



TP n°02 : Le traitement Batch avec Hadoop HDFS

Sommaire

- I. Avant-Propos
- II. Objectif du TP
- III. Hadoop et Docker
- IV. Installation et configuration de l'image Docker
- V. Mémo des commandes HDFS
- VI. Suite du TP : Manipulation des commandes HDFS
- VII. Interfaces web pour Hadoop

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 une VM avec trois conteneurs représentant respectivement :

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

1. ouvrez un terminal ou votre application pour faire une connexion SSH.
2. Entrez les informations fournis : nom_machine, pwd_machine, port_public de la machine associé au port privé 22 (SSH)
3. Entrer dans le conteneur master pour commencer à l'utiliser. **./bash_hadoop_master.sh** (*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

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...

1. 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`

2. Pour afficher le contenu d'un fichier : `hdfs dfs -cat <chemin_src>`

Par exemple:

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

3. Pour créer un répertoire : `hdfs dfs -mkdir <chemin_src>`

Par exemple:

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

4. Pour copier un fichier sur HDFS : `hdfs dfs -put <chemin_src> <chemin_dest_HDFS>`

La commande suivante est réservé seulement au fichier locaux: `hdfs dfs -copyFromLocal <chemin_src> <chemin_dest_HDFS>`

Par exemple:

- `hdfs dfs -put TextFile.txt /user`
- `hdfs dfs -copyFromLocal TextFile.txt /user`

Avec hadoop:

- `hadoop fs -put <chemin_src> <chemin_dest_HDFS>`
- `hadoop fs -copyFromLocal <chemin_src> <chemin_dest_HDFS>`

5. Pour effectuer une copie de fichier : `hdfs dfs -cp <chemin_src> <chemin_dest_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`

6. Pour récupérer un fichier sur HDFS : `hdfs dfs -get <chemin_src> <chemin_dest_HDFS>`

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

Les mêmes syntaxes existent pour effectuer des déplacements:

- *pour déplacer de HDFS vers le volume local* : `hdfs dfs -moveToLocal`
- *pour déplacer du volume local vers HDFS* : `hdfs dfs -moveFromLocal`
- *pour effectuer des déplacements dans HDFS* : `hdfs dfs -mv`

7. Pour supprimer un fichier

- `hdfs dfs -rm <chemin_dest_HDFS>`

Par exemple:

- `hdfs dfs -rm /user/TextFile2.txt`
- `Deleted /user/TextFile2.txt`
- `hadoop fs -rm /user/TextFile2.txt`

8. Pour supprimer un répertoire

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

- `hdfs dfs -rmdir <chemin_dir_empty>`

Par exemple:

- `hdfs dfs -rmdir /user/output2`

Si le répertoire contient des fichiers:

- `hdfs dfs -rm -r <chemin_dir>`

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:

HadoopOverviewDatanodesDatanode Volume FailuresSnapshotStartup ProgressUtilities

Overview'hadoop-master:9000' (active)

Started:Fri Jan 26 11:47:09 UTC 2018

Version:2.7.2, rUnknown

Compiled:2018-05-27T18:05Z by root from Unknown

Cluster ID:CID-3c662456-d44e-4301-bc39-28e479e4dc88

Block Pool ID:BP-431089505-172.17.0.2-1465730089024

Summary

Security is off.

Satellite is off.

20 files and directories, 9 blocks = 28 total filesystem object(s).

Heap Memory used 85.33 MB of 105.5 MB Heap Memory. Max Heap Memory is 980 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:

DFS Used:

Non DFS Used:

125.49 GB

406.85 MB (0.32%)

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>

←→↻🏠


localhost:8088/cluster

***🔖🌟🔍

Rechercher

⬇️🖨️📄⋮

Logged in as: dr.whe



All Applications

Cluster

AboutNodesNode LabelsApplicationsNEWNEW_SAVINGSUBMITTEDACCEPTEDRUNNINGFINISHEDFAILEDKILLEDSchedulerTools

Cluster Metrics

Apps Submitted:0

Apps Pending:0

Apps Running:0

Apps Completed:0

Containers Running:0

Memory Used:0 B

Memory Total:16 GB

Memory Reserved:0 B

VContainers Used:0

VContainers Total:16

VContainers Reserved:0

Active Nodes:2

Decommissioned Nodes:0

Lost Nodes:0

Unhealthy Nodes:0

Rebooted Nodes:0

Scheduler Metrics

Scheduler Type:Capacity Scheduler

Scheduling Resource Type:[MEMORY]

Minimum Allocation:<memory:1024, vCores:1>

Maximum Allocation:<memory:8192, vCores:32>

Show 20 entries

Search:

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI	Blacklisted Nodes
No data available in table											

Showing 0 to 0 of 0 entries

FirstPreviousNextLast

8 / 8