

Using Scala version 2.12.15, Java HotSpot(TM) 64-Bit Server VM, 1.8.0_66
Branch HEAD

Hdfs -version

java version "1.8.0_66"

Java(TM) SE Runtime Environment (build 1.8.0_66-b17)

Java HotSpot(TM) 64-Bit Server VM (build 25.66-b17, mixed mode)

4. Configuration hadoop

Dans le dossier %HADOOP_HOME%\etc\hadoop, vérifier que les fichiers ci-dessous ont la structure suivante:

core-site.xml

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```

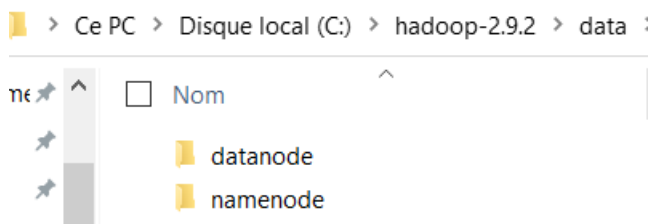
On a spécifié ici le nom du système de fichier. Tous les répertoires et fichiers HDFS seront donc préfixés par hdfs://localhost:9000.

mapred-site.xml

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

Ici nous avons précisé que nous allons utiliser YARN comme implémentation de MapReduce.

Maintenant, on crée un dossier **data** sous C:/hadoop-***, et sous ce dernier, on crée deux dossiers **datanode** et **namenode**.



hdfs-site.xml

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>C:\hadoop-2.9.2\data\namenode</value>
    <final>true</final>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>C:\hadoop-2.9.2\data\datanode</value>
    <final>true</final>
  </property>
</configuration>
```

Le fichier etc/hadoop/hdfs-site.xml contient les paramètres spécifiques au système de fichiers HDFS. On doit aussi paramétrer **YARN** via le fichier etc/hadoop/yarn-site.xml

yarn-site.xml

```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
  </property>
</configuration>
```

Configurer les paramètres l'emplacement de Java dans le fichier **etc/hadoop/hadoop-env.cmd**.
Remplacer set JAVA_HOME=%JAVA_HOME% par

hadoop-env.cmd

