



## Pratique de Hadoop n°02

# Le traitement Batch avec Hadoop HDFS

### Table des matières

I.	Objectif du TP .....	2
II.	Hadoop et Docker.....	2
III.	Installation et configuration de l'image Docker : .....	2
IV.	Mémo des commandes HDFS : .....	5
A.	Commandes Shell courantes pour HDFS.....	5
1.	Quelques commandes courantes pour HDFS .....	5
V.	Suite du TP : Manipulation des commandes HDFS : .....	10
VI.	Interfaces web pour Hadoop.....	11

# I. Avant-Propos :

## Reprise du TP 01 Start Hadoop

# II. Objectif du TP

- Initiation au framework hadoop
- utilisation de docker
- Lancer un cluster hadoop de 3 noeuds.

# III. Hadoop et Docker

Pour déployer le Framework Hadoop, nous allons utiliser des conteneurs Docker. L'utilisation des conteneurs va garantir la consistance entre les environnements de développement et permettra de réduire considérablement la complexité de configuration des machines (dans le cas d'un accès natif) ainsi que la lourdeur d'exécution (si on opte pour l'utilisation d'une machine virtuelle).

# IV. Installation et configuration de l'image Docker :

Nous allons utiliser tout au long de ce TP trois conteneurs représentant respectivement :

- un noeud maître (Namenode)
- deux noeuds esclaves (Datanodes)

Vous devez pour cela avoir installé docker sur votre machine, et l'avoir correctement configuré.

Ouvrir la ligne de commande, et taper les instructions suivantes:

1. Télécharger l'image docker uploadée sur dockerhub:  
**docker pull liliastaxi/spark-hadoop:hv-2.7.2**
2. Créer les trois conteneurs à partir de l'image téléchargée. Pour cela: 2.1.  
Créer un réseau qui permettra de relier les trois conteneurs:  
**docker network create --driver=bridge hadoop**

3. Créer et lancer les trois conteneurs (les instructions -p permettent de faire un mapping entre les ports de la machine hôte et ceux du conteneur):

```
docker run -itd --net=hadoop -p 9070:50070 -p 8088:8088 -p 7077:7077 \  
-p 16010:16010 \  
--name hadoop-master --hostname hadoop-master \  
liliasfaxi/spark-hadoop:hv-2.7.2
```

```
docker run -itd -p 8040:8042 --net=hadoop \  
--name hadoop-slave1 --hostname hadoop-slave1 \  
liliasfaxi/spark-hadoop:hv-2.7.2
```

```
docker run -itd -p 8041:8042 --net=hadoop \  
--name hadoop-slave2 --hostname hadoop-slave2 \  
liliasfaxi/spark-hadoop:hv-2.7.2
```

4. Entrer dans le conteneur master pour commencer à l'utiliser.

```
docker exec -it hadoop-master bash
```

Le résultat de cette exécution sera le suivant:

```
root@hadoop-master:~#
```

Vous vous retrouverez dans le shell du namenode, et vous pourrez ainsi manipuler le cluster à votre guise. La première chose à faire, une fois dans le conteneur, est de lancer Hadoop et Yarn. Un script est fourni pour cela, appelé start-hadoop.sh.

Lancer ce script.

**./start-hadoop.sh**

Le résultat devra ressembler à ce qui suit:

```
root@hadoop-master:~# ./start-hadoop.sh
[
Starting namenodes on [hadoop-master]
hadoop-master: Warning: Permanently added 'hadoop-master,172.22.0.2' (ECDSA) to the list of known hosts.
hadoop-master: starting namenode, logging to /usr/local/hadoop/logs/hadoop-root-namenode-hadoop-master.out
hadoop-slave1: Warning: Permanently added 'hadoop-slave1,172.22.0.3' (ECDSA) to the list of known hosts.
hadoop-slave2: Warning: Permanently added 'hadoop-slave2,172.22.0.4' (ECDSA) to the list of known hosts.
hadoop-slave2: starting datanode, logging to /usr/local/hadoop/logs/hadoop-root-datanode-hadoop-slave2.out
hadoop-slave1: starting datanode, logging to /usr/local/hadoop/logs/hadoop-root-datanode-hadoop-slave1.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: Warning: Permanently added '0.0.0.0' (ECDSA) to the list of known hosts.
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-root-secondarynamenode-hadoop-master.out

starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn--resourcemanager-hadoop-master.out
hadoop-slave2: Warning: Permanently added 'hadoop-slave2,172.22.0.4' (ECDSA) to the list of known hosts.
hadoop-slave1: Warning: Permanently added 'hadoop-slave1,172.22.0.3' (ECDSA) to the list of known hosts.
hadoop-slave2: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-root-nodemanager-hadoop-slave2.out
hadoop-slave1: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-root-nodemanager-hadoop-slave1.out
```

Premiers pas avec Hadoop

Toutes les commandes interagissant avec le système Hadoop commencent par `hadoop fs`. Ensuite, les options rajoutées sont très largement inspirées des commandes Unix standard.

- Créer un répertoire dans HDFS, appelé input. Pour cela, taper:

**hadoop fs -mkdir -p input**

**Si vous avez une erreur :**

Si pour une raison ou une autre, vous n'arrivez pas à créer le répertoire input, avec un message ressemblant à ceci: `ls: `.`: No such file or directory`, veiller à construire l'arborescence de l'utilisateur principal (root), comme suit: **hadoop fs -mkdir -p /user/root**

## V. Mémo des commandes HDFS :

### A. Commandes Shell courantes pour HDFS

#### 1. Quelques commandes courantes pour HDFS

Pour ces commandes, il existe 2 syntaxes possibles:

- Avec hadoop: avec une syntaxe du type `hadoop fs <commande>`,
- Avec hdfs: la syntaxe est `hdfs dfs <commande>`.

Ces commandes sont proche de celles utilisées par le Shell linux comme `ls`, `mkdir`, `rm`, `cat`, etc...

a) *Pour lister le contenu d'un répertoire*

**`hdfs dfs -ls <chemin du répertoire>`**

Par exemple:

**`hdfs dfs -ls /`**

**`hdfs dfs -ls /user`** # pour voir le contenu du répertoire "user"

**Found 2 items**

**`-rw-r--r-- 1 hduser supergroup 3324334 2017-09-16 12:00`**  
**`/user/135-0.txt`**

**`-rw-r--r-- 1 hduser supergroup 3359550 2017-09-16 12:01`**  
**`/user/2600-0.txt`**

On peut utiliser aussi: **`hadoop fs -ls /user`**

b) *Pour afficher le contenu d'un fichier*

**`hdfs dfs -cat <chemin du fichier>`**

Par exemple:

**`hdfs dfs -cat /user/135-0.txt`**

On peut utiliser: **`hadoop fs -cat /user/135-0.txt`**



c) *Pour créer un répertoire*

**hdfs dfs -mkdir <chemin du nouveau répertoire>**

Par exemple:

**hdfs dfs -mkdir /user/output**

d) *Pour copier un fichier sur HDFS*

On peut utiliser:

**hdfs dfs -put <chemin du fichier source>  
<chemin du fichier destination sur HDFS>**

La commande suivante est réservé seulement au fichier locaux:

**hdfs dfs -copyFromLocal <chemin du fichier source>  
<chemin du fichier destination sur HDFS>**

Par exemple:

**hdfs dfs -put TextFile.txt /user**

ou

**hdfs dfs -copyFromLocal TextFile.txt /user**

Les syntaxes équivalentes avec hadoop sont possibles:

**hadoop fs -put <chemin du fichier source>  
<chemin du fichier destination sur HDFS>**

**hadoop fs -copyFromLocal <chemin du fichier source>  
<chemin du fichier destination sur HDFS>**

e) *Pour effectuer une copie de fichier*

```
hdfs dfs -cp <chemin du fichier source sur HDFS>  
<chemin du fichier destination sur HDFS>
```

Par exemple:

```
hdfs dfs -cp /user/TextFile.txt /user/output  
hdfs dfs -cp /user/TextFile.txt /user/TestFile2.txt
```

Avec hadoop:

```
hadoop fs -cp /user/TextFile.txt /user/output  
hadoop fs -cp /user/TextFile.txt /user/TestFile2.txt
```

f) *Pour récupérer un fichier sur HDFS*

```
hdfs dfs -get <chemin du fichier sur HDFS>  
<chemin du fichier en local>
```

Par exemple:

```
hdfs dfs -get /user/TextFile2.txt  
hdfs dfs -get /user/TextFile2.txt LocalTextFile2.txt
```

Cette syntaxe est réservée aux fichiers locaux:

```
hdfs dfs -copyToLocal /user/TextFile2.txt
```

ou

```
hadoop fs -get /user/TextFile2.txt  
hadoop fs -copyToLocal /user/TextFile2.txt
```



Les mêmes syntaxes existent pour effectuer des déplacements:

**hdfs dfs -moveToLocal** pour déplacer de HDFS  
vers le volume local

**hdfs dfs -moveFromLocal** pour déplacer du  
volume local vers HDFS

**hdfs dfs -mv** pour effectuer des déplacements  
dans HDFS

g) *Pour supprimer un fichier*

**hdfs dfs -rm <chemin du fichier sur HDFS>**

Par exemple:

**hdfs dfs -rm /user/TextFile2.txt**

**Deleted /user/TextFile2.txt**

ou

**hadoop fs -rm /user/TextFile2.txt**





h) *Pour supprimer un répertoire*

Si le répertoire est vide, on peut utiliser comme sur le Shell rmdir:

```
hdfs dfs -rmdir <chemin du répertoire vide>
```

Par exemple:

```
hdfs dfs -rmdir /user/output2
```

Si le répertoire contient des fichiers:

```
hdfs dfs -rm -r <chemin du répertoire>
```

Par exemple:

```
hdfs dfs -rm -r /user/output
```

Avec hadoop:

```
hadoop fs -rmdir /user/output2
```

```
hadoop fs -rm -r /user/output
```

## VI. Suite du TP : Manipulation des commandes HDFS :

1. Nous allons utiliser le fichier **purchases.txt** comme entrée pour les futurs traitements MapReduce. Ce fichier se trouve déjà sous le répertoire principal de votre machine master.

2. Charger le fichier purchases dans le répertoire input que vous avez créé:

**hadoop fs -put purchases.txt input**

3. Pour afficher le contenu du répertoire input, la commande est:

**hadoop fs -ls input**

4. Pour afficher les dernières lignes du fichier purchases:

**hadoop fs -tail input/purchases.txt**

5. Le résultat suivant va donc s'afficher:

```
[root@hadoop-master:~# hadoop fs -tail input/purchases.txt
31      17:59  Norfolk Toys      164.34  MasterCard
2012-12-31    17:59  Chula Vista      Music   380.67  Visa
2012-12-31    17:59  Hialeah Toys    115.21  MasterCard
2012-12-31    17:59  Indianapolis    Men's Clothing  158.28  MasterCard
2012-12-31    17:59  Norfolk Garden  414.09  MasterCard
2012-12-31    17:59  Baltimore      DVDs    467.3   Visa
2012-12-31    17:59  Santa Ana      Video Games   144.73  Visa
2012-12-31    17:59  Gilbert Consumer Electronics  354.66  Discover
2012-12-31    17:59  Memphis Sporting Goods  124.79  Amex
2012-12-31    17:59  Chicago Men's Clothing  386.54  MasterCard
2012-12-31    17:59  Birmingham     CDs     118.04  Cash
2012-12-31    17:59  Las Vegas      Health and Beauty  420.46  Amex
2012-12-31    17:59  Wichita Toys   383.9   Cash
2012-12-31    17:59  Tucson Pet Supplies  268.39  MasterCard
2012-12-31    17:59  Glendale      Women's Clothing   68.05  Amex
2012-12-31    17:59  Albuquerque    Toys    345.7   MasterCard
2012-12-31    17:59  Rochester     DVDs    399.57  Amex
2012-12-31    17:59  Greensboro     Baby    277.27  Discover
2012-12-31    17:59  Arlington     Women's Clothing  134.95  MasterCard
2012-12-31    17:59  Corpus Christi DVDs    441.61  Discover
root@hadoop-master:~#
```

## VII. Interfaces web pour Hadoop

Comme vous le savez : Hadoop offre plusieurs interfaces web pour pouvoir observer le comportement de ses différentes composantes. Vous pouvez afficher ces pages en local sur votre machine grâce à l'option `-p` de la commande **docker run**. En effet, cette option permet de publier un port du conteneur sur la machine hôte. Pour pouvoir publier tous les ports exposés, vous pouvez lancer votre conteneur en utilisant l'option `-P`.

