

**Grandes Volúmenes de Datos**



**Actividad MapReduce**

12/10/2021

*Marcos Eladio Somoza*

**ÍNDICE**

[1. Enunciado 4](#_Toc85066240)

[2. Parte 1 5](#_Toc85066241)

[3. Parte 2 5](#_Toc85066242)

[4. Parte 3 6](#_Toc85066243)

[5. Parte 4 8](#_Toc85066244)

[6. Parte 5 9](#_Toc85066245)

# Enunciado

Los objetivos de esta actividad son los siguientes:

* Configurar y arrancar un clúster Hadoop en la nube.
* Aprender los comandos para trabajar con el sistema de archivos distribuido HDFS.
* Entender cómo funciona el algoritmo MapReduce para la ejecución de procesos en paralelo.

Para poder realizar de forma correcta esta actividad es necesario contar con los siguientes requisitos:

* Tener una cuenta en Google Cloud Platform.
* Un equipo con al menos 8Gb de RAM.

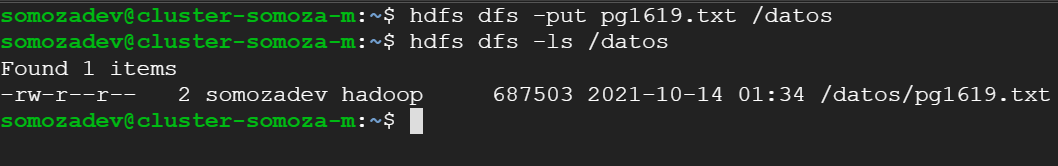
Los enunciados de cada paso son:

* **Paso 1**. Descargar el texto del libro "La Celestina" de la página del proyecto Gutenberg ( http://www.gutenberg.org/ebooks/1619 (Enlaces a un sitio externo.)) y cargarlo en HDFS.
* **Paso 2**. Visualizar los videos del Campus virtual que explican como ejecutar el contador de palabras con MapReduce. Replicar el ejemplo y hacerlo funcionar.
* **Paso 3**. Crear un clúster Dataproc con tres nodos en Google Cloud Platform y ejecutar el contador de palabras. Ver cuánto tiempo tarda. Eliminar el clúster.
* **Paso 4**. Crear un nuevo clúster Dataproc con el doble de nodos y ejecutar el contador de palabras. Ver cuánto tiempo tarda.
* **Paso 5**. Buscar una colección pública de tweets y, a partir del ejemplo del contador de palabras, implementar un programa MapReduce en Java que recibe un fichero de tweets como entrada, y escribe otro fichero con el conteo del número de hashtags (palabras que empiezan por una #) que contienen. Ejecutar el contador de hashtags en el clúster.  
  {\displaystyle \log ,}

# Parte 1

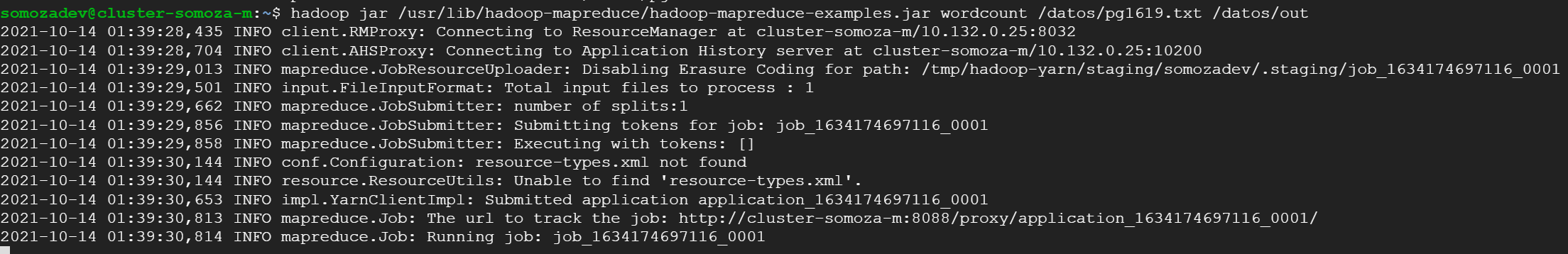
Para descargar el libro deseado en el clúster, usaremos el comando wget. Para poder trabajar con todos los clústers con el libro, se deberá copiar el archivo del libro a HDFS.

