Module : Big Data & Bases de Données NoSQL (M236)

Filière : DUT – Systèmes Informatiques et Réseaux (SIR – S3)

Année universitaire : 2025 / 2026

Compte Rendu de l'Atelier N°02

Manipulation avancée de HDFS et MapReduce sous Docker

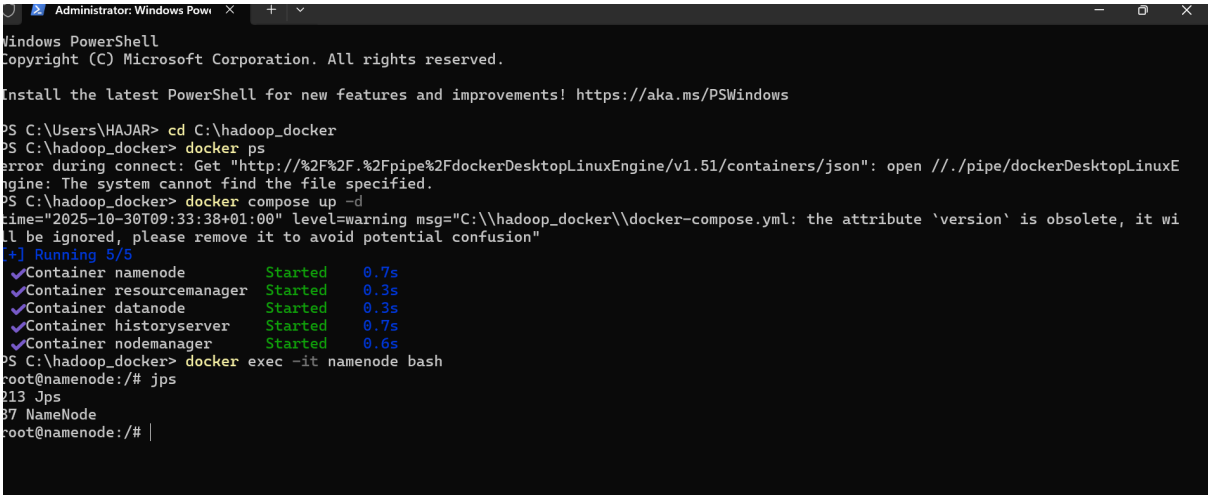
Réalisé par : **Hajar Rachid** (*N°64*)

Introduction:

Cet atelier a pour objectif de se familiariser avec le fonctionnement du système de fichiers distribué **HDFS** et le modèle de traitement parallèle **MapReduce** dans un environnement Hadoop déployé sous **Docker**.

