

Bases de données NOSQL et Big data

TP1 : Initiation à Hadoop

Diplôme National d'Ingénieur en Informatique

Spécialité :

Génie Logiciel

Réalisée par :

Oussama Ben Slama

Année Universitaire 2024/2025

Table des matières

1	Installation et Configuration	2
1.1	Qu'est-ce que Docker ?	2
1.1.1	Installation de Docker	2
1.2	Déployer Hadoop	3
1.2.1	Déploiement	3
2	Commandes de HDFS	6
2.1	Commandes de HDFS	7
3	Création d'une arborescence et téléchargement de fichiers	10
4	État du cluster	12
4.1	Page Overview	12
4.2	Page DataNodes	13
4.3	Page Utilities	13
5	Conclusion	15

Chapitre 1

Installation et Configuration

Afin de réaliser ce TP numéro 1, nous devons installer Hadoop, un framework qui aide à la manipulation de données massives. Ce logiciel peut poser quelques problèmes lors de l'installation, c'est pourquoi, pour éviter les conflits qui pourraient survenir, nous allons utiliser Docker.

1.1 Qu'est-ce que Docker ?

Docker est un logiciel de conteneurisation qui permet la création et l'utilisation de conteneurs Linux. Grâce à ce logiciel, les conteneurs deviennent des machines virtuelles très légères, nous offrant une grande flexibilité pour créer, déployer, copier des conteneurs et les déplacer d'un environnement à un autre.

1.1.1 Installation de Docker

Nous pouvons installer Docker via ce lien : <https://docs.docker.com/desktop/install/windows-install/>, puis démarrer le logiciel installé.

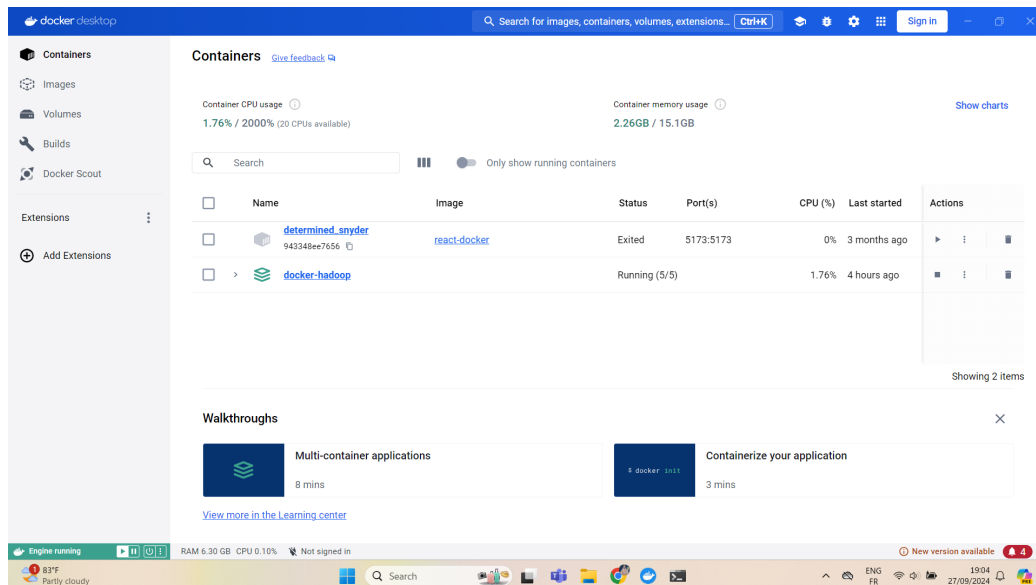


FIGURE 1.1 – Interface de Docker Desktop

1.2 Déployer Hadoop

Comme mentionné précédemment, pour éviter les conflits, nous allons utiliser Docker pour démarrer Hadoop. Il faut d'abord cloner l'image du cluster Hadoop via ce lien : <https://github.com/big-data-europe/docker-hadoop>, en utilisant Git avec cette commande :

```
git clone https://github.com/big-data-europe/docker-hadoop.git
```

FIGURE 1.2 – Clonage du cluster Hadoop

1.2.1 Déploiement

Pour finaliser le processus de déploiement, il faut démarrer le conteneur déjà installé en utilisant la commande suivante de Docker :



