

25/09/2025

TP Hadoop

ESN01

PhilippeVita & Nour ZERABIB
ESN01

Page de garde

- Titre du document : **TP - Hadoop**
- Version : **1.00**
- Date : **25/09/2025**

Société ESN01 - Équipe 01

Membre 1	Philippe VITA
Membre 2	Nour ZERABIB

Table des matières

I.	Se connecter à la machine virtuelle Docker.....	4
1.	Modalité de connexion à la machine virtuelle.....	4
01.	Connexion SSH à l'aide de l'outil PuTTY.....	4
02.	Connexion SFTP à l'aide de l'outil FileZila	4
03.	Paramètres de connexion à renseigner	5
2.	Les étapes de connexion SSH à la machine virtuelle.....	6
01.	Chargement des paramètres de connexion est fait à l'aide du bouton de commande « Load ».....	6
02.	L'ouverture de session se fait à l'aide du bouton de commande « Open ».....	6
3.	Les étapes de connexion SFTP à la machine virtuelle	7
01.	Création d'un nouveau site	7
02.	Chargement des paramètres de connexion est fait à l'aide du bouton de commande « OK » 7	
03.	L'ouverture de session se fait à l'aide du bouton de commande « Connexion ».....	8
II.	Lancer les conteneurs et services Hadoop.....	9
1.	Services nécessaires	9
2.	Commandes pour démarrer les services.....	9
01.	Lancer le conteneur Maître (hadoop-master)	9
02.	Se connecter au conteneur Hadoop maître (hadoop-master)	10
03.	Démarrer les services Hadoop dans le conteneur Hadoop Master	11
III.	Importer les données dans HDFS	13
1.	Méthode d'importation des données	13
01.	Déposer le fichier CSV, source des données, dans le HDFS via l'outil FileZila.....	13
02.	Vérifier que le fichier est bien présent en local sur la machine virtuelle	14

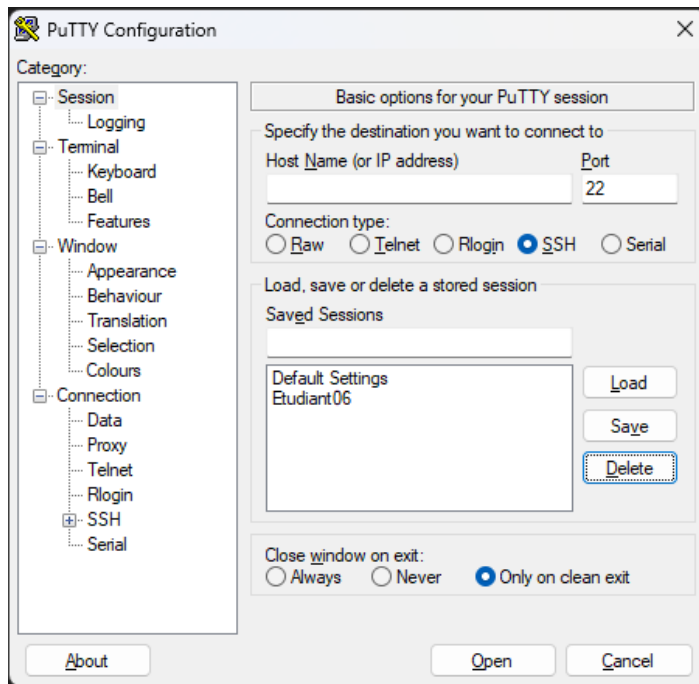
03.	Illustrer la commande d'import du fichier dans le HDFS :.....	15
IV.	Créer et exécuter un job MapReduce.....	17
1.	Structure du job « mapper ».....	17
2.	Structure du job « reducer ».....	18
3.	Commandes exécutables pour soumettre le Job MapReduce sur Hadoop Streaming	18
01.	Vérifier l'emplacement des Jar Streaming.....	18
02.	Application des droits d'exécution sur des jobs MapReduce.....	19
03.	Commande d'exécution du job MapReduce.....	19
V.	Visualiser les résultats	20
1.	Mode de visualisation des résultats du job MapReduce	20
2.	Mode de visualisation des résultats de la Base HBase	21
01.	Démarrer HBase :.....	21
02.	Lancer le Shell HBase.....	21
03.	Créer la table 'tendance_music'	21
VI.	Récupérer les résultats.....	22
1.	Données issues de la base HBase.....	22
2.	Visualisation graphique du Top 10 des Moyennes des streams (nombre de vues)	23
VII.	Annexes et Documentation.....	24
1.	Historique des versions.....	24
2.	Livrables	Erreur ! Signet non défini.

I. Se connecter à la machine virtuelle Docker

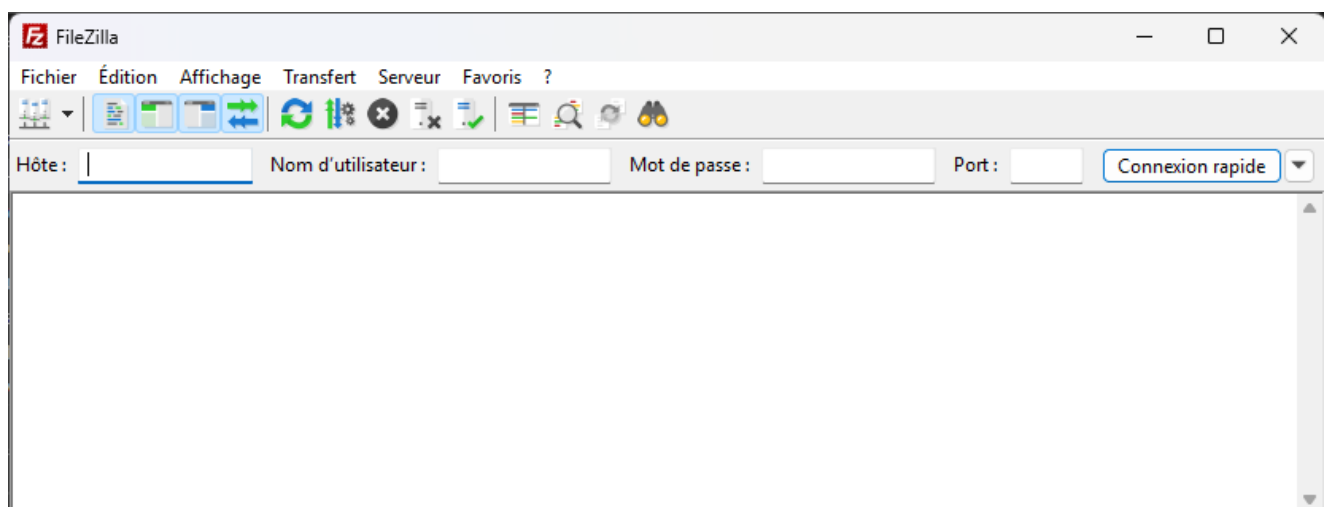
L'outil utilisé pour se connecter à la machine virtuelle est « PuTTY » à partir d'un PC local sous Windows.

1. Modalité de connexion à la machine virtuelle

01. Connexion SSH à l'aide de l'outil PuTTY



02. Connexion SFTP à l'aide de l'outil FileZilla



03. Paramètres de connexion à renseigner

Pour ouvrir une session sur la machine virtuelle distante, que ce soit pour une **session SSH à l'aide de PuTTY**, que ce soit pour une **session SFTP à l'aide FileZila**, il est nécessaire de renseigner les mêmes paramètres de connexion décrits dans le tableau ci-dessous :

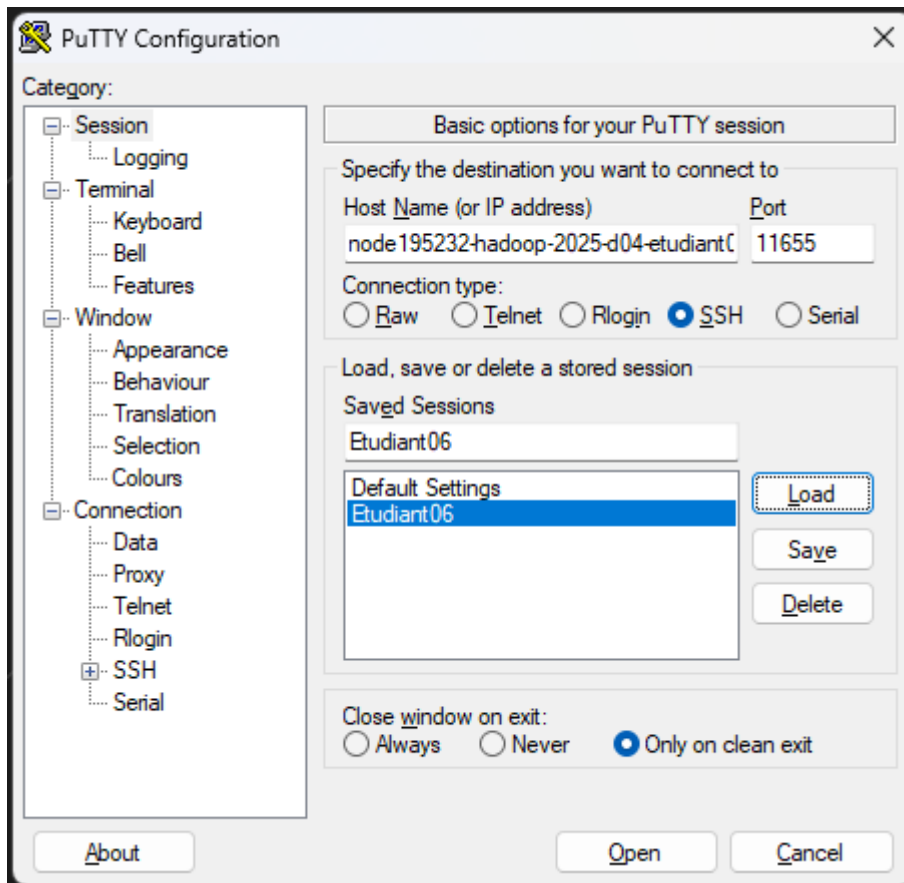
Host Name	node195232-hadoop-2025-d04-etudiant06.sh1.hidora.com → Philippe VITA -OU- node195233-hadoop-2025-d04-etudiant07.sh1.hidora.com → Nour ZERABIB
Port	11655 → Philippe VITA -OU- 11667 → Nour ZERABIB
Protocole	SSH → Pour l'outil PuTTY -OU- SFTP → Pour l'outil FileZila
Login	Root
Password	*****
Nom de session	Etudiant06 → Philippe VITA -OU- Etudiant03 → Nour ZERABIB

Note :

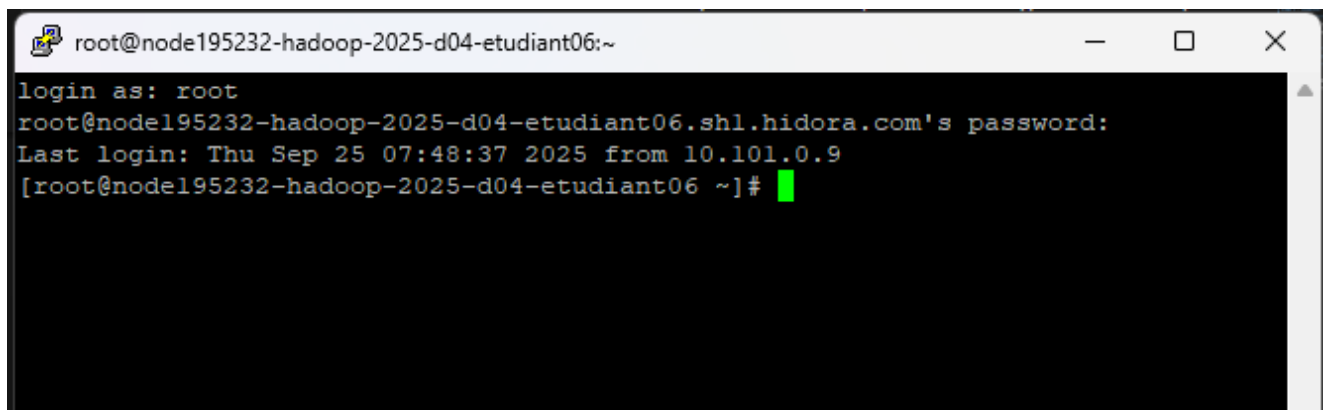
La partie technique du travail a été effectuée sur le Host de Nour ZERABIB.

2. Les étapes de connexion SSH à la machine virtuelle

01. Chargement des paramètres de connexion est fait à l'aide du bouton de commande « Load ».



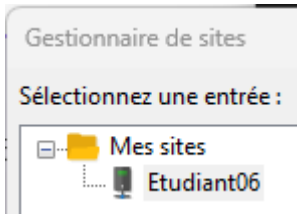
02. L'ouverture de session se fait à l'aide du bouton de commande « Open ».



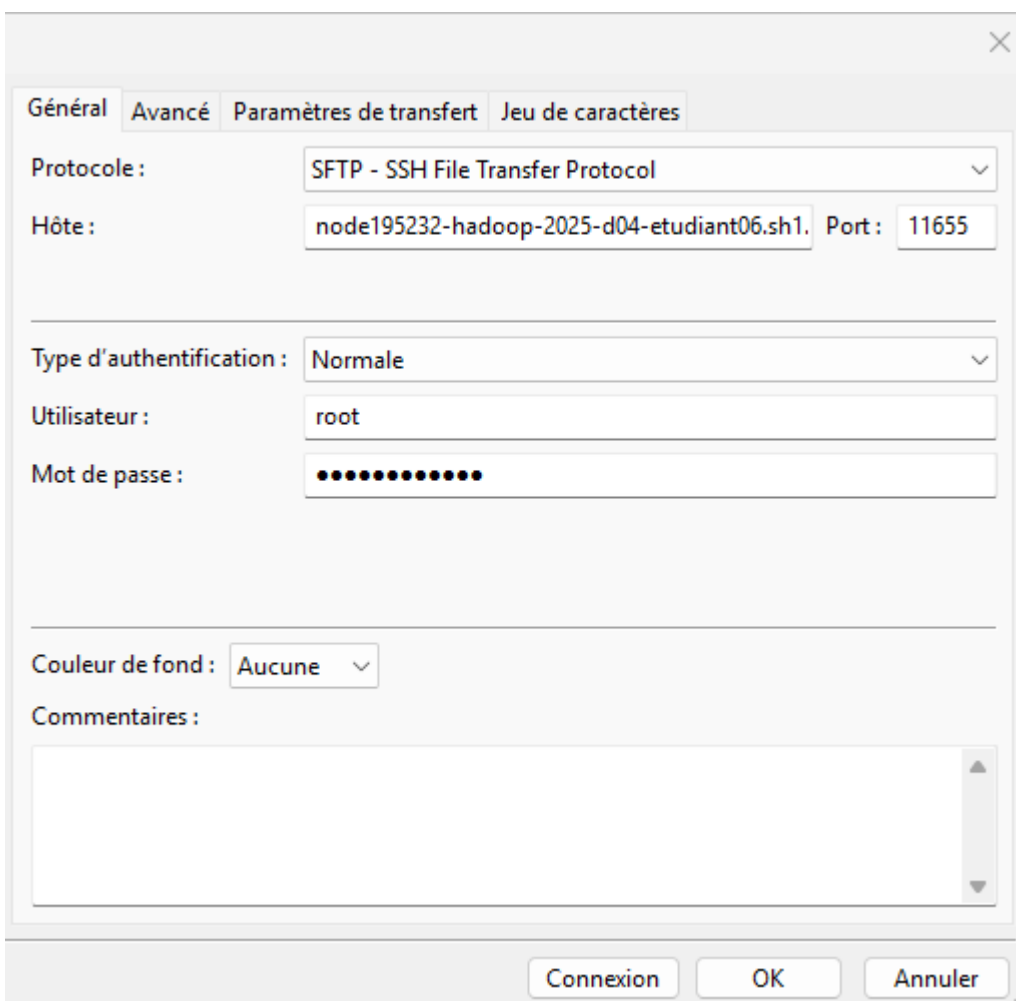
3. Les étapes de connexion SFTP à la machine virtuelle

01. Création d'un nouveau site

- Menu **Fichier**
- Option **Gestionnaire de sites**



02. Chargement des paramètres de connexion est fait à l'aide du bouton de commande « OK »



03. L'ouverture de session se fait à l'aide du bouton de commande « Connexion »

Etudiant06 - sftp://root@node195232-hadoop-2025-d04-etudiant06.sh1.hidora.com:11655 - FileZilla

Fichier Édition Affichage Transfert Serveur Favoris ?

