### **Hadoop**

[Apache Hadoop](https://insatunisia.github.io/TP-BigData/tp1/hadoop.apache.org) est un framework open-source pour stocker et traiter les données volumineuses sur un cluster. Il est utilisé par un grand nombre de contributeurs et utilisateurs. Il a une licence Apache 2.0.



#### **Hadoop et Docker :**

Pour déployer le framework Hadoop, nous allons utiliser des contenaires [Docker](https://www.docker.com/). L'utilisation des contenaires 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).

#### **Installation :**

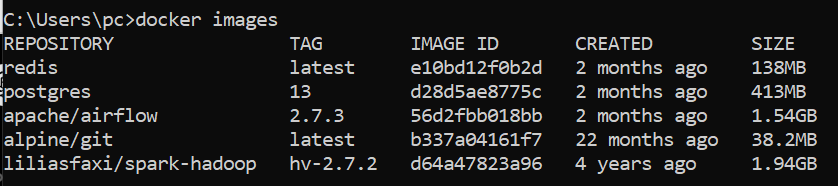
Nous allons utiliser pour ce projet trois contenaires représentant respectivement un noeud maître (Namenode) et deux noeuds esclaves (Datanodes).

Nous avons pour cela installé docker sur notre machine, et l'avoir correctement configuré.

Téléchargement de l’image docker uploadée sur dockerhub:



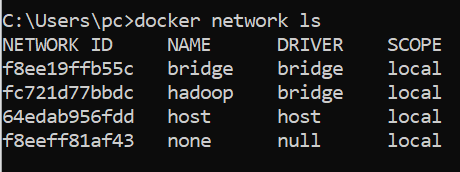
notre image est uploadée :



**Création du réseau nommé hadoop qui permettra de relier les trois contenaires:**

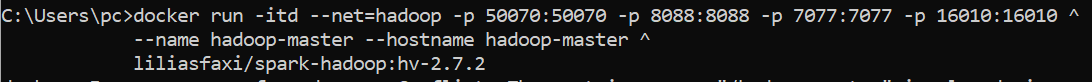
****

le réseau hadoop a été bien crée :

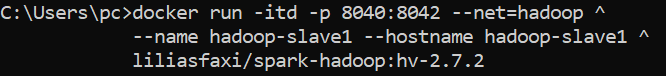


**Création des conteneurs :**

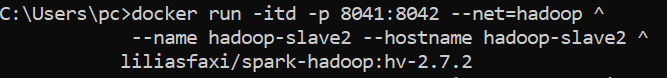
Conteneur **hadoop-master :**



Conteneur **hadoop-slave1 :**

****

Conteneur **hadoop-slave2 :**

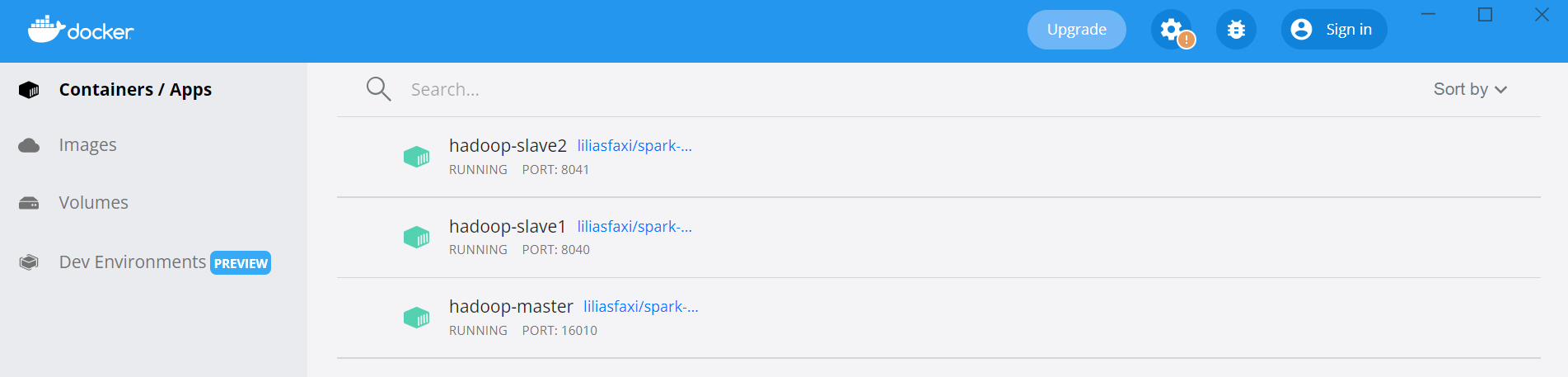
****

les instructions -p permettent de faire un mapping entre les ports de la machine hôte et ceux du contenaire.

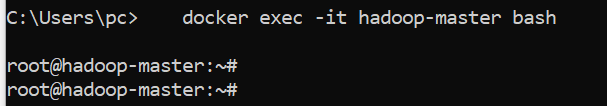
**Les 3 Conteneurs ont été bien crées :**



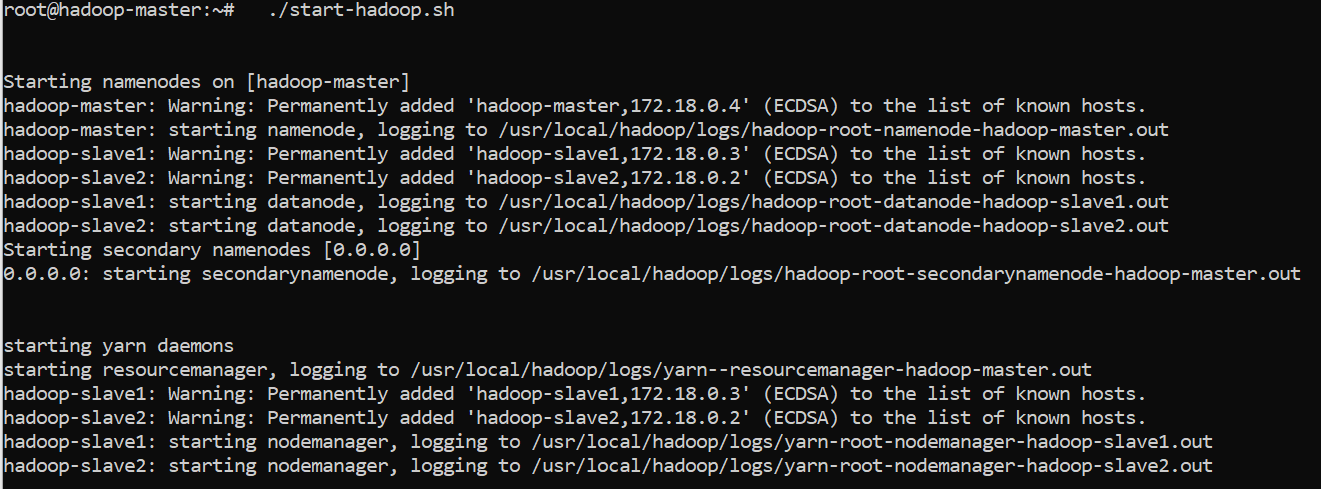
Nos 3 conteneurs sont actives et prêtes à être utilisés :



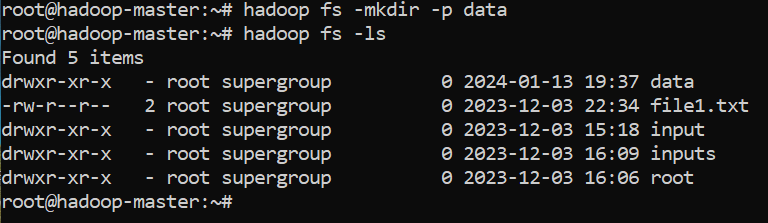
On entre dans le conteneur master pour commencer à l'utiliser :