```
set JAVA_HOME=C:\PROGRA~1\Java\jdk1.8.0_66
```

5. **Exécution** : Tout d'abord, on ouvre une invite de commande en mode administrateur et on tape la commande suivante **hdfs namenode -format**, qui permet de formater le système de fichiers HDFS local.

hdfs namenode -format

6. **Lancer l'environnement hadoop**

Start-dfs (pour lancer Hadoop)

Start-yarn (pour lancer yarn)

Ou **start-all.cmd (pour lancer l'ensemble)**

start-all.cmd

Pour lancer spark

spark-shell

7. **Vérification** : Utiliser l'outil jps pour lister les processus Java en cours d'exécution

jps

17328 DataNode

23792

26720 ResourceManager

27072 Jps

28224 SparkSubmit

31704 NodeManager

3672 NameNode

Interfaces UI Nécessaires

<http://127.0.0.1:8088/> → Hadoop Yarn

<http://localhost:50075/datanode.html/> → Hadoop Data nodes

<http://127.0.0.1:4040/> → Spark UI

Hadoop offre plusieurs interfaces web pour pouvoir observer le comportement de ses différentes composantes. Le port **8088** permet d'afficher les informations du resource manager de Yarn et visualiser le comportement des différents jobs(avancement et résultat) en allant à l'adresse **<http://localhost:8088/cluster>** comme il montre la figure ci-dessous.

←

→

↺

localhost:8088/cluster

🔍

90%


☆

🔒

⬇️

🔗

⋮



All Applications

Logged in as: drw

▼ Cluster

About

Nodes

Node Labels

Applications

NEW

NEW_SAVING

SUBMITTED

ACCEPTED

RUNNING

FINISHED

FAILED

KILLED

Scheduler

Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved
0	0	0	0	0	0 B	0 B	0 B	0	0	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes	Shutdown Nodes
0	0	0	0	1	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

Show 20 ▼ entries

Search

ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCoers	Allocated Memory MB	Reserved CPU VCoers	Reserved Memory MB	% of Queue	% of Cluster	Progress	Tracking UI	Blacklisted Nodes
No data available in table																			

Showing 0 to 0 of 0 entries

First Previous Next Last

Le port **50070** qui permet d’afficher les informations de votre namenode en consultant l’adresse <http://localhost:50070>, comme il montre la figure ci-dessous.

localhost:50070/dfshealth.html#tab-overview

Hadoop

Overview

Datanodes

Datanode Volume Failures

Snapshot

Startup Progress

Utilities

Overview 'localhost:9000' (active)

Started:	Mon Oct 30 11:22:33 +0100 2023
Version:	2.9.2, r826afbeae31ca687bc2f8471dc841b66ed2c6704
Compiled:	Tue Nov 13 13:42:00 +0100 2018 by ajsaka from branch-2.9.2
Cluster ID:	CID-17584a5c-e96a-4a72-a9df-35567cb6dbfd
Block Pool ID:	BP-1119926176-192.168.56.1-1698661314020

Summary

Security is off.

Safemode is off.

1 files and directories, 0 blocks = 1 total filesystem object(s).

Heap Memory used 42.59 MB of 307 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 40.02 MB of 40.84 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	237.15 GB
----------------------	-----------

Rappel : les commandes de Hadoop

hadoop fs -ls	Afficher le contenu du répertoire racine
Hadoop fs -put file.txt	Upload un fichier dans hadoop (à partir du répertoire courant linux)
hadoop fs -get file.txt	Download un fichier à partir de hadoop sur votre disque local
hadoop fs -tail file.txt	Lire les dernières lignes du fichier
hadoop fs -cat file.txt	Affiche tout le contenu du fichier
hadoop fs -mv file.txt newfile.txt	Renommer le fichier
hadoop fs -rm newfile.txt	Supprimer le fichier
hadoop fs -mkdir /myinput	Créer un répertoire
hadoop fs -cat file.txt less	Lire le fichier page par page
hadoop fs -help	Affioche l’aide
hadoop fs -help du	L’aide sur la commande du

Exercice : Travailler avec Hadoop / HDFS et MapReduce

Création des fichiers de démonstration .txt dans le système de fichiers local, afin de le mettre dans l'outil de ligne de commande hdfs.

1. Lister le contenu de la racine HDFS

Exécuter une commande de système de fichiers sur le système de fichiers pris en charge dans Hadoop.

`hadoop fs ou hdfs dfs`

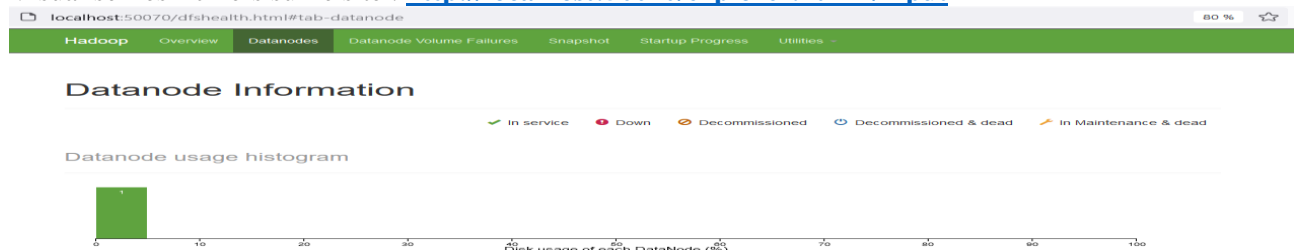
2. Créer le dossier `/user/input` sur la machine virtuelle HDFS
3. Copier les fichiers (*.txt) du rep. Hadoop dans le système de fichiers local, afin de le mettre sur la machine virtuelle HDFS

`hdfs dfs -copyFromLocal %HADOOP_HOME%*.txt /user/input`

4. Afficher le contenu du dossier HDFS `/user/input`
5. Afficher le contenu du fichier `notice.txt`
6. Afficher la taille du fichier `notice.txt`

Pour afficher la taille de l'ensemble des fichiers : **`hdfs dfs -du /input`**

7. Visualiser les fichiers sur le site : <http://localhost:50070/explorer.html#/input>



Dans ce cas, il n'y a qu'un seul nœud (notre ordinateur)

8. Récupérer la taille d'un block HDFS: **`hdfs getconf -confKey dfs.blocksize`**
9. Récupérer le facteur de réplication: **`hdfs getconf -confKey dfs.replication`**
10. Utiliser la commande `hdfs fsck` pour afficher un rapport détaillé sur les fichiers dans HDFS.

`hdfs fsck /input`

- Quel est le nombre de fichier dans hdfs?
- Quel est le nombre de blocs?
- Quel est le nombre de blocs corrompus?
- Quel est le facteur de réplication (Default replication factor)?
- Quel est le nombre de data-nodes contenant les blocs des fichiers du dossier HDFS?
- Quel est le nombre de racks ?

11. Modifier le facteur de réplication, valeur 2, du fichier `notice.txt` dans HDFS:

`hadoop fs -setrep -w 2 /input/notice.txt`

ou `hdfs dfs -setrep -w 2 /input/notice.txt`

N.B: Cette commande prendra beaucoup de temps à s'exécuter si le fichier est volumineux

12. Utiliser la commande **`hdfs fsck`** pour afficher un rapport détaillé sur le dossier HDFS `/user/input`

- Quel est le nombre de blocs?
- Quel est le facteur de réplication (Default replication factor) ?

13. Compter le nombre total de mots des fichiers *.txt disponible dans le répertoire d'entrée `/user/input`, et enregistrer la sortie dans le fichier **`output/part-r00000`**

WordCount est un exemple très simple, l'équivalent du HelloWorld pour les applications de traitement de données. Le Wordcount permet de calculer le nombre de mots dans un fichier donné, en décomposant le calcul en deux étapes (principe de l'algorithme map-reduce):

- L'étape de **Mapping**, qui permet de découper le texte en mots et de délivrer en sortie un flux textuel, où chaque ligne contient le mot trouvé, suivi de la valeur 1 (pour dire que le mot a été trouvé une fois)
- L'étape de **Reducing**, qui permet de faire la somme des 1 pour chaque mot, pour trouver le nombre total d'occurrences de ce mot dans le texte.

`hadoop jar %HADOOP_HOME%\share\hadoop\mapreduce\hadoop-mapreduce-examples-2.9.2.jar`

`wordcount /user/input/*.txt /output`

14. Vérifier les résultats du mapreduce

`hadoop fs -ls /output`

`hadoop fs -cat /output/part-r-00000`