**Máster Universitario en Big Data y Ciencia de Datos**

**ASIGNATURA: 03MBID *Procesamiento de datos masivos***

*Actividad 1 – Tareas portafolio*

Alumno: **Bru Montes, Israel**

Edición **Octubre 2024 – Grupo A** a 8/11/2024

# 1. Instalación de hadoop en docker.

Instalamos en el sistema tanto docker como docker-compose a través del gestor del sistema:

[ibru@bru-laptop ~]$ sudo pacman -S docker docker-compose

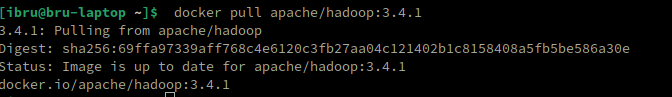
Creamos una carpeta para el intercambio de ficheros entre la máquina del anfitrión.

[ibru@bru-laptop desarrollo]$ mkdir docker\_data

[ibru@bru-laptop desarrollo]$ mkdir docker\_data\hadoop

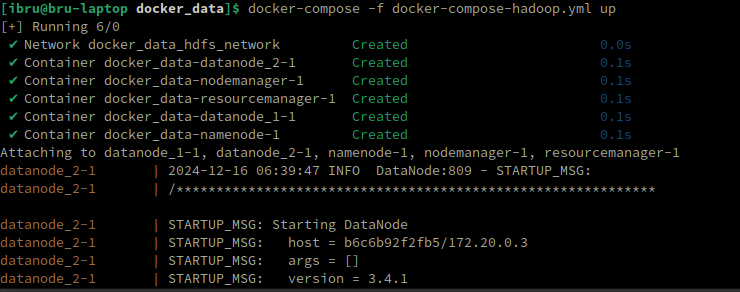
[ibru@bru-laptop desarrollo]$ mkdir docker\_data\hadoop\shared

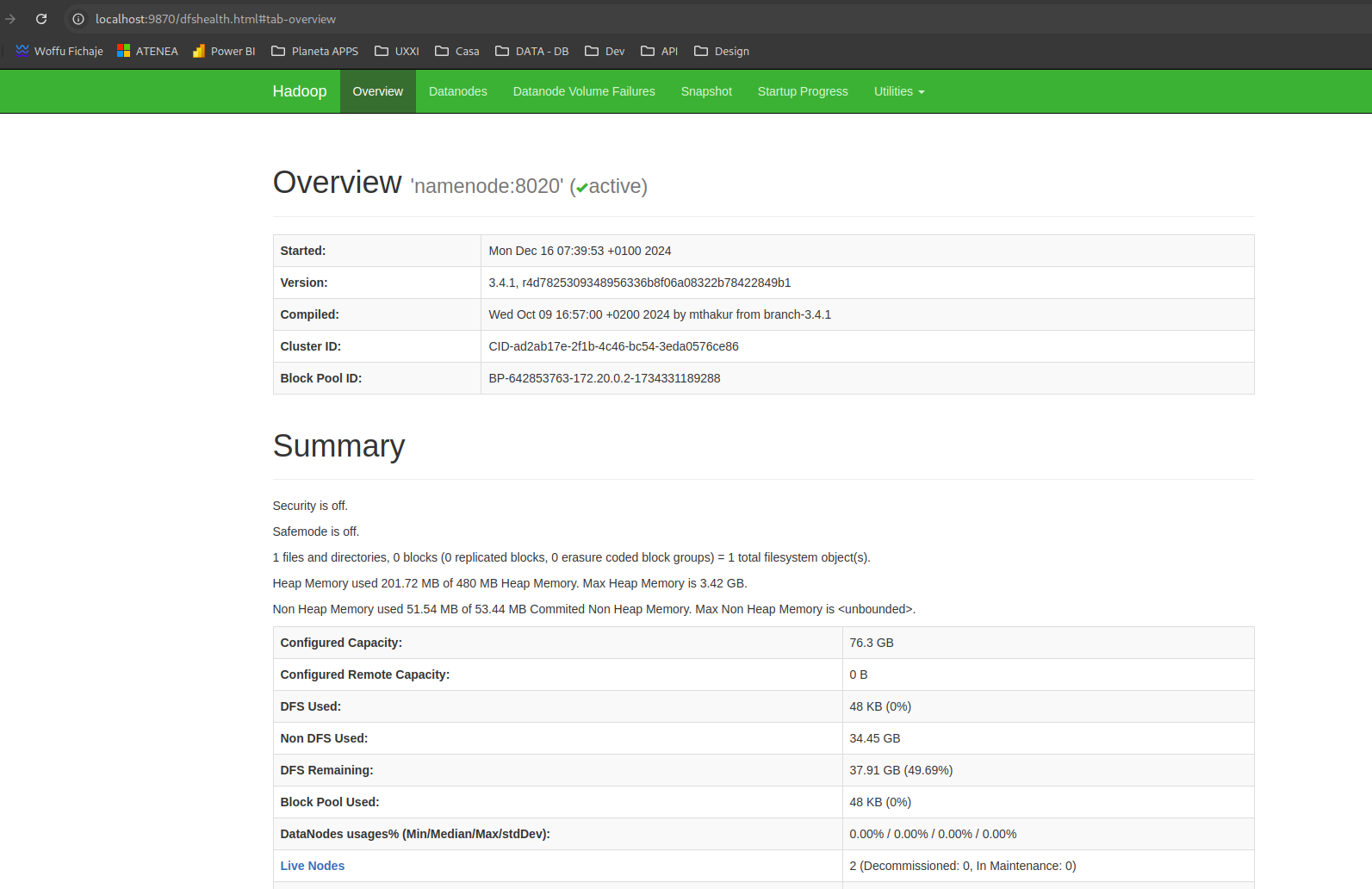
[root@namenode input\_files]# docker pull apache/hadoop:3.4.1

