Taller 1 - Hadoop

Daniel Alejandro Chimbi León

Edna Valentina Henao Barrera

Facultad de Ingeniería, Universidad El Bosque

Big Data Analytics

Fabian Camilo Peña Lozano

14 de septiembre de 2021

PARTE 1 - INTRODUCCIÓN A HADOOP

En la primera parte del presente taller se hará una introducción a Hadoop, el cual es un framework que es de código libre, básicamente sirve para crear y ejecutar apps en clusters, adicionalmente su funcionalidad más característica que es el almacenamiento de datos en forma masiva, para cualquier tipo de datos.

Tiene un procesamiento demasiado eficiente y puede ejecutar trabajos casi de manera ilimitada. Hoy en día hay millones de dispositivos que generan datos de forma enorme, además de la información tan grande que generan las empresas, Hadoop nace de la necesidad que Google tiene para procesar dichos datos en la red. Hadoop posee las siguientes características que hacen que resalte como framework para la gestión de datos.

Dichas características son:

1. **Bajo costo**: Es un framework de código abierto gratuito que utiliza un hardware básico para dicho almacenamiento de datos o información grande.

2. **Escalabilidad**: Facilidad para la creación de sistemas la gestión de los datos, introduciendolos en nodos, es decir, requiere de poca administración.

3. **Flexibilidad**: Hadoop es muy diferente a las bases de datos tradicionales, ya que no es necesario procesar dicha información con antelación para poder realizar un almacenamiento oportuno. Adicionalmente también incluye el almacenamiento de datos no estructurados tales como imágenes y videos.

4. **Tolerancia a fallos**: La información y la gestión de la misma se encuentran protegidos contras fallos de hardware, es decir, si un nodo se disminuye, los trabajos se redirige de forma automática a otros para cumplir con una computación distribuida, creando varias copias que posteriormente se puedan recuperar en caso de algún fallo.

5. **Capacidad y poder de procesamiento**: La computación distribuida del presente framework, procesa de manera rápida los trabajos de Big Data, y es directamente proporcional al número, ya que, entre mayor cantidad de estos, habrá mayor procesamiento.

6. **Capacidad de almacenar y procesar grandes volúmenes de datos de cualquier tipo de datos**: Cada día el volumen de datos que se generan, especialmente en los medios de comunicación social y el internet de las cosas (IoT), es clave para que Hadoop pueda gestionarlos de manera eficiente.

Con la introducción anteriormente escrita, en la primera parte se hará uso de una máquina virtual junto al sistema operativo Linux con la distribución de Ubuntu para poder instalar el framework Hadoop.

1. Primero se crea el nuevo usuario hdoop.

```
edna@edna-VirtualBox:~$ sudo adduser hdoop
Añadiendo el usuario `hdoop' ...
Añadiendo el nuevo grupo `hdoop' (1002) ...
Añadiendo el nuevo usuario `hdoop' (1002) con grupo `hdoop' ...
Creando el directorio personal `/home/hdoop' ...
Copiando los ficheros desde `/etc/skel' ...
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
Cambiando la información de usuario para hdoop
Introduzca el nuevo valor, o presione INTRO para el predeterminado
        Nombre completo []: hdoop
        Número de habitación []:
        Teléfono del trabajo []:
        Teléfono de casa []:
        Otro []:
¿Es correcta la información? [S/n] S
```

2. Se añade el usuario al grupo sudo.

```
edna@edna-VirtualBox:~$ sudo su
root@edna-VirtualBox:/home/edna# usermod -aG sudo hdoop
```

3. Se verifica que el usuario hdoop pertenezca al grupo sudo.

```
root@edna-VirtualBox:/home/edna# groups hdoop
hdoop : hdoop sudo
```

4. Se verifica el acceso al sudo

```
hdoop@edna-VirtualBox:~$ sudo su - hdoop
[sudo] contraseña para hdoop:
hdoop@edna-VirtualBox:~$
```

5. Se hace una copia de seguridad del archivo /.bachrc con el comando

```
edna@edna-VirtualBox:~$ cp ~/.bashrc ~/.bashrc.bak
```

6. Actualice el sistema antes de iniciar una nueva instalación

```
edna@edna-VirtualBox:~$ sudo apt update
[sudo] contraseña para edna:
Obj:1 http://co.archive.ubuntu.com/ubuntu focal InRelease
Des:2 http://security.ubuntu.com/ubuntu focal-security InR
elease [114 kB]
Des:3 http://co.archive.ubuntu.com/ubuntu focal-updates In
Release [114 kB]
```

7. Instalar OpenJDK 8

```
edna@edna-VirtualBox:~$ sudo apt install openjdk-8-jdk -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
```

8. Se verifica la versión de Java

```
edna@edna-VirtualBox:~$ java -version; javac -version
openjdk version "1.8.0_292"
OpenJDK Runtime Environment (build 1.8.0_292-8u292-b10-0ub
untu1~20.04-b10)
OpenJDK 64-Bit Server VM (build 25.292-b10, mixed mode)
javac 1.8.0_292
```

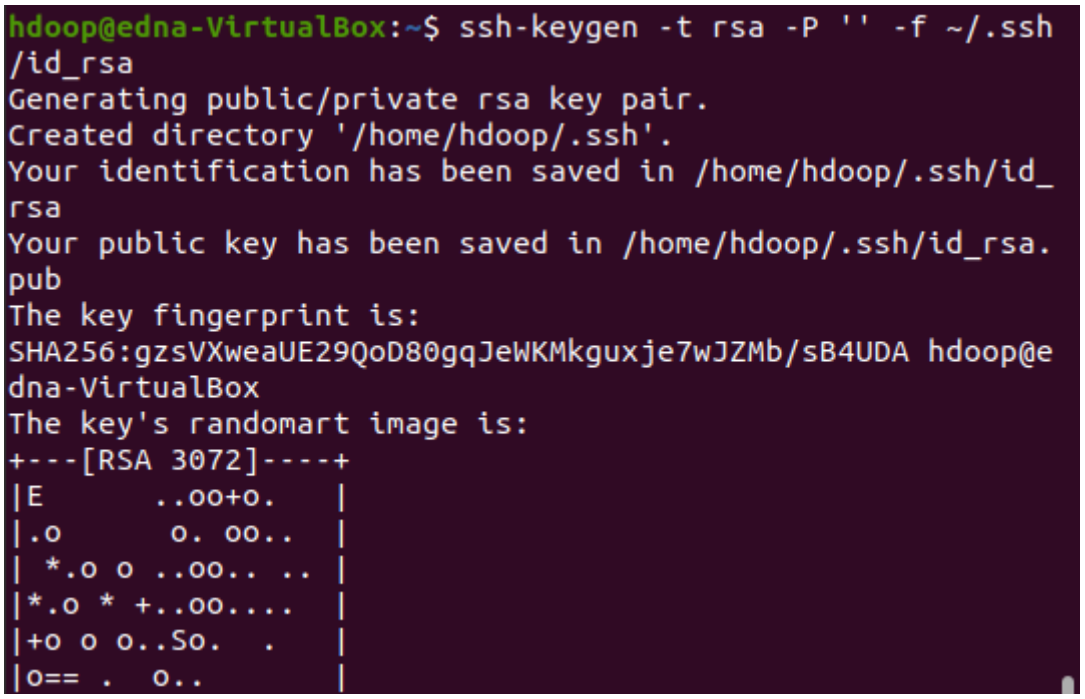
9. Se instala el servidor y el cliente OpenSSH.

```
edna@edna-VirtualBox:~$ sudo apt install openssh-server op
enssh-client -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
openssh-client ya está en su versión más reciente (1:8.2p1
-4ubuntu0.3).
```

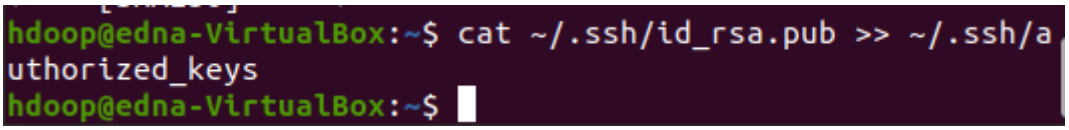
10. Se crea un nuevo usuario de Hadoop y se cambia al usuario recién creado y se ingresa la contraseña correspondiente:

```
edna@edna-VirtualBox:~$ su - hdoop
Contraseña:
hdoop@edna-VirtualBox:~$
```

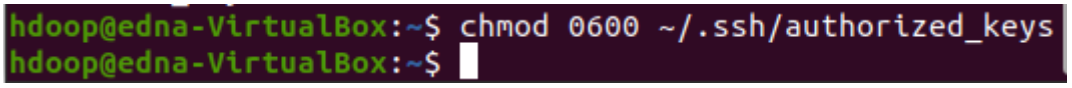
11. Generar un par de claves SSH y definir la ubicación en la que se almacenará



12. Para almacenar la clave pública como authorized_keys en el directorio ssh



13. Se establece los permisos para su usuario con el comando chmod



14. Verifique que todo esté configurado correctamente utilizando el usuario de hdoop para SSH a localhost:

```
hdoop@edna-VirtualBox:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be
established.
ECDSA key fingerprint is SHA256:XKqKTh3ABTcWYrKGkvXh4/HbFT
mVi7PfJSAbpG37jck.
Are you sure you want to continue connecting (yes/no/[fing
erprint])? yes
Warning: Permanently added 'localhost' (ECDSA) to the list
 of known hosts.
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-34-generic
 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
```

15. Use el enlace espejo provisto y descargue el paquete Hadoop con el comando wget

```
hdoop@edna-VirtualBox:~$ wget https://dlcdn.apache.org/had
oop/common/hadoop-3.2.2/hadoop-3.2.2.tar.gz
--2021-09-10 20:28:23--  https://dlcdn.apache.org/hadoop/c
ommon/hadoop-3.2.2/hadoop-3.2.2.tar.gz
Resolviendo dlcdn.apache.org (dlcdn.apache.org)... 151.101
.2.132, 2a04:4e42::644
Conectando con dlcdn.apache.org (dlcdn.apache.org)[151.101
.2.132]:443... conectado.
Petición HTTP enviada, esperando respuesta...
```

16. Extraer los archivos para iniciar la instalación de Hadoop

```
hdoop@edna-VirtualBox:~$ tar xzf hadoop-3.2.2.tar.gz
```

17. Edite el archivo de configuración de shell .bashrc

```
hdoop@edna-VirtualBox:~$ sudo nano .bashrc
[sudo] contraseña para hdoop:
```

```
#Hadoop Related Options
export HADOOP_HOME=/home/hdoop/hadoop-3.2.2
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
```

18. Es vital aplicar los cambios al entorno de ejecución actual mediante el siguiente comando o puede abrir una nueva terminal:

```
hdoop@edna-VirtualBox:~$ source ~/.bashrc
hdoop@edna-VirtualBox:~$
```

19. Al configurar un clúster de Hadoop de un solo nodo, debe definir qué implementación de Java se utilizará. Utilice la variable $ HADOOP_HOME creada anteriormente para acceder al archivo hadoop-env.sh:

```
hdoop@edna-VirtualBox:~$ sudo nano $HADOOP_HOME/etc/hadoo
p/hadoop-env.sh
```

```
# The java implementation to use. By default, this env
# variable is REQUIRED on ALL platforms except OS X!
 export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

20. localice la ruta de Java correcta, use la ruta proporcionada para encontrar el directorio OpenJDK

```
hdoop@edna-VirtualBox:~$ which javac
/usr/bin/javac
hdoop@edna-VirtualBox:~$ readlink -f /usr/bin/javac
/usr/lib/jvm/java-8-openjdk-amd64/bin/javac
hdoop@edna-VirtualBox:~$
```

21. Abra el archivo core-site.xml

```
hdoop@edna-VirtualBox:~$ sudo nano $HADOOP_HOME/etc/hadoo
p/core-site.xml
```

```
<configuration>
<property>
  <name>hadoop.tmp.dir</name>
  <value>/home/hdoop/tmpdata</value>
</property>
<property>
  <name>fs.default.name</name>
  <value>hdfs://127.0.0.1:9000</value>
</property>
</configuration>
```

22. Abre el archivo hdfs-site.xml

```
hdoop@edna-VirtualBox:~$ sudo nano $HADOOP_HOME/etc/hadoo
p/hdfs-site.xml
```

```
<configuration>
<property>
<name>dfs.data.dir</name>
  <value>/home/hdoop/dfsdata/namenode</value>
</property>
<property>
  <name>dfs.data.dir</name>
  <value>/home/hdoop/dfsdata/datanode</value>
</property>
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>
</configuration>
```

23. Usando el comando mkdir para hacer estos directorios

```
hdoop@edna-VirtualBox:~$ mkdir /home/hdoop/dfsdata
hdoop@edna-VirtualBox:~$ mkdir /home/hdoop/dfsdata/nameno
de
```

```
hdoop@edna-VirtualBox:~$ mkdir /home/hdoop/dfsdata/datano
de
```

24. Acceder al archivo mapred-site.xml y definir los valores de MapReduce

```
hdoop@edna-VirtualBox:~$ sudo nano $HADOOP_HOME/etc/hadoo
p/mapred-site.xml
```

```
<configuration>
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>
</configuration>
```

25. Abra el archivo yarn-site.xml

```
hdoop@edna-VirtualBox:~$ sudo nano $HADOOP_HOME/etc/hadoo
p/yarn-site.xml
```

```
<configuration>
<property>
  <name>yarn.nodemanager.aux-services</name>
  <value>mapreduce_shuffle</value>
</property>
<property>
  <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
  <value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
<property>
  <name>yarn.resourcemanager.hostname</name>
  <value>127.0.0.1</value>
</property>
<property>
  <name>yarn.acl.enable</name>
  <value>0</value>
</property>
<property>
  <name>yarn.nodemanager.env-whitelist</name>
```

26. Formatee el NameNode antes de iniciar los servicios de Hadoop por primera vez

```
hdoop@edna-VirtualBox:~$ hdfs namenode -format
2021-09-10 21:55:08,685 INFO namenode.NameNode: STARTUP_
MSG:
/************************************************************
*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = edna-VirtualBox/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 3.2.2
STARTUP_MSG:   classpath = /home/hdoop/hadoop-3.2.2/etc/
```

```
SHUTDOWN_MSG: Shutting down NameNode at edna-VirtualBox/
127.0.1.1
************************************************************
****/
```

27. Se navega hasta la siguiente carpeta, que tiene como directorio hadoop-3.2.2 / sbin y se ejecuta los siguientes comandos para iniciar NameNode y DataNode:

```
hdoop@edna-VirtualBox:~/hadoop-3.2.2/sbin$ ./start-dfs.s
h
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [edna-VirtualBox]
edna-VirtualBox: Warning: Permanently added 'edna-virtua
lbox' (ECDSA) to the list of known hosts.
```

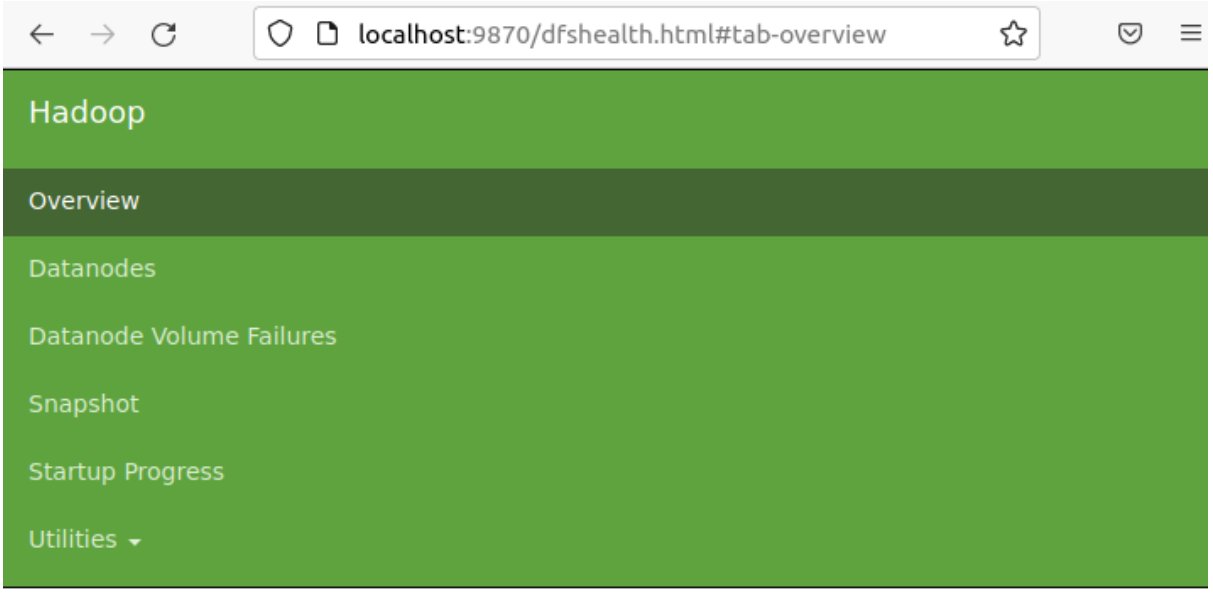
28. Se inicia archivo yarn.sh y los administradores de nodos

```
hdoop@edna-VirtualBox:~/hadoop-3.2.2/sbin$  ./start-yarn
.sh
Starting resourcemanager
Starting nodemanagers
```

29. Para comprobar si todos los programas y se ejecuta como procesos Java

```
hdoop@edna-VirtualBox:~/hadoop-3.2.2/sbin$ jps
19828 NameNode
20825 Jps
19946 DataNode
20491 NodeManager
20366 ResourceManager
20110 SecondaryNameNode
```

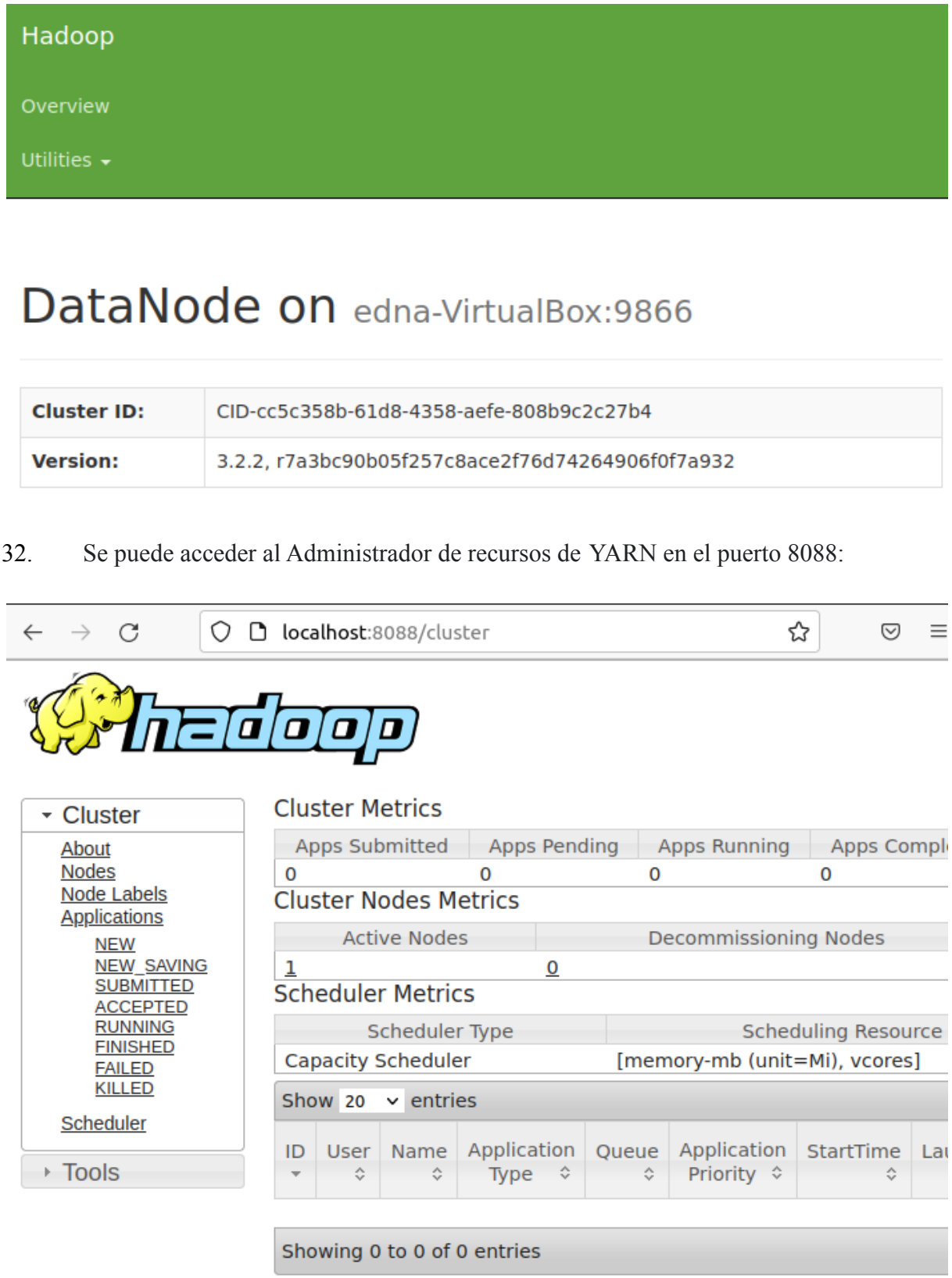
30. Utilizar el navegador y navegar hasta la URL o IP del host local.
    a. El número de puerto predeterminado 9870 le da acceso a la IU de NameNode de Hadoop:



Overview 'localhost:9000' (active)

Started:	Fri Sep 10 22:04:10 -0500 2021

31. El puerto predeterminado 9864 y se utiliza para acceder a DataNodes individuales directamente desde el navegador:



32. Se puede acceder al Administrador de recursos de YARN en el puerto 8088:

Finalmente se puede evidenciar que el proceso de hadoop se está ejecutando en el cluster, como se muestra en la url del navegador.

PARTE 2 - COMPONENTE MAPREDUCE

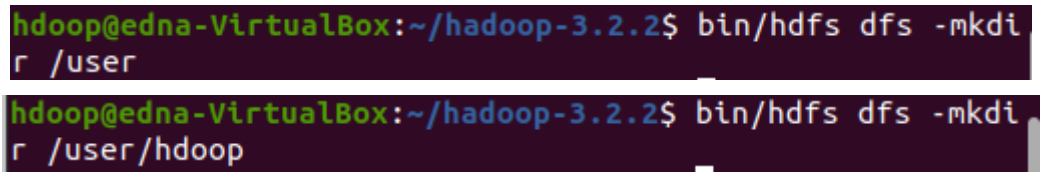
En la segunda parte se estudiará el componente MapReduce tiene la posibilidad de trabajar con varios archivos, ya que los nodos no se interrumpen los proceso a no ser que hagan análisis de clasificación y mezcla.

Los algoritmos iterativos pueden requerir distintas fases de mapeo, mezcla, clasificación y reducción, es decir, que da origen a los múltiples archivos entre las fases del MapReduce. En MapReduce las subtareas se asocian a una tarea que se ejecuta de manera distribuida en los diferentes nodos de procesamiento.

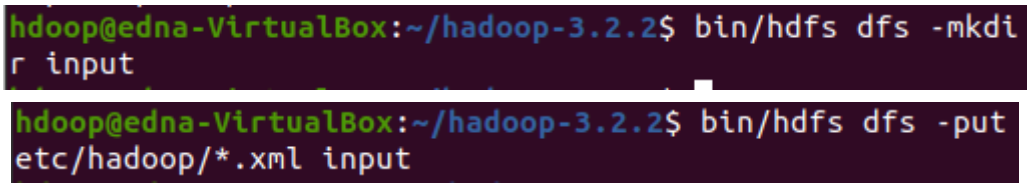
Se controla y gestiona su ejecución mediante proceso Master o Job, el cual es el encargado de aceptar los nuevos trabajos que son enviados al sistema por los clientes.

Luego de explicar el concepto, se procede a crear los siguientes comandos:

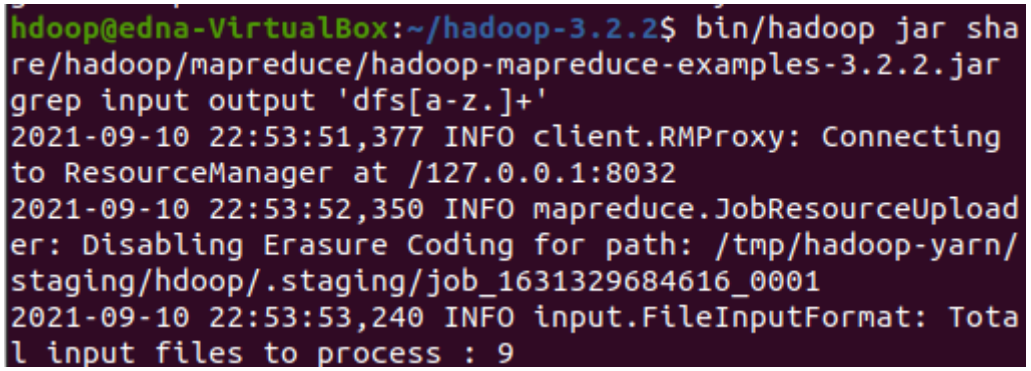
1. Hacer los directorios que hacen referencia a HDFS necesarios para ejecutar los trabajos MapReduce.



2. Copiar los archivos de entrada en el sistema de archivos distribuido.



3. Ejecutar alguno de los ejemplos proporcionados.



```
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=196
        File Output Format Counters
                Bytes Written=54
hdoop@edna-VirtualBox:~/hadoop-3.2.2$
```

4. Examinar los archivos de salida: Se copian los archivos de salida del sistema de archivos distribuido al sistema de archivos local y examínelos

```
hdoop@edna-VirtualBox:~/hadoop-3.2.2$ bin/hdfs dfs -get
output output
hdoop@edna-VirtualBox:~/hadoop-3.2.2$
```

```
hdoop@edna-VirtualBox:~/hadoop-3.2.2$ cat output/*
2       dfsdata
2       dfs.data.dir
1       dfsadmin
1       dfs.replication
```

```
hdoop@edna-VirtualBox:~/hadoop-3.2.2$ bin/hdfs dfs -cat
output/*
2       dfsdata
2       dfs.data.dir
1       dfsadmin
1       dfs.replication
```

¿Qué resultados generó el programa y cuales son los pasos MapReduce que implementa?

En el ejemplo se puede evidenciar que el comando grep espera como argumento una expresión regular, en este caso los resultados que arroja son las palabras que empiezan con las letras “dfs”, se puede evidenciar con los dos comandos para ver la carpeta output que los resultados son los mismos.

5. Posteriormente se realiza el mismo procedimiento con el programa de Word Count

```
hdoop@edna-VirtualBox:~/hadoop-3.2.2$ bin/hadoop jar sha
re/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.2.jar
wordcount input output 'dfs[a-z.]+'
2021-09-11 00:32:05,369 INFO client.RMProxy: Connecting
to ResourceManager at /127.0.0.1:8032
2021-09-11 00:32:06,052 INFO mapreduce.JobResourceUpload
er: Disabling Erasure Coding for path: /tmp/hadoop-yarn/
staging/hdoop/.staging/job_1631329684616_0003
2021-09-11 00:32:06,689 INFO input.FileInputFormat: Tota
l input files to process : 10
2021-09-11 00:32:06,884 INFO mapreduce.JobSubmitter: num
ber of splits:10
```

¿Qué resultados generó el programa y cuales son los pasos MapReduce que implementa?

En el WordCount se puede evidenciar que el comando grep espera como argumento una expresión regular, en este caso los resultados que arroja el conteo todas las palabras que contiene el poema,

PARTE 3 - INTRODUCCIÓN A SPARK

1. Se instala el JDK con los comandos de la parte 1 del taller:

```
danielchimbi@danielchimbi-VirtualBox:~$ sudo apt update
[sudo] contraseña para danielchimbi:
Obj:1 http://co.archive.ubuntu.com/ubuntu focal InRelease
Des:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Des:3 http://co.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Des:4 http://co.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Des:5 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [866
 kB]
```

```
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se pueden actualizar 48 paquetes. Ejecute «apt list --upgr
adable» para verlos.
danielchimbi@danielchimbi-VirtualBox:~$
```

```
danielchimbi@danielchimbi-VirtualBox:~$ sudo apt install o
penjdk-8-jdk -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
openjdk-8-jdk ya está en su versión más reciente (8u292-b1
0-0ubuntu1~20.04).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y
48 no actualizados.
```

```
danielchimbi@danielchimbi-VirtualBox:~$ java -version; jav
ac -version
openjdk version "1.8.0_292"
OpenJDK Runtime Environment (build 1.8.0_292-8u292-b10-0ub
untu1~20.04-b10)
OpenJDK 64-Bit Server VM (build 25.292-b10, mixed mode)
javac 1.8.0_292
```

2. Instalar las dependencias necesarias, las cuales son Git y Scala

```
danielchimbi@danielchimbi-VirtualBox:~$ sudo apt install s
cala git -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
```

```
Configurando scala (2.11.12-4) ...
update-alternatives: utilizando /usr/share/scala-2.11/bin/
scala para proveer /usr/bin/scala (scala) en modo automáti
co
Procesando disparadores para man-db (2.9.1-1) ...
danielchimbi@danielchimbi-VirtualBox:~$
```

3. Se verifican las dependencias instaladas

```
danielchimbi@danielchimbi-VirtualBox:~$ java -version; jav
ac -version; scala -version; git --version
openjdk version "1.8.0_292"
OpenJDK Runtime Environment (build 1.8.0_292-8u292-b10-0ub
untu1~20.04-b10)
OpenJDK 64-Bit Server VM (build 25.292-b10, mixed mode)
javac 1.8.0_292
Scala code runner version 2.11.12 -- Copyright 2002-2017,
LAMP/EPFL
git version 2.25.1
```

4. Se realiza el comando con el enlace directo para descargar el archivo Spark

```
danielchimbi@danielchimbi-VirtualBox:~$ wget https://dlcdn
.apache.org/spark/spark-3.1.2/spark-3.1.2-bin-hadoop3.2.tg
z
--2021-09-15 16:29:21--  https://dlcdn.apache.org/spark/sp
ark-3.1.2/spark-3.1.2-bin-hadoop3.2.tgz
Resolviendo dlcdn.apache.org (dlcdn.apache.org)... 151.101
.2.132, 2a04:4e42::644
Conectando con dlcdn.apache.org (dlcdn.apache.org)[151.101
.2.132]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 228834641 (218M) [application/x-gzip]
Guardando como: “spark-3.1.2-bin-hadoop3.2.tgz.1”

spark-3.1.2-bi 100% 218,23M   612KB/s    en 8m 38s

2021-09-15 16:37:59 (432 KB/s) - “spark-3.1.2-bin-hadoop3.
2.tgz.1” guardado [228834641/228834641]
```

5. Se extrae el archivo mediante el comando tar.

```
danielchimbi@danielchimbi-VirtualBox:~$ tar xvf spark-3.1.2-bin-hadoop3.
2.tgz.1
spark-3.1.2-bin-hadoop3.2/
spark-3.1.2-bin-hadoop3.2/R/
spark-3.1.2-bin-hadoop3.2/R/lib/
spark-3.1.2-bin-hadoop3.2/R/lib/sparkr.zip
```

```
spark-3.1.2-bin-hadoop3.2/jars/RoaringBitmap-0.9.0.jar
spark-3.1.2-bin-hadoop3.2/jars/JTransforms-3.1.jar
spark-3.1.2-bin-hadoop3.2/jars/JLargeArrays-1.5.jar
spark-3.1.2-bin-hadoop3.2/jars/HikariCP-2.5.1.jar
spark-3.1.2-bin-hadoop3.2/RELEASE
danielchimbi@danielchimbi-VirtualBox:~$
```

6. Se mueve el directorio en el directorio spark-3.1.2-bin-hadoop3.2 to the opt/spark.

```
danielchimbi@danielchimbi-VirtualBox:~$ sudo mv spark-3.1.2-bin-hadoop3.
2 /opt/spark
[sudo] contraseña para danielchimbi:
```

7. Se configuran la variables de entorno y dichas configuraciones se añaden a las rutas de exportación

```
danielchimbi@danielchimbi-VirtualBox:~$ python3 --version
Python 3.8.10
danielchimbi@danielchimbi-VirtualBox:~$
```

```
danielchimbi@danielchimbi-VirtualBox:~$ nano .profile
```

```
fi
export SPARK_HOME=/opt/spark

export PATH=$PATH:$SPARK_HOME/bin:$SPARK_HOME/sbin

export PYSPARK_PYTHON=/usr/bin/python3
```

```
danielchimbi@danielchimbi-VirtualBox:~$ whereis python3
python3: /usr/bin/python3 /usr/bin/python3.8 /usr/lib/python3 /usr/lib/p
ython3.9 /usr/lib/python3.8 /etc/python3 /etc/python3.8 /usr/local/lib/p
ython3.8 /usr/include/python3.8 /usr/share/python3 /usr/share/man/man1/p
ython3.1.gz
```

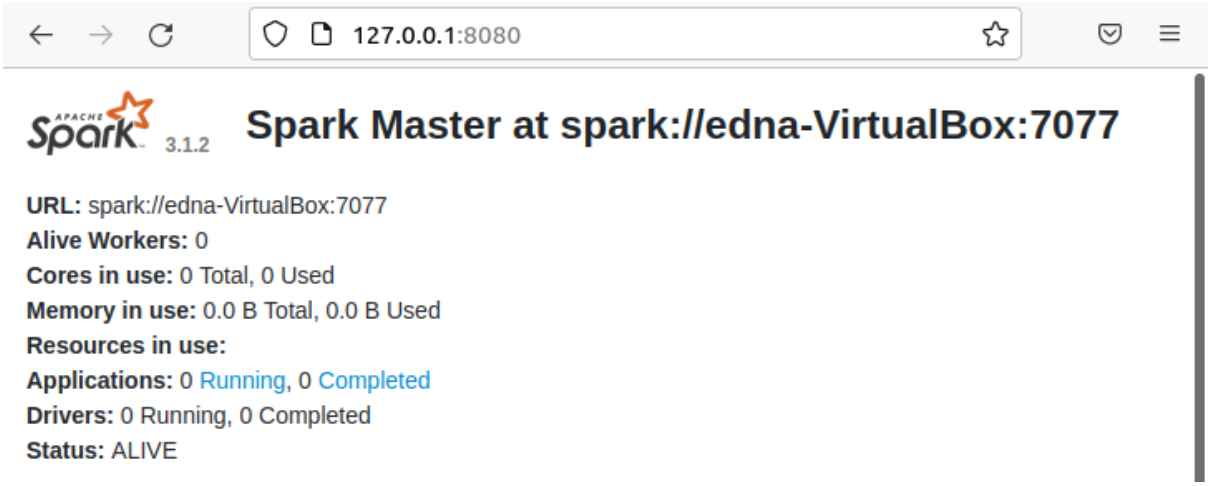
8. Se añaden las rutas y se carga el .profile file

```
danielchimbi@danielchimbi-VirtualBox:~$ source ~/.profile
```

9. Se inicia el servidor.

```
edna@edna-VirtualBox:~$ start-master.sh
starting org.apache.spark.deploy.master.Master, logging to /opt/spark/logs/spar
k-edna-org.apache.spark.deploy.master.Master-1-edna-VirtualBox.out
```

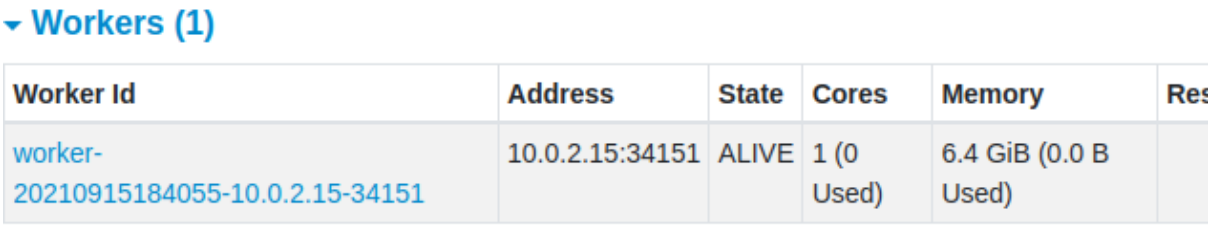
10. Y posteriormente se aparece la interfaz de Spark



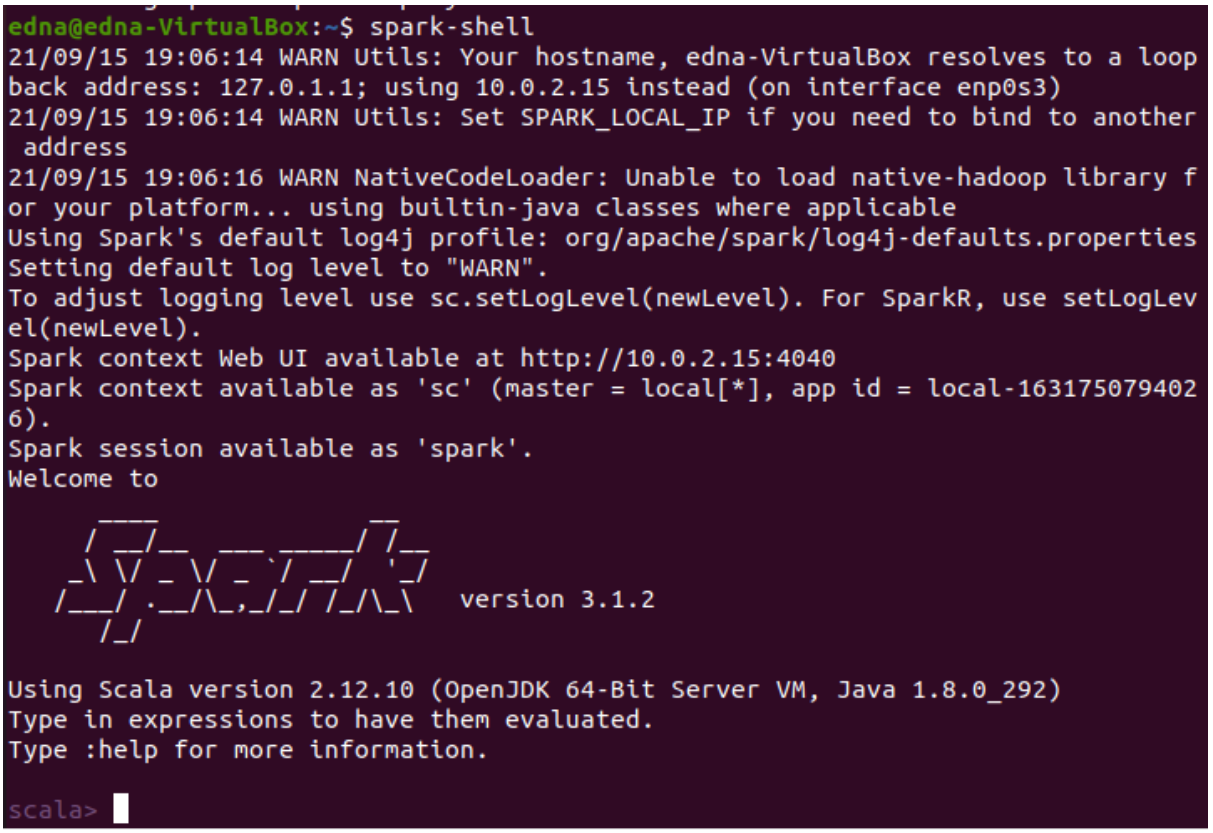
11. Se inicia un servidor esclavo, junto al servidor maestro.

```
edna@edna-VirtualBox:~$ start-slave.sh spark://edna-VirtualBox:7077
This script is deprecated, use start-worker.sh
starting org.apache.spark.deploy.worker.Worker, logging to /opt/spark/logs/spar
k-edna-org.apache.spark.deploy.worker.Worker-1-edna-VirtualBox.out
```

12. Se recarga la vista de Spark y se muestra la lista.



13. Se inicia el servidor tanto esclavo como maestro, y se comprueba si el shell se Spark funciona correctamente.



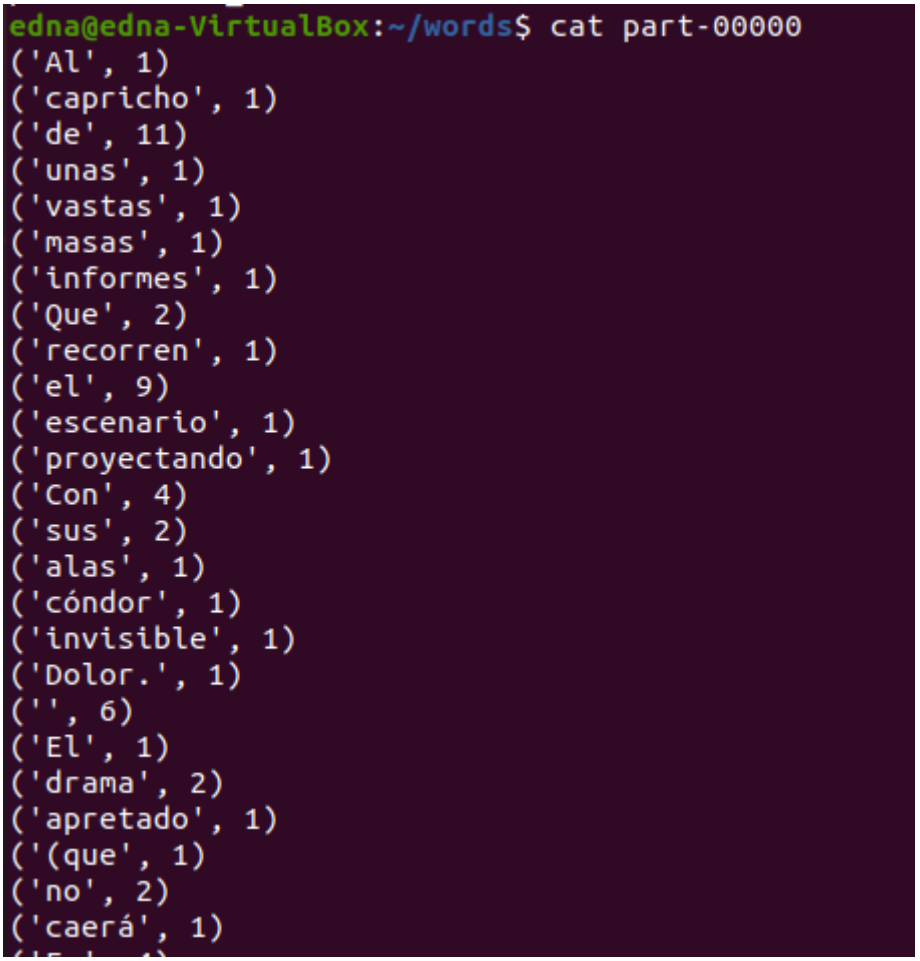
14. Cambiamos a Python.

```
scala> :q
edna@edna-VirtualBox:~$ pyspark
Python 3.8.10 (default, Jun  2 2021, 10:49:15)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
21/09/15 19:09:13 WARN Utils: Your hostname, edna-VirtualBox resolves to a loop
back address: 127.0.1.1; using 10.0.2.15 instead (on interface enp0s3)
21/09/15 19:09:13 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another
 address
21/09/15 19:09:15 WARN NativeCodeLoader: Unable to load native-hadoop library f
or your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLev
el(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.1.2
      /_/

Using Python version 3.8.10 (default, Jun  2 2021 10:49:15)
Spark context Web UI available at http://10.0.2.15:4040
Spark context available as 'sc' (master = local[*], app id = local-163175095920
8).
SparkSession available as 'spark'.
>>>
```

15. Utilizamos algunas transformaciones para construir un conjunto de datos de pares (String, Int) llamado counts y luego lo guardamos en un archivo.

```
>>> text_file = sc.textFile("poem.txt")
>>> counts = text_file.flatMap(lambda line: line.split(" ")).map(lambda word:(w
ord, 1)).reduceByKey(lambda a, b: a + b)
>>> counts.saveAsTextFile("words")
[Stage 0:>                                                          (0 + 1) / 1
[Stage 1:>                                                          (0 + 1) / 1

>>>
```

```
edna@edna-VirtualBox:~$ ls
Descargas   Imágenes    poem.txt                        Vídeos
Documentos  Música      Público                         words
Escritorio  Plantillas  spark-3.1.2-bin-hadoop3.2.tgz
edna@edna-VirtualBox:~$ cd words/
edna@edna-VirtualBox:~/words$ ls
part-00000  _SUCCESS
edna@edna-VirtualBox:~/words$
```



PARTE 4 - INTRODUCCIÓN A JUPYTER

1. Se clona el repositorio en la terminal.

2. Descargamos Anaconda.

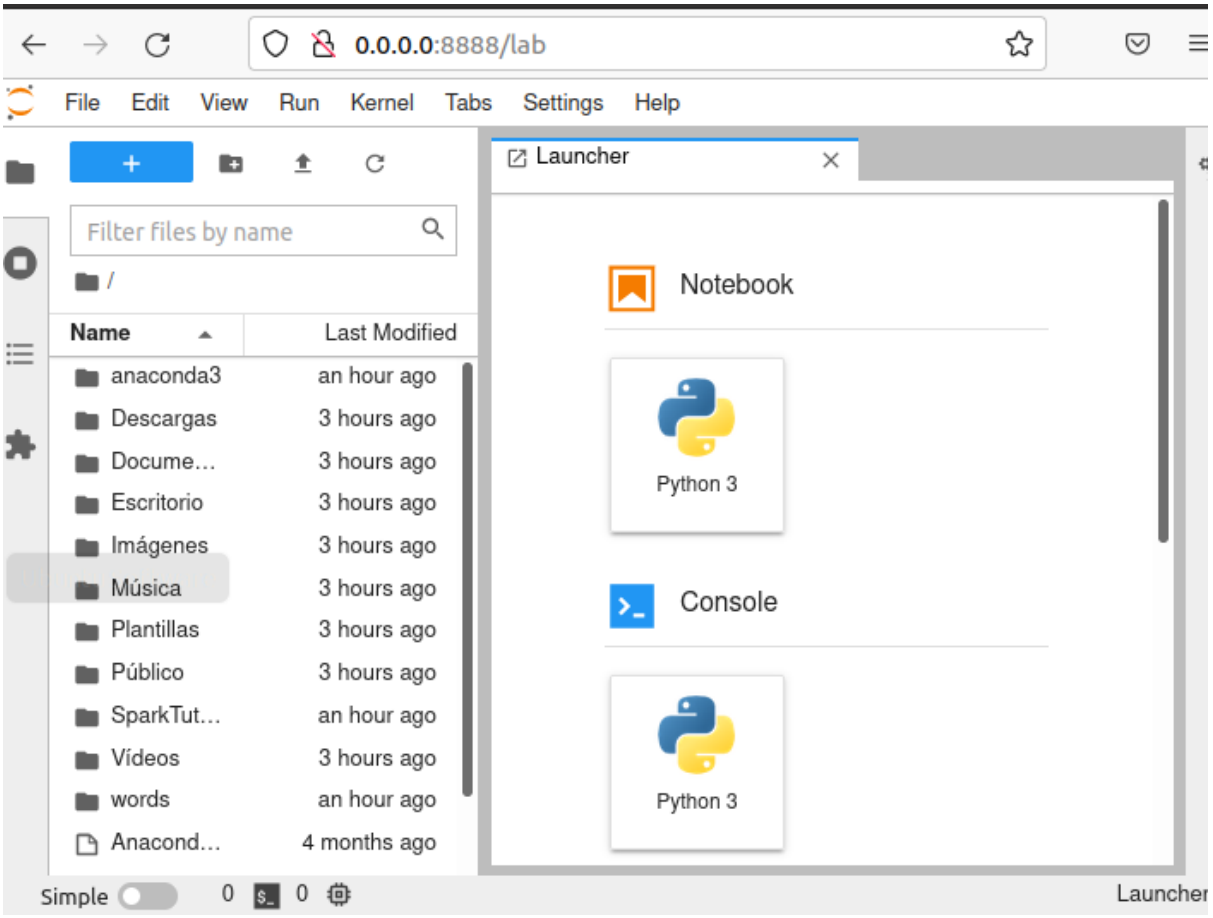
```
edna@edna-VirtualBox:~$ wget https://repo.anaconda.com/archive/Anaconda3-2021.0
5-Linux-x86_64.sh
--2021-09-15 19:57:02--  https://repo.anaconda.com/archive/Anaconda3-2021.05-Li
nux-x86_64.sh
Resolviendo repo.anaconda.com (repo.anaconda.com)... 104.16.131.3, 104.16.130.3
, 2606:4700::6810:8303, ...
Conectando con repo.anaconda.com (repo.anaconda.com)[104.16.131.3]:443... conec
tado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 570853747 (544M) [application/x-sh]
Guardando como: “Anaconda3-2021.05-Linux-x86_64.sh”

        Anaconda3-2  19%[==>                  ] 105,60M  9,45MB/s    eta 47s
```

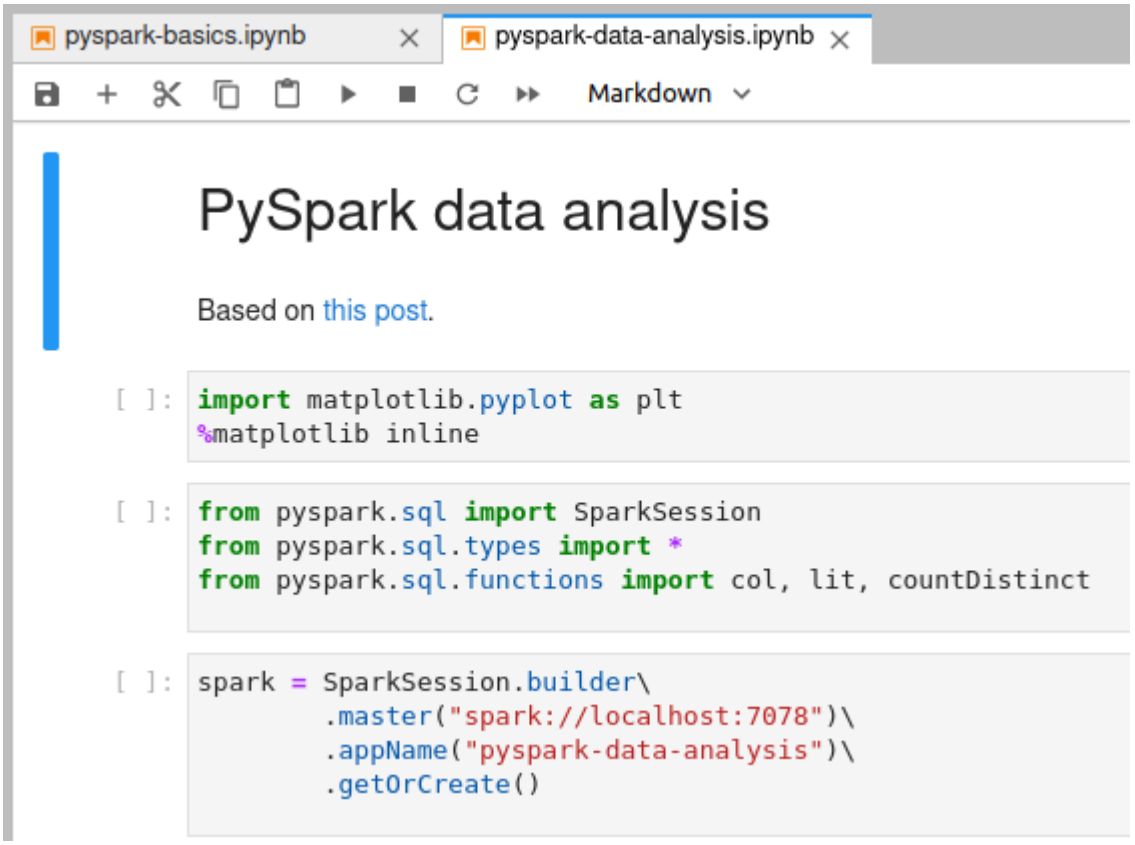
```
edna@edna-VirtualBox:~$ ls
Anaconda3-2021.05-Linux-x86_64.sh
Descargas
Documentos
Escritorio
```

3. Se inicia el Jupyter Lab

```
(base) edna@edna-VirtualBox:~$ jupyter lab --ip=0.0.0.0
[I 2021-09-15 21:02:15.954 ServerApp] jupyterlab | extension was successfully l
inked.
[I 2021-09-15 21:02:15.965 ServerApp] Writing notebook server cookie secret to
/home/edna/.local/share/jupyter/runtime/jupyter_cookie_secret
[I 2021-09-15 21:02:16.141 ServerApp] nbclassic | extension was successfully li
nked.
```

4. Se abre el navegador web en: http://0.0.0.0:8888/lab/. Es posible visualizar la estructura de archivos del repositorio clonado.


0.0.0.0:8888/lab
File Edit View Run Kernel Tabs Settings Help
Launcher
Filter files by name
/
Name
Last Modified
anaconda3 an hour ago
Descargas 3 hours ago
Docume... 3 hours ago
Escritorio 3 hours ago
Imágenes 3 hours ago
Música 3 hours ago
Plantillas 3 hours ago
Público 3 hours ago
SparkTut... an hour ago
Vídeos 3 hours ago
words an hour ago
Anacond... 4 months ago
Notebook
Python 3
Console
Python 3
Simple
Launcher


pyspark-basics.ipynb
pyspark-data-analysis.ipynb
Markdown
PySpark data analysis
Based on this post.
[ ]:
import matplotlib.pyplot as plt
%matplotlib inline
[ ]:
from pyspark.sql import SparkSession
from pyspark.sql.types import *
from pyspark.sql.functions import col, lit, countDistinct
[ ]:
spark = SparkSession.builder\
.master("spark://localhost:7078")\
.appName("pyspark-data-analysis")\
.getOrCreate()