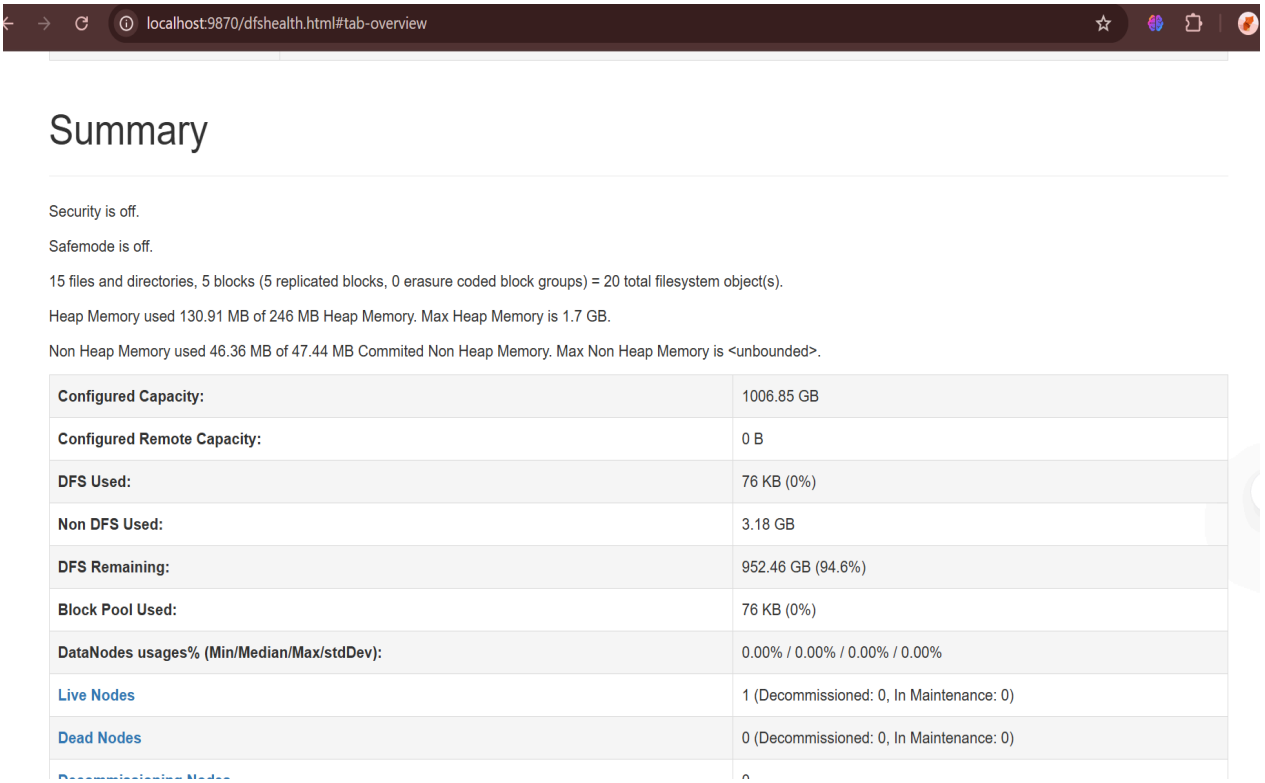
À travers plusieurs étapes, nous avons appris à manipuler les fichiers et répertoires dans HDFS, à observer la **réplication** des données, à exécuter des jobs **MapReduce** (tels que *WordCount* et *grep*), et à analyser les résultats obtenus via les interfaces web d'Hadoop. Les exercices pratiques nous ont également permis de modifier la configuration du cluster, de tester la réplication et de comprendre le comportement d'Hadoop lors de l'exécution de traitements distribués.