Hôte : Nom d'utilisateur : Mot de passe : Port : Connexion rapide ▼

Statut : Connexion à node195232-hadoop-2025-d04-etudiant06.sh1.hidora.com:11655...
 Statut : Using username "root".
 Statut : Connected to node195232-hadoop-2025-d04-etudiant06.sh1.hidora.com
 Statut : Récupération du contenu du dossier...
 Statut : Listing directory /root
 Statut : Contenu du dossier « /root » affiché avec succès

Site local : D:\WorkSpace\GitRepository\Hadoop\ Site distant : /root

Nom de fichier	Taille de fic...	Type de fichier	Nom de fichier	Taille de fi...	Type de fic...	Der
..			..			
archives		Dossier de fich	.ssh		Dossier de ...	18/
dataset		Dossier de fich	installation		Dossier de ...	26/
modele		Dossier de fich	.bash_history	35 731	Fichier BAS...	25/
streaming		Dossier de fich	.bash_logout	18	Fichier sou...	12/
job.sh	0	Fichier source	.bash_profile	176	Fichier sou...	12/
mapper.py	467	Fichier source	.bashrc	176	Fichier sou...	12/
reducer.py	649	Fichier source	.create_cont_hadoop.sh.swp	12 288	Fichier SWP	26/
Spotify_Most_Streamed_Songs.csv	155 593	Fichier source	.cshrc	100	Fichier CS...	12/
Sujet.pdf	48 905	Adobe Acroba	.python_history	7	Fichier PYT...	18/
TP - Hadoop.docx	536 124	Document Mi	.tcshrc	129	Fichier TCS...	12/
TP-Hadoop-Consignes.md	4 215	Fichier MD	.viminfo	532	Fichier VIM...	06/
~\$ - Hadoop.docx	162	Document Mi	.vimrc	145	Fichier VIM...	18/
~WRL3928.tmp	249 622	Fichier TMP	bash_hadoop_master.sh	36	Fichier sou...	22/
			bash_hadoop_slave1.sh	35	Fichier sou...	22/

9 fichiers et 4 dossiers. Taille totale : 995 737 octets

16 fichiers et 2 dossiers. Taille totale : 49 707 octets

Serveur / Fichier local	Direction	Fichier distant	Taille	Prior
-------------------------	-----------	-----------------	--------	-------

II. Lancer les conteneurs et services Hadoop

1. Services nécessaires

Service	Description
HDFS	Hadoop Distributed File System : Gère le stockage distribué des fichiers sur le cluster
YARN	Yet Another Resource Negotiator : Gère les ressources et l'exécution des tâches MapReduce
Zookeeper	Service de coordination distribué, utilisé par HBase pour la gestion des nœuds
HBase	Base de données NoSQL distribuée, construite sur HDFS, utilisée pour stocker les résultats agrégés

2. Commandes pour démarrer les services

01. Lancer le conteneur Maître (hadoop-master)

- Depuis une session SSH, exécuter le script suivant : `$./start_docker_digi.sh`

```
root@node195232-hadoop-2025-d04-etudiant06:~  
[root@node195232-hadoop-2025-d04-etudiant06 ~]# ls -la  
total 124  
dr-xr-x--- 4 root root 4096 Sep 23 10:59 .  
dr-xr-xr-x 19 root root 4096 Sep 25 07:00 ..  
-rw----- 1 root root 35544 Sep 25 08:58 .bash_history  
-rw-r--r-- 1 root root 18 Aug 12 2018 .bash_logout  
-rw-r--r-- 1 root root 176 Aug 12 2018 .bash_profile  
-rw-r--r-- 1 root root 176 Aug 12 2018 .bashrc  
-rw-r--r-- 1 root root 12288 Jan 26 2024 .create_cont_hadoop.sh.swp  
-rw-r--r-- 1 root root 100 Aug 12 2018 .cshrc  
-rw----- 1 root root 7 Jan 18 2024 .python_history  
drwx----- 2 root root 4096 Jan 18 2024 .ssh  
-rw-r--r-- 1 root root 129 Aug 12 2018 .tcshrc  
-rw----- 1 root root 532 Jul 6 2023 .viminfo  
-rw-r--r-- 1 root root 145 Jan 18 2024 .vimrc  
-rw----- 1 root root 36 Jan 22 2024 bash_hadoop_master.sh  
-rw----- 1 root root 35 Jan 22 2024 bash_hadoop_slave1.sh  
-rw----- 1 root root 36 Jan 22 2024 bash_hadoop_slave2.sh  
drwxr-xr-x 2 root root 4096 Jan 26 2024 installation  
-rw----- 1 root root 137 Jan 23 2024 lance_srv_slaves.sh  
-rw----- 1 root root 82 Jan 22 2024 start_docker_digi.sh  
-rw----- 1 root root 79 Jan 22 2024 stop_docker_digi.sh  
[root@node195232-hadoop-2025-d04-etudiant06 ~]#  
[root@node195232-hadoop-2025-d04-etudiant06 ~]# ./start_docker_digi.sh
```

- Résultat : Démarrage automatique des 3 conteneurs Hadoop

```
[root@node195232-hadoop-2025-d04-etudiant06 ~]# ./start_docker_digi.sh  
hadoop-master  
hadoop-slave1  
hadoop-slave2  
[root@node195232-hadoop-2025-d04-etudiant06 ~]#
```

02. Se connecter au conteneur Hadoop maître (hadoop-master)

C'est uniquement dans ce conteneur que tout le reste des opérations se déroulera. Pour ce faire, il convient de s'y connecter :

- Exécuter le script `$./bash_hadoop_master.sh`

```
-rwx----- 1 root root 36 Jan 22 2024 bash_hadoop_master.sh
-rwx----- 1 root root 35 Jan 22 2024 bash_hadoop_slave1.sh
-rwx----- 1 root root 36 Jan 22 2024 bash_hadoop_slave2.sh
drwxr-xr-x 2 root root 4096 Jan 26 2024 installation
-rwx----- 1 root root 137 Jan 23 2024 lance_srv_slaves.sh
-rwx----- 1 root root 82 Jan 22 2024 start_docker_digi.sh
-rwx----- 1 root root 79 Jan 22 2024 stop_docker_digi.sh
[root@node195232-hadoop-2025-d04-etudiant06 ~]#
[root@node195232-hadoop-2025-d04-etudiant06 ~]#
[root@node195232-hadoop-2025-d04-etudiant06 ~]# ./bash_hadoop_master.sh
```

- Résultat attendu : le prompt devient « **root@hadoop-master:~#** »

```
[root@node195232-hadoop-2025-d04-etudiant06 ~]#
[root@node195232-hadoop-2025-d04-etudiant06 ~]#
[root@node195232-hadoop-2025-d04-etudiant06 ~]# ./bash_hadoop_master.sh
root@hadoop-master:~#
```

- Puis, démarrer le service « Hadoop » depuis le conteneur Hadoop Maître :
`$./start-hadoop.sh`

```
root@hadoop-master: ~
-rw-r--r-- 1 root root 482 Sep 24 14:57 hbase1.py
-rw-r--r-- 1 root root 563 Sep 24 15:06 hbase2.py
-rw-r--r-- 1 root root 841 Sep 24 15:07 hbase3.py
-rw-r--r-- 1 root root 309 Sep 24 15:09 hbase4.py
-rw-r--r-- 1 root root 318 Sep 24 15:11 hbase5.py
-rwx----- 1 root root 162 Jan 25 2024 hbase_create.sh
-rwx----- 1 root root 117 Jan 25 2024 hbase_drop.sh
-rwx----- 1 root root 120 Feb 12 2024 hbase_odbc_rest.sh
drwxr-xr-x 1 root root 4096 Jan 26 2024 hdfs
-rw-r--r-- 1 root root 377 Sep 23 09:35 mapper.py
-rw-r--r-- 1 root root 1 Sep 23 15:22 mapper2.py
-rw-r--r-- 1 root root 211312924 Sep 23 11:00 purchases.txt
-rw-r--r-- 1 root root 820 Sep 23 10:14 reducer.py
-rwxr-xr-x 1 root root 723 Jan 28 2025 run-wordcount.sh
-rwx----- 1 root root 46 Jan 22 2024 services_hbase_thrift.sh
-rwxrwx--- 1 root root 1003 Jan 26 2024 setup.sh
-rwxr-xr-x 1 root root 120 Mar 4 2018 start-hadoop.sh
-rwxr-xr-x 1 root root 218 Mar 4 2018 start-kafka-zookeeper.sh
-rw-r--r-- 1 root root 1161 Sep 23 14:07 stats_ventes.py
-rw-r--r-- 1 root root 450 Sep 23 13:37 total_vente
-rw-r--r-- 1 root root 0 Sep 23 13:29 total_ventes.py
root@hadoop-master:~#
root@hadoop-master:~#
root@hadoop-master:~# ./start-hadoop.sh
```

En conséquence :

- Le service **NodeManager** est lancé
- Le service **YARN (yarn daemons)** est lancé

```
root@hadoop-master: ~  
root@hadoop-master:~# ./start-hadoop.sh  
  
Starting namenodes on [hadoop-master]  
hadoop-master: Warning: Permanently added 'hadoop-master,172.18.0.2' (ECDSA) to  
the list of known hosts.  
hadoop-master: namenode running as process 173. Stop it first.  
hadoop-slave1: Warning: Permanently added 'hadoop-slave1,172.18.0.3' (ECDSA) to  
the list of known hosts.  
hadoop-slave2: Warning: Permanently added 'hadoop-slave2,172.18.0.4' (ECDSA) to  
the list of known hosts.  
hadoop-slave1: datanode running as process 72. Stop it first.  
hadoop-slave2: datanode running as process 71. Stop it first.  
Starting secondary namenodes [0.0.0.0]  
0.0.0.0: secondarynamenode running as process 394. Stop it first.  
  
starting yarn daemons  
resourcemanager running as process 588. Stop it first.  
hadoop-slave2: Warning: Permanently added 'hadoop-slave2,172.18.0.4' (ECDSA) to  
the list of known hosts.  
hadoop-slave1: Warning: Permanently added 'hadoop-slave1,172.18.0.3' (ECDSA) to  
the list of known hosts.  
hadoop-slave2: nodemanager running as process 198. Stop it first.  
hadoop-slave1: nodemanager running as process 199. Stop it first.  
  
root@hadoop-master:~#
```

03. Démarrer les services Hadoop dans le conteneur Hadoop Master

a. Démarrer le service « DFS (HDFS) »

- Exécuter le script \$ **start-dfs.sh**

```
root@hadoop-master:~# start-dfs.sh  
Starting namenodes on [hadoop-master]  
hadoop-master: Warning: Permanently added 'hadoop-master,172.18.0.2' (ECDSA) to  
the list of known hosts.  
hadoop-master: namenode running as process 173. Stop it first.  
hadoop-slave2: Warning: Permanently added 'hadoop-slave2,172.18.0.4' (ECDSA) to  
the list of known hosts.  
hadoop-slave1: Warning: Permanently added 'hadoop-slave1,172.18.0.3' (ECDSA) to  
the list of known hosts.  
hadoop-slave2: datanode running as process 71. Stop it first.  
hadoop-slave1: datanode running as process 72. Stop it first.  
Starting secondary namenodes [0.0.0.0]  
0.0.0.0: secondarynamenode running as process 394. Stop it first.  
root@hadoop-master:~#
```

b. Démarrer le service « YARN »

- Exécuter le script \$ **start-yarn.sh**

```
root@hadoop-master:~# start-yarn.sh
starting yarn daemons
resourcemanager running as process 588. Stop it first.
hadoop-slave1: Warning: Permanently added 'hadoop-slave1,172.18.0.3' (ECDSA) to
the list of known hosts.
hadoop-slave2: Warning: Permanently added 'hadoop-slave2,172.18.0.4' (ECDSA) to
the list of known hosts.
hadoop-slave1: nodemanager running as process 199. Stop it first.
hadoop-slave2: nodemanager running as process 198. Stop it first.
root@hadoop-master:~#
```

c. Démarrer les services « HBase » et « ZooKeeper »

- Exécuter le script \$ **start-hbase.sh**

```
root@hadoop-master: ~
root@hadoop-master:~# start-hbase.sh
hadoop-master: Warning: Permanently added 'hadoop-master,172.18.0.2' (ECDSA) to
the list of known hosts.
hadoop-master: running zookeeper, logging to /usr/local/hbase/bin/./logs/hbase-
root-zookeeper-hadoop-master.out
running master, logging to /usr/local/hbase/logs/hbase--master-hadoop-master.out
OpenJDK 64-Bit Server VM warning: ignoring option PermSize=128m; support was rem
oved in 8.0
OpenJDK 64-Bit Server VM warning: ignoring option MaxPermSize=128m; support was
removed in 8.0
: running regionserver, logging to /usr/local/hbase/logs/hbase--regionserver-had
oop-master.out
: OpenJDK 64-Bit Server VM warning: ignoring option PermSize=128m; support was r
emoved in 8.0
: OpenJDK 64-Bit Server VM warning: ignoring option MaxPermSize=128m; support wa
s removed in 8.0
root@hadoop-master:~#
```

d. Vérification des services démarrés

- Exécuter \$ **jps**
- Résultat : Tous les services indispensables Hadoop sont démarrés

```
root@hadoop-master: ~
root@hadoop-master:~# jps
3264 Jps
2693 HMaster
2841 HRegionServer
394 SecondaryNameNode
588 ResourceManager
173 NameNode
2607 HQuorumPeer
root@hadoop-master:~#
```

III. Importer les données dans HDFS

1. Méthode d'importation des données

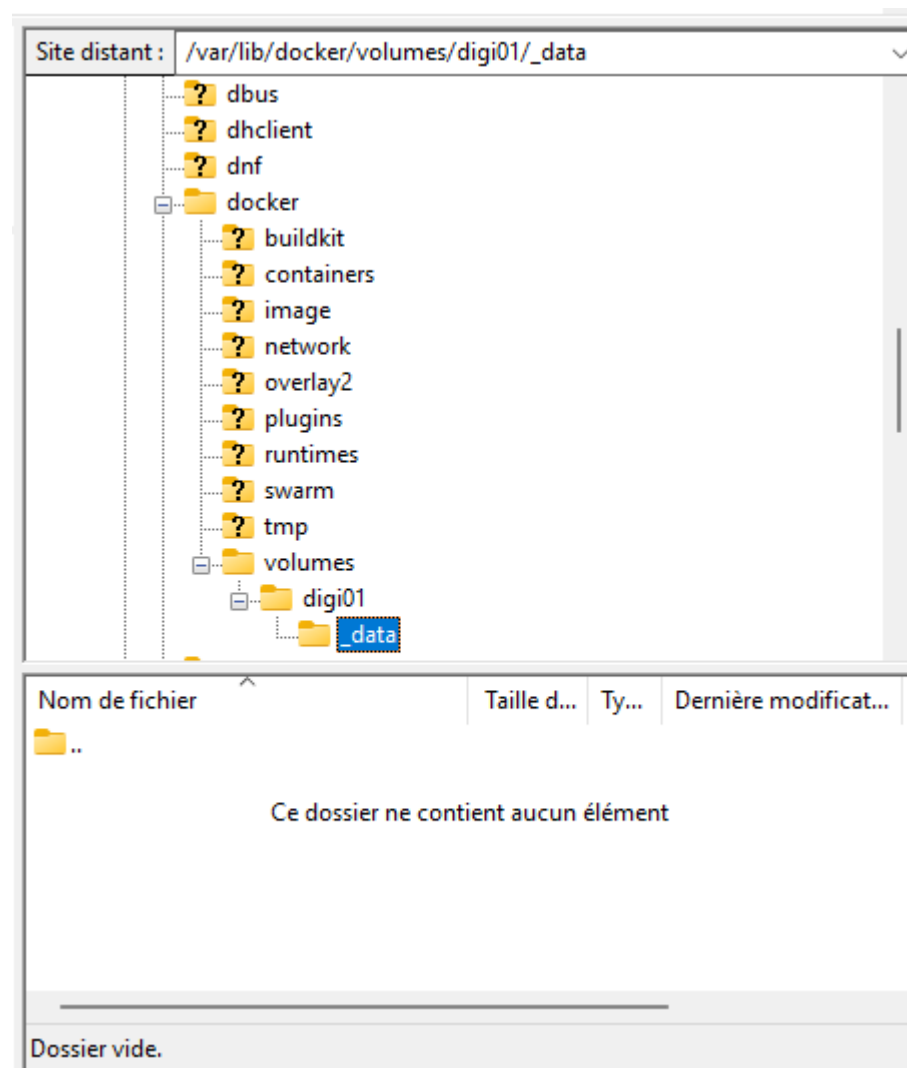
Les données source sont contenues dans un fichier CSV fourni (fichier contenant les colonnes danceability, energy, streams, etc.).

Le nécessaire est fait pour que ces données soient copiées dans le système de fichiers distribué « HDFS », afin d'être accessibles par le job « MapReduce ».

01. Déposer le fichier CSV, source des données, dans le HDFS via l'outil FileZila

Item	Description																																																												
Fichier source	Spotify_Most_Streamed_Songs.csv																																																												
GitRepository	D:\WorkSpace\GitRepository\Hadoop\ <div><div>Site local : D:\WorkSpace\GitRepository\Hadoop\</div><div><div><div>GestionDesVehicules</div><div>Hadoop</div><div>archives</div><div>dataset</div></div></div><table><tr><th>Nom de fichier</th><th>Taille de fic...</th><th>Type de fich...</th><th>Dernière modificat...</th></tr><tr><td>..</td><td></td><td></td><td></td></tr><tr><td>archives</td><td></td><td>Dossier de fi...</td><td>25/09/2025 10:14:17</td></tr><tr><td>dataset</td><td></td><td>Dossier de fi...</td><td>24/09/2025 10:15:46</td></tr><tr><td>modele</td><td></td><td>Dossier de fi...</td><td>24/09/2025 17:45:58</td></tr><tr><td>streaming</td><td></td><td>Dossier de fi...</td><td>23/09/2025 10:17:26</td></tr><tr><td>job.sh</td><td>0</td><td>Fichier sour...</td><td>24/09/2025 17:41:56</td></tr><tr><td>mapper.py</td><td>467</td><td>Fichier sour...</td><td>25/09/2025 11:10:39</td></tr><tr><td>reducer.py</td><td>649</td><td>Fichier sour...</td><td>25/09/2025 11:11:09</td></tr><tr><td>Spotify_Most_Streamed_Songs.csv</td><td>155 593</td><td>Fichier sour...</td><td>25/09/2025 09:35:41</td></tr><tr><td>Sujet.pdf</td><td>48 905</td><td>Adobe Acro...</td><td>25/09/2025 09:35:41</td></tr><tr><td>TP - Hadoop.docx</td><td>655 770</td><td>Document ...</td><td>25/09/2025 15:01:04</td></tr><tr><td>TP-Hadoop-Consignes.md</td><td>4 215</td><td>Fichier MD</td><td>25/09/2025 11:09:03</td></tr><tr><td>~\$ - Hadoop.docx</td><td>162</td><td>Document ...</td><td>25/09/2025 10:39:09</td></tr><tr><td>~WRL3928.tmp</td><td>249 622</td><td>Fichier TMP</td><td>25/09/2025 11:26:55</td></tr></table><div>Sélection de 1 fichier. Taille totale : 155 593 octets</div></div>	Nom de fichier	Taille de fic...	Type de fich...	Dernière modificat...	..				archives		Dossier de fi...	25/09/2025 10:14:17	dataset		Dossier de fi...	24/09/2025 10:15:46	modele		Dossier de fi...	24/09/2025 17:45:58	streaming		Dossier de fi...	23/09/2025 10:17:26	job.sh	0	Fichier sour...	24/09/2025 17:41:56	mapper.py	467	Fichier sour...	25/09/2025 11:10:39	reducer.py	649	Fichier sour...	25/09/2025 11:11:09	Spotify_Most_Streamed_Songs.csv	155 593	Fichier sour...	25/09/2025 09:35:41	Sujet.pdf	48 905	Adobe Acro...	25/09/2025 09:35:41	TP - Hadoop.docx	655 770	Document ...	25/09/2025 15:01:04	TP-Hadoop-Consignes.md	4 215	Fichier MD	25/09/2025 11:09:03	~\$ - Hadoop.docx	162	Document ...	25/09/2025 10:39:09	~WRL3928.tmp	249 622	Fichier TMP	25/09/2025 11:26:55
Nom de fichier	Taille de fic...	Type de fich...	Dernière modificat...																																																										
..																																																													
archives		Dossier de fi...	25/09/2025 10:14:17																																																										
dataset		Dossier de fi...	24/09/2025 10:15:46																																																										
modele		Dossier de fi...	24/09/2025 17:45:58																																																										
streaming		Dossier de fi...	23/09/2025 10:17:26																																																										
job.sh	0	Fichier sour...	24/09/2025 17:41:56																																																										
mapper.py	467	Fichier sour...	25/09/2025 11:10:39																																																										
reducer.py	649	Fichier sour...	25/09/2025 11:11:09																																																										
Spotify_Most_Streamed_Songs.csv	155 593	Fichier sour...	25/09/2025 09:35:41																																																										
Sujet.pdf	48 905	Adobe Acro...	25/09/2025 09:35:41																																																										
TP - Hadoop.docx	655 770	Document ...	25/09/2025 15:01:04																																																										
TP-Hadoop-Consignes.md	4 215	Fichier MD	25/09/2025 11:09:03																																																										
~\$ - Hadoop.docx	162	Document ...	25/09/2025 10:39:09																																																										
~WRL3928.tmp	249 622	Fichier TMP	25/09/2025 11:26:55																																																										
Path du dépôt	FS : /var/lib/docker/volumes/digi01/_data																																																												

Link : /datavolume1/



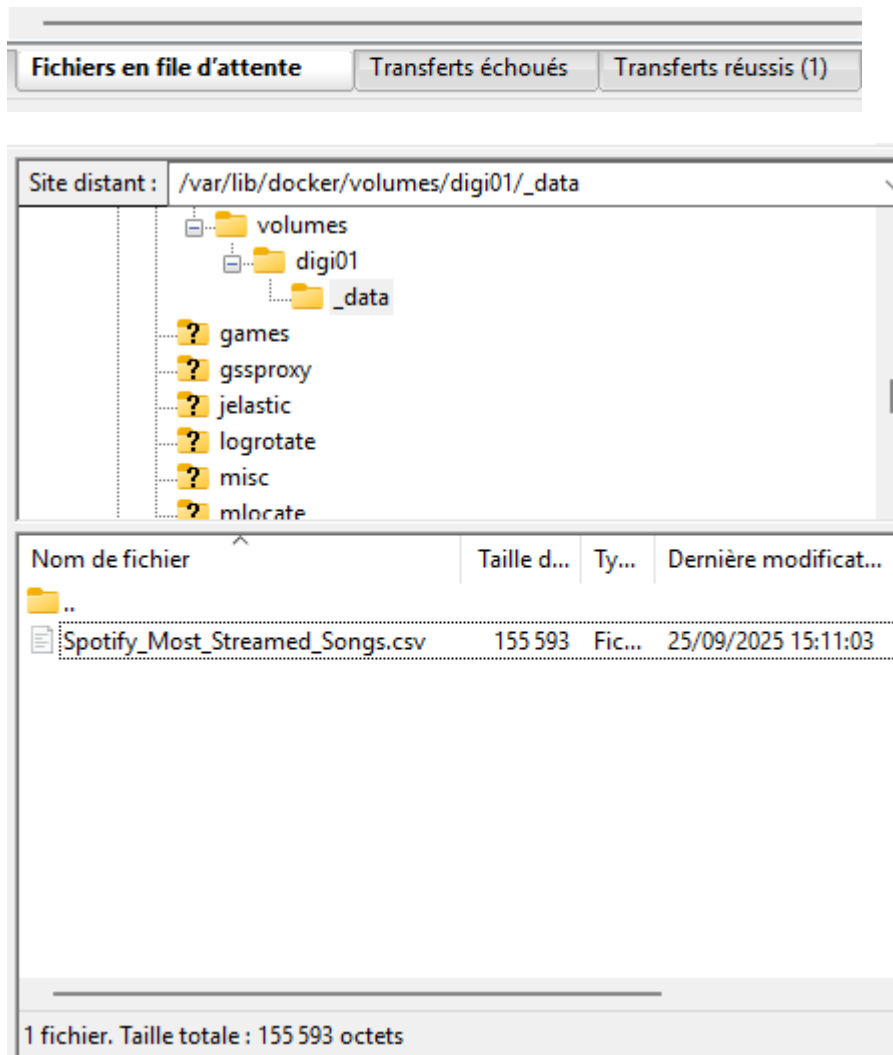
**Path
d'exploitation**

FS : /root

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

02. Vérifier que le fichier est bien présent en local sur la machine virtuelle

- Le fichier CSV déposé en local sur la machine virtuelle (hadoop-master) apparaît bien dans le dossier de dépôt :



- Il est également bien visible depuis le lien symbolique du dossier de dépôt :

```

root@hadoop-master: ~
root@hadoop-master:~# ls -la /datavolumel
total 164
drwxr-xr-x 2 root root 4096 Sep 25 13:19 .
drwxr-xr-x 1 root root 4096 Jan 28 2025 ..
-rw-r--r-- 1 root root 155593 Sep 25 13:19 Spotify_Most_Streamed_Songs.csv
root@hadoop-master:~#

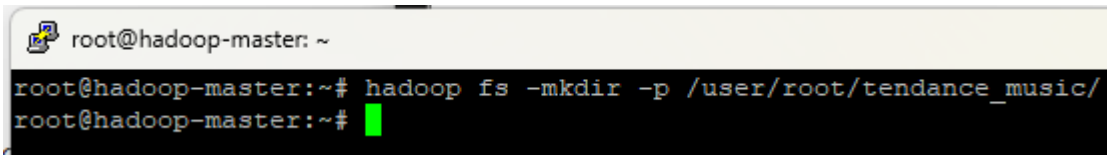
```

03. Illustrer la commande d'import du fichier dans le HDFS :

Le fichier source Spotify_Most_Streamed_Songs.csv contient des colonnes pertinentes pour l'analyse musicale telles que danceability_%, energy_%, et streams. Ces données sont importées dans HDFS via la commande suivante :

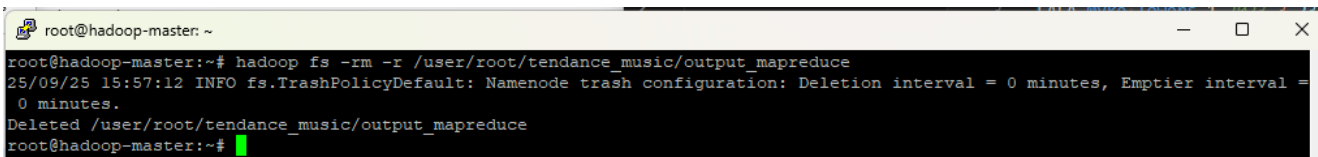
a. *Créer le répertoire HDFS cible*

\$ `hadoop fs -mkdir -p /user/root/tendance_music/`



```
root@hadoop-master: ~  
root@hadoop-master:~# hadoop fs -mkdir -p /user/root/tendance_music/  
root@hadoop-master:~#
```

b.



```
root@hadoop-master: ~  
root@hadoop-master:~# hadoop fs -rm -r /user/root/tendance_music/output_mapreduce  
25/09/25 15:57:12 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Empty interval = 0 minutes.  
Deleted /user/root/tendance_music/output_mapreduce  
root@hadoop-master:~#
```

c. *Récupérer le fichier source depuis le dossier de dépôt vers le HDFS cible*

\$ `hadoop fs -put /datavolume1/Spotify_Most_Streamed_Songs.csv /user/root/tendance_music/`

IV. Créer et exécuter un job MapReduce

1. Structure du job « mapper »

```
1  #!/usr/bin/env python3
2  import sys, csv
3
4
5  reader = csv.DictReader(sys.stdin)          # Lecture du fichier Spotify
6
7  for row in reader:
8      try:
9          d = row.get('danceability_')      # Récupération de 'danceability_'
10         e = row.get('energy_')             # Récupération de 'energy'
11         s = row.get('streams')             # Récupération de 'streams'
12         d = int(float(d))                  # Conversion
13         e = int(float(e))
14         s = int(s)
15         print(d, "\t", e, "\t", s)
16     except Exception:
17
18         # Ignorer les lignes invalides
19         continue
```

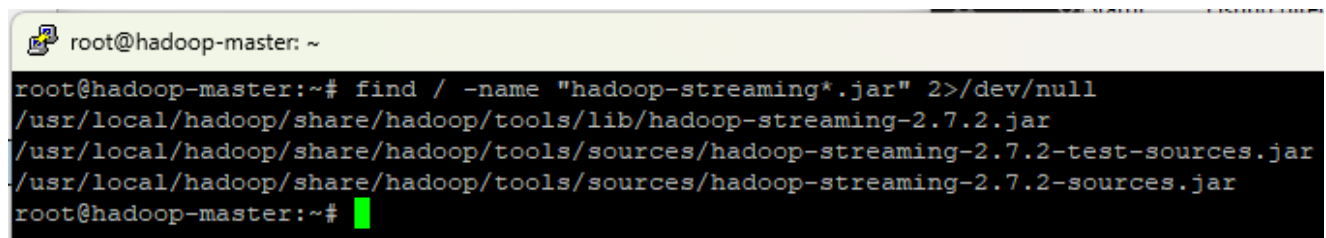
2. Structure du job « reducer »

```
1  #!/usr/bin/env python3
2  import sys
3
4  current_key = None
5  sum_streams = 0
6  count = 0
7
8  for line in sys.stdin:          # Lecture du flux entrant
9      line = line.strip()        # Nettoyage des données brutes
10     if not line:               # Sauter la ligne vide
11         continue
12     d, e, s = line.split('\t')  # Récupérer 'danceability', 'energy', 'streams'
13     key= d+" "+e                # Créer la clé : Pair ('danceability', 'energy')
14     val = int(s)                # Créer la valeur (streams)
15
16     if current_key is None:     # Initialiser les clés et valeurs
17         current_key = key
18         sum_streams = val
19         count = 1
20         continue
21
22     if key == current_key:      # Agrégation des données recueillies
23         sum_streams += val
24         count += 1
25     else:
26         avg = sum_streams / count if count else 0 # Fonction d'agrégation appliquée
27         print(current_key, "\tsum=", sum_streams, "\tcount=", count, "\tavg=", avg)
28         current_key = key
29         sum_streams = val
30         count = 1
31
32 # Pour la dernière clé
33 if current_key is not None:    # Si la clé courante n'est pas nulle
34     avg = sum_streams / count if count else 0
35     print(current_key, "\tsum=", sum_streams, "\tcount=", count, "\tavg=", avg)
```

3. Commandes exécutables pour soumettre le Job MapReduce sur Hadoop Streaming

01. Vérifier l'emplacement des Jar Streaming

```
$ find / -name "hadoop-streaming*.jar" 2>/dev/null
```



```
root@hadoop-master: ~
root@hadoop-master:~# find / -name "hadoop-streaming*.jar" 2>/dev/null
/usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.7.2.jar
/usr/local/hadoop/share/hadoop/tools/sources/hadoop-streaming-2.7.2-test-sources.jar
/usr/local/hadoop/share/hadoop/tools/sources/hadoop-streaming-2.7.2-sources.jar
root@hadoop-master:~#
```

02. Application des droits d'exécution sur des jobs MapReduce

```
root@hadoop-master:~# chmod +x /root/mapper.py
root@hadoop-master:~# chmod +x /root/reducer.py
root@hadoop-master:~#
```

03. Commande d'exécution du job MapReduce

```
hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.7.2.jar
  -input /user/root/tendance_music/Spotify_Most_Streamed_Songs.csv \
  -output /user/root/tendance_music/output_mapreduce \
  -mapper mapper.py \
  -reducer reducer.py \
  -file /root/mapper.py \
  -file /root/reducer.py
```

