Manipulations sur HDFS

Voici quelques commandes à essayer dans le terminal.
Connectez-vous sur le compte « hduser » (mot de passe : hduser) avec la commande suivante :

```
sparkpython@sparkpython-VirtualBox:~$ su hduser
Mot de passe :
hduser@sparkpython-VirtualBox:/home/sparkpython$ cd
hduser@sparkpython-VirtualBox:~$
```

Démarrer hdfs et yarn avec les deux commandes suivantes (le mot de passe est toujours « hduser ») :

```
hduser@sparkpython-VirtualBox:~$ start-dfs.sh
hduser@sparkpython-VirtualBox:~$ start-yarn.sh
hduser@sparkpython-VirtualBox:~$ jps
5617 DataNode
6358 Jps
5480 NameNode
6238 NodeManager
5934 ResourceManager
5791 SecondaryNameNode
hduser@sparkpython-VirtualBox:~$
```

`hdfs dfs -ls` : elle n'affiche rien pour l'instant car elle s'adresse par défaut à votre dossier HDFS personnel /user/votre_login qui est vide (ici /user/hduser) .

`hdfs dfs -ls /` : affiche ce qu'il y a à la racine HDFS. Vous pouvez descendre inspecter les dossiers que vous voyez. Exemple hdfs dfs -ls /user. Il n'y a pas de commande équivalente à cd, parce qu'il n'y a pas de notion de dossier courant dans HDFS, donc à chaque fois, il faut remettre le chemin complet. C'est une habitude à prendre.

`hdfs dfs -ls -R -h /nomDossier` : affiche les fichiers des sous-dossiers, avec une taille arrondie en Ko, Mo ou Go.

`hdfs dfs -mkdir fichiers` : crée un dossier dans votre espace HDFS, c'est-à-dire /user/hduser/fichiers. Notez que la taille d'un dossier sera toujours 0.

Créez un fichier appelé **bonjour.txt** dans votre compte Linux et contenant le mot « bonjour ».
Copier ce fichier sur HDFS par **`hdfs dfs -put bonjour.txt`**

Utilisez **`hdfs dfs -ls`** pour vérifier.

`hdfs dfs -cat bonjour.txt` : affiche le contenu. Il n'y a pas de commande more mais vous pouvez faire **`hdfs dfs -cat bonjour.txt | more`**

`hdfs dfs -tail bonjour.txt` : affiche le dernier Ko du fichier.

Supprimer ce fichier de HDFS par **`hdfs dfs -rm bonjour.txt`**.

Remettre à nouveau ce fichier par **`hdfs dfs -copyFromLocal bonjour.txt`** (vérifier avec **`hdfs dfs -ls`**). Cette commande est similaire à **`hdfs dfs -put`**.

`hdfs dfs -chmod go+w bonjour.txt` (vérifier son propriétaire, son groupe et ses droits avec **`hdfs dfs -ls`**)

`hdfs dfs -chmod go-r bonjour.txt` (vérifier les droits)

`hdfs dfs -mv bonjour.txt fichiers/bonjour.txt` (vérifier avec **`hdfs dfs -ls`**)

`hdfs dfs -get fichiers/bonjour.txt transf.txt` : transfère le fichier de HDFS vers votre compte Linux en lui changeant son nom. Cette commande ne serait pas à faire avec de vraies méga-données !

`hdfs dfs -cp fichiers/bonjour.txt fichiers/salut.txt` (vérifier)

`hdfs dfs -count -h /user/hduser` : affiche le nombre de sous-dossiers, fichiers et octets occupés.

`hdfs dfs -rm fichiers/bonjour.txt` (vérifier avec hdfs dfs -ls fichiers).

`hdfs dfs -rmr fichiers` (vérifier avec **`hdfs dfs -ls`**).

Supprimer les fichiers locaux **`bonjour.txt`** et **`transf.txt`**.

Téléchargez le fichier arbres.csv dans votre compte :

```
sparkpython@sparkpython-VirtualBox:~$
URL=https://forge.scilab.org/index.php/p/rdataset/source/file/master/csv/datasets/Titanic.csv

sparkpython@sparkpython-VirtualBox:~$ wget $URL
```

Copiez ce fichier vers HDFS par : **`hdfs dfs -put Titanic.csv`**. Vérifiez sa présence et sa taille sur HDFS puis supprimez-le de votre compte local ainsi que de HDFS.

Voici une autre manière de placer un fichier sur HDFS sans le stocker dans votre compte :
`wget -O - $URL | hdfs dfs -put - Titanic.csv`

Voici une dernière commande, juste pour la curiosité :
`hdfs fsck /user/hduser -files -blocks` : affiche la liste des blocs utilisés par vos fichiers. Sachant que les blocs font 64 Mo (256 ou 512 Mo sur un cluster pro).