Partie 1: Vérification et préparation de l'environnement

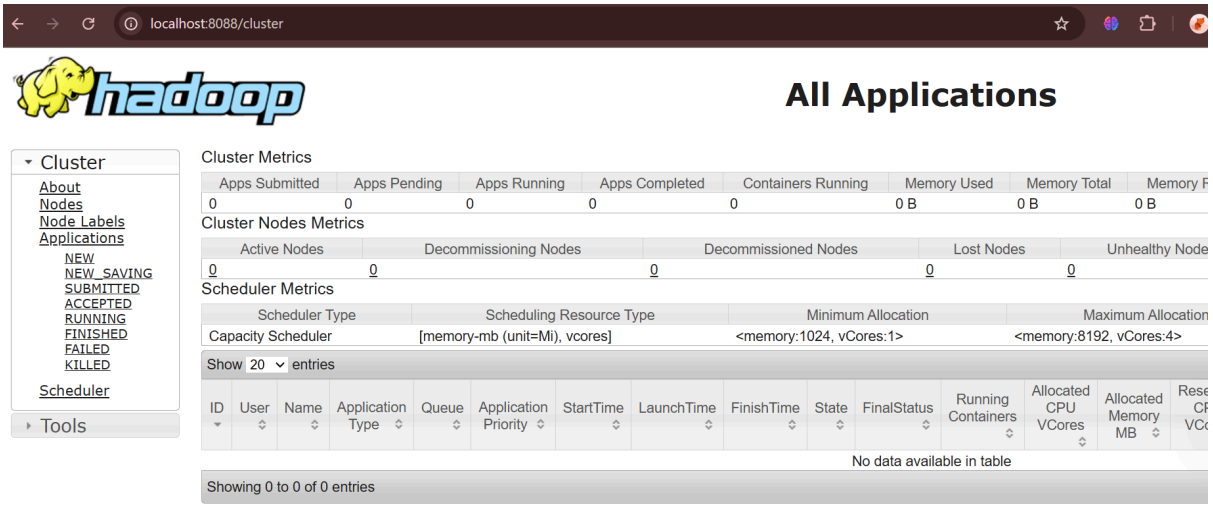


➔ Étape 4 : Accès aux interfaces Web

- **NameNode UI :**



• **YARN UI :**



Partie 2 — Manipulation du système de fichiers HDFS

→ Étape 1 : Créer une arborescence de travail:

```
PS C:\hadoop_docker> docker exec -it namenode bash
root@namenode:/# jps
213 Jps
87 NameNode
root@namenode:/# hdfs dfs -mkdir -p /user/etudiant/hdfs_test
root@namenode:/# hdfs dfs -mkdir -p /data/textes
root@namenode:/# hdfs dfs -ls /
Found 5 items
drwxr-xr-x   - root supergroup          0 2025-10-23 15:15 /data
drwxrwxrwx   - root root                0 2025-10-23 14:37 /input
drwxr-xr-x   - root supergroup          0 2025-10-23 15:15 /user
drwxr-xr-x   - root supergroup          0 2025-10-23 14:47 /wordcount_input
drwxr-xr-x   - root supergroup          0 2025-10-23 14:51 /wordcount_output
root@namenode:/#
```

→ Étape 2 : Ajouter des fichiers dans HDFS :

```
drwxr-xr-x   - root supergroup          0 2025-10-23 14:51 /wordcount_output
e1.txtamenode:/# echo "Hadoop est un système distribué" > texte
root@namenode:/# echo "Le Big Data repose sur HDFS" > texte2.txt
root@namenode:/# hdfs dfs -put -f texte1.txt texte2.txt /data/textes
2025-10-30 08:37:57,368 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted =
false
2025-10-30 08:37:58,303 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted =
false
root@namenode:/#
```

→ Étape 3 : Explorer le contenu :

```
false
root@namenode:/# hdfs dfs -ls /data/textes
Found 2 items
-rw-r--r--   3 root supergroup         34 2025-10-30 08:37 /data/textes/texte1.txt
-rw-r--r--   3 root supergroup         28 2025-10-30 08:37 /data/textes/texte2.txt
root@namenode:/# hdfs dfs -cat /data/textes/texte1.txt
2025-10-30 08:39:46,064 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted =
false
Hadoop est un système distribué
root@namenode:/#
```

→ Étape 4 : Vérifier la réplication

```
Hadoop est un système distribué
root@namenode:/# hdfs fsck /data/textes/texte1.txt -files -blocks -locations
Connecting to namenode via http://namenode:9870/fsck?ugi=root&files=1&blocks=1&locations=1&path=%2Fdata%2Ftextes%2Ftexte1.txt
FSCK started by root (auth:SIMPLE) from /172.18.0.2 for path /data/textes/texte1.txt at Thu Oct 30 08:40:35 UTC 2025
/data/textes/texte1.txt 34 bytes, replicated: replication=3, 1 block(s):  Under replicated BP-344549181-172.18.0.2-1760719874156:blk_
1073741830_1006. Target Replicas is 3 but found 1 live replica(s), 0 decommissioned replica(s), 0 decommissioning replica(s).
0. BP-344549181-172.18.0.2-1760719874156:blk_1073741830_1006 len=34 Live_repl=1  [DatanodeInfoWithStorage[172.18.0.4:9866,DS-090a4567
-2a85-498b-85db-1e7e7cf4ec82,DISK]]


Status: HEALTHY
 Number of data-nodes:  1
 Number of racks:               1
 Total dirs:                    0
 Total symlinks:                0

Replicated Blocks:
 Total size:    34 B
 Total files:   1
 Total blocks (validated):      1 (avg. block size 34 B)
 Minimally replicated blocks:   1 (100.0 %)
 Over-replicated blocks:        0 (0.0 %)
 Under-replicated blocks:       1 (100.0 %)
 Mis-replicated blocks:         0 (0.0 %)
 Default replication factor:    3
 Average block replication:     1.0
 Missing blocks:                0
 Corrupt blocks:                0
 Missing replicas:              2 (66.666664 %)

Erasure Coded Block Groups:
 Total size:    0 B
 Total files:   0
 Total block groups (validated):        0
 Minimally erasure-coded block groups:  0
```

Partie 3 — Commandes HDFS essentielles

→ Copie et déplacement :

```
The filesystem under path '/data/textes/texte1.txt' is HEALTHY
root@namenode:/# hdfs dfs -cp /data/textes/texte1.txt /user/etudiant/hdfs_test/
2025-10-30 08:41:26,834 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted =
false
2025-10-30 08:41:26,932 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted =
false
root@namenode:/# hdfs dfs -mv /user/etudiant/hdfs_test/texte1.txt /data/
root@namenode:/#
```

→ Suppression :

```
root@namenode:/# hdfs dfs -mv /user/etudiant/hdfs_test/texte1.txt /data/
root@namenode:/# hdfs dfs -rm /data/textes/texte2.txt
Deleted /data/textes/texte2.txt
root@namenode:/#
```

→ Informations sur l'espace :

```
root@namenode:/# hdfs dfs -rm /data/textes/texte2.txt
Deleted /data/textes/texte2.txt
root@namenode:/# hdfs dfs -du -h /
68  204  /data
15  45   /input
0   0    /user
57  171  /wordcount_input
58  174  /wordcount_output
root@namenode:/# hdfs dfsadmin -report
Configured Capacity: 1081101176832 (1006.85 GB)
Present Capacity: 1022692352006 (952.46 GB)
DFS Remaining: 1022692274176 (952.46 GB)
DFS Used: 77830 (76.01 KB)
DFS Used%: 0.00%
Replicated Blocks:
        Under replicated blocks: 5
        Blocks with corrupt replicas: 0
        Missing blocks: 0
        Missing blocks (with replication factor 1): 0
        Low redundancy blocks with highest priority to recover: 5
        Pending deletion blocks: 0
Erasure Coded Block Groups:
        Low redundancy block groups: 0
        Block groups with corrupt internal blocks: 0
        Missing block groups: 0
        Low redundancy blocks with highest priority to recover: 0
        Pending deletion blocks: 0

-------------------------------------------------
Live datanodes (1):

Name: 172.18.0.4:9866 (datanode.hadoop_docker_hadoop-net)
Hostname: datanode
Decommission Status : Normal
Configured Capacity: 1081101176832 (1006.85 GB)
```

→ Changer les permissions :

```
root@namenode:/# hdfs dfs -chmod -R 755 /data
root@namenode:/# hdfs dfs -chown -R root:root /data
root@namenode:/#
```

Partie 4 — Lancer et analyser des jobs MapReduce

→ Étape 1 : Job WordCount (compte de mots) :

```
root@namenode:/# hdfs dfs -mkdir -p /wordcount_input
root@namenode:/# echo "Hadoop simplifie le traitement des grandes données" >
wc1.txt
umes" > wc2.txt# echo "MapReduce permet d'analyser de gros volu
root@namenode:/# hdfs dfs -put -f wc1.txt wc2.txt /wordcount_input
2025-10-30 08:45:03,597 INFO sasl.SaslDataTransferClient: SASL encryption tru
st check: localHostTrusted = false, remoteHostTrusted = false
2025-10-30 08:45:03,675 INFO sasl.SaslDataTransferClient: SASL encryption tru
st check: localHostTrusted = false, remoteHostTrusted = false
root@namenode:/#
```

Supprimer la sortie précédente :

```
root@namenode:/# hdfs dfs -rm -r /wordcount_output
Deleted /wordcount_output
root@namenode:/#
```

Exécuter le job :

```
root@namenode:/# hadoop jar /opt/hadoop-3.2.1/share/hadoop/mapreduce/hadoop-mapreduceexamples-3.2.1.jar \
> wordcount /wordcount_input /wordcount_output
JAR does not exist or is not a normal file: /opt/hadoop-3.2.1/share/hadoop/mapreduce/hadoop-mapreduceexamples-3.2.1.jar
root@namenode:/#
```

Afficher le résultat :

```
false
root@namenode:/# hdfs dfs -rm -r /wordcount_output
Deleted /wordcount_output
root@namenode:/# hadoop jar /opt/hadoop-3.2.1/share/hadoop/mapreduce/hadoop-mapreduceexamples-3.2.1.jar \
> wordcount /wordcount_input /wordcount_output
JAR does not exist or is not a normal file: /opt/hadoop-3.2.1/share/hadoop/mapreduce/hadoop-mapreduceexamples-3.2.1.jar
root@namenode:/# hdfs dfs -cat /wordcount_output/part-r-00000
cat: `/wordcount_output/part-r-00000': No such file or directory
root@namenode:/# hdfs dfs -rm -r /grep_output
rm: `/grep_output': No such file or directory
root@namenode:/# hadoop jar /opt/hadoop-3.2.1/share/hadoop/mapreduce/hadoop-mapreduceexamples-3.2.1.jar \
> grep /wordcount_input /grep_output "Hadoop"
JAR does not exist or is not a normal file: /opt/hadoop-3.2.1/share/hadoop/mapreduce/hadoop-mapreduceexamples-3.2.1.jar
root@namenode:/# hdfs dfs -cat /grep_output/part-r-00000
cat: `/grep_output/part-r-00000': No such file or directory
root@namenode:/#
```