- Résultat obtenu :

```
root@hadoop-master: ~
root@hadoop-master:~# hadoop fs -rm -r /user/root/tendance_music/output_mapreduce
25/09/25 15:57:12 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Empty interval = 0 minutes.
Deleted /user/root/tendance_music/output_mapreduce
root@hadoop-master:~# clear
root@hadoop-master:~# hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.7.2.jar -input /user/root/tendance_music/Spotify_Most_Streamed_Songs.csv -output /user/root/tendance_music/output_mapreduce -mapper mapper.py -reducer reducer.py -file /root/mapper.py -file /root/reducer.py
25/09/25 15:57:51 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.
packageJobJar: [/root/mapper.py, /root/reducer.py, /tmp/hadoop-unjar554588936036812369/] [] /tmp/streamjob6064713556845998292.jar tmpDir=null
25/09/25 15:57:52 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/172.18.0.2:8032
25/09/25 15:57:52 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/172.18.0.2:8032
25/09/25 15:57:52 INFO mapred.FileInputFormat: Total input paths to process : 1
25/09/25 15:57:52 INFO mapreduce.JobSubmitter: number of splits:2
25/09/25 15:57:52 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1758786623458_0002
25/09/25 15:57:53 INFO impl.YarnClientImpl: Submitted application application_1758786623458_0002
25/09/25 15:57:53 INFO mapreduce.Job: The url to track the job: http://hadoop-master:8088/proxy/application_1758786623458_0002/
25/09/25 15:57:53 INFO mapreduce.Job: Running job: job_1758786623458_0002
25/09/25 15:57:58 INFO mapreduce.Job: Job job_1758786623458_0002 running in uber mode : false
25/09/25 15:57:58 INFO mapreduce.Job: map 0% reduce 0%
25/09/25 15:58:01 INFO mapreduce.Job: Task Id : attempt_1758786623458_0002_m_0000001_0, Status : FAILED
Error: java.lang.RuntimeException: PipeMapRed.waitOutputThreads(): subprocess failed with code 127
    at org.apache.hadoop.streaming.PipeMapRed.waitOutputThreads(PipeMapRed.java:322)
    at org.apache.hadoop.streaming.PipeMapRed.mapRedFinished(PipeMapRed.java:535)
    at org.apache.hadoop.streaming.PipeMapper.close(PipeMapper.java:130)
    at org.apache.hadoop.mapred.MapRunner.run(MapRunner.java:61)
    at org.apache.hadoop.streaming.PipeMapRunner.run(PipeMapRunner.java:34)
    at org.apache.hadoop.mapred.MapTask.runOldMapper(MapTask.java:453)
    at org.apache.hadoop.mapred.MapTask.run(MapTask.java:343)
    at org.apache.hadoop.mapred.YarnChild$2.run(YarnChild.java:164)
    at java.security.AccessController.doPrivileged(Native Method)
    at javax.security.auth.Subject.doAs(Subject.java:422)
    at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1657)
    at org.apache.hadoop.mapred.YarnChild.main(YarnChild.java:158)
25/09/25 15:58:01 INFO mapreduce.Job: Task Id : attempt_1758786623458_0002_m_0000000_0, Status : FAILED
Error: java.lang.RuntimeException: PipeMapRed.waitOutputThreads(): subprocess failed with code 127
    at org.apache.hadoop.streaming.PipeMapRed.waitOutputThreads(PipeMapRed.java:322)
    at org.apache.hadoop.streaming.PipeMapRed.mapRedFinished(PipeMapRed.java:535)
    at org.apache.hadoop.streaming.PipeMapper.close(PipeMapper.java:130)
    at org.apache.hadoop.mapred.MapRunner.run(MapRunner.java:61)
    at org.apache.hadoop.streaming.PipeMapRunner.run(PipeMapRunner.java:34)
    at org.apache.hadoop.mapred.MapTask.runOldMapper(MapTask.java:453)
    at org.apache.hadoop.mapred.MapTask.run(MapTask.java:343)
    at org.apache.hadoop.mapred.YarnChild$2.run(YarnChild.java:164)
    at java.security.AccessController.doPrivileged(Native Method)
    at javax.security.auth.Subject.doAs(Subject.java:422)
    at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1657)
    at org.apache.hadoop.mapred.YarnChild.main(YarnChild.java:158)
```

1. Mode de visualisation des résultats du job MapReduce

\$ Commande : **hadoop fs -cat /user/root/tendance_music/output_mapreduce/part-00000 | head**

Cette commande affiche les premières lignes du fichier de sortie, contenant :

- La clé agrégée `danceability_energy`
- Le total des streams
- Le nombre d'occurrences
- La moyenne des streams

```
root@hadoop-master: ~  
File Input Format Counters  
  Bytes Read=159689  
File Output Format Counters  
  Bytes Written=24738  
25/09/25 10:26:35 INFO streaming.StreamJob: Output directory: tp_music/out  
root@hadoop-master:~# hdfs dfs -cat tp_music/out/part-00000  
23 _ 25      sum= 395591396  count= 1      avg= 395591396.0  
24 _ 60      sum= 663832097  count= 1      avg= 663832097.0  
25 _ 30      sum= 297328960  count= 1      avg= 297328960.0  
32 _ 74      sum= 705469769  count= 1      avg= 705469769.0  
33 _ 61      sum= 460492795  count= 1      avg= 460492795.0  
33 _ 71      sum= 1840364617 count= 1      avg= 1840364617.0  
34 _ 51      sum= 726434358  count= 1      avg= 726434358.0  
34 _ 63      sum= 1449779435 count= 1      avg= 1449779435.0  
34 _ 76      sum= 242767149  count= 1      avg= 242767149.0  
34 _ 56      sum= 265882712  count= 1      avg= 265882712.0  
34 _ 57      sum= 600976848  count= 1      avg= 600976848.0  
35 _ 30      sum= 2355719893 count= 1      avg= 2355719893.0  
35 _ 23      sum= 807561936  count= 1      avg= 807561936.0  
36 _ 15      sum= 389771964  count= 1      avg= 389771964.0  
36 _ 57      sum= 1947371785 count= 1      avg= 1947371785.0  
36 _ 28      sum= 284908316  count= 1      avg= 284908316.0  
37 _ 47      sum= 841749534  count= 1      avg= 841749534.0  
39 _ 43      sum= 838586769  count= 1      avg= 838586769.0  
40 _ 56      sum= 2420461338 count= 1      avg= 2420461338.0  
40 _ 48      sum= 571386359  count= 1      avg= 571386359.0  
40 _ 64      sum= 284785823  count= 1      avg= 284785823.0  
41 _ 61      sum= 872137015  count= 1      avg= 872137015.0  
41 _ 25      sum= 338564981  count= 1      avg= 338564981.0  
42 _ 86      sum= 284819874  count= 1      avg= 284819874.0  
43 _ 66      sum= 1755214421 count= 1      avg= 1755214421.0  
43 _ 74      sum= 51985779   count= 1      avg= 51985779.0  
43 _ 55      sum= 882831184  count= 1      avg= 882831184.0  
44 _ 41      sum= 117747907  count= 1      avg= 117747907.0  
44 _ 32      sum= 988515741  count= 1      avg= 988515741.0  
44 _ 77      sum= 184308753  count= 1      avg= 184308753.0  
44 _ 9       sum= 30546883   count= 1      avg= 30546883.0  
45 _ 62      sum= 621660989  count= 1      avg= 621660989.0  
45 _ 24      sum= 446390129  count= 1      avg= 446390129.0  
45 _ 54      sum= 1813673666 count= 1      avg= 1813673666.0  
45 _ 60      sum= 1449799467 count= 1      avg= 1449799467.0  
45 _ 59      sum= 1089402494 count= 1      avg= 1089402494.0  
45 _ 37      sum= 1410088830 count= 1      avg= 1410088830.0  
46 _ 64      sum= 556585270  count= 1      avg= 556585270.0
```

2. Mode de visualisation des résultats de la Base HBase

01. Démarrer HBase :

```
$ start-hbase.sh
```

```
$ hbase-daemon.sh start thrift
```

02. Lancer le Shell HBase

```
$ hbase shell
```

03. Créer la table 'tendance_music'

```
$ create 'tendance_music', 'data'
```

```
hbase(main):001:0> list
TABLE
bibliotheque
sales_ledger
tendance_music
3 row(s) in 0.1700 seconds
=> ["bibliotheque", "sales_ledger", "tendance_music"]
hbase(main):002:0> █
```

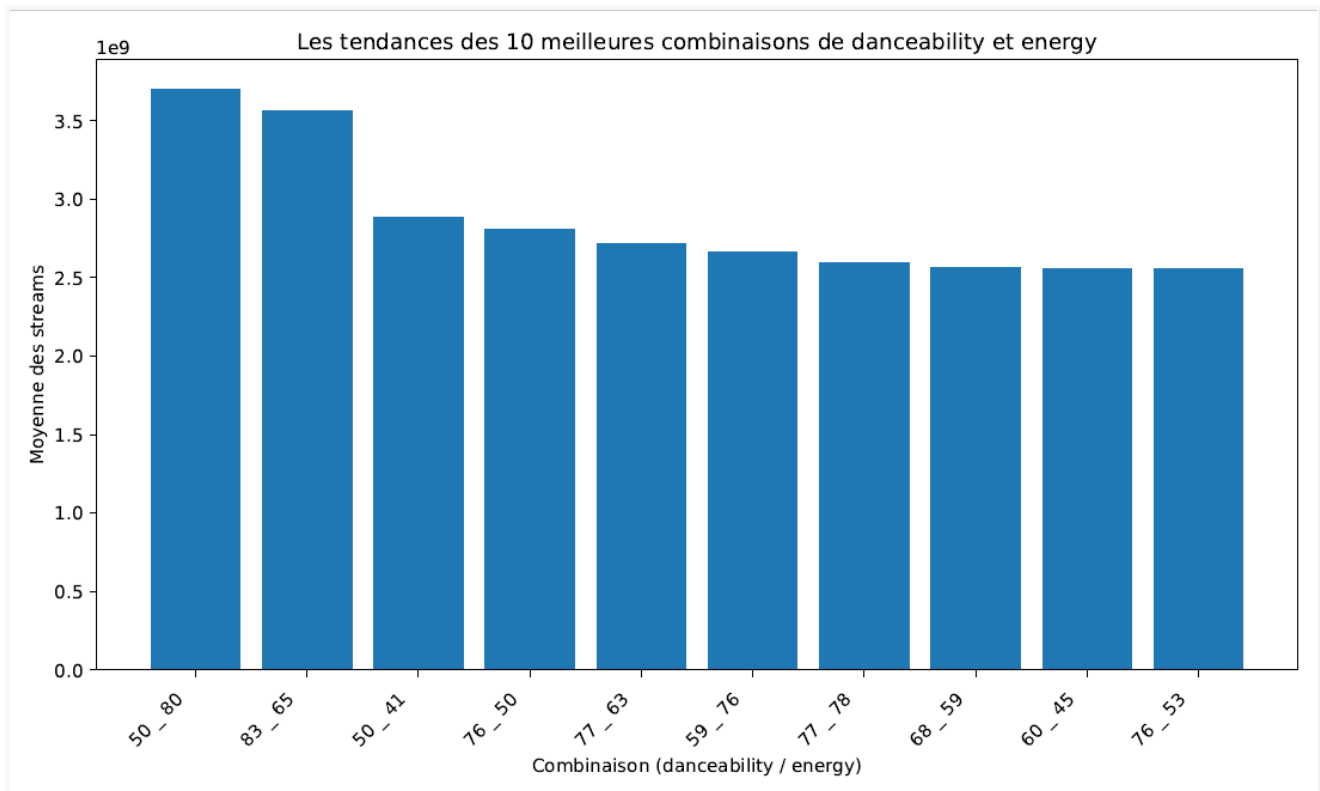

VI. Récupérer les résultats

1. Données issues de la base HBase

On injecte les données dans la table « tendance_music » depuis le fichier de sortie du Job reducer « part-00000 » qu'on a récupéré en local :

```
=> ["bibliotheque", "sales_ledger", "tendance_music"]
hbase(main):002:0> scan 'tendance_music'
COLUMN+CELL
ROW      23      25      column=data:avg_streams, timestamp=1758800455561, value=395591396.00
ROW      23      25      column=data:count, timestamp=1758800455561, value=1
ROW      23      25      column=data:total_streams, timestamp=1758800455561, value=395591396
ROW      24      60      column=data:avg_streams, timestamp=1758800455586, value=663832097.00
ROW      24      60      column=data:count, timestamp=1758800455586, value=1
ROW      24      60      column=data:total_streams, timestamp=1758800455586, value=663832097
ROW      25      30      column=data:avg_streams, timestamp=1758800455610, value=297328960.00
ROW      25      30      column=data:count, timestamp=1758800455610, value=1
ROW      25      30      column=data:total_streams, timestamp=1758800455610, value=297328960
ROW      32      74      column=data:avg_streams, timestamp=1758800455633, value=705469769.00
ROW      32      74      column=data:count, timestamp=1758800455633, value=1
ROW      32      74      column=data:total_streams, timestamp=1758800455633, value=705469769
ROW      33      61      column=data:avg_streams, timestamp=1758800455656, value=460492795.00
ROW      33      61      column=data:count, timestamp=1758800455656, value=1
ROW      33      61      column=data:total_streams, timestamp=1758800455656, value=460492795
ROW      33      71      column=data:avg_streams, timestamp=1758800455683, value=1840364617.00
ROW      33      71      column=data:count, timestamp=1758800455683, value=1
ROW      33      71      column=data:total_streams, timestamp=1758800455683, value=1840364617
ROW      34      51      column=data:avg_streams, timestamp=1758800455711, value=726434358.00
ROW      34      51      column=data:count, timestamp=1758800455711, value=1
ROW      34      51      column=data:total_streams, timestamp=1758800455711, value=726434358
ROW      34      56      column=data:avg_streams, timestamp=1758800455784, value=265882712.00
ROW      34      56      column=data:count, timestamp=1758800455784, value=1
ROW      34      56      column=data:total_streams, timestamp=1758800455784, value=265882712
ROW      34      57      column=data:avg_streams, timestamp=1758800455818, value=600976848.00
ROW      34      57      column=data:count, timestamp=1758800455818, value=1
ROW      34      57      column=data:total_streams, timestamp=1758800455818, value=600976848
...
92      51      column=data:total_streams, timestamp=1758800466856, value=122763672
92      58      column=data:avg_streams, timestamp=1758800466912, value=183706234.00
92      58      column=data:count, timestamp=1758800466912, value=1
92      58      column=data:total_streams, timestamp=1758800466912, value=183706234
92      62      column=data:avg_streams, timestamp=1758800466773, value=782369383.00
92      62      column=data:count, timestamp=1758800466773, value=1
92      62      column=data:total_streams, timestamp=1758800466773, value=782369383
92      66      column=data:avg_streams, timestamp=1758800466881, value=1687664027.00
92      66      column=data:count, timestamp=1758800466881, value=1
92      66      column=data:total_streams, timestamp=1758800466881, value=1687664027
92      70      column=data:avg_streams, timestamp=1758800466834, value=153372011.00
92      70      column=data:count, timestamp=1758800466834, value=1
92      70      column=data:total_streams, timestamp=1758800466834, value=153372011
93      47      column=data:avg_streams, timestamp=1758800466934, value=162887075.00
93      47      column=data:count, timestamp=1758800466934, value=1
93      47      column=data:total_streams, timestamp=1758800466934, value=162887075
93      65      column=data:avg_streams, timestamp=1758800466956, value=11956641.00
93      65      column=data:count, timestamp=1758800466956, value=1
93      65      column=data:total_streams, timestamp=1758800466956, value=11956641
94      61      column=data:avg_streams, timestamp=1758800466982, value=190490915.00
94      61      column=data:count, timestamp=1758800466982, value=1
94      61      column=data:total_streams, timestamp=1758800466982, value=190490915
95      52      column=data:avg_streams, timestamp=1758800467059, value=335074782.00
95      52      column=data:count, timestamp=1758800467059, value=1
95      52      column=data:total_streams, timestamp=1758800467059, value=335074782
95      66      column=data:avg_streams, timestamp=1758800467031, value=1424589568.00
95      66      column=data:count, timestamp=1758800467031, value=1
95      66      column=data:total_streams, timestamp=1758800467031, value=1424589568
95      69      column=data:avg_streams, timestamp=1758800467082, value=294352144.00
95      69      column=data:count, timestamp=1758800467082, value=1
95      69      column=data:total_streams, timestamp=1758800467082, value=294352144
95      89      column=data:avg_streams, timestamp=1758800467005, value=428685680.00
95      89      column=data:count, timestamp=1758800467005, value=1
95      89      column=data:total_streams, timestamp=1758800467005, value=428685680
439 Row(s) in 0.2760 seconds
```

2. Visualisation graphique du Top 10 des Moyennes des streams (nombre de vues)



Ce graphique a été réalisé après avoir accédé aux données de la table HBase « **tendance_music** », puis on récupère les 10 meilleures moyennes de streams qu'on a triées dans l'ordre décroissant pour enfin l'enregistrer dans la table « **rows** ».

Le résultat est sauvegardé sur fichier PDF, à l'aide de la commande :

```
$ plt.save.fig("top10.pdf")
```


1. Historique des versions

Version	Date	Rédacteur	Objet de la modification
1.00	25/09/2025	Philippe VITA	Initialisation du document

2. Livrables

Les livrables sont à déposer sur un espace GitHub.

Support	Description
GitHub de la Formatrice	Amina MARIE Compte : Aminamarie
Livrables attendus	<pre>PhilippeVita@PC-VITA-01 MINGW64 /D/WorkSpace/GitRepository/Hadoop (main) \$ git status On branch main No commits yet Changes to be committed: (use "git rm --cached <file>..." to unstage) new file: Consignes-TP-Hadoop.md new file: README.md new file: Rapport-TP-Hadoop.pdf new file: Spotify_Most_Streamed_Songs.csv new file: gitignore new file: load_hbase.py new file: mapper.py new file: reducer.py new file: top10.pdf</pre>