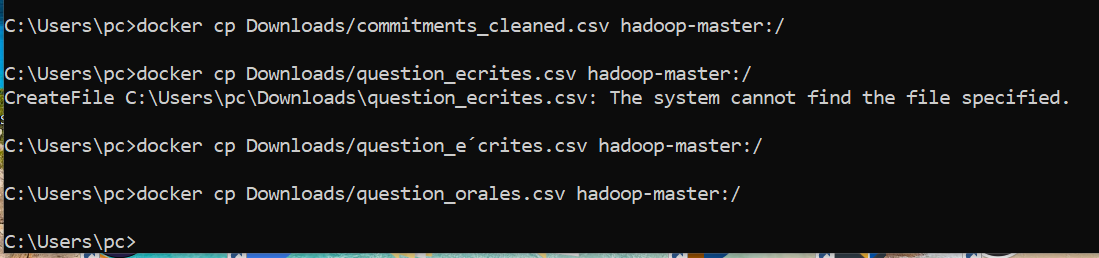
Nous nous retrouvons dans le shell du namenode, et nous pourrons maintenent manipuler le cluster à notre guise. La première chose à faire, une fois dans le conteneur, est de lancer hadoop et yarn :



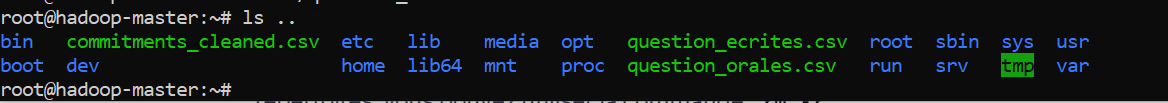
Création du répertoire data dans HDFS :



Copier les fichiers csv scrapés de ma machine locale vers le conteneur **hadoop-master :**



On revient maintenant vers hadoop-master pour vérifier si les fichiers existent bien dans le namenode :



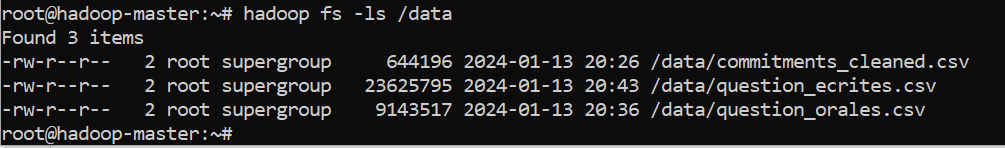
Copier les fichiers csv du **hadoop-master** vers **HDFS** dans le répertoire **data**:







Nos fichiers existent bien dans le répertoire **data** dans HDFS**:**

****

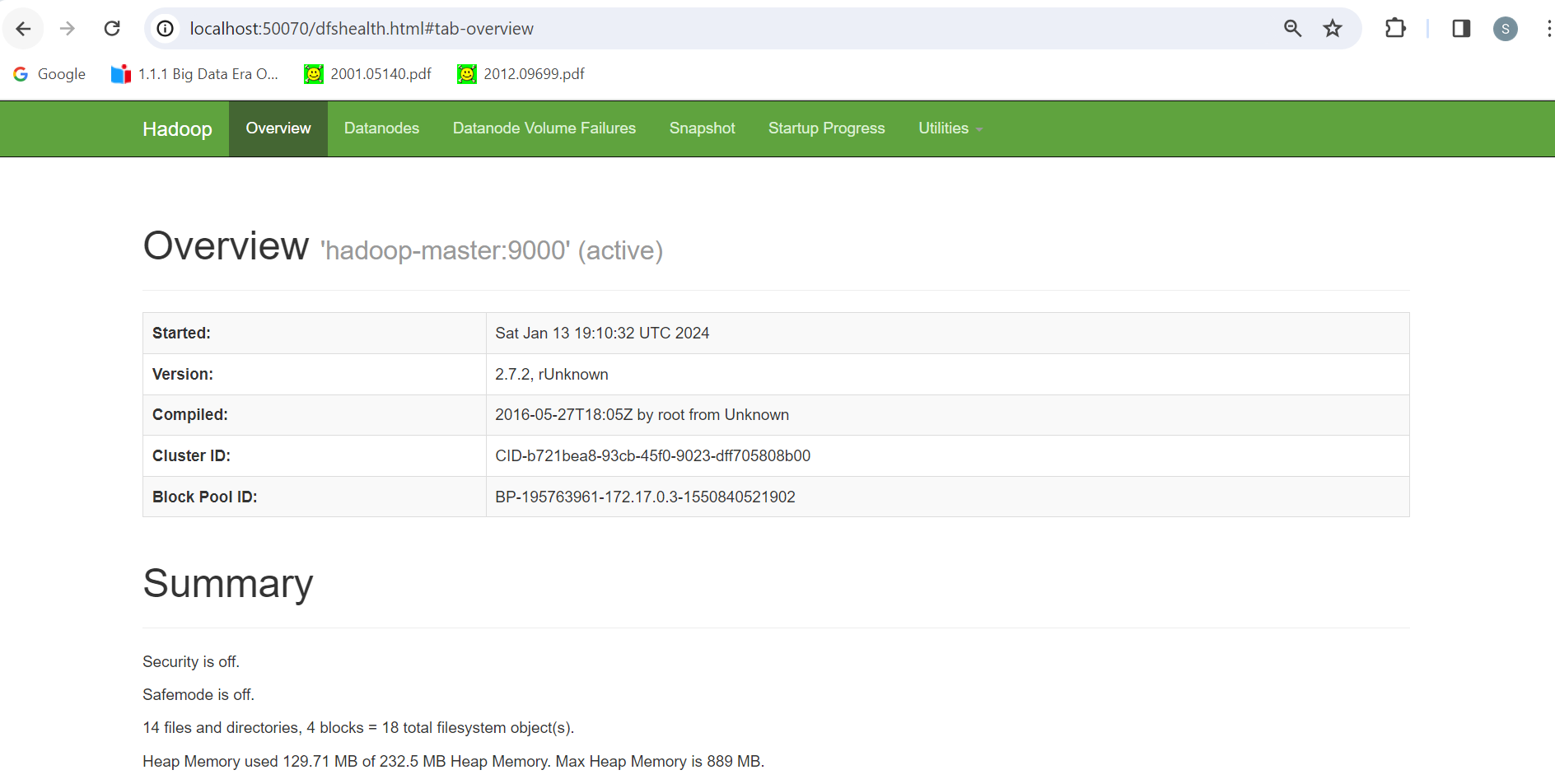
**Interfaces web pour Hadoop :**

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 contenaire sur la machine hôte. Pour pouvoir publier tous les ports exposés, vous pouvez lancer votre contenaire 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:

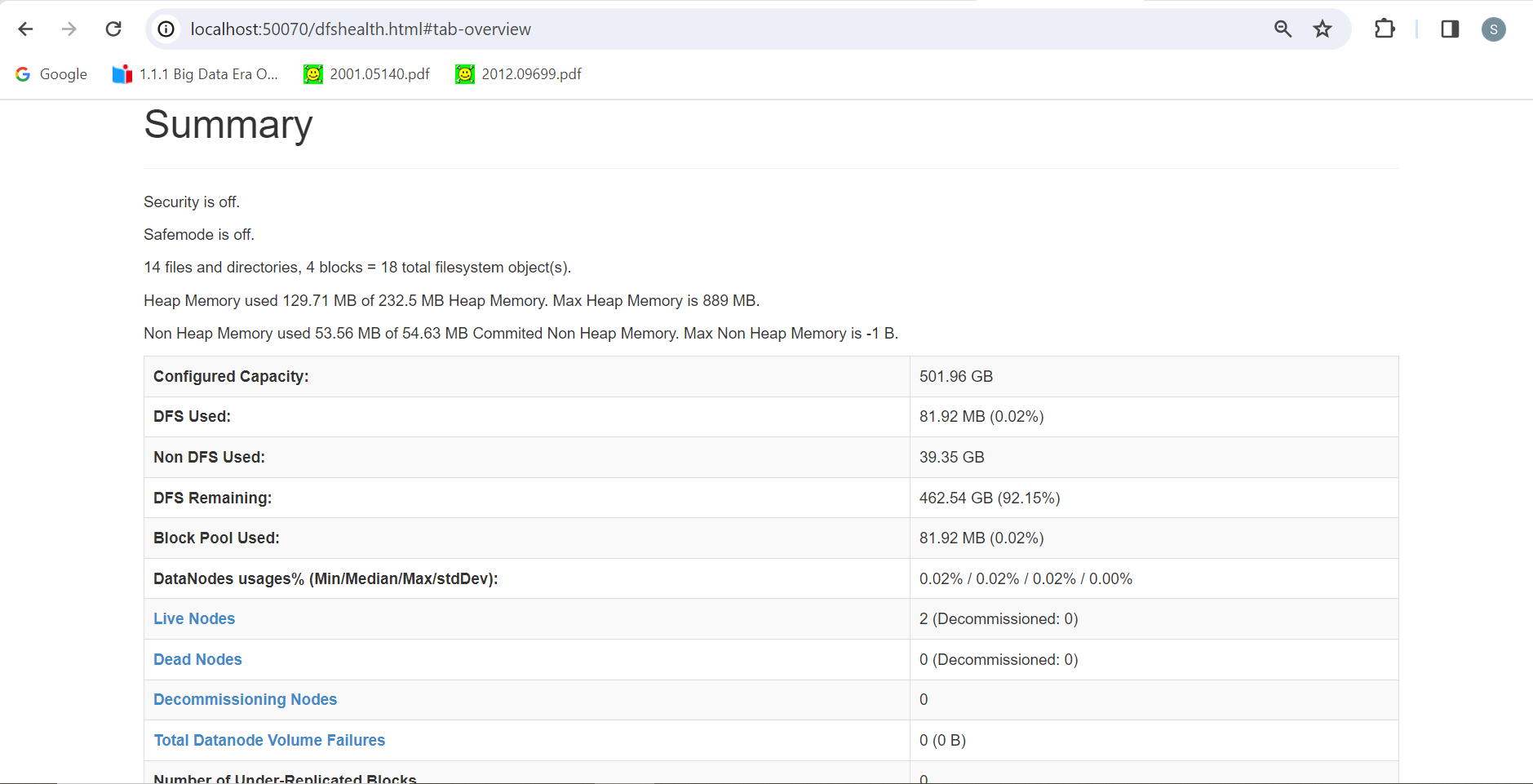
* Le port 50070: qui permet d'afficher les informations du namenode.
* Le port 8088: qui permet d'afficher les informations du resource manager de Yarn et visualiser le comportement des différents jobs.

Notre cluster est lancé et prêt à l'emploi sur : <http://localhost:50070>.

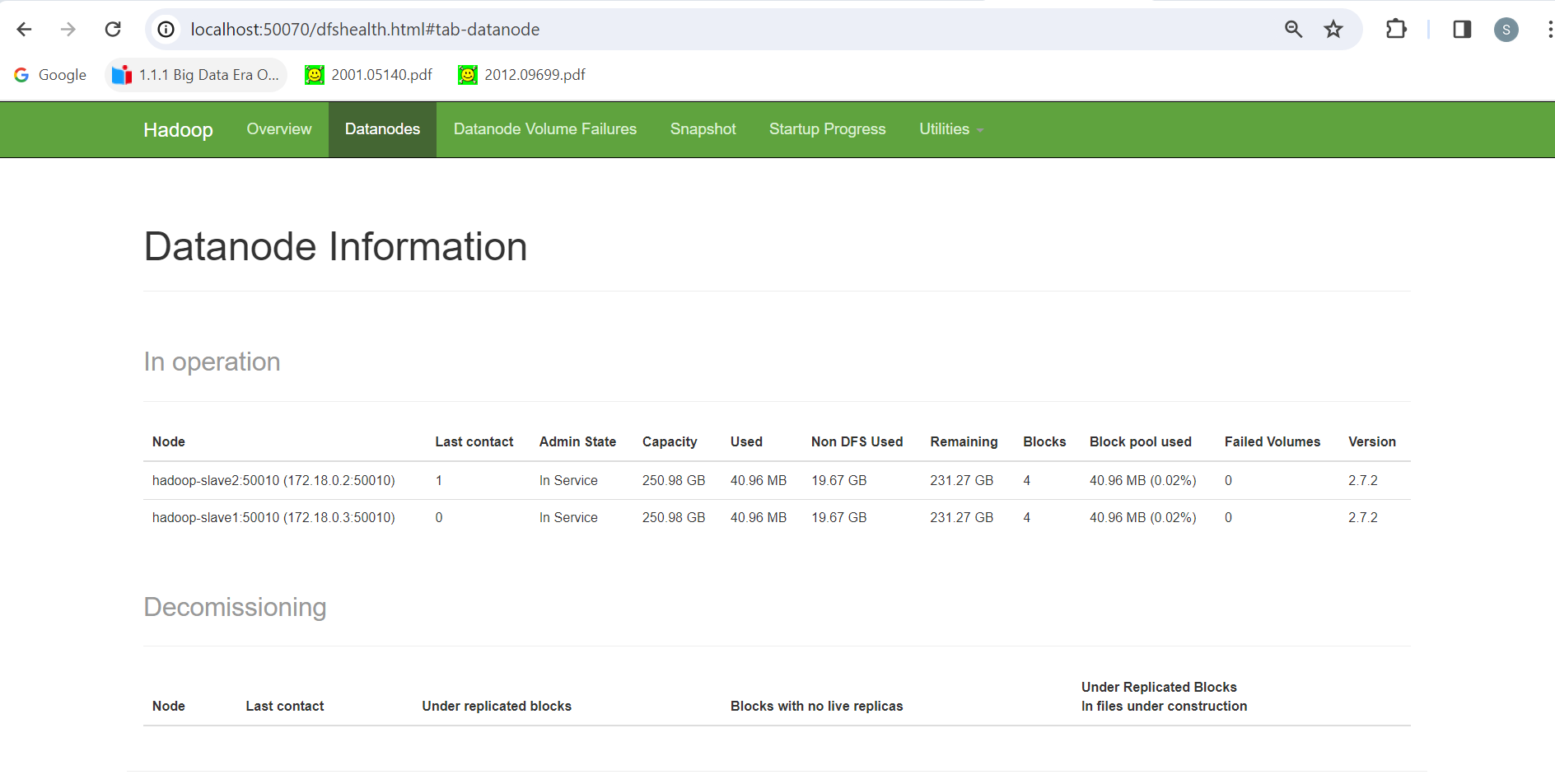


Dans la section summary on obtient des informations supplémentaires telles que :

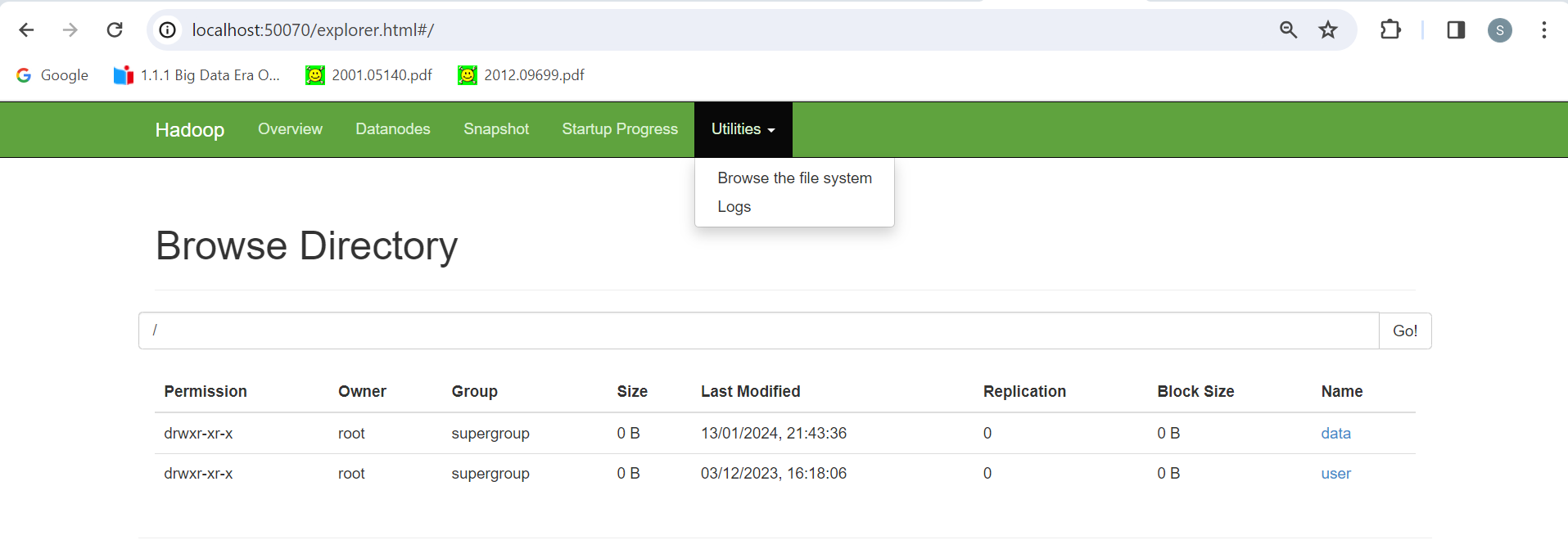
* Distributed File System(DFS) used.
* DFS remaining
* live nodes
* nombre de blocs



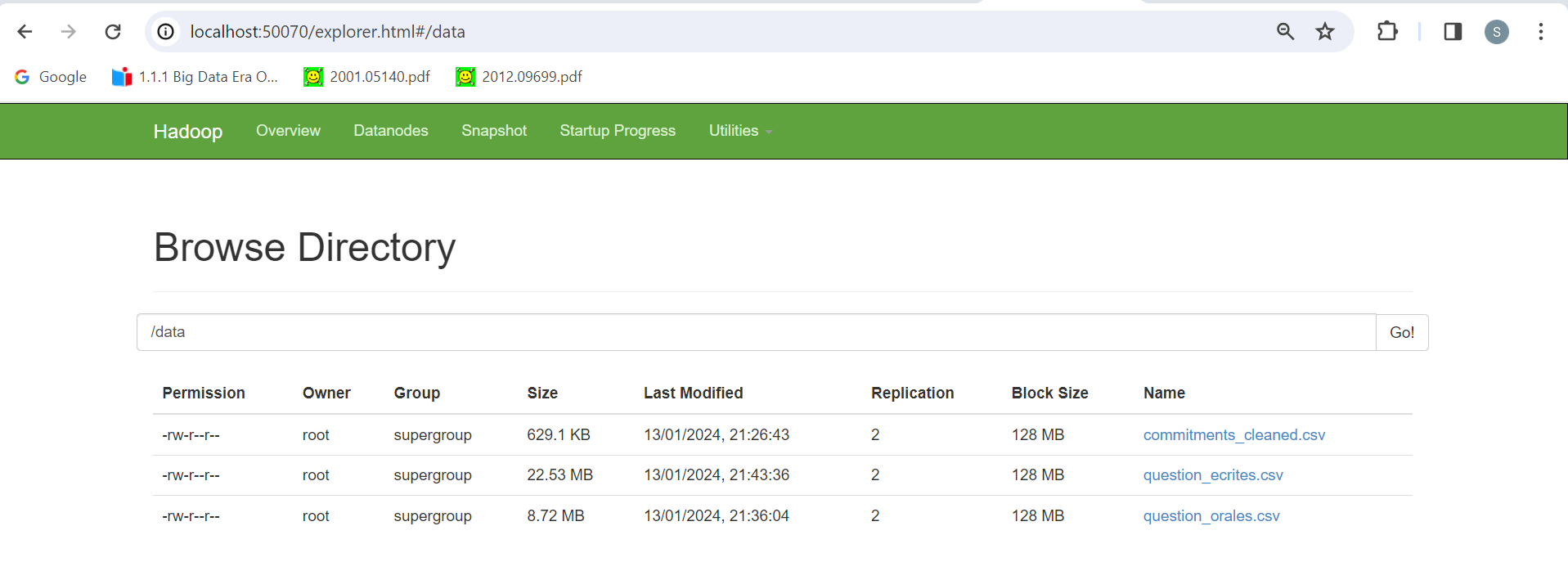
**Datanodes :**

****

Sous **Utilities** on peut voir les informations de notre **DFS** :



Sous le repertoire **data** on trouve nos fichiers csv :



On peut également visualiser l'avancement et les résultats de nos Jobs sur : <http://localhost:8088>