Partie 5 — Exercices pratiques

Exercice 1:

Apprendre à manipuler les fichiers et répertoires dans le système de fichiers HDFS. Cet exercice permet de comprendre les opérations essentielles : création de dossiers, ajout de fichiers, suppression et vérification de l'espace occupé

```
root@namenode:/# hdfs dfs -mkdir -p /etudiant/donnees
root@namenode:/# echo "Premier fichier" > a.txt
root@namenode:/# echo "Deuxième fichier" > b.txt
root@namenode:/# echo "Troisième fichier" > c.txt
root@namenode:/# hdfs dfs -put -f a.txt b.txt c.txt /etudiant/donnees
2025-10-30 08:51:14,824 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted =
false
2025-10-30 08:51:14,902 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted =
false
2025-10-30 08:51:15,326 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted =
false
root@namenode:/# hdfs dfs -du -h /etudiant/donnees
16  48  /etudiant/donnees/a.txt
18  54  /etudiant/donnees/b.txt
19  57  /etudiant/donnees/c.txt
root@namenode:/# hdfs dfs -rm /etudiant/donnees/a.txt
Deleted /etudiant/donnees/a.txt
root@namenode:/# hdfs dfs -ls /etudiant/donnees
Found 2 items
-rw-r--r--   3 root supergroup         18 2025-10-30 08:51 /etudiant/donnees/b.txt
-rw-r--r--   3 root supergroup         19 2025-10-30 08:51 /etudiant/donnees/c.txt
root@namenode:/#
```

Exercice 2 :

Comprendre le mécanisme de réplication et la tolérance aux pannes dans Hadoop. L'objectif est de modifier le facteur de réplication dans le fichier de configuration hdfs-site.xml, de redémarrer Hadoop, puis de vérifier la nouvelle politique de réplication appliquée aux fichiers.

```
root@namenode:/# cat >> /opt/hadoop-3.2.1/etc/hadoop/hdfs-site.x
ml <<EOL
> <property>
>   <name>dfs.replication</name>
>   <value>2</value>
> </property>
> EOL
root@namenode:/# cat /opt/hadoop-3.2.1/etc/hadoop/hdfs-site.xml
```

```
PS C:\hadoop_docker> docker compose down
time="2025-10-26T12:16:07+01:00" level=warning msg="C:\\hadoop_d
ocker\\docker-compose.yml: the attribute `version` is obsolete,
it will be ignored, please remove it to avoid potential confusio
n"
[+] Running 6/6
 ✔Container datanode                Removed                1.6s
 ✔Container nodemanager             Removed                0.1s
 ✔Container historyserver           Removed                1.6s
 ✔Container resourcemanager         Removed                1.5s
 ✔Container namenode                Removed                1.8s
 ✔Network hadoop_docker_hadoop-net  Removed                0.3s
PS C:\hadoop_docker> docker compose up -d
time="2025-10-26T12:16:18+01:00" level=warning msg="C:\\hadoop_d
ocker\\docker-compose.yml: the attribute `version` is obsolete,
it will be ignored, please remove it to avoid potential confusio
n"
[+] Running 6/6
 ✔Network hadoop_docker_hadoop-net  Created                0.1s
 ✔Container namenode                Started                1.9s
 ✔Container datanode                Started                2.1s
 ✔Container resourcemanager         Started                2.1s
 ✔Container historyserver           Started                2.3s
 ✔Container nodemanager             Started                2.2s
PS C:\hadoop_docker>
```

```
PS C:\hadoop_docker> docker exec -it namenode bash
root@namenode:/# hdfs fsck /data -files -blocks -locations

Connecting to namenode via http://namenode:9870/fsck?ugi=root&fi
les=1&blocks=1&locations=1&path=%2Fdata
FSCK started by root (auth:SIMPLE) from /172.18.0.2 for path /da
ta at Sun Oct 26 11:17:47 UTC 2025
/data <dir>
/data/texte1.txt 34 bytes, replicated: replication=3, 1 block(s)
:  Under replicated BP-1955090828-172.18.0.2-1760719912866:blk_1
073741830_1006. Target Replicas is 3 but found 1 live replica(s)
, 0 decommissioned replica(s), 0 decommissioning replica(s).
0. BP-1955090828-172.18.0.2-1760719912866:blk_1073741830_1006 le
n=34 Live_repl=1  [DatanodeInfoWithStorage[172.18.0.3:9866,DS-d0
8f746a-5f49-4c64-89b7-c1bc36030021,DISK]]

/data/textes <dir>
/data/textes/texte1.txt 34 bytes, replicated: replication=3, 1 b
lock(s):  Under replicated BP-1955090828-172.18.0.2-176071991286
6:blk_1073741828_1004. Target Replicas is 3 but found 1 live rep
lica(s), 0 decommissioned replica(s), 0 decommissioning replica(
s).
0. BP-1955090828-172.18.0.2-1760719912866:blk_1073741828_1004 le
n=34 Live_repl=1  [DatanodeInfoWithStorage[172.18.0.3:9866,DS-d0
8f746a-5f49-4c64-89b7-c1bc36030021,DISK]]
```