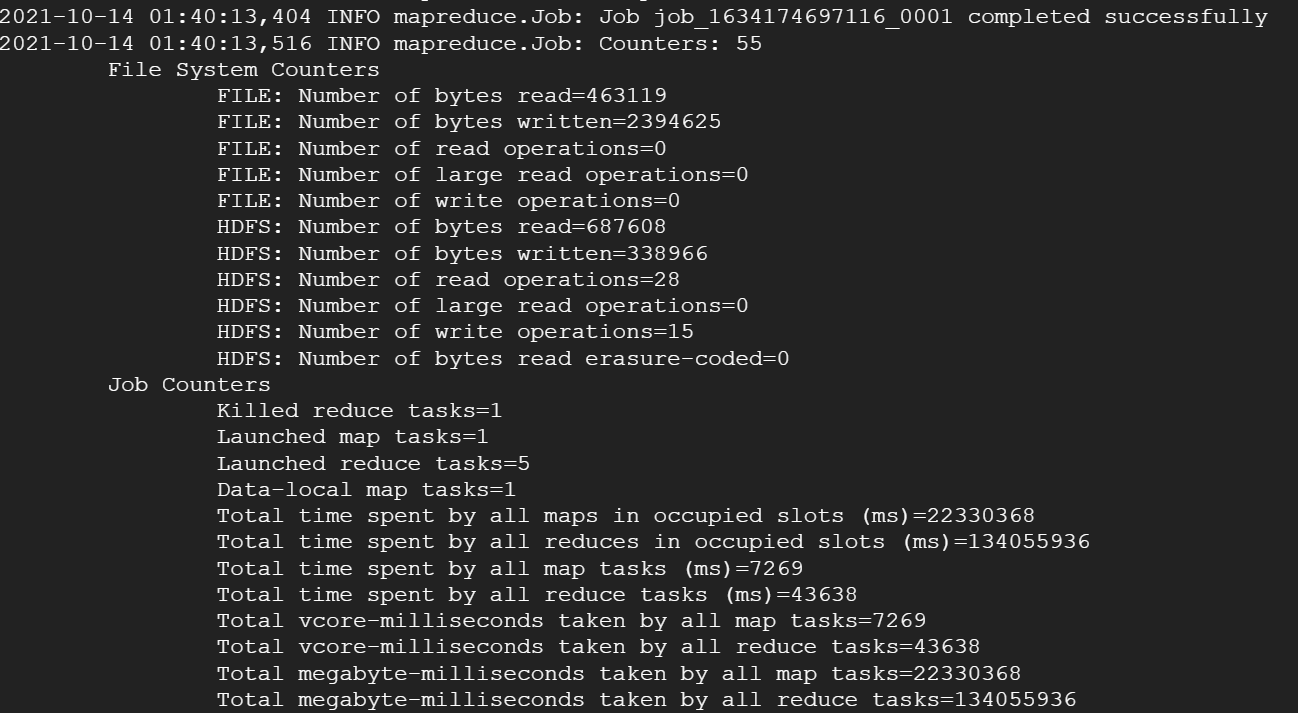
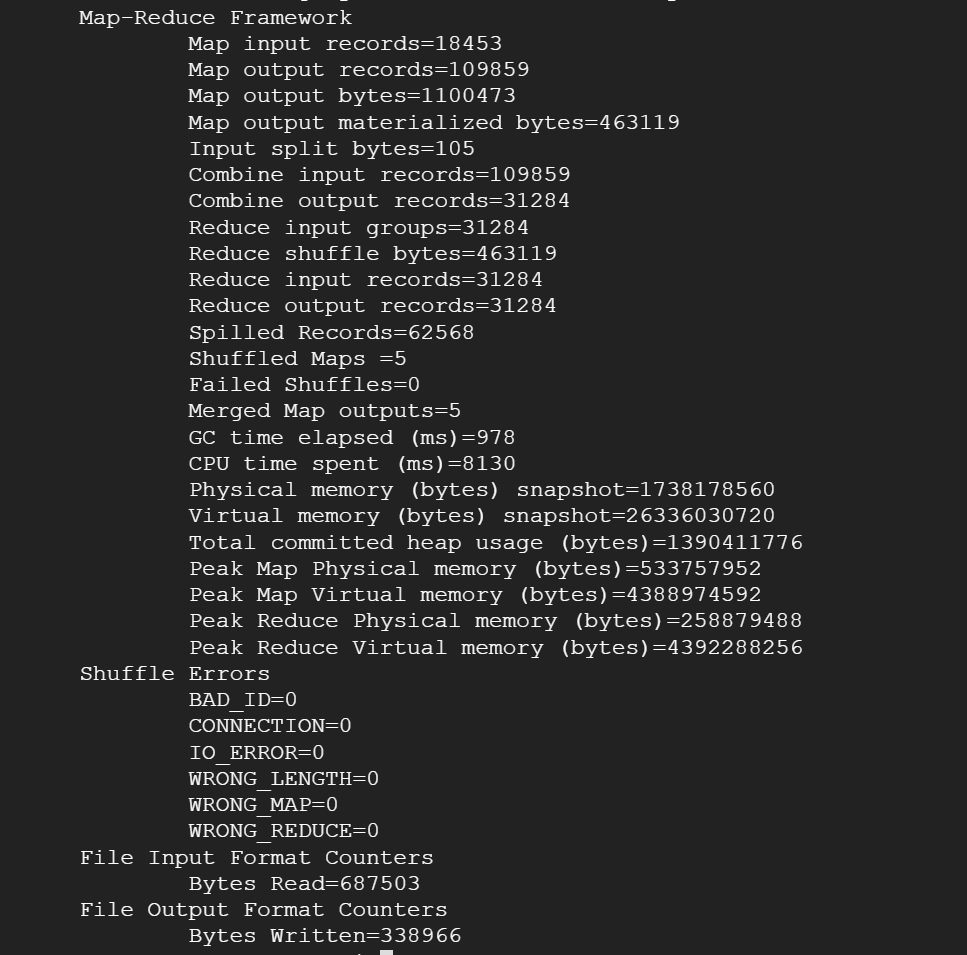