En regardant le contenu du fichier **start-container.sh** fourni dans le projet, vous verrez que deux ports de la machine maître ont été exposés:

- Le port 9070: qui permet d'afficher les informations de votre namenode.
- Le port 8088: qui permet d'afficher les informations du Resource Manager de Yarn et visualiser le comportement des différents jobs.

Une fois votre cluster lancé et prêt à l'emploi, vous pouvez, sur votre navigateur préféré de votre machine hôte, aller à : `http://localhost:9070`. Vous obtiendrez le résultat suivant:

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities

### Overview 'hadoop-master:9000' (active)

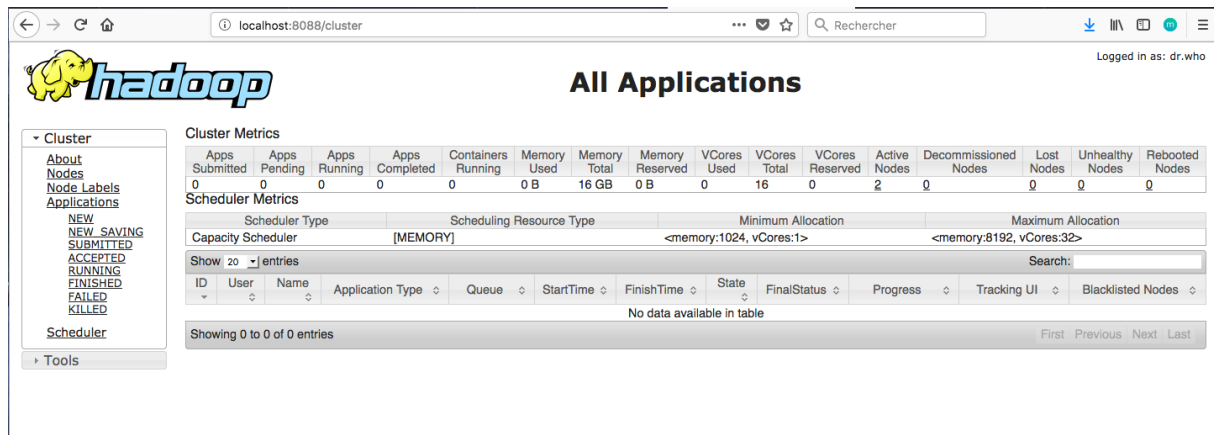
Started:	Fri Jan 26 11:47:09 UTC 2018
Version:	2.7.2, rUnknown
Compiled:	2016-05-27T18:05Z by root from Unknown
Cluster ID:	CID-3c662456-d44e-4301-bc39-28e479c4dc88
Block Pool ID:	BP-431089505-172.17.0.2-1465730089024

### Summary

Security is off.  
Safemode is off.  
20 files and directories, 8 blocks = 28 total filesystem object(s).  
Heap Memory used 85.33 MB of 165.5 MB Heap Memory. Max Heap Memory is 889 MB.  
Non Heap Memory used 37.42 MB of 38.44 MB Committed Non Heap Memory. Max Non Heap Memory is 214 MB.

Configured Capacity:	125.49 GB
DFS Used:	406.85 MB (0.32%)
Non DFS Used:	34.1 GB

Vous pouvez également visualiser l'avancement et les résultats de vos Jobs (Map Reduce ou autre) en allant à l'adresse: <http://localhost:8088>



The screenshot shows the Hadoop YARN web interface at localhost:8088/cluster. The page title is "All Applications". On the left, there is a sidebar with a "Cluster" menu containing links for "About", "Nodes", "Node Labels", "Applications", and "Scheduler". The "Applications" link is selected. The main content area displays "Cluster Metrics" and "Scheduler Metrics".

**Cluster Metrics**

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
0	0	0	0	0	0 B	16 GB	0 B	0	16	0	2	0	0	0	0

**Scheduler Metrics**

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:8192, vCores:32>

Below the scheduler metrics, there is a table with columns: ID, User, Name, Application Type, Queue, StartTime, FinishTime, State, FinalStatus, Progress, Tracking UI, and Blacklisted Nodes. The table is currently empty, showing "Showing 0 to 0 of 0 entries".