Exercice 3 :

Mettre en pratique l'exécution d'un job MapReduce afin de compter le nombre de mots présents dans des fichiers texte. L'exercice vise aussi à observer l'effet d'une modification du fichier d'entrée sur les résultats du job, pour mieux comprendre le fonctionnement du traitement distribué

```
root@namenode:/# hdfs dfs -mkdir -p /livres
root@namenode:/# echo "Chapitre 1: Hadoop est puissant" > livre1.txt
root@namenode:/# echo "Chapitre 2: MapReduce est utile" > livre2.txt
root@namenode:/# hdfs dfs -put -f livre1.txt livre2.txt /livres
2025-10-30 09:00:20,455 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted =
false
2025-10-30 09:00:20,533 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted =
false
root@namenode:/# hdfs dfs -rm -r /livres_output
rm: `/livres_output': No such file or directory
root@namenode:/# hadoop jar /opt/hadoop-3.2.1/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.1.jar wordcount /livres /livres_ou
tput
2025-10-30 09:00:43,373 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2025-10-30 09:00:43,434 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2025-10-30 09:00:43,434 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2025-10-30 09:00:43,692 INFO input.FileInputFormat: Total input files to process : 2
2025-10-30 09:00:43,718 INFO mapreduce.JobSubmitter: number of splits:2
2025-10-30 09:00:43,841 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local1786309763_0001
2025-10-30 09:00:43,841 INFO mapreduce.JobSubmitter: Executing with tokens: []
2025-10-30 09:00:43,970 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2025-10-30 09:00:43,972 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2025-10-30 09:00:43,973 INFO mapreduce.Job: Running job: job_local1786309763_0001
2025-10-30 09:00:43,985 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-10-30 09:00:43,985 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:f
alse, ignore cleanup failures: false
2025-10-30 09:00:43,986 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2025-10-30 09:00:44,022 INFO mapred.LocalJobRunner: Waiting for map tasks
2025-10-30 09:00:44,025 INFO mapred.LocalJobRunner: Starting task: attempt_local1786309763_0001_m_000000_0
2025-10-30 09:00:44,045 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-10-30 09:00:44,045 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:f
alse, ignore cleanup failures: false
2025-10-30 09:00:44,059 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
2025-10-30 09:00:44,063 INFO mapred.MapTask: Processing split: hdfs://namenode:9000/livres/livre1.txt:0+32
2025-10-30 09:00:44,160 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2025-10-30 09:00:44,160 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
```

```
root@namenode:/# hdfs dfs -cat /livres_output/part-r-00000
2025-10-30 09:00:57,032 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted =
false
1:      1
2:      1
Chapitre        2
Hadoop  1
MapReduce       1
est     2
puissant        1
utile   1
root@namenode:/# echo "Hadoop est vraiment puissant" > livre1.txt
root@namenode:/# hdfs dfs -put -f livre1.txt /livres
2025-10-30 09:01:16,500 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted =
false
root@namenode:/# hdfs dfs -rm -r /livres_output
Deleted /livres_output
root@namenode:/# hadoop jar /opt/hadoop-3.2.1/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.1.jar wordcount /livres /livres_ou
tput
2025-10-30 09:01:54,778 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2025-10-30 09:01:54,821 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2025-10-30 09:01:54,821 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2025-10-30 09:01:55,018 INFO input.FileInputFormat: Total input files to process : 2
2025-10-30 09:01:55,033 INFO mapreduce.JobSubmitter: number of splits:2
2025-10-30 09:01:55,109 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local315776936_0001
2025-10-30 09:01:55,109 INFO mapreduce.JobSubmitter: Executing with tokens: []
2025-10-30 09:01:55,186 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2025-10-30 09:01:55,187 INFO mapreduce.Job: Running job: job_local315776936_0001
2025-10-30 09:01:55,187 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2025-10-30 09:01:55,193 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-10-30 09:01:55,193 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:f
alse, ignore cleanup failures: false
2025-10-30 09:01:55,194 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2025-10-30 09:01:55,226 INFO mapred.LocalJobRunner: Waiting for map tasks
2025-10-30 09:01:55,226 INFO mapred.LocalJobRunner: Starting task: attempt_local315776936_0001_m_000000_0
```

```
                Combine output records=9
                Reduce input groups=8
                Reduce shuffle bytes=127
                Reduce input records=9
                Reduce output records=8
                Spilled Records=18
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=10
                Total committed heap usage (bytes)=1171783680
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=61
        File Output Format Counters
                Bytes Written=73
root@namenode:/# hdfs dfs -cat /livres_output/part-r-00000
2025-10-30 09:02:08,166 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted =
false
2:      1
Chapitre        1
Hadoop  1
MapReduce       1
est     2
puissant        1
utile   1
vraiment        1
root@namenode:/#
```