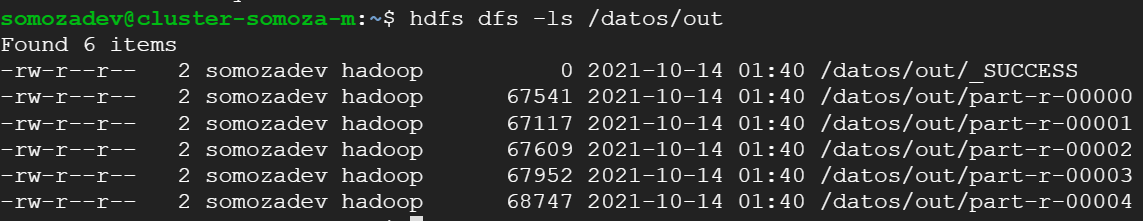
Crearemos una carpeta datos en HDFS mediante el comando hdfs dfs -mkdir /datos. Ahora, copiaremos el libro a dicha carpeta:

# Parte 2

Para aplicar el contador de palabras, se deberá llamar a wordcount disponible en los ejemplos de hadoop. Para ello, se usará el siguiente comando:

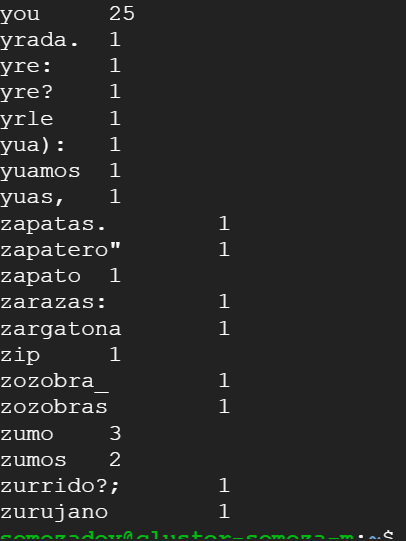
Pasando como primer parámetro el libro (el fichero que se desea contar) y como segundo el directorio de salida. 

Cuando termine, se guardará el resultado en la carpeta indicada /datos/out como se puede ver: 

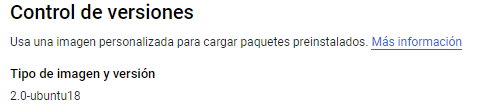
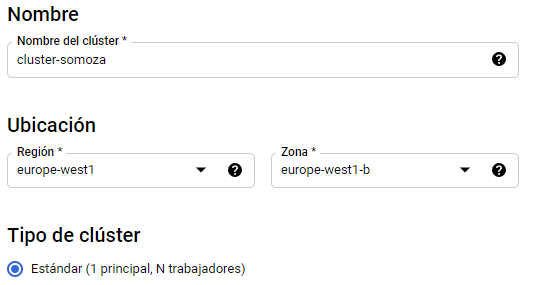
Para comprobar que funcionó correctamente, se pueden visualizan los ficheros con -cat:



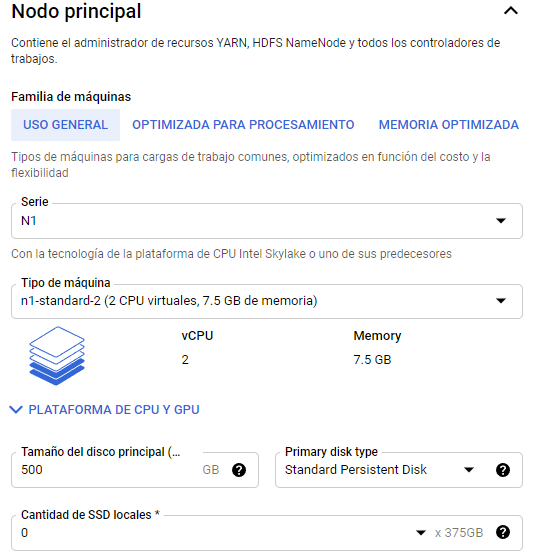
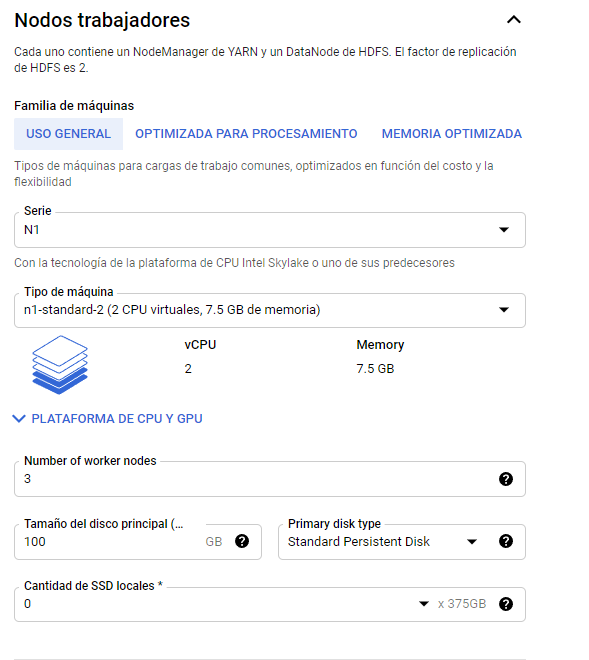


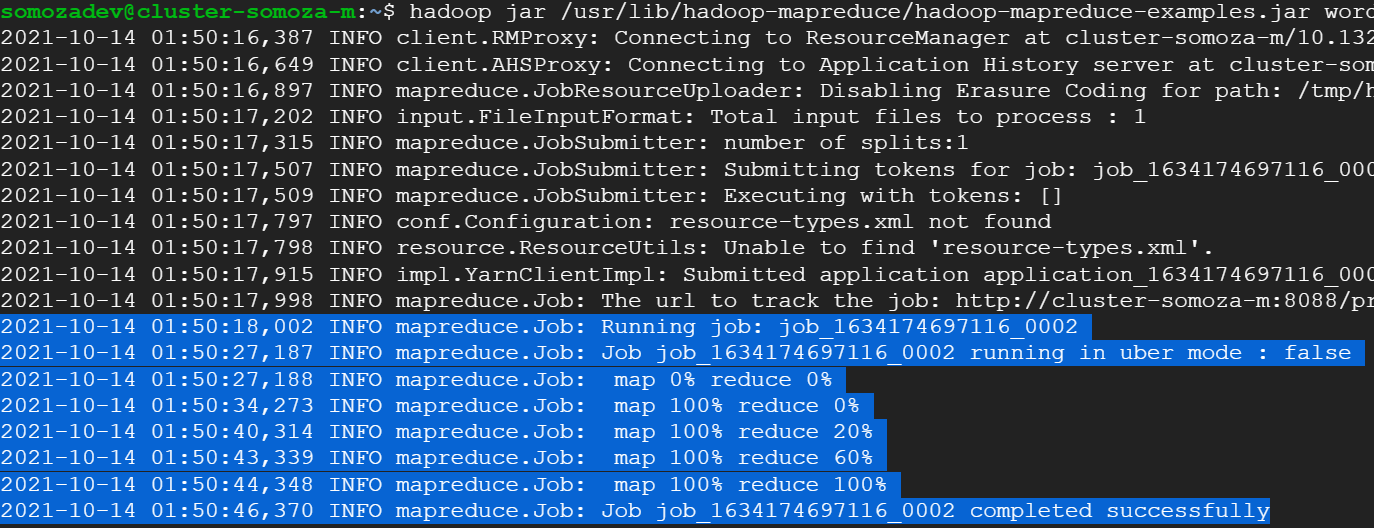
# Parte 3

Para crear el clúster, hay que asignare la región adecuada, en nuestro caso (al ser europeos) europe-west1 y la zona predeterminada vale, europe-west1-b. En cuanto a la imagen del clúster, se utilizará Ubuntu (más concretamente 2.0-ubuntu18), dado que se ha estado utilizando Ubuntu durante la asignatura.



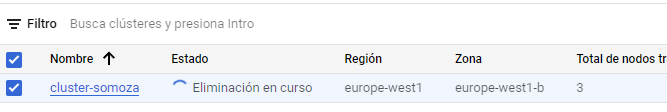
En cuanto a la arquitectura del clúster, se usará un main node con 2CPUs, 7.5Gb de ram y 500Gb de disco. También, tres worker nodes con 2CPUs, 7.5Gb de ram y 200Gb de disco cada uno.



