

Alumno: Alter Caimi – DATA-PT06

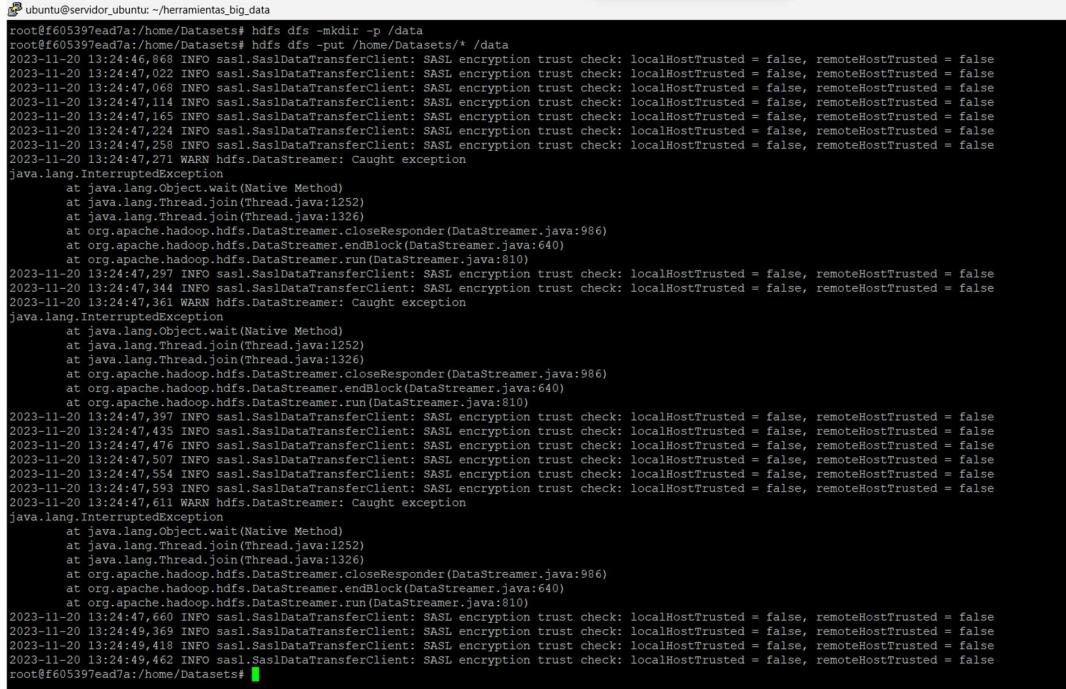
Parte 1



A screenshot of a terminal window titled "Ubuntu" showing the output of the command "sudo docker ps". The window lists several Docker containers running Hadoop components. The columns are: CONTAINER ID, IMAGE, NAMES, COMMAND, CREATED, STATUS, and PORTS. The ports listed include 9864/tcp, 8188/tcp, 0.0.0.0:9870->9870/tcp, 0.0.0.0:9870->9870/tcp, and 8042/tcp.

CONTAINER ID	IMAGE	NAMES	COMMAND	CREATED	STATUS	PORTS
80c572b1ec9f	bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8	/entrypoint.sh /run..."	7 minutes ago	Up 7 minutes (healthy)	0.0.0.0:9864->9864/tcp, :::9864->9864/tcp	
9a1dc3f6c565	bde2020/hadoop-historyserver:2.0.0-hadoop3.2.1-java8	/entrypoint.sh /run..."	7 minutes ago	Up 7 minutes (healthy)	8188/tcp	
f605397ead7a	bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8	/entrypoint.sh /run..."	7 minutes ago	Up 7 minutes (healthy)	0.0.0.0:9870->9870/tcp, :::9870->9870/tcp, 0.0.0.0:9870->9870/tcp	
9010>9000/tcp	bde2020/hadoop-nodemanager:2.0.0-hadoop3.2.1-java8	/entrypoint.sh /run..."	7 minutes ago	Up 7 minutes (healthy)	8042/tcp	
5b8112eb5b45	bde2020/hadoop-resourcemanager:2.0.0-hadoop3.2.1-java8	/entrypoint.sh /run..."	7 minutes ago	Up 7 minutes (healthy)	8086/tcp	
eac5ed15b279	bde2020/hadoop-resourcemanager:2.0.0-hadoop3.2.1-java8	"entrypoint.sh /run..."	7 minutes ago	Up 7 minutes (healthy)	8086/tcp	

Archivos cargados en el HDFS de Hadoop



A screenshot of a terminal window titled "Ubuntu" showing the output of the command "hdfs dfs -mkdir -p /data". The window then shows the results of "hdfs dfs -put /home/datasets/* /data". The logs show multiple INFO and WARN messages from the SaslDataTransferClient indicating SASL encryption trust checks and DataStreamer errors. The logs also mention java.lang.InterruptedIOException and java.lang.Object.wait() calls.

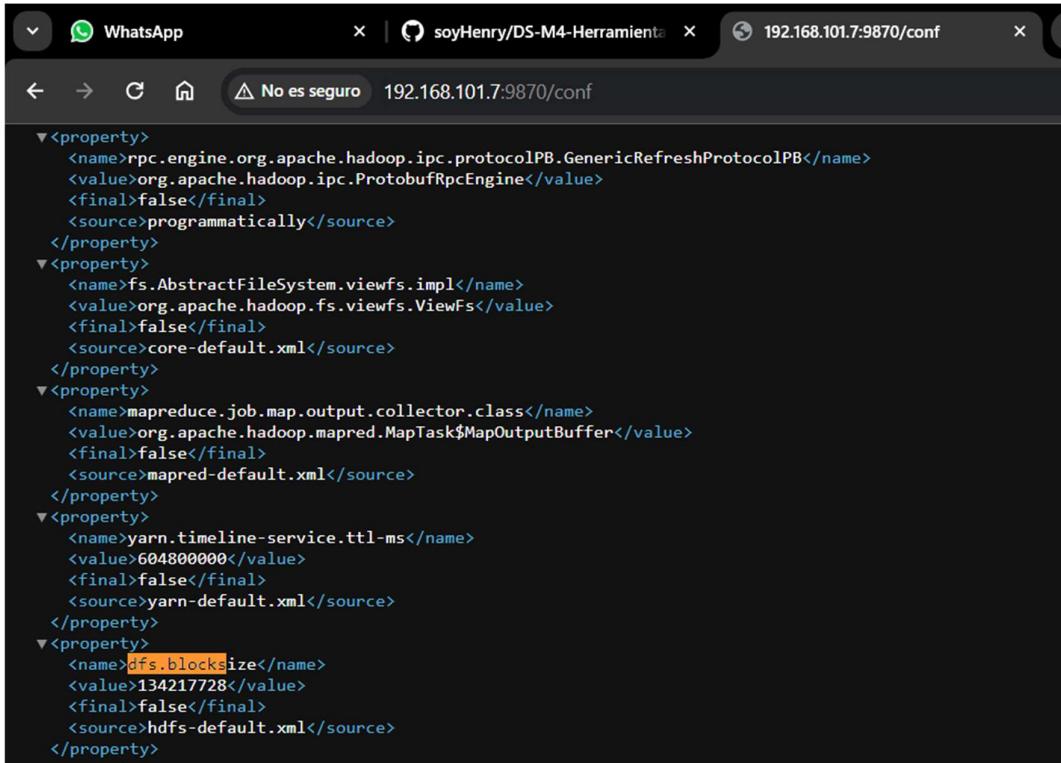
```
root@f605397ead7a:/home/datasets# hdfs dfs -mkdir -p /data
root@f605397ead7a:/home/datasets# hdfs dfs -put /home/datasets/* /data
2023-11-20 13:24:46.868 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2023-11-20 13:24:47.022 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2023-11-20 13:24:47.068 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2023-11-20 13:24:47.114 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2023-11-20 13:24:47.165 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2023-11-20 13:24:47.224 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2023-11-20 13:24:47.258 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2023-11-20 13:24:47.271 WARN hdfs.DataStreamer: Caught exception
java.lang.InterruptedIOException
at java.lang.Object.wait(Native Method)
at java.lang.Thread.join(Thread.java:1252)
at java.lang.Thread.join(Thread.java:1326)
at org.apache.hadoop.hdfs.DataStreamer.closeResponder(DataStreamer.java:986)
at org.apache.hadoop.hdfs.DataStreamer.endBlock(DataStreamer.java:640)
at org.apache.hadoop.hdfs.DataStreamer.run(DataStreamer.java:810)
2023-11-20 13:24:47.297 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2023-11-20 13:24:47.344 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2023-11-20 13:24:47.361 WARN hdfs.DataStreamer: Caught exception
java.lang.InterruptedIOException
at java.lang.Object.wait(Native Method)
at java.lang.Thread.join(Thread.java:1252)
at java.lang.Thread.join(Thread.java:1326)
at org.apache.hadoop.hdfs.DataStreamer.closeResponder(DataStreamer.java:986)
at org.apache.hadoop.hdfs.DataStreamer.endBlock(DataStreamer.java:640)
at org.apache.hadoop.hdfs.DataStreamer.run(DataStreamer.java:810)
2023-11-20 13:24:47.397 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2023-11-20 13:24:47.435 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2023-11-20 13:24:47.476 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2023-11-20 13:24:47.507 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2023-11-20 13:24:47.554 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2023-11-20 13:24:47.593 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2023-11-20 13:24:47.611 WARN hdfs.DataStreamer: Caught exception
java.lang.InterruptedIOException
at java.lang.Object.wait(Native Method)
at java.lang.Thread.join(Thread.java:1252)
at java.lang.Thread.join(Thread.java:1326)
at org.apache.hadoop.hdfs.DataStreamer.closeResponder(DataStreamer.java:986)
at org.apache.hadoop.hdfs.DataStreamer.endBlock(DataStreamer.java:640)
at org.apache.hadoop.hdfs.DataStreamer.run(DataStreamer.java:810)
2023-11-20 13:24:47.660 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2023-11-20 13:24:49.369 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2023-11-20 13:24:49.418 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2023-11-20 13:24:49.462 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
root@f605397ead7a:/home/datasets#
```

```

ubuntu@servidor_ubuntu:~/herramientas_big_data
ubuntu@servidor_ubuntu:~/herramientas_big_data$ sudo docker exec -it namenode bash
root@f605397ead7a:/# hdfs dfs -ls /data
Found 17 items
-rw-r--r-- 3 root supergroup 16308 2023-11-20 13:24 /data/airports.csv
drwxr-xr-x - root supergroup 0 2023-11-20 13:24 /data/calendario
drwxr-xr-x - root supergroup 0 2023-11-20 13:24 /data/canaldeventa
drwxr-xr-x - root supergroup 0 2023-11-20 13:24 /data/cliente
drwxr-xr-x - root supergroup 0 2023-11-20 13:24 /data/compra
drwxr-xr-x - root supergroup 0 2023-11-20 13:24 /data/data_nvo
drwxr-xr-x - root supergroup 0 2023-11-20 13:24 /data/empleado
drwxr-xr-x - root supergroup 0 2023-11-20 13:24 /data/gasto
-rw-r--r-- 3 root supergroup 4813 2023-11-20 13:24 /data/iris.csv
-rw-r--r-- 3 root supergroup 15802 2023-11-20 13:24 /data/iris.json
-rw-r--r-- 3 root supergroup 94 2023-11-20 13:24 /data/personal.csv
drwxr-xr-x - root supergroup 0 2023-11-20 13:24 /data/producto
drwxr-xr-x - root supergroup 0 2023-11-20 13:24 /data/proveedor
-rw-r--r-- 3 root supergroup 69772435 2023-11-20 13:24 /data/raw-flight-data.csv
drwxr-xr-x - root supergroup 0 2023-11-20 13:24 /data/sucursal
drwxr-xr-x - root supergroup 0 2023-11-20 13:24 /data/tipodegasto
drwxr-xr-x - root supergroup 0 2023-11-20 13:24 /data/venta
root@f605397ead7a:/#

