1. Accéder à la VM

Si vous avez un message d'erreur avec le namenode ou avec Hive :
Entrer : hadoop dfsadmin -safemode leave dans l'invite de commande
Cela arrive lorsque le namenode entre en safemode (pour une raison inconnue sur certains pcs)

Téléchargement de la vm : https://www.cloudera.com/downloads/quickstart_vms/5-12.html

1. Lancer la VM cloudera depuis les postes de travail :
Menu des applications > VMcatalog ou Autre > 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

3. Accéder à Hue
Hue est disponible sur le port 8888. Vous pouvez y accéder avec les credentials :
cloudera/cloudera
Aller sur localhost:8888
cloudera

1. 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.
Utiliser les commandes "hdfs dfs -ls" pour retrouver son répertoire
Et "hdfs dfs -ls /" pour lister les éléments courants
Utiliser la commande "hdfs dfs -put" pour mettre un fichier

Vous pouvez récupérer le fichier à déplacer avec "wget". le fichier est disponible à l'adresse suivante :
https://s3.eu-central-1.amazonaws.com/telecom-hadoop/elus_mun2014.zip

Le dataset provient de
https://www.data.gouv.fr/fr/datasets/les-elus-municipaux-version-enrichie/

Dézipper le fichier.

hdfs command guide :
https://hadoop.apache.org/docs/r2.7.1/hadoop-project-dist/hadoop-hdfs/HDFSCommands.html

Réponse :
wget https://s3.eu-central-1.amazonaws.com/telecom-hadoop/elus_mun2014.zip
unzip elus_mun2014.zip
hdfs dfs -put elus_mun2014.csv /user/cloudera/raw_data

Automatisation des interactions avec 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".

1. Créer le fichier loaddata.sh sur votre systeme de gestion fichier client

touch loaddata.sh

2. Mettre les droits en exécution sur le fichier avec la commande "chmod"
chmod u+x loaddata.sh

3. Utiliser un éditeur pour écrire dans le fichier (nano, vim, vi, gedit, etc.)
> vim loaddata.sh
"i" pour insérer du text, echap pour sortir du mode "insert", ":wq" pour sauver + quitter
ls
4. Tester votre poste de travail avec un hello world
echo "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".

5. Tester votre poste de travail avec un hello world

Réponse pour hello world :
touch loaddata.sh
chmod u+x loaddata.shcsv

vim loaddata.sh
script :
#!/bin/bash
echo "Hello, world!"

execution :
./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.

Indice : Pour écrire le timestamp epoch dans une variable :
now=$(date +"%s")

Reponse :
hdfs dfs -put $1 raw_data/tp_hadoop_`date +%s`_.csv

Pour aller plus loin
6. vérifier que le fichier n'existe pas dans le répertoire hdfs avant de l'insérer. Sinon, mettre un index au nouveau fichier à insérer.
Exemple :
tp_hadoop_[timestamp]_2 etc…

Proposition de solution :

```
#!/bin/bash
filename=$1
echo $filename
DATE_WITH_TIME=`date +%Y%m%d`
new_filename="tp_hadoop_"$DATE_WITH_TIME"_.csv"

echo $new_filename

liste_fichiers_deja_presents=`hdfs dfs -ls raw_data`
echo $liste_fichiers_deja_presents

nb_files_found=`hdfs dfs -ls raw_data | grep "tp_hadoop_"$DATE_WITH_TIME"_" | wc -l`

if [ "$nb_files_found" -ge "1" ] ; then
    echo "File already found, adding index"
    new_filename="tp_hadoop_"$DATE_WITH_TIME"_"$nb_files_found".csv"
fi

hdfs dfs -put ./$filename raw_data"/"$new_filename
```

Ressources supplémentaires :
Pour s'améliorer sur les commandes HDFS, ou la compréhension de HDFS :
La page Hdfs Hortonworks explique les mécanismes HDFS et propose des tutoriaux :
https://fr.hortonworks.com/apache/hdfs/#section_1
https://hadoop.apache.org/docs/r2.7.1/hadoop-project-dist/hadoop-hdfs/HDFSCommands.html (liste des commandes HDFS)
Linux/Unix tutoriaux : https://www.tutorialspoint.com/unix/index.htm