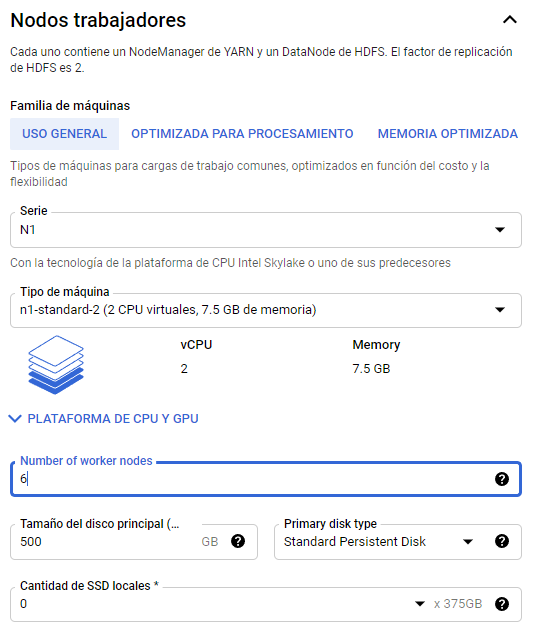
Al usar wordcount con La Celestina con esta configuración de clústers, vemos que ha empezado en el tiempo 01:50:18, y ha terminado en 01:50:46. Esto nos indica que ha tardado ***28 segundos*** en realizar el MapReduce con un master node y tres workers.

Ahora borraremos el clúster.

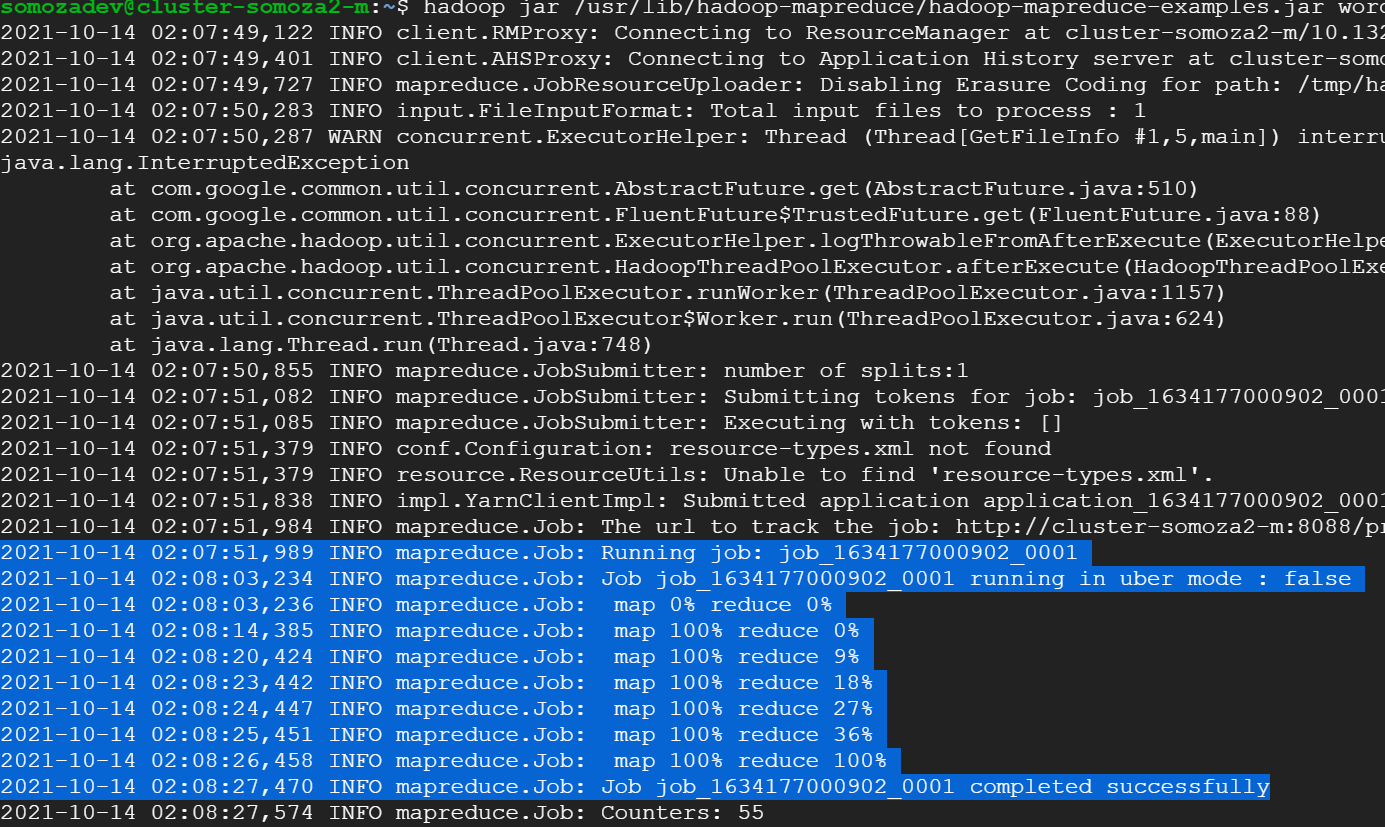


# Parte 4

Se ha creado un clúster igual al anterior pero con el doble de nodos trabajadores ( es decir, con un master node pero seis workers).