Exercice 4 :

Utiliser le programme “grep” fourni par Hadoop pour rechercher les lignes contenant un mot spécifique (ici “données”). Ce travail permet d’illustrer comment Hadoop peut effectuer des recherches textuelles efficaces sur de grands volumes de données stockées dans HDFS.

```
root@namenode:/# hdfs dfs -rm -r /grep_output
rm: `/grep_output': No such file or directory
root@namenode:/# hadoop jar /opt/hadoop-3.2.1/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.1.jar grep /livres /grep_output "d
onnées"
2025-10-30 09:04:24,508 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2025-10-30 09:04:24,552 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2025-10-30 09:04:24,552 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2025-10-30 09:04:24,730 INFO input.FileInputFormat: Total input files to process : 2
2025-10-30 09:04:24,744 INFO mapreduce.JobSubmitter: number of splits:2
2025-10-30 09:04:24,810 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local13426451_0001
2025-10-30 09:04:24,810 INFO mapreduce.JobSubmitter: Executing with tokens: []
2025-10-30 09:04:24,893 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2025-10-30 09:04:24,893 INFO mapreduce.Job: Running job: job_local13426451_0001
2025-10-30 09:04:24,894 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2025-10-30 09:04:24,898 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-10-30 09:04:24,898 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:f
alse, ignore cleanup failures: false
2025-10-30 09:04:24,899 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter
2025-10-30 09:04:24,928 INFO mapred.LocalJobRunner: Waiting for map tasks
2025-10-30 09:04:24,929 INFO mapred.LocalJobRunner: Starting task: attempt_local13426451_0001_m_000000_0
2025-10-30 09:04:24,942 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2025-10-30 09:04:24,942 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output directory:f
alse, ignore cleanup failures: false
2025-10-30 09:04:24,952 INFO mapred.Task:  Using ResourceCalculatorProcessTree : [ ]
2025-10-30 09:04:24,954 INFO mapred.MapTask: Processing split: hdfs://namenode:9000/livres/livre2.txt:0+32
2025-10-30 09:04:25,032 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)
2025-10-30 09:04:25,032 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100
```

```
                HDFS: Number of read operations=41
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=14
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Map input records=0
                Map output records=0
                Map output bytes=0
                Map output materialized bytes=6
                Input split bytes=128
                Combine input records=0
                Combine output records=0
                Reduce input groups=0
                Reduce shuffle bytes=6
                Reduce input records=0
                Reduce output records=0
                Spilled Records=0
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=0
                Total committed heap usage (bytes)=973078528
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=86
        File Output Format Counters
                Bytes Written=0
root@namenode:/# hdfs dfs -cat /grep_output/part-r-00000
root@namenode:/#
```

Brève conclusion :

À la fin de cet atelier, nous avons obtenu des résultats concrets confirmant la bonne configuration du cluster Hadoop sous Docker. Les différentes manipulations sur **HDFS** et l'exécution des **jobs MapReduce** ont permis de comprendre en profondeur le fonctionnement d'un système distribué. L'expérience a renforcé la compréhension du Big Data et de la manière dont Hadoop gère le stockage et le traitement des grandes quantités de données.