|  |
| --- |
| Máster en Big Data |
| Tecnologías de Almacenamiento |
| 1. Hands-On: Ejecución MapReduce |

Índice

[1. Introducción 3](#_Toc96241410)

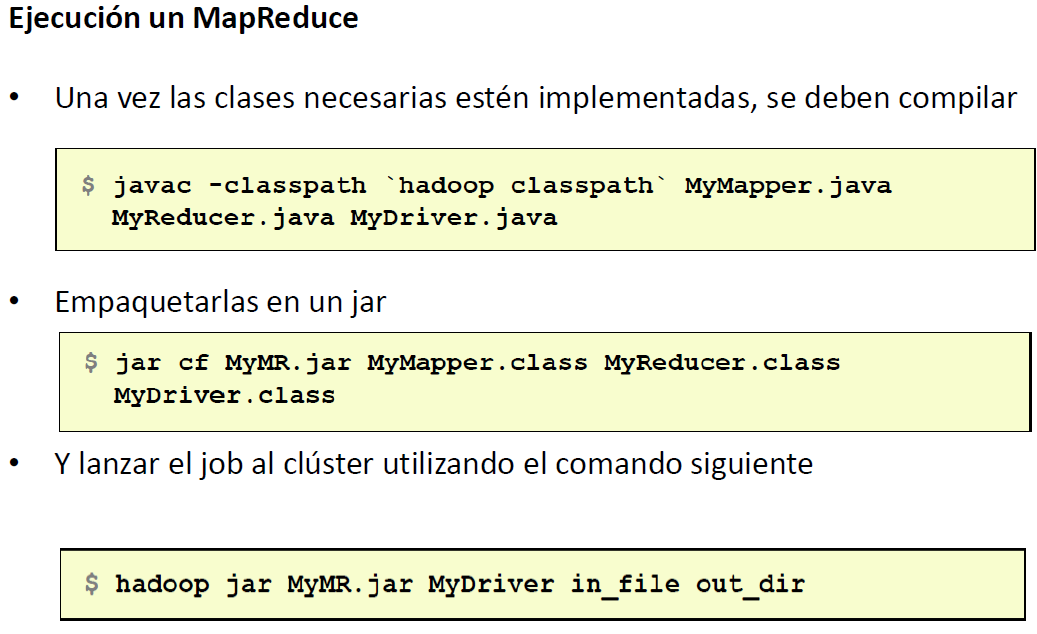
[2. Ejecutando un Job 3](#_Toc96241411)

[3. Finalizando un Job 7](#_Toc96241412)

# Introducción

El objetivo de este Hands-On es aprender a ejecutar jobs de MapReduce por línea de comandos y entender su comportamiento. El entorno de ejecución será la máquina desplegada en el Hands-On anterior.

# Ejecutando un Job



Todas las instrucciones deben introducirse mediante Shell de Linux.

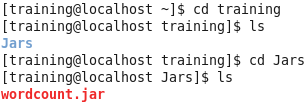
1. Ejecutar el job del proyecto WordCount ya compilado en la carpeta */home/training/training/Jars* (la clase de entrada y su package es *nextret.WordCount*)

• Usamos *cd* para salir de las carpetas. Usamos *ls* para listar lo que tenemos allí. Usmos *cd + nombre carpeta* para entrar en la carpeta.



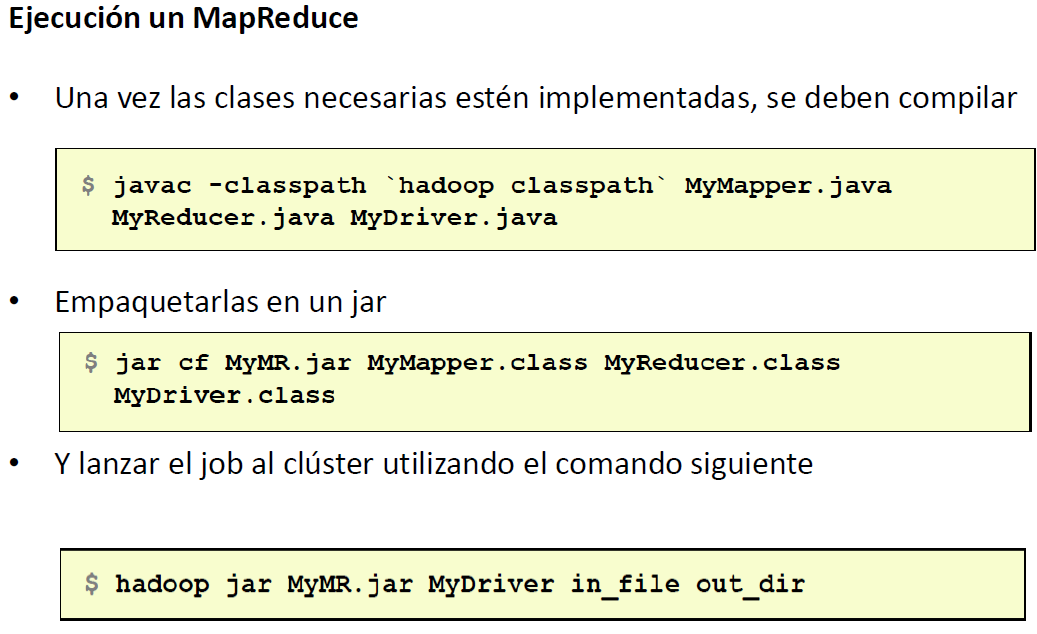






**De este modo en el siguiente paso no hace falta poner la ruta.**

• Ejecutamos el proyecto wordcount.jar. Para hacerlo usamos el siguiente esquema



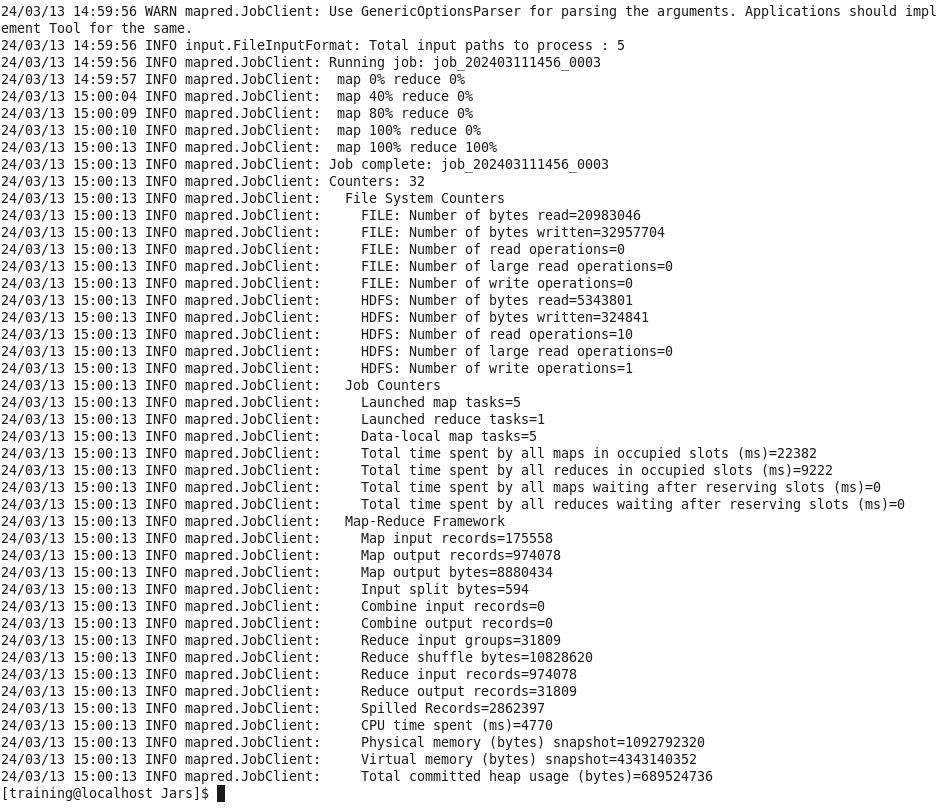
Donde:

* MyMR es el jar con su path. En nuestro caso como ya estamos en el directorio no hace falta poner el path.
* MyDriver es nextret.WordCount
* In\_file son los datos a leer. En este caso leemos el texto de Shakespeare
* Out\_dir es el directorio de salida. Tiene que ser uno que no exista, como por ejemplo una carpeta llamada “Output” dentro de traning.

Vemos:



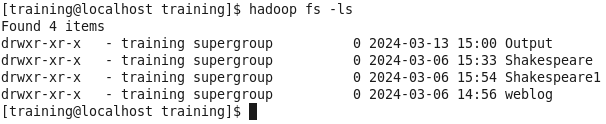
Como respuesta obtenemos:



1. Revisa los archivos generados y visualiza los resultados. Explica los resultados.

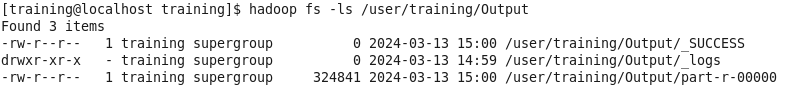
Visitamos la nube para ver si se creó la carpeta *Output*.





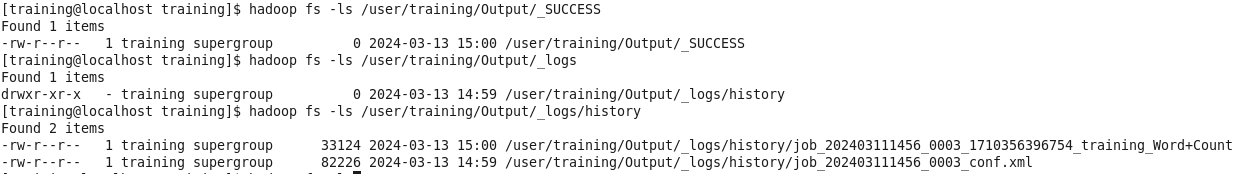
Vemos lo que hay dentro de la carpeta.





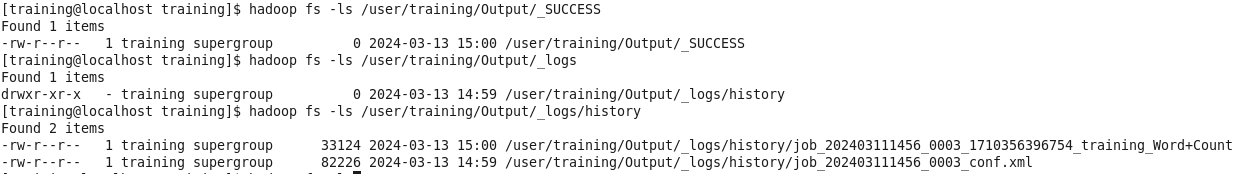
Dentro de la carpeta \_*logs* hay otra carpeta *history*





Dentro de la carpeta *history* hay 2 archivos.

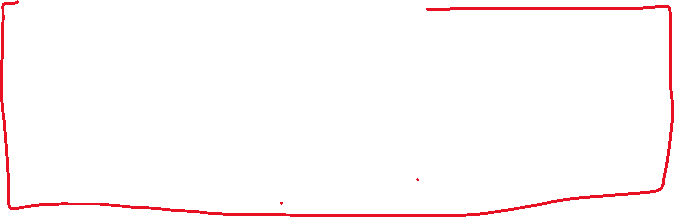




Entonces explicamos esquemáticamente lo que hay en Output

• Archivo \_SUCCESS que no tiene peso. Simplemente es un archivo que indica que se ha realizado correctamente la ejecución.

• Carpeta \_logs.



* Carpeta history
  + Archivo job\_202403111456\_0003\_1710356396754\_training\_Word+Count
  + Archivo job\_202403111456\_0003\_conf.xml

• Archivo part-r-00000.

Leemos lo que hay en el archivo part-r-00000





Y la respuesta es muy larga (1400 páginas de Word) así que reducimos a las últimas palabras.

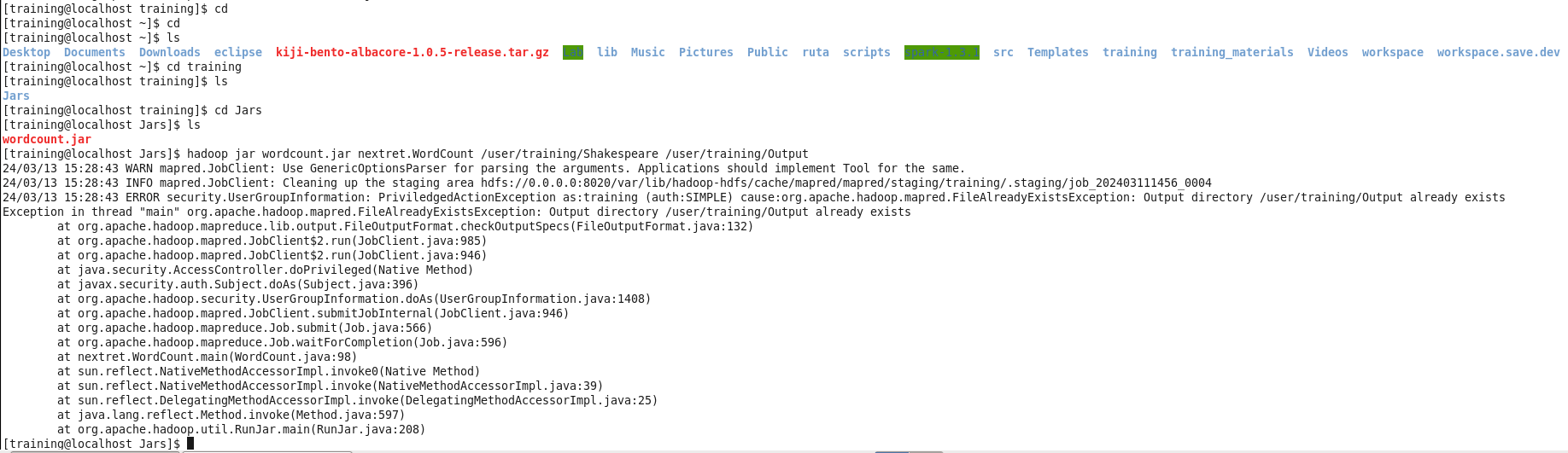




Tenemos el conteo de cada palabra. Key, tab, conteo.

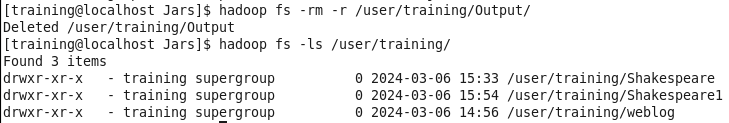
1. Vuelve a ejecutar el mismo comando para ejecutar el Job, ¿qué ha pasado?

Volvemos a hacer los comandos anteriores y la respuesta es que “FileAlreadyEsistsExemption: Output already exists.” Porque ya existe la carpeta “Output” que acabamos de crear.



1. Elimina todos los archivos de esta última ejecución

Eliminamos y vemos que ya no aparece la carpeta “Output” en ese path.



# Finalizando un Job

1. Abre dos terminales de Linux. En uno, vuelve a ejecutar el Job con el volcado del resultado en un nuevo directorio. En el otro, lista los Jobs en ejecución *(mapred job -list)* y mata el Job actual *(mapred job -kill <jobId>)*







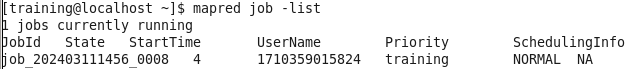


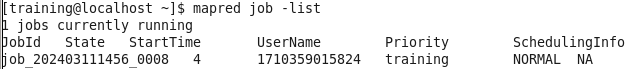
En una consola preparamos el siguiente comando:





En otra consola de preparamos “mapred job”. Primero ejecutamos lo anterior y después este :



Que devuelve 

Y de forma rápida (antes que termine) escribimos:



Donde el número de job es el que ha devuelto anteriormente. Para hacerlo exitosamente nos damos cuenta que solo cambia el valor final de 005 a 006 a 007 para cada jo basí que tenemos preparado previamente el comando y con solo ↑ ya lo tenemos escrito.

La respuesta en la consola donde lo matamos es:



La respuesta donde se está ejecutando el job es:

