TP Hadoop HDFS : Manipulation des fichiers et HA

Mohamed Koubaa

September 30, 2025

Objectif

Ce TP a pour objectif de vous familiariser avec la manipulation des fichiers et des répertoires dans HDFS ainsi que de tester la haute disponibilité (HA) d'un cluster Hadoop. Vous apprendrez à :

- Lancer un cluster Hadoop HA via Docker Compose.
- Vérifier l'état du cluster et des services.
- Manipuler des fichiers HDFS (import/export, lecture, suppression, etc.).
- Tester la disponibilité du système après une panne du NameNode actif.

Pré-requis

- Docker et Docker Compose installés.
- Accès à un terminal pour exécuter les commandes Docker et HDFS.
- Notions de base sur HDFS et la haute disponibilité.

1 Lancer le cluster Hadoop HA avec Docker Compose

1. Télécharger le fichier **hadoop-ha-docker.zip** à partir de votre espace moodle.
2. Décompresser le fichier puis ouvrir un terminal à partir du répertoire contenant le fichier `docker-compose.yml`.
3. Démarrer tous les conteneurs du cluster :

```
docker-compose up -d
```

4. Vérifier que tous les conteneurs sont démarrés :

```
docker ps
```

5. Vérifier les logs pour s'assurer que NameNodes, DataNodes, JournalNodes et ResourceManagers sont opérationnels :

```
docker logs namenode1
docker logs namenode2
docker logs datanode1
docker logs datanode2
docker logs resourcemanager1
docker logs resourcemanager2
docker logs journalnode1
docker logs journalnode2
```

6. Vérifier les processus Java à l'intérieur des conteneurs :

```
docker exec -it namenode1 jps
docker exec -it datanode1 jps
docker exec -it resourcemanager1 jps
docker exec -it journalnode1 jps
```

2 Tester la haute disponibilité (HA) de HDFS

1. Vérifier quel NameNode est actif :

```
docker exec -it namenode1 hdfs haadmin -getServiceState nn1
docker exec -it namenode2 hdfs haadmin -getServiceState nn2
# ou bien
docker exec -it namenode2 hdfs haadmin -getAllServiceState
```

2. Simuler l'arrêt du NameNode actif :

```
docker stop namenode1   # si namenode1 est actif
```

3. Vérifier que le NameNode standby prend le relais automatiquement :

```
docker exec -it namenode2 hdfs haadmin -getServiceState nn2
```

4. Tester que les DataNodes continuent de fonctionner et que HDFS est accessible :

```
docker exec -it datanode1 hdfs dfs -ls /
docker exec -it datanode2 hdfs dfs -ls /
```

5. Redémarrer le NameNode initialement actif pour revenir à l'état normal :

```
docker start namenode1
```

3 Gestion de fichiers et répertoires dans HDFS

Toutes les commandes suivantes doivent être exécutées depuis le conteneur du **NameNode actif**. Pour cela, connectez-vous au conteneur (par exemple `namenode1`) :

```
# Ouvrir un terminal sur namenode 1
docker exec -it namenode1 bash
```

Puis lancez les commandes `hdfs dfs ...`.

3.1 Gestion de l'arborescence HDFS

```
# Créer un répertoire
hdfs dfs -mkdir /tp-hdfs
hdfs dfs -mkdir -p /tp_hdfs/data
```

```
# Lister le contenu d'un répertoire
hdfs dfs -ls /
hdfs dfs -ls -R /tp-hdfs   # liste récursive
```

```
# Supprimer un répertoire
hdfs dfs -rm -r /tp_hdfs/data
```

3.2 Import de fichiers dans HDFS

Créez d'abord un fichier local simple dans le conteneur :

```
# Créer un fichier en local
echo "Bonjour HDFS" > exemple.txt
```

```
# Copier un fichier local vers HDFS
hdfs dfs -put exemple.txt /tp-hdfs/data/
# Un autre exemple
hdfs dfs -copyFromLocal /etc/hosts /tp-hdfs/data/
```

```
# Copier plusieurs fichiers d'un coup
hdfs dfs -put *.txt /tp-hdfs/data/
```

```
# Déplacer un fichier local vers HDFS
hdfs dfs -moveFromLocal exemple.txt /tp-hdfs/data/
```

3.3 Lister les fichiers dans HDFS

```
# Lister le contenu sur HDFS
hdfs dfs -ls /tp_hdfs/data
```

Exemple de sortie :

```
-rw-r--r--   1 root supergroup         13 2025-09-28 20:12 /tp_hdfs/data/exemple.txt
```

3.4 Afficher le contenu d'un fichier

```
hdfs dfs -cat /tp_hdfs/data/exemple.txt
```

Résultat attendu :

```
Bonjour HDFS
```

3.5 Vérifier l'espace utilisé : `hdfs dfs -du -h`

Ajoutons un deuxième fichier un peu plus gros :

```
echo "1,Jean,23" > donnees.csv
echo "2,Sarah,31" >> donnees.csv
echo "3,Ali,27" >> donnees.csv

hdfs dfs -put donnees.csv /tp_hdfs/data/
```

Puis surveillons l'espace utilisé :

```
hdfs dfs -du -h /tp_hdfs/data
```

Exemple de sortie :

```
13   /tp_hdfs/data/exemple.txt
45   /tp_hdfs/data/donnees.csv
```

3.6 Export de fichiers depuis HDFS

```
# Copier de HDFS vers local
hdfs dfs -get /tp_hdfs/data/fichier.txt .
hdfs dfs -copyToLocal /tp_hdfs/data/* /tmp/
```

```
# Déplacer de HDFS vers local
hdfs dfs -moveToLocal /tp_hdfs/data/fichier2.txt .
```

3.7 Lecture et exploration des fichiers HDFS

```
# Afficher le contenu
hdfs dfs -cat /tp_hdfs/data/fichier.txt
hdfs dfs -text /tp_hdfs/data/fichier.gz
```

```
# Lire seulement une partie
hdfs dfs -tail /tp_hdfs/data/fichier.txt
```

```
# Compter lignes, mots, octets
hdfs dfs -count /tp_hdfs/data
hdfs dfs -du -h /tp_hdfs/data
```

3.8 Manipulation avancée

```
# Déplacer / Renommer un fichier
hdfs dfs -mv /tp_hdfs/data/fichier.txt /tp_hdfs/data/input/fichier_renomme.txt

# Copier un fichier à l'intérieur d'HDFS
hdfs dfs -cp /tp_hdfs/data/fichier.txt /tp_hdfs/data/fichier_copy.txt

# Supprimer un fichier
hdfs dfs -rm /tp_hdfs/data/fichier_copy.txt

# Vérifier permissions et réplication
hdfs dfs -ls /tp_hdfs/data
hdfs dfs -stat %r /tp_hdfs/data/fichier.txt
```

4 Exercices pratiques

1. Créer un répertoire `/etudiants/nom` dans HDFS.
2. Copier deux fichiers locaux (`notes.csv`, `planning.csv`) vers `/etudiants/nom`.
3. Afficher leur contenu dans HDFS sans les rapatrier.
4. Déplacer `notes.csv` vers `/etudiants/nom/archives`.
5. Copier `planning.csv` en `planning_backup.csv`.
6. Supprimer `planning.csv`.
7. Télécharger `planning_backup.csv` vers votre `/home/user`.
8. Vérifier l'espace utilisé dans `/etudiants/nom`.