FIGURE 1.3 – Démarrage du conteneur

Après avoir exécuté la commande précédente, tous les conteneurs du cluster Hadoop seront démarrés :

```
C:\Windows\System32\cmd.exe x + -
C:\Users\bensl\Desktop\Education\Ing-2\TP-Ing2\S1\Big-Data>git clone https://github.com/big-data-europe/docker-hadoop.git
Cloning into 'docker-hadoop'...
remote: Enumerating objects: 539, done.
remote: Counting objects: 100% (189/189), done.
remote: Compressing objects: 100% (23/23), done.
Receiving objects: 100% (539/539), 100.00 KiB | 15.00 MiB/s, done.
Resolving deltas: 3% (8/251) reused 166 (delta 166), pack-reused 350 (from 1)Resolving deltas: 0% (0/251)
Resolving deltas: 100% (251/251), done.
C:\Users\bensl\Desktop\Education\Ing-2\TP-Ing2\S1\Big-Data>docker-compose up -d
no configuration file provided: not found
C:\Users\bensl\Desktop\Education\Ing-2\TP-Ing2\S1\Big-Data>dir
Volume in drive C is OS
Volume Serial Number is DESA-6ADD

Directory of C:\Users\bensl\Desktop\Education\Ing-2\TP-Ing2\S1\Big-Data

27/09/2024 14:34 <DIR>      .
27/09/2024 14:33 <DIR>      ..
27/09/2024 14:34 <DIR>      docker-hadoop
                0 File(s)          0 bytes
                3 Dir(s) 163 868 282 880 bytes free






C:\Users\bensl\Desktop\Education\Ing-2\TP-Ing2\S1\Big-Data>cd docker-hadoop
C:\Users\bensl\Desktop\Education\Ing-2\TP-Ing2\S1\Big-Data\docker-hadoop>docker-compose up -d
time="2024-09-27T14:35:16+01:00" level=warning msg="C:\\Users\\bensl\\Desktop\\Education\\Ing-2\\TP-Ing2\\S1\\Big-Data\\docker-hadoop\\
docker-compose.yml: 'version' is obsolete"
[+] Running 22/28
```

FIGURE 1.4 – Résultat de la commande Docker

Q Search


☰

☑ Only show running containers

<input type="checkbox"/>	Name	Image	Status	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	<div><div> determined_snyder</div><div>943348ee7656 </div></div>	react-docker	Exited	5173:5173	0%	3 months ago	<div><div>▶</div><div>⋮</div><div></div></div>
<input type="checkbox"/>	> <div><div> docker-hadoop</div></div>		Running (5/5)		3.17%	2 minutes ago	<div><div>■</div><div>⋮</div><div></div></div>


Showing 2 items

Walkthroughs



Multi-container applications

8 mins



Containerize your application

3 mins


[View more in the Learning center](#)

RAM 5.31 GB CPU 0.70% Signed in

New version available

3

Q Search



ENG FR

14:59

27/09/2024

FIGURE 1.5 – Liste des conteneurs Docker

Chapitre 2

Commandes de HDFS

Parmi les conteneurs du cluster Hadoop, il y a le *namenode*, auquel nous allons nous connecter en utilisant la commande suivante :

```
docker exec -it namenode bash
```

FIGURE 2.1 – Démarrer le *namenode*

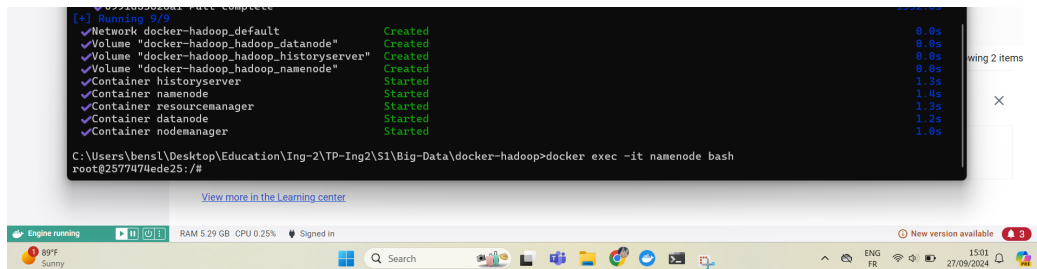


FIGURE 2.2 – Résultat de la commande

Afin de continuer à tester HDFS, nous allons créer un fichier texte nommé *bonjour.txt* en utilisant la commande : **echo "Bonjour Hadoop et HDFS"**
> **bonjour.txt**

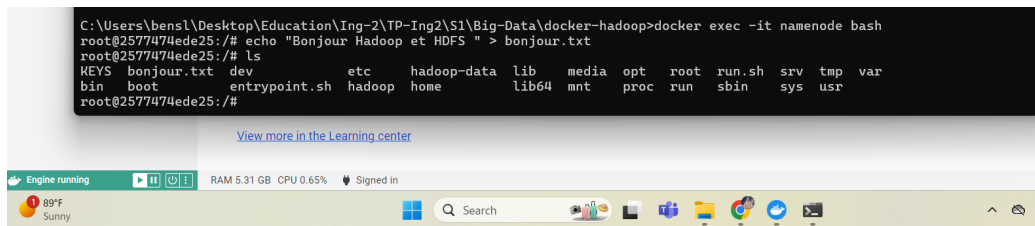


FIGURE 2.3 – Création du fichier *bonjour.txt*

Pour interagir avec le système Hadoop, nous utilisons des commandes qui commencent par *hdfs dfs*, avec des options inspirées des commandes du système Unix. Dans cette section, nous allons tester quelques commandes afin de manipuler les fichiers et dossiers sur le disque local du conteneur *namenode* ainsi que les fichiers *HDFS*, visibles en exécutant *hdfs dfs -ls*.

2.1 Commandes de HDFS

hdfs dfs -ls Cette commande permet d'afficher la liste des fichiers et dossiers présents dans le système HDFS.

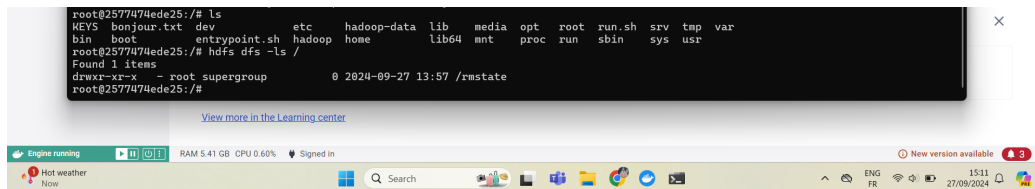


FIGURE 2.4 – Affichage des fichiers

hdfs dfs -put bonjour.txt Cette commande permet de copier le fichier du disque local du conteneur vers le système HDFS.

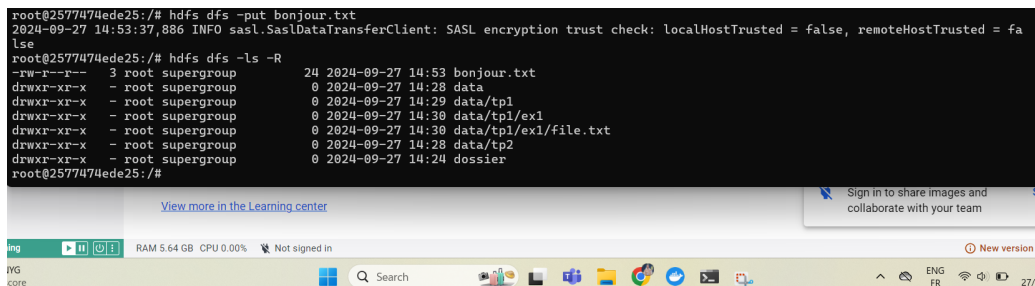


FIGURE 2.5 – Copie du fichier dans HDFS

hdfs dfs -cat bonjour.txt Cette commande permet d'afficher le contenu du fichier texte *bonjour.txt* situé dans HDFS.

```
root@2577474ede25:/# hdfs dfs -cat bonjour.txt
2024-09-27 14:59:26,075 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
lse
Bonjour Hadoop et HDFS
root@2577474ede25:/#
```

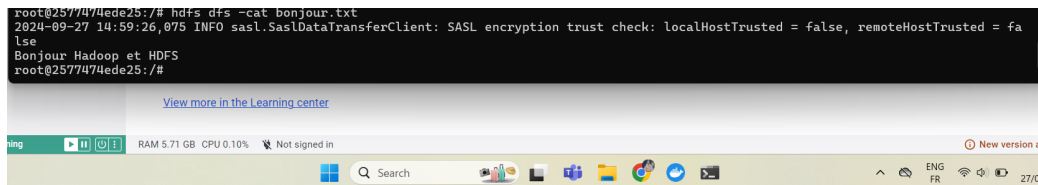


FIGURE 2.6 – Affichage du contenu du fichier

hdfs dfs -rm bonjour.txt Cette commande permet de supprimer le fichier texte *bonjour.txt* du système HDFS.

```
root@2577474ede25:/# hdfs dfs -rm bonjour.txt
Deleted bonjour.txt
root@2577474ede25:/# hdfs dfs -ls -R
drwxr-xr-x - root supergroup 0 2024-09-27 14:28 data
drwxr-xr-x - root supergroup 0 2024-09-27 14:29 data/tp1
drwxr-xr-x - root supergroup 0 2024-09-27 14:30 data/tp1/ex1
drwxr-xr-x - root supergroup 0 2024-09-27 14:30 data/tp1/ex1/file.txt
drwxr-xr-x - root supergroup 0 2024-09-27 14:28 data/tp2
drwxr-xr-x - root supergroup 0 2024-09-27 14:24 dossier
root@2577474ede25:/#
```

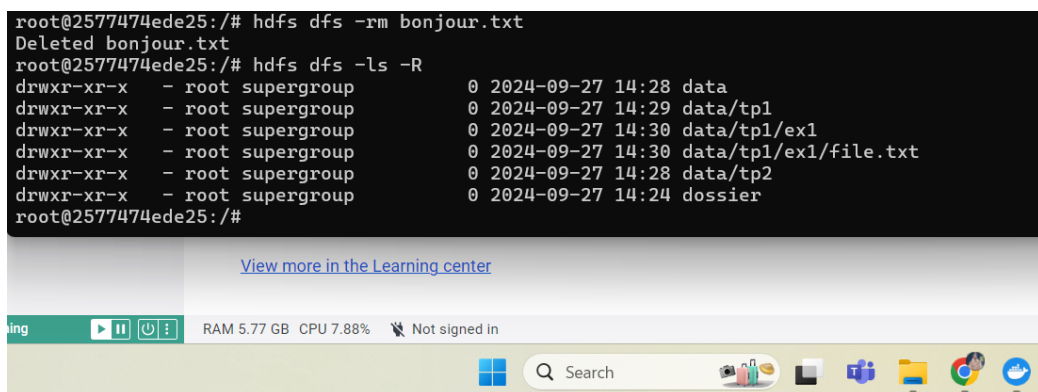


FIGURE 2.7 – Suppression du fichier

hdfs dfs -copyFromLocal bonjour.txt Cette commande est similaire à *hdfs dfs -put bonjour.txt*, permettant de copier un fichier local vers HDFS.

```
root@2577474ede25:/# hdfs dfs -copyFromLocal bonjour.txt
2024-09-27 15:06:30,880 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
lse
root@2577474ede25:/# hdfs dfs -ls -R
-rw-r--r-- 3 root supergroup 24 2024-09-27 15:06 bonjour.txt
drwxr-xr-x - root supergroup 0 2024-09-27 14:28 data
drwxr-xr-x - root supergroup 0 2024-09-27 14:29 data/tp1
drwxr-xr-x - root supergroup 0 2024-09-27 14:30 data/tp1/ex1
drwxr-xr-x - root supergroup 0 2024-09-27 14:30 data/tp1/ex1/file.txt
drwxr-xr-x - root supergroup 0 2024-09-27 14:28 data/tp2
drwxr-xr-x - root supergroup 0 2024-09-27 14:24 dossier
root@2577474ede25:/#
```

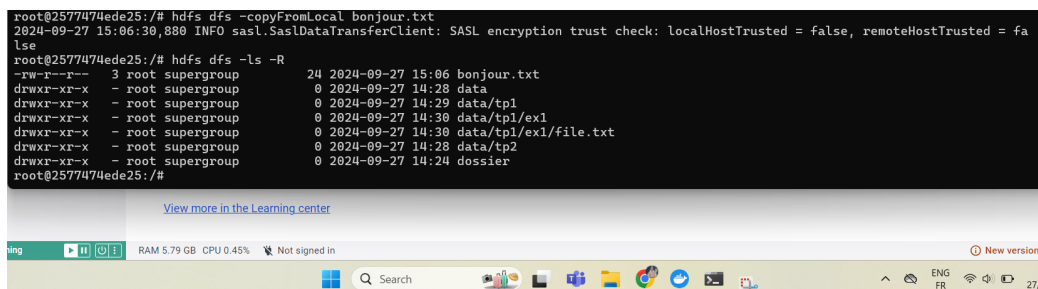


FIGURE 2.8 – Copie du fichier depuis le local

hdfs dfs -chmod go+w bonjour.txt Cette commande permet de configurer les droits d'accès pour le fichier *bonjour.txt*.

```
root@2577474ede25:/# hdfs dfs -chmod go+w bonjour.txt
root@2577474ede25:/# hdfs dfs -ls
Found 3 items
-rw-rw-rw-   3 root supergroup      24 2024-09-27 15:06 bonjour.txt
drwxr-xr-x   - root supergroup      0 2024-09-27 14:28 data
drwxr-xr-x   - root supergroup      0 2024-09-27 14:24 dossier
root@2577474ede25:/#
```

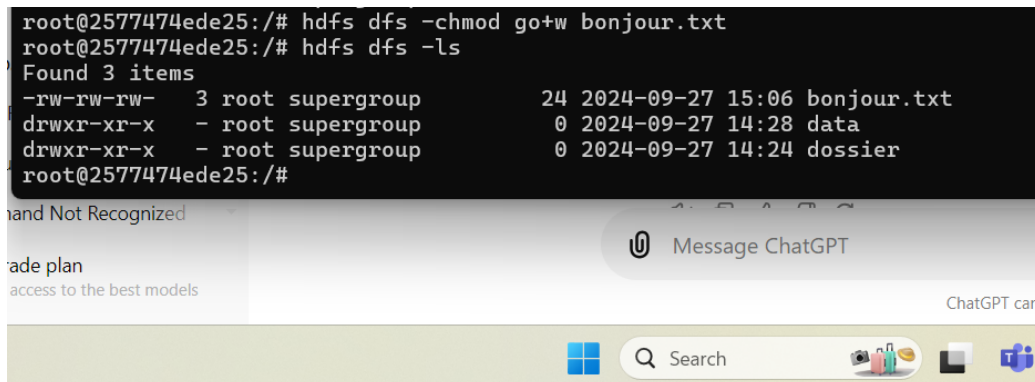


FIGURE 2.9 – Configuration des droits d'accès

hdfs dfs -mv bonjour.txt dossier/bonjour.txt Cette commande permet de déplacer le fichier *bonjour.txt* vers le dossier spécifié.

```
root@2577474ede25:/# hdfs dfs -chmod go-r bonjour.txt
root@2577474ede25:/# hdfs dfs -ls
Found 3 items
-rw--w--w-   3 root supergroup      24 2024-09-27 15:06 bonjour.txt
drwxr-xr-x   - root supergroup      0 2024-09-27 14:28 data
drwxr-xr-x   - root supergroup      0 2024-09-27 14:24 dossier
root@2577474ede25:/# hdfs dfs -mv bonjour.txt dossier/bonjour.txt
root@2577474ede25:/# hdfs dfs -ls -R
drwxr-xr-x   - root supergroup      0 2024-09-27 14:28 data
drwxr-xr-x   - root supergroup      0 2024-09-27 14:29 data/tp1
drwxr-xr-x   - root supergroup      0 2024-09-27 14:30 data/tp1/ex1
drwxr-xr-x   - root supergroup      0 2024-09-27 14:30 data/tp1/ex1/file.txt
drwxr-xr-x   - root supergroup      0 2024-09-27 14:28 data/tp2
drwxr-xr-x   - root supergroup      0 2024-09-27 15:15 dossier
-rw--w--w-   3 root supergroup      24 2024-09-27 15:06 dossier/bonjour.txt
root@2577474ede25:/#
```

FIGURE 2.10 – Déplacement du fichier

Chapitre 3

Création d'une arborescence et téléchargement de fichiers

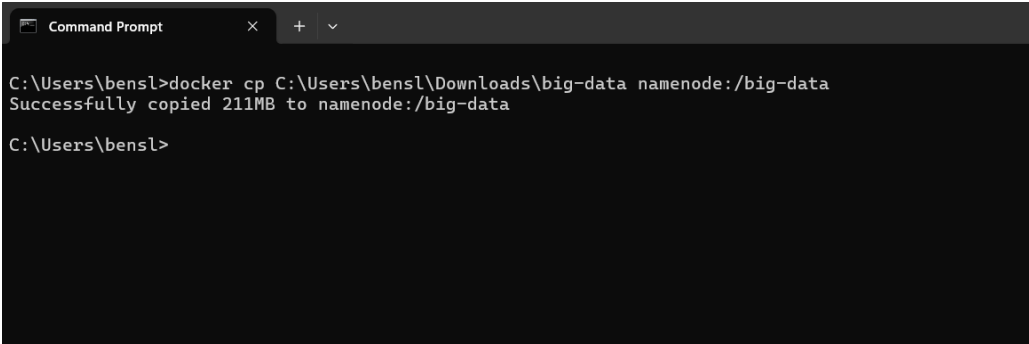
Dans ce chapitre, nous allons télécharger le fichier *purchases.txt* via ce lien <https://www.kaggle.com/datasets/dsfelix/purchases.txt>, ainsi que le fichier *pg4300.txt* via ce lien <https://www.gutenberg.org/cache/epub/4300/pg4300.txt>.

Pour copier un fichier de la machine hôte vers le conteneur *namenode*, nous utilisons la commande suivante :

A dark-themed terminal window with three colored window control buttons (red, yellow, green) in the top-left corner. The text inside the terminal is the Docker command to copy a file from the host to a container.

```
docker cp C:\Users\bensl\Downloads\big-data namenode:/big-data
```

FIGURE 3.1 – Commande Docker cp

A Windows Command Prompt window titled "Command Prompt" with standard window controls. It shows the execution of the Docker command and its successful outcome.

```
C:\Users\bensl>docker cp C:\Users\bensl\Downloads\big-data namenode:/big-data
Successfully copied 211MB to namenode:/big-data
C:\Users\bensl>
```

FIGURE 3.2 – Résultat de la commande

Nous répétons la même opération avec le fichier *pg4300.txt* :

```
C:\Users\bensl>docker cp C:\Users\bensl\Downloads\pg4300.txt namenode:/pg4300.txt
Successfully copied 1.59MB to namenode:/pg4300.txt

C:\Users\bensl>
```

FIGURE 3.3 – Résultat de la commande

Ensuite, nous déplaçons le fichier du conteneur vers le système HDFS en utilisant la commande **hdfs dfs -put file.txt** :

```
root@2577474ede25:/# hdfs dfs -put pg4300.txt
2024-09-27 17:04:03,937 INFO sasl.SaslDataTransferClient: SASL encryption trust check:
lse
root@2577474ede25:/# hdfs dfs -ls -R
drwxr-xr-x - root supergroup 0 2024-09-27 14:28 data
drwxr-xr-x - root supergroup 0 2024-09-27 14:29 data/tp1
drwxr-xr-x - root supergroup 0 2024-09-27 14:30 data/tp1/ex1
drwxr-xr-x - root supergroup 0 2024-09-27 14:30 data/tp1/ex1/file.txt
drwxr-xr-x - root supergroup 0 2024-09-27 14:28 data/tp2
drwxr-xr-x - root supergroup 0 2024-09-27 15:24 dossier
-rw-r--r-- 3 root supergroup 1586382 2024-09-27 17:04 pg4300.txt
root@2577474ede25:/# hdfs dfs -mv pg4300.txt data/pg4300.txt
root@2577474ede25:/# hdfs dfs -ls -R
drwxr-xr-x - root supergroup 0 2024-09-27 17:06 data
-rw-r--r-- 3 root supergroup 1586382 2024-09-27 17:04 data/pg4300.txt
drwxr-xr-x - root supergroup 0 2024-09-27 14:29 data/tp1
drwxr-xr-x - root supergroup 0 2024-09-27 14:30 data/tp1/ex1
drwxr-xr-x - root supergroup 0 2024-09-27 14:30 data/tp1/ex1/file.txt
drwxr-xr-x - root supergroup 0 2024-09-27 14:28 data/tp2
drwxr-xr-x - root supergroup 0 2024-09-27 15:24 dossier
root@2577474ede25:/#
```

FIGURE 3.4 – Résultat de la commande

Voici maintenant l'arborescence de notre système de fichiers Hadoop :

```
root@2577474ede25:/# hdfs dfs -ls -R data/
-rw-r--r-- 3 root supergroup 1586382 2024-09-27 17:04 data/pg4300.txt
drwxr-xr-x - root supergroup 0 2024-09-27 14:29 data/tp1
drwxr-xr-x - root supergroup 0 2024-09-27 14:30 data/tp1/ex1
drwxr-xr-x - root supergroup 0 2024-09-27 14:30 data/tp1/ex1/file.tx
drwxr-xr-x - root supergroup 0 2024-09-27 14:28 data/tp2
root@2577474ede25:/#
```

FIGURE 3.5 – Arborescence du système de fichiers Hadoop

Chapitre 4

État du cluster

Après toutes ces opérations, nous pouvons mentionner que les services Hadoop génèrent des pages web automatiquement pour permettre de suivre leur fonctionnement. En cliquant sur ce lien <http://localhost:9870/dfshealth.html#>, nous pouvons accéder à plusieurs pages liées aux différents services Hadoop, telles que :

4.1 Page Overview

Il y a un tableau *Summary* où l'on peut voir l'espace total, l'espace utilisé et l'espace libre dans ce conteneur.

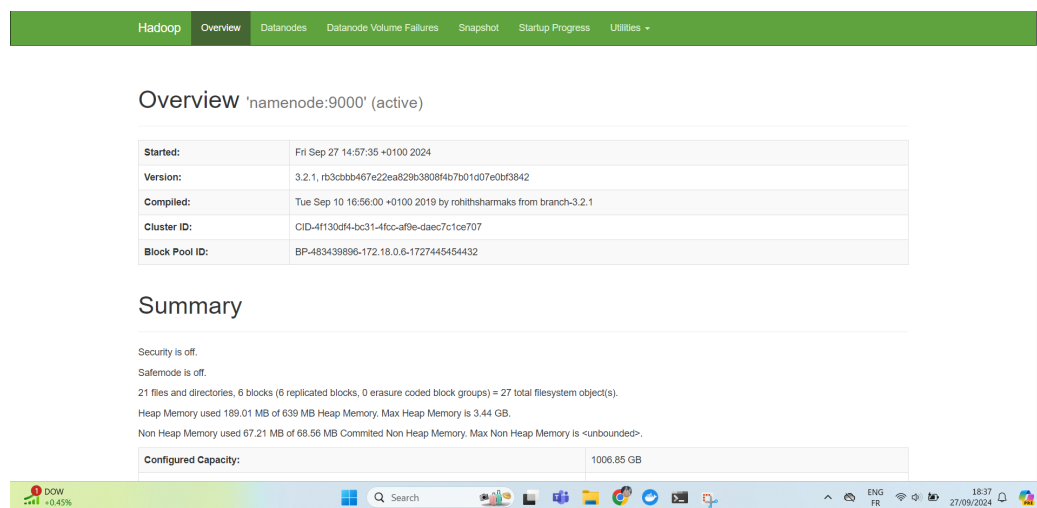


FIGURE 4.1 – Page Overview du cluster

4.2 Page DataNodes

Cette page affiche la capacité et la charge de chaque DataNode.

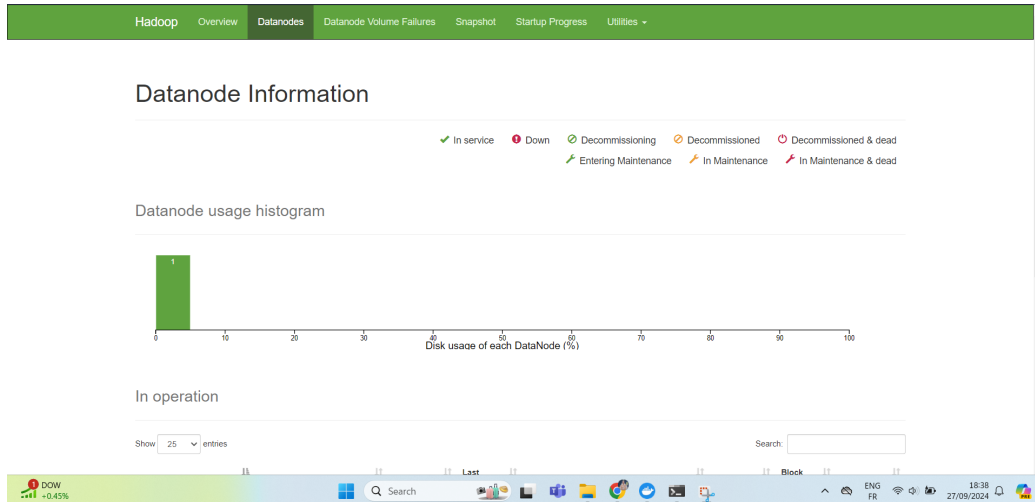


FIGURE 4.2 – Page DataNodes

4.3 Page Utilities

Nous pouvons parcourir l'arborescence des fichiers HDFS. En cliquant sur le nom d'un fichier, des informations sur les blocs et les machines contenant ce fichier sont affichées.

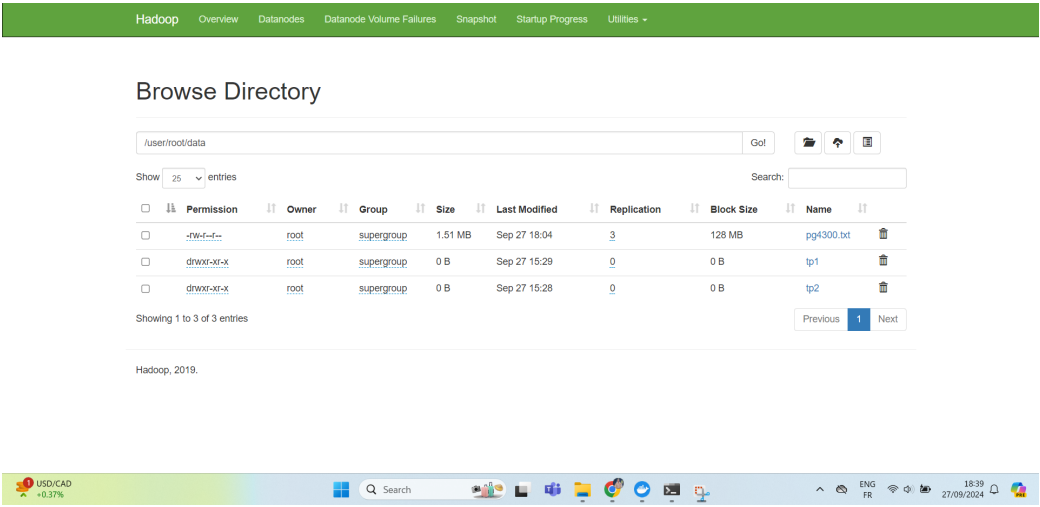


FIGURE 4.3 – Page Utilities

Chapitre 5

Conclusion

À la fin de ce TP, il est important de retenir quelques points clés. L'utilisation de Docker est un excellent moyen d'éviter les conflits d'installation, car elle permet de bénéficier d'images prêtes à l'emploi, simplifiant ainsi la configuration de l'environnement.

Manipuler les fichiers entre le conteneur et le système de fichiers Hadoop en utilisant des commandes inspirées des commandes Unix rend le premier contact avec HDFS plus intuitif et accessible.