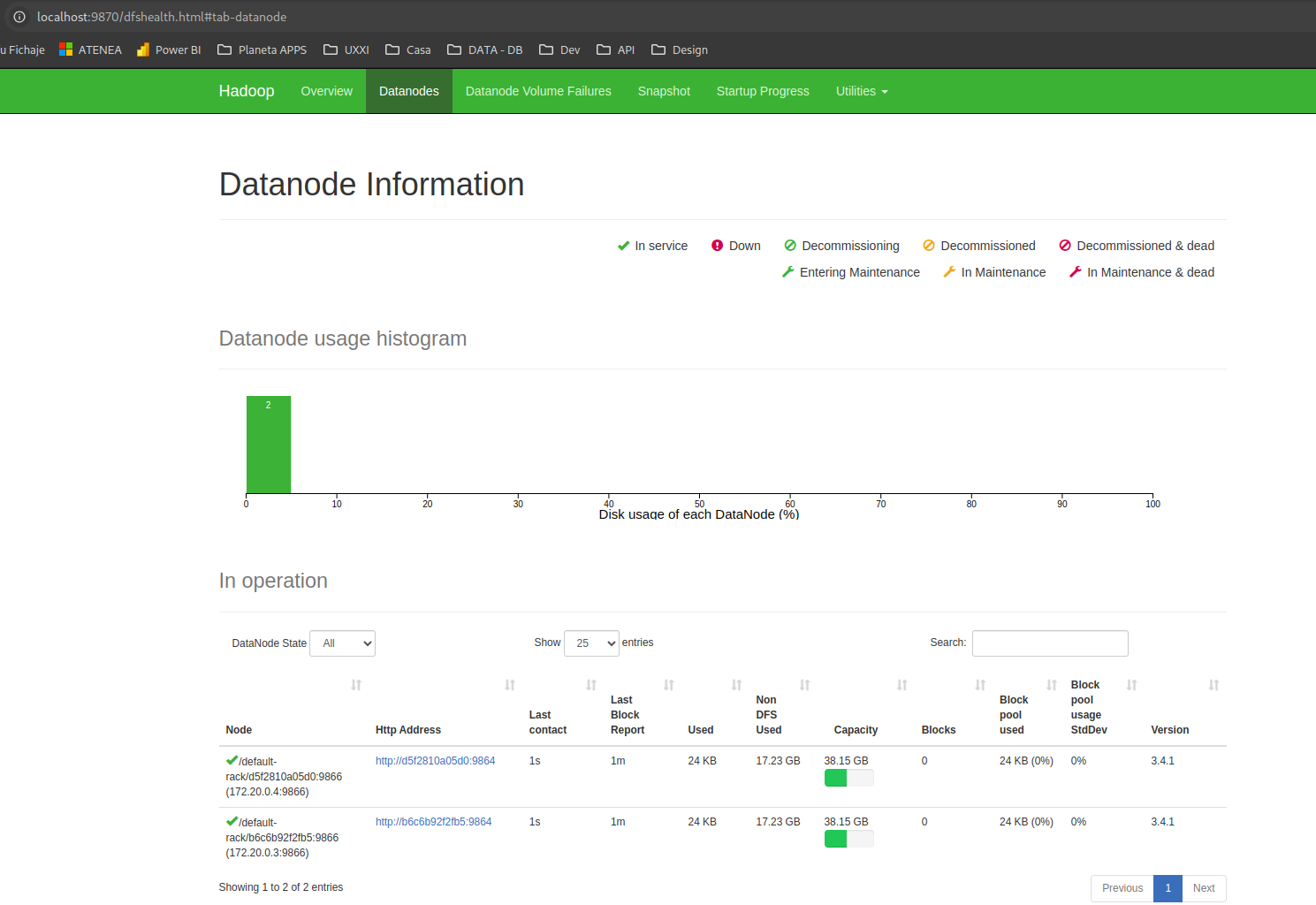


https://medium.com/@bayuadiwibowo/deploying-a-big-data-ecosystem-dockerized-hadoop-spark-hive-and-zeppelin-654014069c82

[ibru@bru-laptop ~]$ docker-compose -f docker-compose-hadoop.yml up





[ibru@bru-laptop ~]$ docker container ls

[root@namenode input\_files]# hadoop fs -mkdir /user

[root@namenode input\_files]# hadoop fs -mkdir /user/root

[root@namenode input\_files]# hdfs dfs -put medidas.txt

2. Ejercicio 1