```

Tamaño de bloque:



```

<property>
  <name>rpc.engine.org.apache.hadoop.ipc.protocolPB.GenericRefreshProtocolPB</name>
  <value>org.apache.hadoop.ipc.ProtobufRpcEngine</value>
  <final>false</final>
  <source>programmatically</source>
</property>
<property>
  <name>fs.AbstractFileSystem.viewfs.impl</name>
  <value>org.apache.hadoop.fs.viewfs.ViewFs</value>
  <final>false</final>
  <source>core-default.xml</source>
</property>
<property>
  <name>mapreduce.job.map.output.collector.class</name>
  <value>org.apache.hadoop.mapred.MapTask$MapOutputBuffer</value>
  <final>false</final>
  <source>mapred-default.xml</source>
</property>
<property>
  <name>yarn.timeline-service.ttl-ms</name>
  <value>60480000</value>
  <final>false</final>
  <source>yarn-default.xml</source>
</property>
<property>
  <name>dfs.blocksize</name>
  <value>134217728</value>
  <final>false</final>
  <source>hdfs-default.xml</source>
</property>

```

Factor de réplica

```

<property>
  <name>yarn.sharedcache.store.in-memory.check-period-mins</name>
  <value>720</value>
  <final>false</final>
  <source>yarn-default.xml</source>
</property>
<property>
  <name>fs.s3a.multipart.threshold</name>
  <value>2147483647</value>
  <final>false</final>
  <source>core-default.xml</source>
</property>
<property>
  <name>dfs.namenode.checkpoint.period</name>
  <value>3600s</value>
  <final>false</final>
  <source>hdfs-default.xml</source>
</property>
<property>
  <name>hadoop.security.kms.client.encrypted.key.cache.low-watermark</name>
  <value>0.3f</value>
  <final>false</final>
  <source>core-default.xml</source>
</property>
<property>
  <name>dfs.replication</name>
  <value>3</value>
  <final>false</final>
  <source>hdfs-default.xml</source>
</property>

```

Parte 2

Containers creados:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
fd7cheic5903	bde2020/hive:2.3.2-postgresql-metastore	"entrypoint.sh /opt/..."	55 seconds ago	Up 53 seconds	0.0.0.0:10000->10000/tcp, :::10000->
e059575fa01	bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8	"entrypoint.sh /run..."	About a minute ago	Up 54 seconds (healthy)	0.0.0.0:9870->9870/tcp, :::9870->9870
0/tcp	0.0.0.0:9010->9000/tcp				
f41a5d703ea9	bde2020/hadoop-nodemanager:2.0.0-hadoop3.2.1-java8	"entrypoint.sh /run..."	About a minute ago	Up 58 seconds (health: starting)	8042/tcp
2558547fe7e1	bde2020/hive:2.3.2-postgresql-metastore	"entrypoint.sh /opt/..."	About a minute ago	Up 55 seconds	10000/tcp, 0.0.0.0:9003->9003/tcp, :::9003->9003
:9803->9003/tcp					
c2323bb0f1198	bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8	"entrypoint.sh /run..."	About a minute ago	Up 55 seconds (healthy)	0.0.0.0:9864->9864/tcp, :::9864->9864
4/tcp					
65fd0d276fd4	bde2020/hadoop-historyserver:2.0.0-hadoop3.2.1-java8	"entrypoint.sh /run..."	About a minute ago	Up 58 seconds (health: starting)	8188/tcp
a5a004f120f	bde2020/hive-metastore-postgresql:2	"docker-entrypoint..."	About a minute ago	Up 55 seconds	0.0.0.0:5432->5432/tcp, :::5432->5432
2/tcp					
eb3408ef15fd	bde2020/hadoop-resourcemanager:2.0.0-hadoop3.2.1-java8	"entrypoint.sh /run..."	About a minute ago	Up 19 seconds (health: starting)	8088/tcp

Ejecutando script de creación de tablas:

```

ubuntu@servidor_ubuntu:~/herramientas_big_data
root@fd7cbe1c8503:/opt# hive -f Pas002.hql
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop-2.7.4/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in file:/opt/hive/conf/hive-log4j2.properties Async: true
Could not open input file for reading. (File file:/opt/Pas002.hql does not exist)
root@fd7cbe1c8503:/opt# ls -l
total 16
drwxr-xr-x 1 20415 input 4096 Feb  5 2018 hadoop-2.7.4
drwxr-xr-x 1 root  root  4096 Nov 20 14:16 hive
root@fd7cbe1c8503:/opt# cd hive
root@fd7cbe1c8503:/opt/hive# ls
LICENSE NOTICE Pas002.hql RELEASE_NOTES.txt bin binary-package-licenses conf examples hcatalog jdbc lib scripts
root@fd7cbe1c8503:/opt/hive# ls -l
total 84
-rw-r--r-- 1 root  staff  20798 Nov  9 2017 LICENSE
-rw-r--r-- 1 root  staff  230 Nov  9 2017 NOTICE
-rw-r--r-- 1 1000  1000  4679 Nov 20 13:08 Pas002.hql
-rw-r--r-- 1 root  staff  1979 Nov  9 2017 RELEASE_NOTES.txt
drwxr-xr-x 3 root  root  4096 Feb  5 2018 bin
drwxr-xr-x 2 root  root  4096 Feb  5 2018 binary-package-licenses
drwxr-xr-x 1 root  root  4096 Nov 20 13:49 conf
drwxr-xr-x 4 root  root  4096 Feb  5 2018 examples
drwxr-xr-x 7 root  root  4096 Feb  5 2018 hcatalog
drwxr-xr-x 2 root  root  4096 Feb  5 2018 jdbc
drwxr-xr-x 4 root  root  16384 Feb  5 2018 lib
drwxr-xr-x 4 root  root  4096 Feb  5 2018 scripts
root@fd7cbe1c8503:/opt/hive# clear
root@fd7cbe1c8503:/opt/hive# hive -f Pas002.hql
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop-2.7.4/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in file:/opt/hive/conf/hive-log4j2.properties Async: true
OK
Time taken: 1.924 seconds
OK
Time taken: 0.085 seconds
OK
Time taken: 0.18 seconds
OK
Time taken: 0.383 seconds
OK
Time taken: 0.016 seconds
OK
Time taken: 0.076 seconds
OK

```

Tablas creadas:

```

ubuntu@servidor_ubuntu:~/herramientas_big_data
root@fd7cbe1c8503:/opt/hive# hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop-2.7.4/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in file:/opt/hive/conf/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
hive> use integrador;
OK
Time taken: 0.586 seconds
hive> show tables;
OK
calendario
canal_venta
cliente
compra
empleado
piso
producto
proveedor
sucursal
tipo_gasto
venta
Time taken: 0.132 seconds, Fetched: 11 row(s)
hive> 

```

Consulta:

```

SELECT AVG(precio * cantidad) AS venta_promedio, idsucursal FROM venta
GROUP BY idsucursal
ORDER BY venta_promedio DESC;

```

```
ubuntu@servidor_ubuntu: ~/herramientas_big_data
2023-11-20 14:33:32,694 Stage-1 map = 0%,  reduce = 0%
2023-11-20 14:33:33,724 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local11057317954_0001
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2023-11-20 14:33:35,065 Stage-2 map = 100%,  reduce = 100%
Ended Job = job_local1462372067_0002
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 5157892 HDFS Write: 0 SUCCESS
Stage-Stage-2:  HDFS Read: 5157892 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
venta_promedio  idsucursal
100258.78663694278      31
32976.98284560569      10
19754.544860583042     7
18274.348849315073     14
14564.586478405332     5
13998.286618773946     3
10214.726294583888     16
9953.812500000013     4
9936.55436946149      6
9858.066563483742     2
9357.612461538452     9
8795.02870005058      23
8743.018868999192      30
8597.748954918026     15
8265.324677685952     22
8050.08905041032      8
8006.459319470695     20
8000.039580952381     21
7935.565281076803     18
7900.035384252727     24
7556.244771528999     1
7461.261488641867     26
7346.727169650474     12
7167.582071713144     17
6798.589407799711     25
6651.844915966382     29
5738.406808813071     27
5638.821593274617     13
5347.6253525046395    11
3751.3727988338173    19
Time taken: 4.823 seconds, Fetched: 30 row(s)
hive>
```

Paso 03

Creación de tablas Parquet-Snappy

```

ubuntu@servidor_ubuntu:~/herramientas_big_data
Moving data to directory hdfs://namenode:9000/data2/sucursal/.hive-staging_hive_2023-11-20_19-18-23_021_7092461627262767470-1/-ext-10000
Loading data to table integrador2.sucursal
MapReduce Jobs Launched:
Stage-Stage-1: HDFS Read: 4398417 HDFS Write: 987475 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 1.77 seconds
OK
Time taken: 0.018 seconds
OK
Time taken: 0.066 seconds
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = root_20231120191824_59994906-ea19-4731-85d4-3f5db89cdfe65
Total Job = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Job running in-process (local Hadoop)
2023-11-20 19:18:26,500 Stage-1 map = 100%, reduce = 0%
Ended Job = job_local004042203_0014
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://namenode:9000/data2/calendario/.hive-staging_hive_2023-11-20_19-18-24_883_5401752822395510836-1/-ext-10000
Loading data to table integrador2.calendario
MapReduce Jobs Launched:
Stage-Stage-1: HDFS Read: 4508277 HDFS Write: 1607969 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 1.898 seconds
OK
Time taken: 0.01 seconds
OK
Time taken: 0.052 seconds
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = root_20231120191826_e6ba9daf-bb4e-4705-84f8-772adaae6724
Total Job = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Job running in-process (local Hadoop)
2023-11-20 19:18:28,177 Stage-1 map = 100%, reduce = 0%
Ended Job = job_local004042203_0014
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://namenode:9000/data2/proveedor/.hive-staging_hive_2023-11-20_19-18-26_850_4037659467268885478-1/-ext-10000
Loading data to table integrador2.proveedor
MapReduce Jobs Launched:
Stage-Stage-1: HDFS Read: 4509473 HDFS Write: 1010173 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
Time taken: 1.567 seconds
root@fd7cbe1c503:/opt/hive# [REDACTED]

ubuntu@servidor_ubuntu:~/herramientas_big_data
root@fd7cbe1c503:/opt/hive$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [/opt/hive/libexec/jars/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [/opt/hive/libexec/jars/hadoop-common-2.7.4/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in file:/opt/hive/conf/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
hive> use integrador2;
OK
Time taken: 0.495 seconds
hive> show tables;
OK
calendario
canal_venta
cliente
compra
empleado
gasto
producto
proveedor
sucursal
tipo_gasto
venta
Time taken: 0.126 seconds, Fetched: 11 row(s)
hive. [REDACTED]

```

SET hive.cli.print.header=true; (comando para que muestre los nombres de columna).

Ejecutamos la QUERY:

```
hive> SELECT AVG(cantidad * precio) AS venta_promedio, idcanal FROM venta
```

```
> GROUP BY idcanal
```

```
> ORDER BY venta_promedio DESC;
```

```

ubuntu@servidor_ubuntu:~/herramientas/big_data
$ hive -e "SELECT idSucursal, sum(cantidad * precio) AS venta_promedio, idcanal FROM venta
    GROUP BY idcanal
    ORDER BY venta_promedio DESC;
"
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
1 2018-03-09 2018-03-17 3 969 13 1674 4281.5 812.12 2
2 2018-03-09 2018-03-17 2 954 13 1674 4281.5 543.18 3
3 2016-03-28 2016-03-31 2 1722 13 1674 42837 430.32 1
4 2017-10-23 2017-10-24 3 2876 13 1674 42834 818.84 2
5 2017-11-22 2017-11-25 2 678 13 1674 42825 554.18 3
6 2016-01-01 2016-01-02 2 3263 13 1674 42850 152.1 1
7 2015-03-25 2015-03-26 3 2903 13 1674 42849 150.5 1
8 2017-07-10 2017-07-18 2 201 13 1674 42940 2162.0 2
9 2018-04-03 2018-04-06 2 1006 13 1674 42905 456.0 3
10 2019-03-11 2019-03-17 1 1003 13 1674 42894 515.0 2
Time taken: 1.549 seconds, Fetched: 10 row(s)
hive> SELECT idSucursal, sum(cantidad * precio) AS venta_promedio, idcanal FROM venta
    GROUP BY idcanal
    ORDER BY venta_promedio DESC;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = root_20231120193850_96a35684-5df3-4d38-b0f7-bc0832efec97
Total Jobs = 1
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2023-11-20 19:38:52,432 Stage-1 map = 0%, reduce = 0%
2023-11-20 19:38:52,432 Stage-1 map = 100%, reduce = 100%
Ended Job = job_local11012303_0001
Launching Job 2 out of 2
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
2023-11-20 19:38:52,432 Stage-2 map = 100%, reduce = 100%
Ended Job = job_local11012303_0002
MapReduce Jobs Launched:
Stage-Stage-1: HDFS Read: 1306762 HDFS Write: 0 SUCCESS
Stage-Stage-2: HDFS Read: 1306762 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
venta_promedio idcanal
14567.69969682513 3
12118.735389057414 2
9916.872401237091 1
Time taken: 4.728 seconds, Fetched: 3 row(s)
hive> 

```

Paso 04

Creamos el índice

```

Logging initialized using configuration in file:/opt/hive/conf/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Hive> use integrador2;
OK
Time taken: 0.534 seconds
hive> use integrador2;
OK
Time taken: 0.007 seconds
hive> CREATE INDEX index_venta_sucursal ON TABLE venta(IdSucursal) AS 'org.apache.hadoop.hive.index.compact.CompactIndexHandler' WITH DEFERRED REBUILD;
OK

```

Ejecutamos query

```
hive> select idsucursal, sum(precio * cantidad) from venta group by idsucursal;
```

```
ubuntu@servidor_ubuntu: ~/herramientas_big_data
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Job running in-process (local Hadoop)
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
2023-11-20 19:43:21,935 Stage-1 map = 0%,  reduce = 0%
2023-11-20 19:43:22,942 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local1896423402_0001
MapReduce Jobs Launched:
Stage-Stage-1: HDFS Read: 162062 HDFS Write: 0 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
1      1.7198013113830566E7
2      1.8789474888427734E7
3      1.4614211233947754E7
4      2.034559274472046E7
5      1.3151821603469849E7
6      1.4576925257827759E7
7      6.234534359597778E7
8      1.3733451917617798E7
9      7907182.54208374
10     6.941654890026855E7
11     5764740.125061035
12     8617710.976959229
13     7042888.17578125
14     2.668054930819702E7
15     1.6782805977355957E7
16     1.546509561416626E7
17     1.0794378599456787E7
18     1.0022618956039429E7
19     1286720.8696594238
20     8470833.9553833
21     1.2600062337417603E7
22     5000521.427429199
23     1.7387771740737915E7
24     1.6755975042282104E7
25     1.4120670222351074E7
26     1.5437350034942627E7
27     8073938.3884887695
29     6332556.36026001
30     1.0745170186523438E7
31     7.870314750735474E7
Time taken: 4.862 seconds, Fetched: 30 row(s)
hive>
```

Parte 5

HBase

Contenedores levantados:

```
ubuntu@servidor_ubuntu:~/herramientas_big_data$ sudo docker ps
CONTAINER ID        IMAGE
183b3569f3        bde2020/hive:2.3.2-postgresql-metastore
002/tcp
024045a26e        bde2020/hbase-master:1.0.0-hbase1.0.0
010/tcp
d23aa8b3018a      bde2020/hbase-regionserver:1.0.0-hbase1.2.6
030/tcp
7e52c03b950       bde2020/hadoop-resourcemanager:2.0.0-hadoop3.2.1-java8
040/tcp
sr365297cd0       bde2020/hive:2.3.2-postgresql-metastore
tcp, 10002/tcp
f0490406680       bde2020/hadoop-historyserver:2.0.0-hadoop3.2.1-java8
d30c2b653d40      apache/zeppelin:0.9.0
59f18e8ffff69     bde2020/hive-metastore-postgresql:2.3.0
0:9010->9000/tcp, :::9010->9000/tcp
9eeeddfef8        neodjilistat
cp, 0.0.0.0:7687->7687/tcp, ::7687->7687/tcp, neo4j
28af2c1eb8da      bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8
f2eedcc3f67a      bde2020/hadoop-nodemanager:2.0.0-hadoop3.2.1-java8
b4108d459343      zookeeper:3.4.10
cp, 3888/tcp
ubuntu@servidor_ubuntu:~/herramientas_big_data$
```

Comando: scan 'personal'. Muestra los datos de la tabla creada

```
=> ["personal"]
hbase(main):024:0> get 'personal', '4'
COLUMN          CELL
personal_data:city   timestamp=1700828795522, value=Caracas
personal_data:name  timestamp=1700828789793, value=Eliecer
2 row(s) in 0.0610 seconds

hbase(main):025:0> scan 'personal'
ROW              COLUMN+CELL
1               column=personal_data:age, timestamp=1700828754704, value=25
1               column=personal_data:city, timestamp=1700828747143, value=C\xC3\xB3rdoba
1               column=personal_data:name, timestamp=1700828738703, value=Juan
2               column=personal_data:age, timestamp=1700828773188, value=32
2               column=personal_data:city, timestamp=1700828767328, value=Lima
2               column=personal_data:name, timestamp=1700828761656, value=Franco
3               column=personal_data:age, timestamp=1700828784271, value=34
3               column=personal_data:name, timestamp=1700828778841, value=Ivan
4               column=personal_data:city, timestamp=1700828795522, value=Caracas
4               column=personal_data:name, timestamp=1700828789793, value=Eliecer
4 row(s) in 0.1110 seconds

hbase(main):026:0>
```

Luego de pasar el archivo personal.csv al HDFS, y de ejecutar la importación de los datos de personal.csv a una tabla Hbase con el comando ImportTsv. Luego mostramos la tabla con el comando scan 'personal'.

```
HDFS: Number of write operations=0
Map-Reduce Framework
  Map input records=4
  Map output records=4
  Input split bytes=109
  Shuffles=0
  Failed Shuffles=0
  Merged Map outputs=0
  GC time elapsed (ms)=9
  CPU time spent (ms)=0
  Physical memory (bytes) snapshot=0
  Virtual memory (bytes) snapshot=0
  Total committed heap usage (bytes)=62849024

ImportTsv
  End Lines=0
  File Input Format Counters
    Bytes Read=94
  File Output Format Counters
    Bytes Written=0
root@hbase-master:/# hbase shell
2023-11-24 12:52:43,233 WARN [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
HBase Shell; enter 'help|RETURN' for list of supported commands.
Type "exit|RETURN" to leave the HBase Shell
Version 1.2.6, rUnknown, Mon May 29 02:25:32 CDT 2017

hbase(main):001:0> scan 'personal'
ROW              COLUMN+CELL
1               column=personal_data:age, timestamp=1700828754704, value=25
1               column=personal_data:city, timestamp=1700828747143, value=C\xC3\xB3rdoba
1               column=personal_data:name, timestamp=1700828738703, value=Juan
2               column=personal_data:age, timestamp=1700828773188, value=32
2               column=personal_data:city, timestamp=1700828767328, value=Lima
2               column=personal_data:name, timestamp=1700828761656, value=Franco
3               column=personal_data:age, timestamp=1700828784271, value=34
3               column=personal_data:name, timestamp=1700828778841, value=Ivan
4               column=personal_data:city, timestamp=1700828795522, value=Caracas
4               column=personal_data:name, timestamp=1700828789793, value=Eliecer
5               column=personal_data:age, timestamp=1700830340086, value=24
5               column=personal_data:city, timestamp=1700830340086, value=Buenos Aires
5               column=personal_data:name, timestamp=1700830340086, value=Olivia
6               column=personal_data:age, timestamp=1700830340086, value=27
6               column=personal_data:city, timestamp=1700830340086, value=Buenos Aires
6               column=personal_data:name, timestamp=1700830340086, value=Juana
7               column=personal_data:age, timestamp=1700830340086, value=26
7               column=personal_data:city, timestamp=1700830340086, value=Mexico DF
7               column=personal_data:name, timestamp=1700830340086, value=Orlando
8               column=personal_data:age, timestamp=1700830340086, value=30
8               column=personal_data:city, timestamp=1700830340086, value=Ontario
8               column=personal_data:name, timestamp=1700830340086, value=Ricardo
8 row(s) in 0.6520 seconds

hbase(main):002:0>
```

Creamos la tabla 'album', poblamos con datos, y mostramos la familia de columnas 'label1':

```

hbase(main):002:0> create 'album','label','image'
0 row(s) in 1.4240 seconds

=> Hbase::Table - album
hbase(main):003:0> put 'album','label1','label:size','10'
0 row(s) in 0.1370 seconds

hbase(main):004:0> put 'album','label1','label:color','255:255:255'
0 row(s) in 0.0460 seconds

hbase(main):005:0> put 'album','label1','label:text','Family album'
0 row(s) in 0.0330 seconds

hbase(main):006:0> put 'album','label1','image:name','holiday'
0 row(s) in 0.0150 seconds

hbase(main):007:0> put 'album','label1','image:source','/tmp/pic1.jpg'
0 row(s) in 0.0150 seconds

hbase(main):008:0> get 'album','label1'
COLUMN                                CELL
  image:name                           timestamp=1700831080983, value=holiday
  image:source                          timestamp=1700831086689, value=/tmp/pic1.jpg
  label:color                           timestamp=1700831067882, value=255:255:255
  label:size                            timestamp=1700831056007, value=10
  label:text                           timestamp=1700831074342, value=Family album
5 row(s) in 0.0630 seconds

hbase(main):009:0> █

```

MongoDB

Para esta parte tuve que editar el .yml para usar la versión mongodb:4.4.6

Una vez levantados los contenedores, ejecutamos un mongoimport

```

root@2882c91cb5be:/#
ubuntu@servidor_ubuntu:~/herramientas_big_data/Datasets$ sudo docker exec -it mongodb bash
root@2882c91cb5be:/# mongoimport /data/iris.csv --type csv --headerline -d dataprueba -c iris_csv
2023-11-24T14:40:11.474+0000      connected to: mongodb://localhost/
2023-11-24T14:40:11.540+0000      150 document(s) imported successfully. 0 document(s) failed to import.
root@2882c91cb5be:/# █

```

Luego otro mongoimport desde un json:

```

root@2882c91cb5be:/#
root@2882c91cb5be:/# mongoimport --db dataprueba --collection iris_json --file /data/iris.json --jsonArray
2023-11-24T14:43:40.377+0000      connected to: mongodb://localhost/
2023-11-24T14:43:40.424+0000      150 document(s) imported successfully. 0 document(s) failed to import.
root@2882c91cb5be:/# █

```

En lugar de usar mongosh, usamos la línea de comandos de mongo, debido a que no reconoce el comando mongosh. Luego usamos la base de datos creada y mostramos los documentos presentes en iris_csv

```

root@2882c91cb5be:/# mongo
MongoDB shell version v4.4.6
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&kgssapiServiceName=mongodb
Implicit session: session { "id" : UUID("e3abb386-a3e1-4c0b-bc89-996554744742") }
MongoDB server version: 4.4.6
The server generated these startup warnings when booting:
2023-11-24T14:35:02.466+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2023-11-24T14:35:05.890+00:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).
The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> use dataprueba
switched to db dataprueba
> show collections
iris
iris_json
> db.iris_csv.find()
{
  "_id": ObjectId("6560b5cb13eb637dac22ff"),
  "fila": "fila1",
  "sepal length": 5.1,
  "sepal width": 3.5,
  "petal length": 1.4,
  "petal width": 0.2,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b5cb13eb637dac2300"),
  "fila": "fila2",
  "sepal length": 4.9,
  "sepal width": 3,
  "petal length": 1.4,
  "petal width": 0.2,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b5cb13eb637dac2301"),
  "fila": "fila3",
  "sepal length": 4.7,
  "sepal width": 3.2,
  "petal length": 1.3,
  "petal width": 0.2,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b5cb13eb637dac2302"),
  "fila": "fila4",
  "sepal length": 4.6,
  "sepal width": 3.1,
  "petal length": 1.5,
  "petal width": 0.2,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b5cb13eb637dac2303"),
  "fila": "fila5",
  "sepal length": 5.4,
  "sepal width": 3.9,
  "petal length": 1.4,
  "petal width": 0.4,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b5cb13eb637dac2304"),
  "fila": "fila6",
  "sepal length": 5.4,
  "sepal width": 3.4,
  "petal length": 1.7,
  "petal width": 0.3,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b5cb13eb637dac2305"),
  "fila": "fila7",
  "sepal length": 5,
  "sepal width": 3.4,
  "petal length": 1.5,
  "petal width": 0.2,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b5cb13eb637dac2306"),
  "fila": "fila8",
  "sepal length": 5,
  "sepal width": 3.4,
  "petal length": 1.5,
  "petal width": 0.2,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b5cb13eb637dac2307"),
  "fila": "fila9",
  "sepal length": 4.6,
  "sepal width": 3.0,
  "petal length": 1.4,
  "petal width": 0.1,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b5cb13eb637dac2308"),
  "fila": "fila10",
  "sepal length": 5.4,
  "sepal width": 3.4,
  "petal length": 1.5,
  "petal width": 0.2,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b5cb13eb637dac2309"),
  "fila": "fila11",
  "sepal length": 5.4,
  "sepal width": 3.4,
  "petal length": 1.6,
  "petal width": 0.2,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b5cb13eb637dac2310"),
  "fila": "fila12",
  "sepal length": 4.8,
  "sepal width": 3.0,
  "petal length": 1.4,
  "petal width": 0.1,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b5cb13eb637dac2311"),
  "fila": "fila13",
  "sepal length": 4.8,
  "sepal width": 3.4,
  "petal length": 1.4,
  "petal width": 0.1,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b5cb13eb637dac2312"),
  "fila": "fila14",
  "sepal length": 4.3,
  "sepal width": 3.0,
  "petal length": 1.1,
  "petal width": 0.1,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b5cb13eb637dac2308"),
  "fila": "fila15",
  "sepal length": 5.8,
  "sepal width": 4,
  "petal length": 1.2,
  "petal width": 0.2,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b5cb13eb637dac2309"),
  "fila": "fila16",
  "sepal length": 5.7,
  "sepal width": 4.4,
  "petal length": 1.5,
  "petal width": 0.4,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b5cb13eb637dac2310"),
  "fila": "fila17",
  "sepal length": 5.4,
  "sepal width": 3.9,
  "petal length": 1.3,
  "petal width": 0.4,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b5cb13eb637dac2311"),
  "fila": "fila18",
  "sepal length": 5.1,
  "sepal width": 3.5,
  "petal length": 1.4,
  "petal width": 0.3,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b5cb13eb637dac2312"),
  "fila": "fila19",
  "sepal length": 5.7,
  "sepal width": 3.8,
  "petal length": 1.7,
  "petal width": 0.3,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b5cb13eb637dac2313"),
  "fila": "fila20",
  "sepal length": 5.1,
  "sepal width": 3.8,
  "petal length": 1.5,
  "petal width": 0.3,
  "species": "setosa"
}
Type "it" for more
>

```

Mostramos los documentos de iris_json:

```

> db.iris_json.find()
{
  "_id": ObjectId("6560b69ca1004d86f80bd592"),
  "sepalLength": 5.1,
  "sepalWidth": 3.5,
  "petalLength": 1.4,
  "petalWidth": 0.2,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b69ca1004d86f80bd593"),
  "sepalLength": 4.9,
  "sepalWidth": 3,
  "petalLength": 1.4,
  "petalWidth": 0.2,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b69ca1004d86f80bd594"),
  "sepalLength": 4.7,
  "sepalWidth": 3.2,
  "petalLength": 1.3,
  "petalWidth": 0.2,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b69ca1004d86f80bd595"),
  "sepalLength": 4.6,
  "sepalWidth": 3.1,
  "petalLength": 1.5,
  "petalWidth": 0.2,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b69ca1004d86f80bd596"),
  "sepalLength": 5,
  "sepalWidth": 3.6,
  "petalLength": 1.4,
  "petalWidth": 0.2,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b69ca1004d86f80bd597"),
  "sepalLength": 5.4,
  "sepalWidth": 3.9,
  "petalLength": 1.5,
  "petalWidth": 0.4,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b69ca1004d86f80bd598"),
  "sepalLength": 4.6,
  "sepalWidth": 3.4,
  "petalLength": 1.4,
  "petalWidth": 0.3,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b69ca1004d86f80bd599"),
  "sepalLength": 5,
  "sepalWidth": 3.4,
  "petalLength": 1.5,
  "petalWidth": 0.2,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b69ca1004d86f80bd59a"),
  "sepalLength": 4.4,
  "sepalWidth": 2.9,
  "petalLength": 1.4,
  "petalWidth": 0.2,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b69ca1004d86f80bd59b"),
  "sepalLength": 4.3,
  "sepalWidth": 3.0,
  "petalLength": 1.4,
  "petalWidth": 0.1,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b69ca1004d86f80bd59c"),
  "sepalLength": 5.4,
  "sepalWidth": 3.7,
  "petalLength": 1.5,
  "petalWidth": 0.2,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b69ca1004d86f80bd59d"),
  "sepalLength": 4.6,
  "sepalWidth": 3.7,
  "petalLength": 1.6,
  "petalWidth": 0.2,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b69ca1004d86f80bd59e"),
  "sepalLength": 4.6,
  "sepalWidth": 3,
  "petalLength": 1.4,
  "petalWidth": 0.1,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b69ca1004d86f80bd59f"),
  "sepalLength": 4.3,
  "sepalWidth": 3.0,
  "petalLength": 1.7,
  "petalWidth": 0.1,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b69ca1004d86f80bd5a0"),
  "sepalLength": 5.6,
  "sepalWidth": 4,
  "petalLength": 1.2,
  "petalWidth": 0.2,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b69ca1004d86f80bd5a1"),
  "sepalLength": 5.7,
  "sepalWidth": 4.4,
  "petalLength": 1.5,
  "petalWidth": 0.4,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b69ca1004d86f80bd5a2"),
  "sepalLength": 5.4,
  "sepalWidth": 3.9,
  "petalLength": 1.3,
  "petalWidth": 0.4,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b69ca1004d86f80bd5a3"),
  "sepalLength": 5.1,
  "sepalWidth": 3.5,
  "petalLength": 1.4,
  "petalWidth": 0.3,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b69ca1004d86f80bd5a4"),
  "sepalLength": 5.7,
  "sepalWidth": 3.8,
  "petalLength": 1.7,
  "petalWidth": 0.3,
  "species": "setosa"
}
{
  "_id": ObjectId("6560b69ca1004d86f80bd5a5"),
  "sepalLength": 5.1,
  "sepalWidth": 3.8,
  "petalLength": 1.5,
  "petalWidth": 0.3,
  "species": "setosa"
}
Type "it" for more
>

```

Aplicamos un primero mongoexport:

```

root@2882c91cb5be:/# mongoexport --db dataprueba --collection iris_csv --fields sepal_length,sepal_width,petal_length,petal_width,species --type=csv --out /data/iris_export.csv
2023-11-24T16:46:10.532+0000    connected to: mongodb://localhost/
2023-11-24T16:46:10.638+0000    exported 150 records
root@2882c91cb5be:#

```

Aplicamos un segundo mongoexport:

```

root@2882c91cb5be:/# mongoexport --db dataprueba --collection iris_json --fields sepal_length,sepal_width,petal_length,petal_width,species --type=json --out /data/iris_export.json
2023-11-24T16:46:59.506+0000    connected to: mongodb://localhost/
2023-11-24T16:46:59.541+0000    exported 150 records
root@2882c91cb5be:#

```

Copiamos los archivos .jar dentro de hive-server:

```

ubuntu@servidor_ubuntu:~/herramientas_big_data/Mongo
ubuntu@servidor_ubuntu:~/herramientas_big_data$ ls
Datasets          Pas001.sh          Pas004_ConConsulta.hql          docker-compose-kafka.yml      docker-compose-v5.yml      hbase-distributed-local.env
Generacion_Ventas.ipynb  Pas002.hql          Pas005.py          docker-compose-v1.yml      docker-compose-v1.yml      iris.hql
MapReduce          Pas003.hql          Pas006_IncrementalVentas.py  docker-compose-v2.yml      docker-compose-v2.yml      pruebasPySpark.py
Parquet           Pas001.hql          Pas006_IncrementalVentas.py  docker-compose-v3.yml      docker-compose-v3.yml      pruebasScala.scala
Pas000.sh          Pas004.hql          README.md          docker-compose-v4.yml      docker-compose-v4.yml      pyspark-ETL.ipynb
ubuntu@servidor_ubuntu:~/herramientas_big_data$ cd Mongo/
ubuntu@servidor_ubuntu:~/herramientas_big_data/Mongo$ ls
mongo-hadoop-core-2.0.2.jar  mongo-hadoop-spark-2.0.2.jar  mongo-java-driver-3.12.11.jar  mongo-scala-driver-0.8.15.jar
ubuntu@servidor_ubuntu:~/herramientas_big_data/Mongo$ sudo docker cp mongo-hadoop-hive-2.0.2.jar hive-server:/opt/hive/lib/mongo-hadoop-hive-2.0.2.jar
ubuntu@servidor_ubuntu:~/herramientas_big_data/Mongo$ sudo docker cp mongo-hadoop-core-2.0.2.jar hive-server:/opt/hive/lib/mongo-hadoop-core-2.0.2.jar
ubuntu@servidor_ubuntu:~/herramientas_big_data/Mongo$ sudo docker cp mongo-hadoop-spark-2.0.2.jar hive-server:/opt/hive/lib/mongo-hadoop-spark-2.0.2.jar
ubuntu@servidor_ubuntu:~/herramientas_big_data/Mongo$ sudo docker cp mongo-java-driver-3.12.11.jar hive-server:/opt/hive/lib/mongo-java-driver-3.12.11.jar
ubuntu@servidor_ubuntu:~/herramientas_big_data/Mongo$ 

```

Copiamos iris.hql en hive-server:

```
ubuntu@servidor_ubuntu:~/herramientas_big_data
ubuntu@servidor_ubuntu:~/herramientas_big_data/Mongo$ cd ..
ubuntu@servidor_ubuntu:~/herramientas_big_data$ ls
datasets
Paso01.sh          Paso04_ConConsulta.hql      docker-compose-kafka.yml  docker-compose-v5.yml   hbase-distributed-local.env
Generacion_Ventas.ipynb  Paso02_hql            docker-compose-v1.yml    docker-compose.yml   iris.hql
Mongo               Paso02_ConConsultas.hql    docker-compose-v2.yml    ejemploHive04.txt  pruebaPySpark.py
Parquet              Paso03_hql            Paso06_IncrementalVentas.py  docker-compose-v3.yml  pruebaScala.scala
Paso00.sh           Paso04.hql            README.md                docker-compose-v4.yml  pyspark-ETL.ipynb
```

Paso 06

Generamos el entorno compose-v4.

```
ubuntu@servidor_ubuntu:~/herramientas_big_data$ sudo docker-compose -f docker-compose-v4.yml up -d
Pulling spark-master (bde2020/spark-master:3.0.0-hadoop3.2)...
3.0.0-hadoop3.2: Pulling from bde2020/spark-master
21c83c524219: Pull complete
29291ffb8890: Pull complete
e77f5ea3e590: Pull complete
c7d4c1dd90df: Pull complete
d8d1136f7620: Pull complete
ca4884ea2462: Pull complete
040ed820351c: Pull complete
Digest: sha256:64a883c48788816d839f463299190d58b9c63dc0162b6c10de1b28a8ab83e4aa
Status: Downloaded newer image for bde2020/spark-master:3.0.0-hadoop3.2
Pulling spark-worker-1 (bde2020/spark-worker:3.0.0-hadoop3.2)...
3.0.0-hadoop3.2: Pulling from bde2020/spark-worker
21c83c524219: Already exists
29291ffb8890: Already exists
e77f5ea3e590: Already exists
c7d4c1dd90df: Already exists
d8d1136f7620: Already exists
ca4884ea2462: Already exists
fe0000ab8b42: Pull complete
Digest: sha256:fe69e7b8039abe4e397876c2a346d06cd6c1795193f0073db56842340a060944
Status: Downloaded newer image for bde2020/spark-worker:3.0.0-hadoop3.2
Creating historyserver ...
Creating namenode ...
Creating nodemanager ...
Creating resourcemanager ...
Creating datanode ...
Creating historyserver
Creating nodemanager
Creating namenode
Creating resourcemanager
Creating datanode ... done
Creating spark-master ...
Creating spark-master ... done
Creating spark-worker-1 ...
Creating spark-worker-1 ... done
ubuntu@servidor_ubuntu:~/herramientas_big_data$
```

Entramos al bash de spark-master y ejecutamos pyspark

Importamos `pyspark.sql.types` *, luego definimos el esquema y cargamos los datos del csv (previamente colocado en HDFS) en spark. Luego mostramos la tabla:

```
ubuntu@servidor_ubuntu:~/herramientas_big_data
Welcome to
version 3.0.0

Using Python version 2.7.18 (default, May  3 2020 22:58:12)
SparkSession available as 'spark'.
>>> from pyspark.sql.types import *
>>> flightSchema = StructType([StructField("DayoMonth", IntegerType(), False),StructField("DayoWeek", IntegerType(0), False),StructField("Carrier", StringType(), False),StructField("OriginAirportID", StringType(), False),StructField("DestAirportID", StringType(), False),StructField("DepDelay", IntegerType(), False),StructField("ArrDelay", IntegerType(), False)])
>>> flights = spark.read.csv('hdfs://namenode:9000/data/flights/raw-flight-data.csv', schema=flightSchema, header=True)
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
File "/usr/local/lib/python2.7/dist-packages/pyspark/sql/utils.py", line 55, in __init__
    self._jutils = self._load_jar()
File "/usr/local/lib/python2.7/dist-packages/pyspark/sql/utils.py", line 55, in _load_jar
    self._jutils = SparkUtils(jarPath='spark://lib/py4j-0.10.9-rtc.zip/py4j/java_gateway.py')
File "/usr/local/lib/python2.7/dist-packages/pyspark/sql/utils.py", line 137, in __call__
    raise_from(converted)
File "/usr/local/lib/python2.7/dist-packages/pyspark/sql/utils.py", line 137, in deco
    raise from converted
File "/usr/local/lib/python2.7/dist-packages/pyspark/sql/utils.py", line 33, in raise_from
    raise exception
py4j.protocol.Py4JJavaError: An error occurred while calling o111.schema.
:java.util.NoSuchElementException: Path does not exist: hdfs://namenode:9000/data/flights/raw-flight-data.csv;
>>> flights = spark.read.csv('hdfs://namenode:9000/data/flights/raw-flight-data.csv', schema=flightSchema, header=True)
>>> flights.show()

+-----+-----+-----+-----+-----+-----+-----+
|DayoMonth|DayoWeek|Carrier|OriginAirportID|DestAirportID|DepDelay|ArrDelay|
+-----+-----+-----+-----+-----+-----+-----+
|      1|       5|     DL|        11433|       13303|      -3|      1|
|      1|       5|     DL|        14869|       12478|      0|     -8|
|      1|       5|     DL|        14057|       14869|     -4|     -15|
|      1|       5|     DL|        11136|       14869|     20|      0|
|      1|       5|     DL|        11193|       12892|     -6|     -11|
|      1|       5|     DL|        10397|       15016|     -11|     -19|
|      1|       5|     DL|        15016|       10397|      0|     -11|
|      1|       5|     DL|        10397|       14869|     15|      24|
|      1|       5|     DL|        10397|       104869|     33|      34|
|      1|       5|     DL|        12781|       10397|     323|     323|
|      1|       5|     DL|        14107|       13487|     -7|     -13|
|      1|       5|     DL|        11433|       11298|     22|      41|
|      1|       5|     DL|        11298|       11433|     40|      20|
|      1|       5|     DL|        11353|       12451|     -2|     -7|
|      1|       5|     DL|        13357|       12451|     71|     75|
|      1|       5|     DL|        12451|       10397|     75|      57|
|      1|       5|     DL|        12953|       10397|     -11|     10|
|      1|       5|     DL|        11433|       12953|     -31|     -10|
|      1|       5|     DL|        10397|       14771|     31|     38|
|      1|       5|     DL|        13204|       10397|     6|     23|
+-----+-----+-----+-----+-----+-----+-----+
only showing top 20 rows
>>> 
```

Ahora ejecutamos flights.describe():

```
>>> flights.show()
+-----+-----+-----+-----+-----+
| DayofMonth | DayOfWeek | Carrier | OriginAirportID | DestAirportID | DepDelay | ArrDelay |
+-----+-----+-----+-----+-----+
| 19 | 5 | DL | 11433 | 13303 | -3 | 1 |
| 19 | 5 | DL | 14869 | 12478 | 0 | -8 |
| 19 | 5 | DL | 14057 | 14869 | -4 | -15 |
| 19 | 5 | DL | 15016 | 11433 | 28 | 24 |
| 19 | 5 | DL | 11193 | 12892 | -6 | -11 |
| 19 | 5 | DL | 10397 | 15016 | -11 | -19 |
| 19 | 5 | DL | 15016 | 10397 | 0 | -11 |
| 19 | 5 | DL | 10397 | 14869 | 15 | 24 |
| 19 | 5 | DL | 10397 | 10423 | 33 | 34 |
| 19 | 5 | DL | 11278 | 10277 | 322 | 322 |
| 19 | 5 | DL | 14107 | 13487 | -7 | -13 |
| 19 | 5 | DL | 11433 | 11298 | 22 | 41 |
| 19 | 5 | DL | 11298 | 11433 | 40 | 20 |
| 19 | 5 | DL | 11433 | 12892 | -2 | -7 |
| 19 | 5 | DL | 10397 | 12451 | 71 | 75 |
| 19 | 5 | DL | 12451 | 10397 | 75 | 57 |
| 19 | 5 | DL | 12953 | 10397 | -1 | 10 |
| 19 | 5 | DL | 11433 | 12953 | -3 | -10 |
| 19 | 5 | DL | 10397 | 14771 | 31 | 38 |
| 19 | 5 | DL | 13204 | 10397 | 8 | 25 |
+-----+-----+-----+-----+
only showing top 20 rows

>>> flights.describe()
+-----+-----+
| DayofMonth: string, DayOfWeek: string, Carrier: string, OriginAirportID: string, DestAirportID: string, DepDelay: string, ArrDelay: string |
+-----+-----+
```

Nos colocamos en la línea de comandos de spark-master y comenzamos scala:

Creamos la clase flightschema, cargamos los datos desde el csv almacenado en el hdfs y luego mostramos la tabla:

```

ubuntu@ubuntu: ~$ herramientas/bigdata
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://61.234.95.189:4040
Spark context available as 'sc' (master = spark://spark-master:7077, app id = app-20231124190135-0001).
Spark session available as 'spark'.
Welcome to

version 3.0.0

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 1.8.0_252)
Type in expressions to have them evaluated.
Type :help for more information.

scala> case class flightSchema(DayOfMonth:String, DayOfWeek:String, Carrier:String, OriginAirportID:String, DestAirportID:String, DepDelay:String, ArrDelay:String)
defined class flightSchema

scala> val flights = spark.read.format("csv").option("sep", ",").option("header", "true").load("hdfs://namenode:9000/data/flights/raw-flight-data.csv").as[flightSchema]
flights: org.apache.spark.sql.Dataset[flightSchema] = [DayOfMonth: string, DayOfWeek: string ... 5 more fields]

scala> flights.show()
+-----+-----+-----+-----+-----+
|DayOfMonth|DayOfWeek|Carrier|OriginAirportID|DestAirportID|DepDelay|ArrDelay|
+-----+-----+-----+-----+-----+
|19|1|DL|11433|13303|-3|1|  

|19|5|DL|14869|124781|0|-8|  

|19|5|DL|140571|14869|41|-15|  

|19|5|DL|15016|11433|28|24|  

|19|5|DL|11193|12892|6|-11|  

|19|5|DL|10397|15016|-1|-19|  

|19|5|DL|15016|10397|0|-11|  

|19|5|DL|10397|14869|13|24|  

|19|5|DL|10397|11433|33|34|  

|19|5|DL|11278|10397|23|322|  

|19|5|DL|14107|13487|-7|13|  

|19|5|DL|11433|11298|22|41|  

|19|5|DL|11298|11433|40|20|  

|19|5|DL|11433|12892|-2|-7|  

|19|5|DL|10397|12451|71|75|  

|19|5|DL|12451|10397|75|57|  

|19|5|DL|12451|11433|-1|-10|  

|19|5|DL|11433|12953|-3|-10|  

|19|5|DL|10397|14771|31|38|  

|19|5|DL|13204|10397|8|25|
+-----+-----+-----+-----+-----+
only showing top 20 rows

scala>

```

KAFKA

Levantamos el entorno con Docker-compose-kafka:

```
ubuntu@servidor_ubuntu:/heramientas_big_data$ ls
datasets          Parque  GeneracionVentasPorDia.py  docker-compose-v3.yml  docker-compose.yml  hadoop.env    pruehaPySpark.py
Generacion_Ventas.lynnb  Passo0.sh  IncrementalVentas.py  docker-compose-v4.yml  ejemplodeNeo4j.txt  hbase-distributed-local.env  pruehaScala.scala
Mongo             Passo0.py  docker-compose-kafka.yml  docker-compose-v5.yml  hadoop-hive.env   iris.hql      pyspark-ETL.ipynb
ubuntu@servidor_ubuntu:/heramientas_big_data$ sudo docker-compose -f docker-compose-kafka.yml up -d
Pulling zookeeper (wurstmeister/zookeeper:3.4.6)...
3.4.6: Pulling from wurstmeister/zookeeper
Digest: sha256:c00a2a0a2a0a2a0a2a0a2a0a2a0a2a0a
Status: Downloaded newer image for wurstmeister/zookeeper:3.4.6
Pulling kafka (wurstmeister/kafka-manager:latest)...
latest: Pulling from sheepkiller/kafka-manager
Digest: sha256:29469bb80a614ed3128969b5b535c480e84704d826cdf73e790b5a6e63fc
Status: Downloaded newer image for wurstmeister/zookeeper:3.4.6
Pulling kafka (wurstmeister/kafka:2.12-2.1.1)...
2.1.1: Pulling from wurstmeister/kafka
Digest: sha256:e15fb3b99438aha2d5fdcb3fb750a5990ba9260c8fb3fd29c7e776e8c150518b78
Status: Downloaded newer image for sheepkiller/kafka-manager:latest
Pulling kafka (wurstmeister/kafka:2.12-2.1.1)...
2.1.1: Pulling from wurstmeister/kafka
Digest: sha256:615fb3b99438aha2d5fdcb3fb750a5990ba9260c8fb3fd29c7e776e8c150518b78
Status: Downloaded newer image for sheepkiller/kafka-manager:latest
Pulling kafka (wurstmeister/kafka:2.12-2.2.1)...
2.2.1: Pulling from wurstmeister/kafka
Digest: sha256:1ab3df91a1e807e013a10ld98c1df41d19a4ebad3ab98266c7902866fbfdf5
Status: Downloaded newer image for wurstmeister/kafka:2.12-2.2.1
Creating zookeeper_container ...
Creating zookeeper_container ... done
Creating kafka_container ...
Creating kafka_manager ...
Creating kafka_manager ... done
Creating kafka_container ...
Creating kafka_container ... done
Creating kafka_manager ...
Creating kafka_manager ... done
ubuntu@servidor_ubuntu:/heramientas_big_data$
```

Entramos a Kafka container:

```
ubuntu@servidor_ubuntu:~/herramientas_big_data
ubuntu@servidor_ubuntu:~/herramientas_big_data$ sudo docker exec -it kafka_container bash
bash-4.4# ls
bash-4.4# cd /opt/kafka/bin
bash-4.4# ls
connect-distributed.sh          kafka-consumer-groups.sh      kafka-preferred-replica-election.sh  kafka-streams-application-reset.sh  zookeeper-server-start.sh
connect-explain.sh               kafka-consumer-perf-test.sh  kafka-producer-perf-test.sh        kafka-rogue.sh                      zookeeper-server-stop.sh
kafka-acks.sh                   kafka-delegation-tokens.sh   kafka-reassign-partitions.sh       kafka-verifiable-consumer.sh      zookeeper-shell.sh
kafka-broker-api-versions.sh    kafka-delete-record.sh       kafka-replica-validation.sh     kafka-verifiable-producer.sh
kafka-configs.sh                kafka-dump-log.sh          kafka-run-class.sh              trogrod.sh
kafka-console-consumer.sh        kafka-log-dir.sh          kafka-server-start.sh         windows
kafka-console-producer.sh       kafka-mirror-maker.sh      kafka-server-stop.sh        zookeeper-security-migration.sh
bash-4.4#
```

```
sh kafka-topics.sh --create --bootstrap-server kafka:9092 --replication-factor 1 --partitions 100 -  
-topic demo
```

Con el comando anterior creamos un nuevo tópico llamado topic demo. Con el siguiente comando listamos los tópicos creados para ver si efectivamente se creó:

```
sh kafka-topics.sh --list --bootstrap-server kafka:9092
```

Luego con el comando siguiente describimos el tópico topic demo, donde vemos que hay 100 particiones (imagen cortada)

```
sh kafka-topics.sh --describe --bootstrap-server kafka:9092 --topic demo
```

```
bash-4.4# sh kafka-topics.sh --create --bootstrap-server kafka:9092 --replication-factor 1 --partitions 100 --topic demo
bash-4.4# sh kafka-topics.sh --list --bootstrap-server kafka:9092
demo
bash-4.4# sh kafka-topics.sh --describe --bootstrap-server kafka:9092 --topic demo
Topic: demo      PartitionCount:100      ReplicationFactor:1      Configs:segment.bytes=1073741824
Topic: demo      Partition: 0      Leader: 1001      Replicas: 1001 Isr: 1001
Topic: demo      Partition: 1      Leader: 1001      Replicas: 1001 Isr: 1001
Topic: demo      Partition: 2      Leader: 1001      Replicas: 1001 Isr: 1001
Topic: demo      Partition: 3      Leader: 1001      Replicas: 1001 Isr: 1001
Topic: demo      Partition: 4      Leader: 1001      Replicas: 1001 Isr: 1001
Topic: demo      Partition: 5      Leader: 1001      Replicas: 1001 Isr: 1001
Topic: demo      Partition: 6      Leader: 1001      Replicas: 1001 Isr: 1001
Topic: demo      Partition: 7      Leader: 1001      Replicas: 1001 Isr: 1001
Topic: demo      Partition: 8      Leader: 1001      Replicas: 1001 Isr: 1001
Topic: demo      Partition: 9      Leader: 1001      Replicas: 1001 Isr: 1001
Topic: demo      Partition: 10     Leader: 1001      Replicas: 1001 Isr: 1001
Topic: demo      Partition: 11     Leader: 1001      Replicas: 1001 Isr: 1001
Topic: demo      Partition: 12     Leader: 1001      Replicas: 1001 Isr: 1001
Topic: demo      Partition: 13     Leader: 1001      Replicas: 1001 Isr: 1001
Topic: demo      Partition: 14     Leader: 1001      Replicas: 1001 Isr: 1001
Topic: demo      Partition: 15     Leader: 1001      Replicas: 1001 Isr: 1001
Topic: demo      Partition: 16     Leader: 1001      Replicas: 1001 Isr: 1001
Topic: demo      Partition: 17     Leader: 1001      Replicas: 1001 Isr: 1001
Topic: demo      Partition: 18     Leader: 1001      Replicas: 1001 Isr: 1001
Topic: demo      Partition: 19     Leader: 1001      Replicas: 1001 Isr: 1001
Topic: demo      Partition: 20     Leader: 1001      Replicas: 1001 Isr: 1001
Topic: demo      Partition: 21     Leader: 1001      Replicas: 1001 Isr: 1001
Topic: demo      Partition: 22     Leader: 1001      Replicas: 1001 Isr: 1001
```

```
sh kafka-console-consumer.sh --bootstrap-server kafka:9092 --topic demo --from-beginning
```

Con este comando indicamos consumir mensajes del tópico topic demo, desde el principio

```
ubuntu@servidor_ubuntu: ~
ubuntu@servidor_ubuntu:~$ sudo docker exec -it kafka_container bash
[sudo] password for ubuntu:
bash-4.4# cd /opt/kafka/bin
bash-4.4# sh kafka-console-consumer.sh --bootstrap-server kafka:9092 --topic demo --from-beginning
```

En una consola aparte abrimos el bash de cakfca_container y ejecutamos el productor:

```
ubuntu@servidor_ubuntu: ~
bash-4.4# cd /opt/kafka/bin
bash-4.4# ls
connect-distributed.sh          kafka-consumer-groups.sh    kafka-preferred-replica-election.sh  kafka-streams-application-reset.sh  zookeeper-server-start.sh
connect-standalone.sh           kafka-consumer-perf-test.sh  kafka-producer-perf-test.sh        kafka-topics.sh                zookeeper-server-stop.sh
kafka-acls.sh                  kafka-delegation-tokens.sh  kafka-reassign-partitions.sh       kafka-verifiable-consumer.sh   zookeeper-shell.sh
kafka-broker-api-versions.sh   kafka-delete-records.sh   kafka-replica-verification.sh    kafka-verifiable-producer.sh
kafka-configs.sh               kafka-dump-log.sh        kafka-run-class.sh              kafka-verifier.sh
kafka-console-consumer.sh       kafka-log-dir.sh       kafka-server-start.sh         trogdror.sh
kafka-console-producer.sh       kafka-mirror-maker.sh  kafka-server-stop.sh        windows
kafka-console-producer.sh       kafka-mirror-maker.sh  kafka-server-stop.sh        zookeeper-security-migration.sh
bash-4.4# sh kafka-console-producer.sh --broker-list localhost:9092 --topic demo
>Esto es una prueba 1
```

En la imagen anterior se observa que enviamos el mensaje 'Esto es una prueba 1'

Ahora vamos del lado del consumidor y observamos que aparece el mensaje:

```
ubuntu@servidor_ubuntu: ~
ubuntu@servidor_ubuntu:~$ sudo docker exec -it kafka_container bash
[sudo] password for ubuntu:
bash-4.4# cd /opt/kafka/bin
bash-4.4# sh kafka-console-consumer.sh --bootstrap-server kafka:9092 --topic demo --from-beginning
Esto es una prueba 1
```

Enviamos un nuevo mensaje:

```

ubuntu@servidor_ubuntu: ~
bash-4.4# cd /opt/kafka/bin
bash-4.4# ls
connect-distributed.sh          kafka-consumer-groups.sh    kafka-preferred-replica-election.sh  kafka-streams-application-reset.sh   zookeeper-server-start.sh
connect-singleton.sh            kafka-consumer-perf-test.sh  kafka-primer-perf-test.sh           kafka-topics.sh                   zookeeper-server-stop.sh
KafkaACLs.sh                   kafka-delegation-tokens.sh  kafka-partition-partitions.sh        kafka-verifiable-consumer.sh       zookeeper-shell.sh
kafka-broker-api-versions.sh   kafka-delete-records.sh   kafka-replica-validation.sh        kafka-verifiable-producer.sh      trogrodor.sh
kafka-configs.sh               kafka-dump-log.sh       kafka-run-class.sh                 windows
kafka-console-consumer.sh       kafka-log-dir.sh       kafka-server-start.sh             zookeeper-security-migration.sh
kafka-console-producer.sh      kafka-mirror-maker.sh  kafka-server-stop.sh
bash-4.4# sh kafka-console-producer.sh --broker-list localhost:9092 --topic demo
>Esto es una prueba 1
>Estamos probando Kafka en el módulo 4 de Henry :D

```

Vemos que aparece del lado del consumidor:

```

ubuntu@servidor_ubuntu: ~
ubuntu@servidor_ubuntu:~$ sudo docker exec -it kafka_container bash
[sudo] password for ubuntu:
bash-4.4# cd /opt/kafka/bin
bash-4.4# sh kafka-console-consumer.sh --bootstrap-server kafka:9092 --topic demo --from-beginning
Esto es una prueba 1
Estamos probando Kafka en el módulo 4 de Henry :D

```

Ahora creamos un cluster en Databricks para poder usar spark:

The screenshot shows the Databricks cluster configuration interface for the 'Integrador M4' cluster. The configuration tabs include Configuration, Cuadernos (0), Bibliotecas, Log de eventos, IU de Spark, Logs del driver, Métricas, Aplicaciones, and IU de cómputo Spark - Máster. The configuration section shows the Databricks Runtime version as 12.2 LTS (includes Apache Spark 3.3.2, Scala 2.12). The driver type is set to Community Optimized with 15.3 GB of memory, 2 cores, and 1 DBU. The instance section specifies 15 GB of free memory and notes that the cluster will automatically terminate after one or two hours of inactivity. The availability zone is set to us-west-2b.

Creamos el stream y vemos el esquema:

databricks

Cuaderno sin nombre 2023-11-26 14:28:34 Python ⚡

Archivo Editar Ver Ejecutar Ayuda Última edición hace 7 minutos Provide feedback

Interrumpir Integrador M4 Compartir Publicar

Cmd 1

```
1 %scala
2 val df = spark.readStream
3   .format("kafka")
4     .option("kafka.bootstrap.servers", "192.168.101.11:9092")
5     .option("subscribe", "demo")
6     .option("startingOffsets", "earliest") // From starting
7   .load()
```

df: org.apache.spark.sql.DataFrame = [key: binary, value: binary ... 5 campos adicionales]

df: org.apache.spark.sql.DataFrame = [key: binary, value: binary ... 5 more fields]

Comando ejecutado en 1,98 segundos -- por altercaimi@hotmail.com el 26/11/2023, 14:38:48 en <Integrador M4>

Cmd 2

```
1 %scala
2 df.printSchema()
```

root

```
 |-- key: binary (nullable = true)
|   |-- value: binary (nullable = true)
|   |-- topic: string (nullable = true)
|   |-- partition: integer (nullable = true)
|   |-- offset: long (nullable = true)
|   |-- timestamp: timestamp (nullable = true)
|   |-- timestampType: integer (nullable = true)
```

Comando ejecutado en 0,46 segundos -- por altercaimi@hotmail.com el 26/11/2023, 14:38:56 en <Integrador M4>