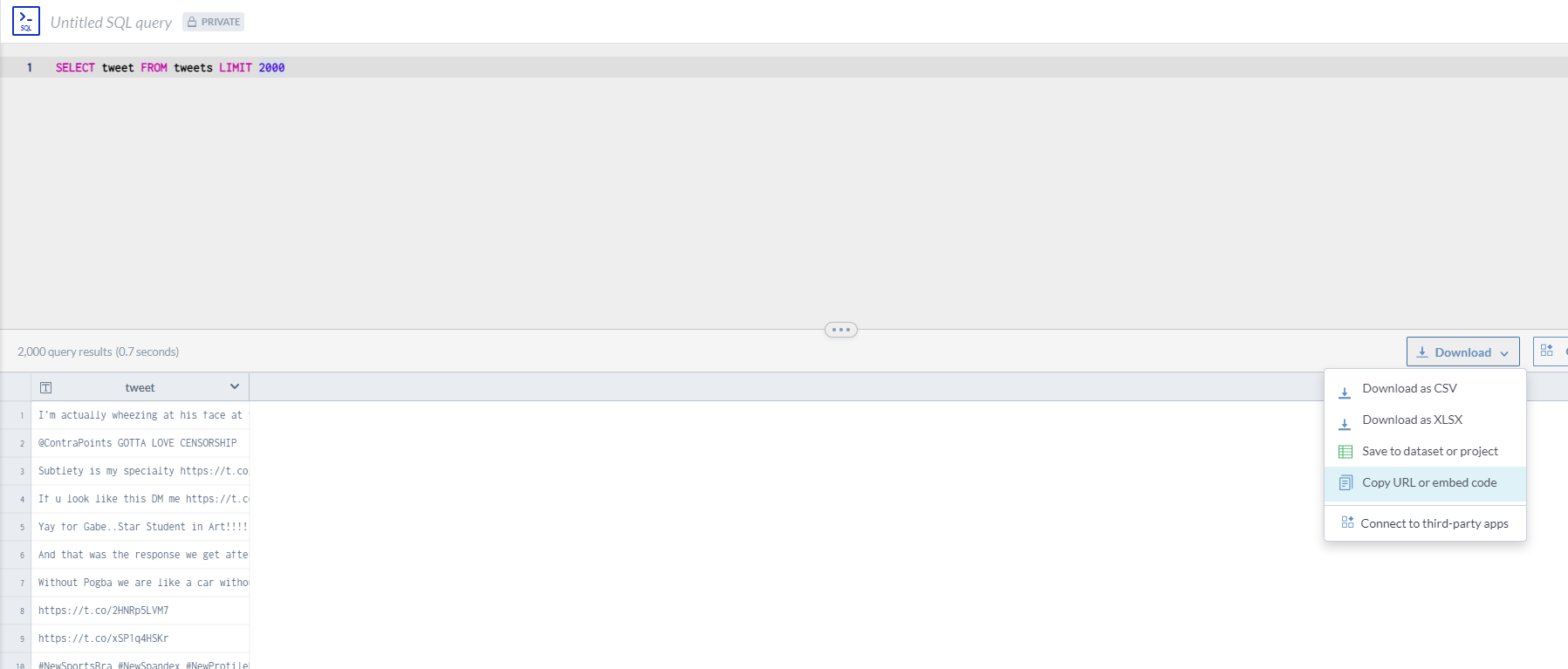
Al ejecutar MapReduce con La Celestina, vemos que :



Ha empezado en el tiempo 02:07:51, y ha terminado en 02:08:27. Esto nos indica que ha tardado ***36 segundos*** en realizar el MapReduce con un master node y seis workers. Ha tardado más que con 3 workers dado que tarda más tiempo en repartir el trabajo entre los workers del que gana trabajando con 6 en lugar de con 3.

# Parte 5

Para conseguir la colección de tweets, se ha utilizado la página <https://data.world/> en donde se ha encontrado un dataset de tweets. Gracias a la query **SELECT tweet FROM tweets LIMIT 2000** se han aislado 2000 tweets de esta y guardado para descargar en <https://download.data.world/s/ntjoxziox6ytgl62odcvdlttxojys2>.



Ya con el input disponible, se descargará al clúster con wget. También, se creará el fichero WordCount.java. Para hacer que cuente el número de “#” habrá que modificar dos líneas de código en el map:

public void map(Object key, Text value, Context context

                    ) throws IOException, InterruptedException {

      StringTokenizer itr = new StringTokenizer(value.toString());

      while (itr.hasMoreTokens()) {

          String str = itr.nextToken();

          if(str.substring(0,1).equals("#")){

            word.set(str);

            context.write(word, one);

          }

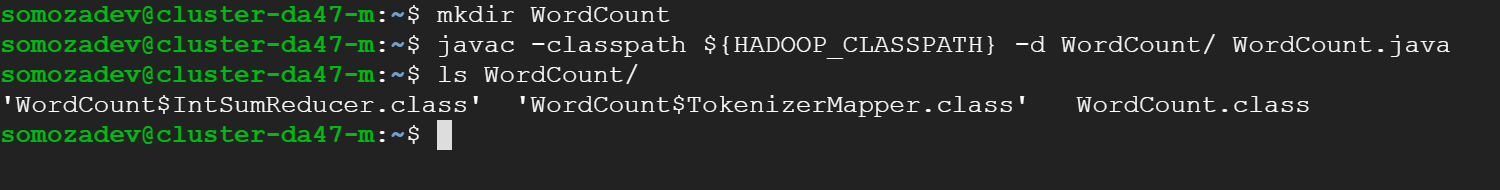
      }

    }

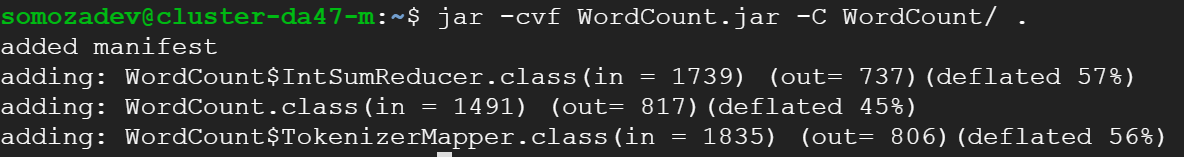
Para poder compilar el WordCount, primero hay que asignar el HADOOP\_CLASPATH con:

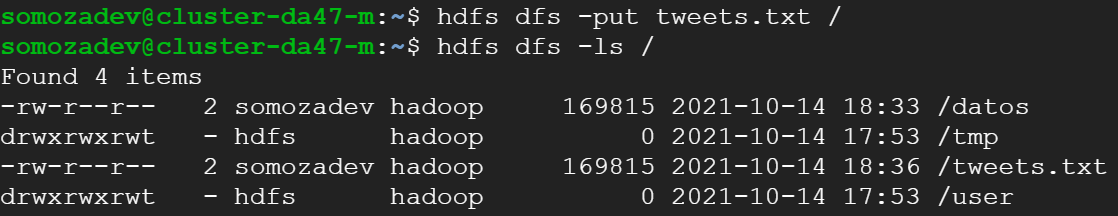


Así, se podrá compilar con el siguiente comando:

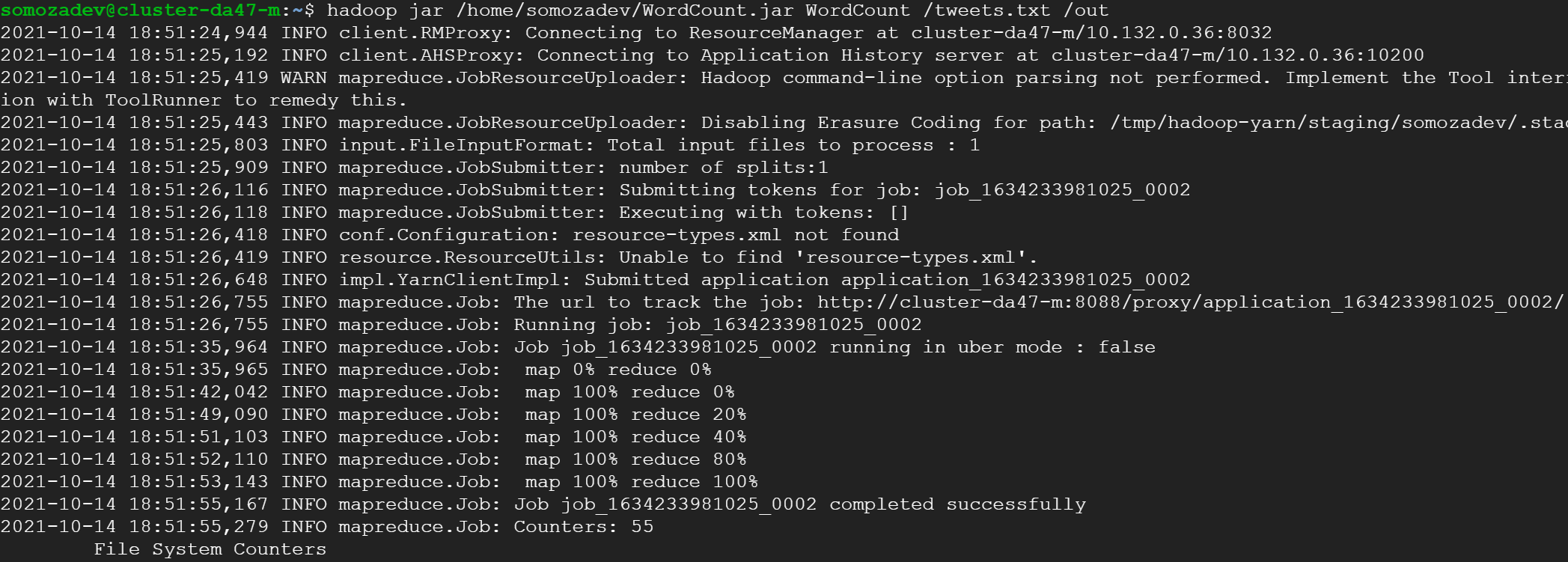


Y finalmente:

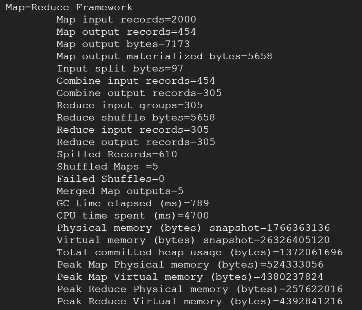


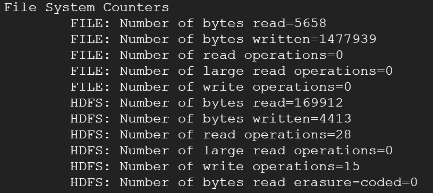
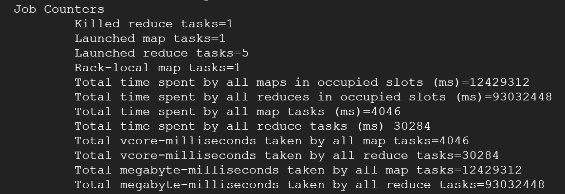
Para poder hacer el WordCount, habrá que copiar el fichero tweets.txt a HDFS 

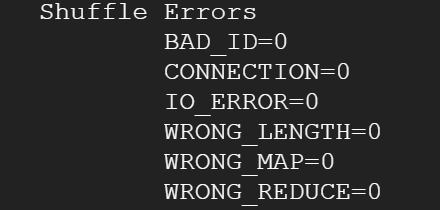
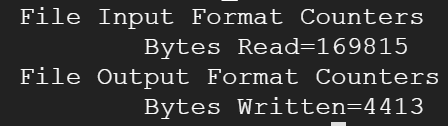
Y finalmente, hacer la llamada al jar mediante hadoop:

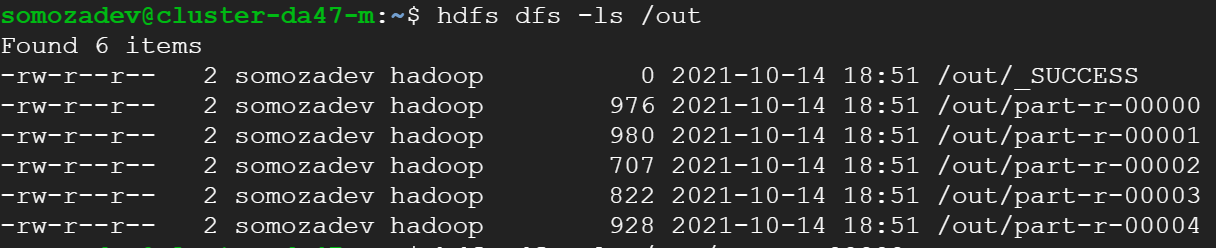


Como se puede ver, el proceso MapReduce empezó a las 18:51:26 y terminó a las 18:51:55, tardando ***29 segundos*** en hacer el conteo.









Como apartado extra, me he tomado la libertad de realizar de nuevo el WordCounter pero asignando como salida un Google Cloud Storage, a fin de poder entregar los ficheros de salida en el zip. Para ello, he cambiado el directorio de salida /out a gs://digdata-bucket-somoza/out 

Además de añadir también el fichero tweets.txt y el WordCounter.java.