(7,0 ptos) Dado un dataset que contenga entradas con la forma “persona;tienda;gasto”, crea un programa llamado mediaGastadoPersonaTienda que para cada persona indique su gasto medio por tienda, siguiendo el formato persona;tienda;gastomedio. Ejemplo:

Entrada Salida

Alice;Tienda1;50 Alice;Tienda1;75

Alice;Tienda2;20 Alice;Tienda2;20

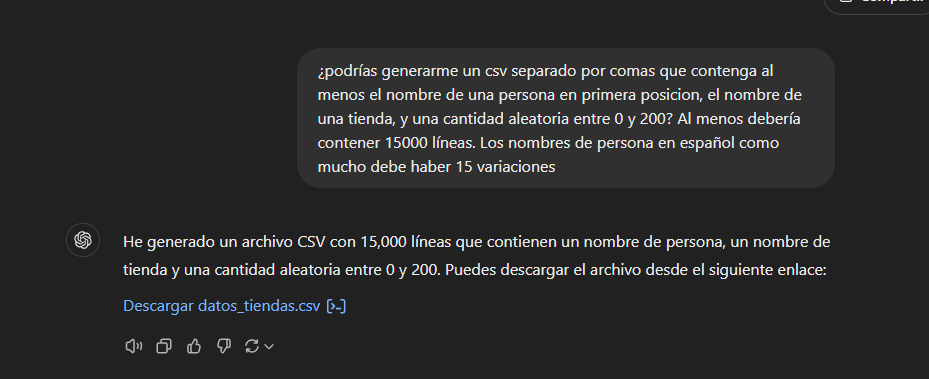
Bob;Tienda1;30 Bob;Tienda1;25

Alice;Tienda1;100

Bob;Tienda1;2

Generamos con ChatGPT un archivo csv con las condiciones que se muestran en la siguiente captura:



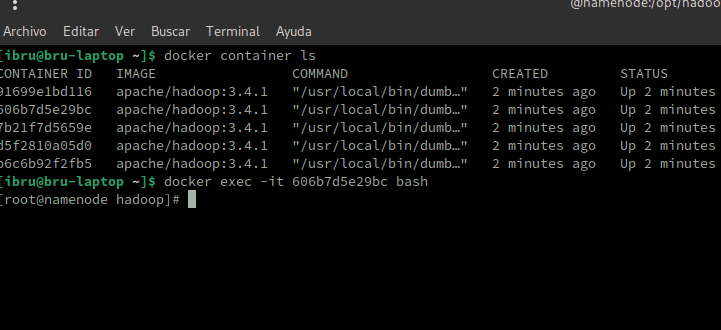


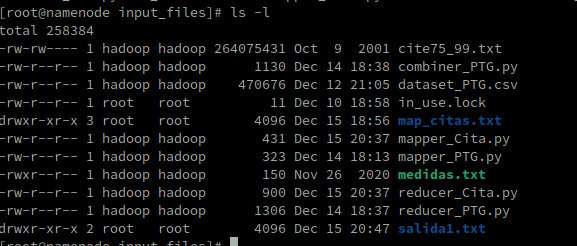
Se genera este archivo para tener más datos.

Generamos 3 archivos para procesar el archivo csv anterior. El mapper únicamente procesa el archivo y crea un nuevo archivo de salida formando una clave cliente-tienda y separado por una tabulación.

[

[ibru@bru-laptop ~]$ docker exec -it 606b7d5e29bc bash

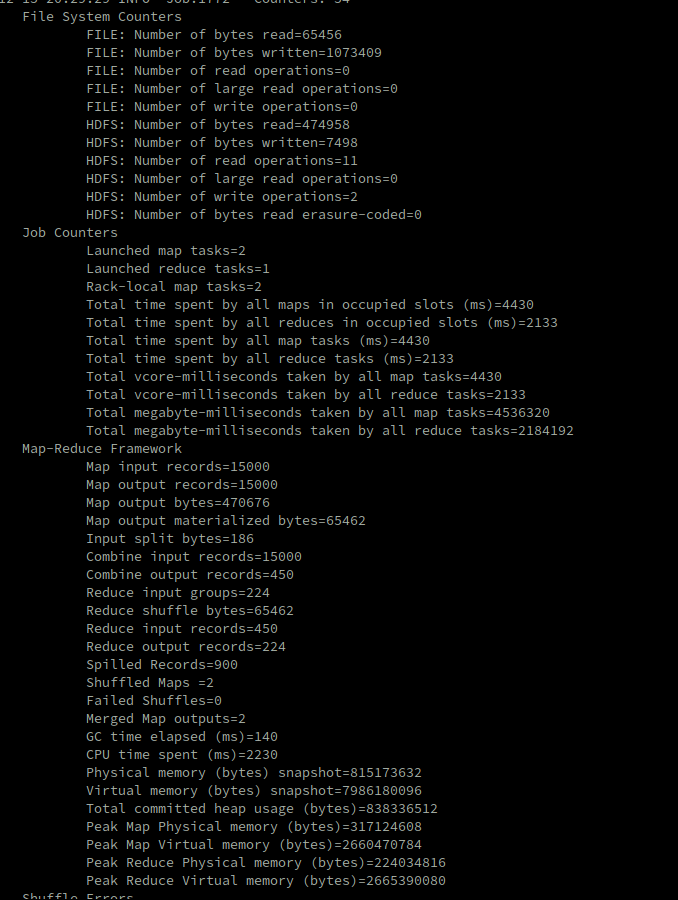




ç

[root@namenode hadoop]# cd input\_files/

root@namenode input\_files]# hadoop jar ../share/hadoop/tools/lib/hadoop-streaming-3.4.1.jar -files mapper\_PTG.py,reducer\_PTG.py,combiner\_PTG.py -mapper mapper\_PTG.py -combiner combiner\_PTG.py -reducer reducer\_PTG.py -input dataset\_PTG.csv -output ptg\_out.txt



2. (7,0 ptos) Para este ejercicio utilizarán el fichero de entrada cite75\_99.txt que puede ser descargado del National Bureau of Economic Research (NBER) de EEUU (<http://www.nber.org/patents/>). Una descripción detallada de este fichero puede encontrarse en:

Hall, B. H., A. B. Jaffe, and M. Trajtenberg (2001). "The NBER Patent Citation Data File: Lessons, Insights and Methodological Tools." NBER Working Paper 8498.

Este fichero contiene citas de patentes emitidas entre 1975 y 1990 en los EEUU. Es un fichero CSV (comma-separated values) con más de 16,5 millones de filas, y las primeras líneas son como sigue:

"CITING","CITED"

3858241,956203

3858241,1324234

3858241,3398406

3858241,3557384

3858241,3634889

3858242,1515701

3858242,3319261

3858242,3668705

[root@namenode hadoop]# cd input\_files/

[root@namenode input\_files]# hdfs dfs -put cite75\_99.txt

put: `.': No such file or directory: `hdfs://namenode/user/root'

[root@namenode input\_files]# hdfs dfs -mkdir /user

^[[A[root@namenode input\_files]# hdfs dfs -mkdir /user/root

[root@namenode input\_files]# hdfs dfs -put cite75\_99.txt

hadoop jar ../share/hadoop/tools/lib/hadoop-streaming-3.4.1.jar -files mapper\_Cita.py,reducer\_Cita.py -mapper mapper\_Cita.py -input cite75\_99.txt -reducer reducer\_Cita.py -output salida1.txt

root@namenode input\_files]# hadoop jar ../share/hadoop/tools/lib/hadoop-streaming-3.4.1.jar -files mapper\_Cita.py,reducer\_Cita.py -mapper mapper\_Cita.py -input cite75\_99.txt -reducer reducer\_Cita.py -output salida1.txt



[root@namenode input\_files]# hdfs dfs -get salida1.txt .

[root@namenode input\_files]# cat salida1.txt/part-00000 |more

10000 3964859,4647229

100000 4539112

1000006 5031388

1000007 4714284

1000011 4766693

1000017 5033339

1000026 3908629

1000033 4043055

1000043 4190903,4975983

1000044 4091523

1000045 4055371,4082383

1000046 4290571

1000049 5525001,5918892

1000051 5996916

1000054 4541310

1000065 4946631

1000067 4748968

1000070 4944640,5071294,5312208

1000073 4928425,5009029

1000076 4107819,5474494

1000083 4867716,5845593

1000084 5322091,5566726

1000086 4182197,4683770

1000089 4178246,4217220,4686189,4839046

1000094 5277853,5395228,5503546,5505607,5505610,5505611,5540869,5544405,5571464,5807591

10001 4897975,4920718,5713167

1000102 4228735

1000108 5120183,5791855

1000118 4399627

100012 4919466,5029926,5058767

3. (6,0 ptos) Dado el siguiente programa Big Data que tiene defectos de diseño, se debe entender cuál es el defecto y hacer un informe que contenga: (1) Nombre y apellidos, (2) Tiempo empleado por el alumno en entender cuál es el defecto, y (3) Descripción del defecto. Todos los archivos del programa se encuentran en la carpeta: “PersonasQueCompranEnMuchasTiendas”

Nombre y apellidos: Israel Bru Montes

Tiempo:

Descripción del defecto: