAY);

grunt> STORE A INTO 'elus_municipaux' USING
org.apache.hive.hcatalog.pig.HCatStorer();
```

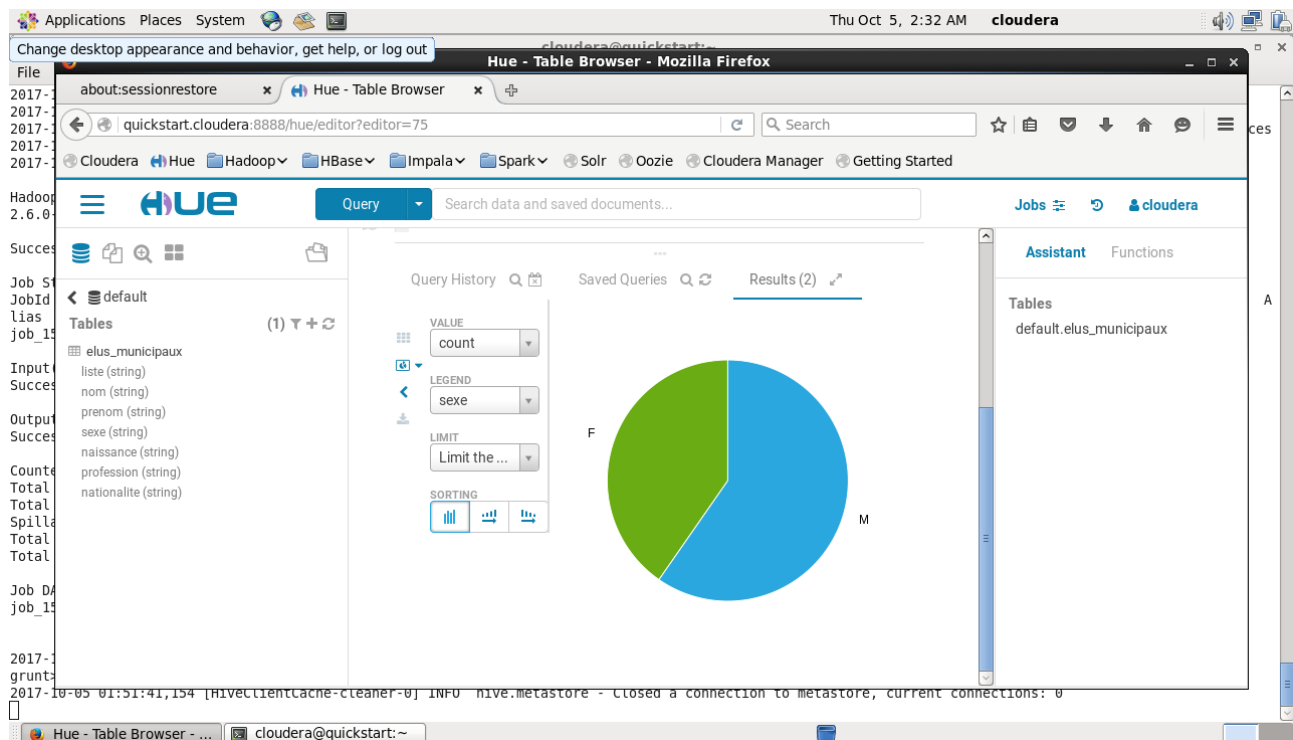
Dans Query > Editor > Pig

```
# Afficher les 10 premières lignes
SELECT * FROM elus_municipaux LIMIT 10;

# Compter le nombre d'ingénieurs
SELECT COUNT(*) FROM elus_municipaux
WHERE profession='Ingénieur';
```

```
# Afficher le top 5 des prénoms les plus utilisés parmi les femmes et le nombre d'occurrence associé
SELECT prenom, COUNT(*) as count FROM elus_municipaux
WHERE sexe='F'
GROUP BY prenom
ORDER BY count
LIMIT 5;

# Afficher la part d'hommes et de femmes, utilisez les charts pour une représentation plus visuelle.
SELECT sexe, COUNT(*) as cnt FROM elus_municipaux
GROUP BY sexe
ORDER BY cnt DESC
LIMIT 2;
```



```
hive
hive> add FILE /home/cloudera/transformation.py;
Added resources: [/home/cloudera/transformation.py]
hive> list FILE; # See the added files
/home/cloudera/transformation.py
hive> select transform (liste, nom) using 'python transformation.py' as (liste,
l_name) from elus_municipaux;
```



```

Applications Places System Thu Oct 5, 2:54 AM cloudera
Change desktop appearance and behavior, get help, or log out cloudera@quickstart:~
File Edit View Search Terminal Help
Union Nationale pour L'Indépendance tein-fouin
Union Nationale pour L'Indépendance dounehote
Union Nationale pour L'Indépendance diemene
Union Nationale pour L'Indépendance dounezek
Dynamique Vôôk-S'unir pour agir autrement wacalie
FLNKS atinoua
FLNKS atti
FLNKS aourigna Ép. atinoua
FLNKS ouetcho
FLNKS tara
FLNKS akapo
Liste Coordination Commune-Tribus avec Rheebu-Nuu vama
ENTENTE POUR UN RENOUVEAU MUNICIPAL A YATE koroma
FLNKS vama Ép. agourere
FLNKS vandegou
FLNKS digoue
FLNKS newedou
FLNKS atiti
FLNKS koroma
FLNKS gouetcha
YATE AUTREMENT tara
ENTENTE POUR UN RENOUVEAU MUNICIPAL A YATE ouetcho
FLNKS waru
FLNKS newedou Ép. koroma
TERRE COMMUNE ET AVENIR DE KAWIIPA bogihondiha
TERRE COMMUNE ET AVENIR DE KAWIIPA nimoiri
TERRE COMMUNE ET AVENIR DE KAWIIPA meboede
TERRE COMMUNE ET AVENIR DE KAWIIPA uhila
FLNKS Unitaire diaion Ép. wathia
TERRE COMMUNE ET AVENIR DE KAWIIPA hari
TERRE COMMUNE ET AVENIR DE KAWIIPA thomas
TERRE COMMUNE ET AVENIR DE KAWIIPA kenerou
TERRE COMMUNE ET AVENIR DE KAWIIPA ponga
TERRE COMMUNE ET AVENIR DE KAWIIPA hari Ép. moly
DIVER-CITÉ KAWIPA diaion
TERRE COMMUNE ET AVENIR DE KAWIIPA debaoue
FLNKS Unitaire chagui
DIVER-CITÉ KAWIPA mereatu
TERRE COMMUNE ET AVENIR DE KAWIIPA thavoavianon
Time taken: 69.239 seconds, Fetched: 525177 row(s)
hive>

```

```

# Afficher les personnes de moins de 60 ans étant retraité
import sys
import datetime

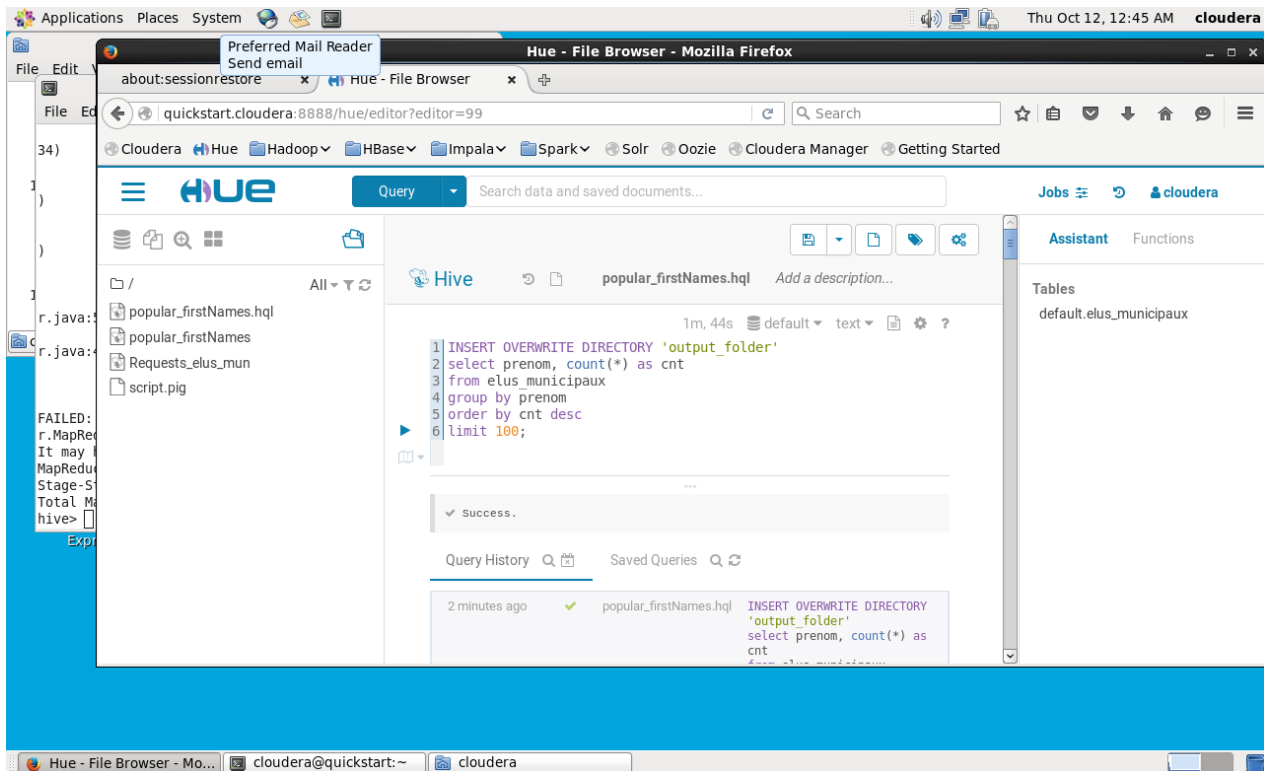
now = datetime.datetime.now()
nb_days_60_years = 60 * 365

for line in sys.stdin:
    line = line.strip()
    nom, prenom, naissance, profession = line.split('\t')
    if "retrait" not in profession:
        continue
    age_in_timespan = now - datetime.datetime.strptime(naissance, '%Y-%m-%d')
    if age_in_timespan.days > nb_days_60_years:
        continue
    print ('\t'.join([nom, prenom, naissance, profession]))

```

Automatisation avec Oozie

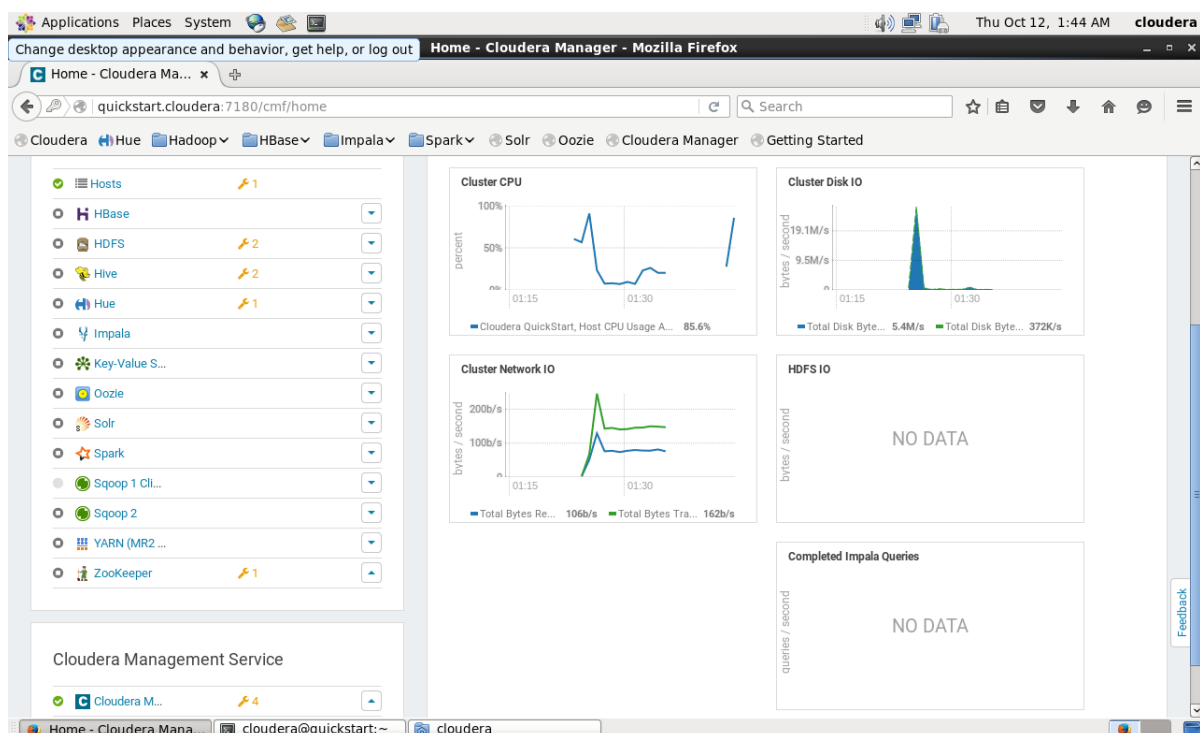
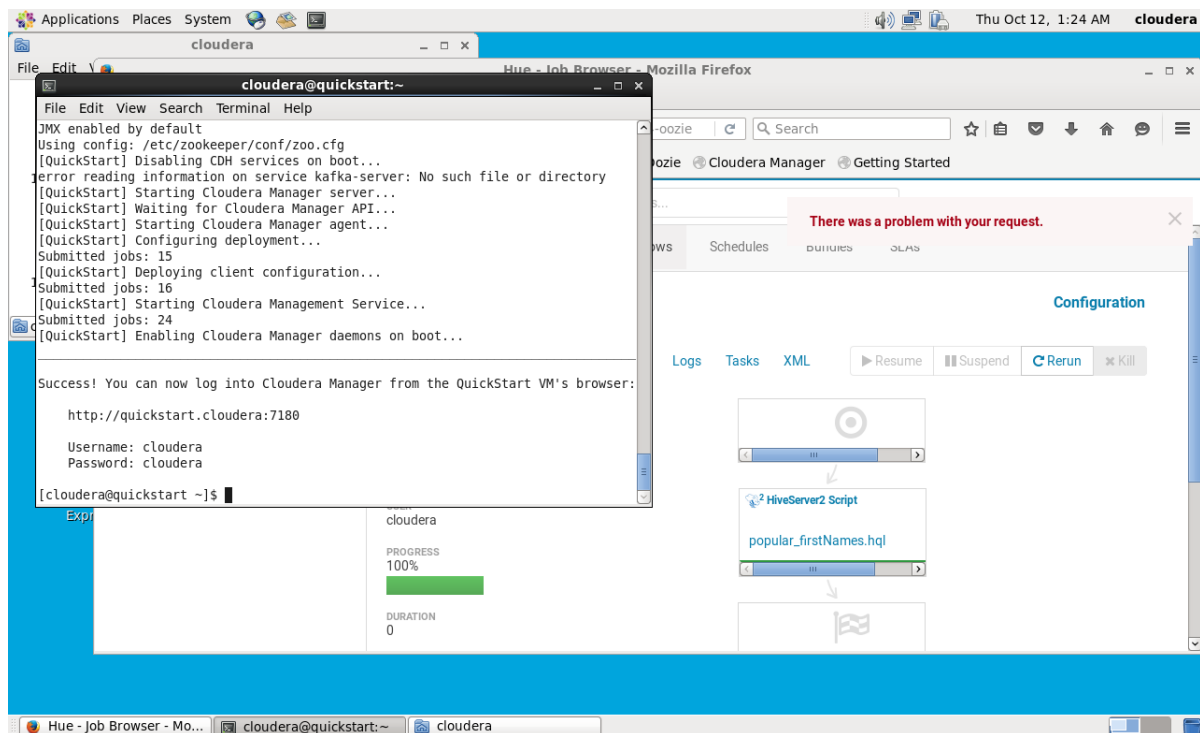
```
INSERT OVERWRITE DIRECTORY 'output_folder'  
SELECT prenom, count(*) AS cnt  
FROM elus_municipaux  
GROUP BY prenom  
ORDER BY cnt DESC  
LIMIT 100;
```



Gestion des services Hadoop via Cloudera Manager

Pour lancer Cloudera Manager sur la vm, il faut exécuter la commande suivante :

```
sudo /home/cloudera/cloudera-manager --force --express
```



Kafka

```
thaianthantrong@MacBook-Pro-de-Thai:~$ brew install kafka
Warning: You are using OS X 10.12.
We do not provide support for this pre-release version.
You may encounter build failures or other breakages.
==> Installing dependencies for kafka: zookeeper
==> Installing kafka dependency: zookeeper
==> Downloading https://www.apache.org/dyn/closer.cgi?path=zookeeper/zookeeper-3
Already downloaded: /Users/thaianthantrong/Library/Caches/Homebrew/
zookeeper-3.4.8.tar.gz
==> ./configure --prefix=/usr/local/Cellar/zookeeper/3.4.8 --without-cppunit
==> make install
==> Caveats
To have launchd start zookeeper now and restart at login:
  brew services start zookeeper
Or, if you don't want/need a background service you can just run:
  zkServer start
==> Summary
🍺 /usr/local/Cellar/zookeeper/3.4.8: 237 files, 17.6M, built in 27 seconds
==> Installing kafka
==> Downloading http://mirrors.ibiblio.org/apache/kafka/0.9.0.1/kafka-0.9.0.1-src.tar.gz
##### 100,0%
==> gradle
==> gradle jar
==> Caveats
To have launchd start kafka now and restart at login:
  brew services start kafka
Or, if you don't want/need a background service you can just run:
  zookeeper-server-start /usr/local/etc/kafka/zookeeper.properties; kafka-
server-start /usr/local/etc/kafka/server.properties
==> Summary
🍺 /usr/local/Cellar/kafka/0.9.0.1: 4,880 files, 76.8M, built in 3 minutes 1
second
```

Dans un nouveau terminal

```
thaianthantrong@eduroam-1-186:~$ cd Downloads/kafka_2.11-0.11.0.1
thaianthantrong@eduroam-1-186:~/Downloads/kafka_2.11-0.11.0.1$ bin/kafka-
topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --
partitions 1 --topic fr.telecom.tpkafka.kafkatopicstest
WARNING: Due to limitations in metric names, topics with a period ('.') or
underscore ('_') could collide. To avoid issues it is best to use either, but
not both.
Created topic « fr.telecom.tpkafka.kafkatopicstest".
```

Trois terminaux : Zookeeper, Broker, et un autre pour s'interfacer.

Terminal interface

```
thaianthantrong@eduroam-1-186:~/Downloads/kafka_2.11-0.11.0.1$ bin/kafka-
console-producer.sh --broker-list localhost:9092 --topic
fr.telecom.tpkafka.kafkatopicstest
>My very first message with Kafka
>
```

Quatrième terminal

```
thaianthantrong@eduroam-1-186:~/Downloads/kafka_2.11-0.11.0.1$ bin/kafka-
console-consumer.sh --bootstrap-server localhost:9092 --topic
fr.telecom.tpkafka.kafkatopicstest --from-beginning
My very first message with Kafka
```

Le consumer possède un offset sur le topic Kafka qui permet de récupérer les derniers messages non lus. Cependant, l'option `--from-beginning` permet de re-parcourir l'ensemble des données disponibles dans le topic, sans se soucier de l'offset. Ces deux commandes possèdent plusieurs options affichables en les lançant sans arguments.

```
thaianthantrong@eduroam-1-186:~/Downloads/kafka_2.11-0.11.0.1$ bin/kafka-
console-consumer.sh --bootstrap-server
Option bootstrap-server requires an argument
Option                                Description
-----
[.]
--from-beginning                      If the consumer does not already have
                                     an established offset to consume
                                     from, start with the earliest
                                     message present in the log rather
                                     than the latest message.
[.]
```

Pour l'instant nous n'avons utilisé qu'un seul broker. Pour tester le parallélisme et la haute tolérance aux pannes de Kafka nous allons créer un cluster de 3 brokers.

Un broker est défini par un fichier de configuration, dans lequel est spécifié l'id du broker, le port sur lequel le broker écoute, et le dossier dans lequel les données seront écrites.

Afin de spécifier les deux autres brokers, créez deux nouveaux fichiers propriétés en partant du fichier `server.properties`.

```
cp config/server.properties config/server-1.properties
cp config/server.properties config/server-2.properties
```

Dans ces deux fichiers, modifiez les propriétés énoncées plus haut afin d'avoir:

```
config/server-1.properties:
  broker.id=1
  listeners=PLAINTEXT://:9093
  log.dirs=/tmp/kafka-logs-1

config/server-2.properties:
  broker.id=2
  listeners=PLAINTEXT://:9094
  log.dirs=/tmp/kafka-logs-2
```

Un de nos brokers tourne déjà. Nous allons maintenant lancer les deux autres brokers.

```
bin/kafka-server-start.sh config/server-1.properties &
bin/kafka-server-start.sh config/server-2.properties &
```

Créez maintenant un nouveau topic, de 1 partition, avec un facteur de réplication de 3 :

```
bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 3
--partitions 1 --topic fr.telecom.tpkafka.topicreplication
```

Pour voir la façon dont sont réparties les partitions et les réplicas sur un topic, tapez la commande ci-dessous

```
bin/kafka-topics.sh --describe --zookeeper localhost:2181 --topic fr.telecom.tpkafka.topicreplication
```

```
thaianthantrong@eduroam-1-186:~/Downloads/kafka_2.11-0.11.0.1$ bin/kafka-topics.sh --describe --zookeeper localhost:2181 --topic fr.telecom.tpkafka.topicreplication
Topic:fr.telecom.tpkafka.topicreplication PartitionCount:1 ReplicationFactor:3
  Configs:
    Topic: fr.telecom.tpkafka.topicreplication Partition: 0 Leader:
2 Replicas: 2,0,1 Isr: 2,0,1
```

On voit que le leader est le broker numéro 2. Les réplicas sont les partitions, leader ou non sur lesquelles les données sont écrites. ISR est le In Sync Replica, il s'agit des partitions qui sont bien synchronisées avec la partition leader.

Nous allons maintenant tester la tolérance aux pannes.

```
bin/kafka-console-producer.sh --broker-list localhost:9092 --topic fr.telecom.tpkafka.topicreplication
```

Kill le broker sur lequel est la partition leader. (vous pouvez quitter le terminal du broker correspondant, ou utiliser `ps faux | grep kafka` pour retrouver l'id puis tuer le processus avec `kill -9`)

```
ps faux | grep server-2.properties
```