PRÁCTICA Nº 2.9
Práctica guiada Oozie

- Nombre y apellidos: Alvaro Lucio-Villegas de Cea

Índice

1. Definir origen de datos

Los datos se han generado utilizando https://www.mockaroo.com/

Los ficheros clientes.sql y compras.csv se encuentran en

/home/cloudera/Desktop/BLQ2-Tarea1

1.1. Subir compras.csv a HDFS

>hdfs dfs -mkdir blq2-tarea1

>hdfs dfs -put /home/cloudera/Desktop/BLQ2-Tarea1/compras.csv blq2-tarea1

```
[cloudera@quickstart ~]$ hdfs dfs -mkdir blq2-tarea1
[cloudera@quickstart ~]$ hdfs dfs -put /home/cloudera/Downloads/c
client.asc    clientes.sql  compras.csv
[cloudera@quickstart ~]$ hdfs dfs -put /home/cloudera/Downloads/compras.csv
[cloudera@quickstart ~]$ hdfs dfs -put /home/cloudera/Downloads/compras.csv blq2-tarea1
[cloudera@quickstart ~]$
```

1.2. Crear tabla clientes y cargar datos

>mysql -u root -p (passwd cloudera)

>source /home/cloudera/Dowloads/clientes.sql

```
Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.00 sec)

Query OK, 1 row affected (0.01 sec)

mysql> source /home/cloudera/Downloads/clientes.sql
```

2. Tratamiento Sqoop

Origen: tabla clientes de la base de datos prueba

Destino: /user/cloudera/blq2-tarea1/tabla_clientes

2.1. Definir sentencia Sqoop

```
sqoop import \
--connect jdbc:mysql://localhost/practica_oozie \
--username root \
--password cloudera \
--table clientes \
--target-dir blq2-tarea1/tabla_clientes
```

```
[cloudera@quickstart ~]$ sqoop import \
> --connect jdbc:mysql://localhost/practica_oozie \
> --username root \
> --password cloudera \
> --table clientes \
> --target-dir blq2-tarea1/tabla_clientes
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
23/05/03 10:36:19 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
23/05/03 10:36:19 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
23/05/03 10:36:19 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
23/05/03 10:36:19 INFO tool.CodeGenTool: Beginning code generation
23/05/03 10:36:20 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `clientes` AS t LIMIT 1
23/05/03 10:36:20 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `clientes` AS t LIMIT 1
23/05/03 10:36:20 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-mapreduce
Note: /tmp/sqoop-cloudera/compile/90f8b00cda09e6b30b8a9063d59e9c25/clientes.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
23/05/03 10:36:21 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-cloudera/compile/90f8b00cda09e6b30b8a9063d59e9c25/
clientes.jar
23/05/03 10:36:21 WARN manager.MySQLManager: It looks like you are importing from mysql.
23/05/03 10:36:21 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
23/05/03 10:36:21 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
23/05/03 10:36:21 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
23/05/03 10:36:21 INFO mapreduce.ImportJobBase: Beginning import of clientes
23/05/03 10:36:21 INFO Configuration.deprecation: mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
23/05/03 10:36:21 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
23/05/03 10:36:22 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
23/05/03 10:36:22 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
23/05/03 10:36:24 INFO db.DBInputFormat: Using read commited transaction isolation
```

```
        Job Counters
                Launched map tasks=4
                Other local map tasks=4
                Total time spent by all maps in occupied slots (ms)=18964
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=18964
                Total vcore-milliseconds taken by all map tasks=18964
                Total megabyte-milliseconds taken by all map tasks=19419136
        Map-Reduce Framework
                Map input records=1000
                Map output records=1000
                Input split bytes=408
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=606
                CPU time spent (ms)=3570
                Physical memory (bytes) snapshot=883064832
                Virtual memory (bytes) snapshot=6357393408
                Total committed heap usage (bytes)=1092616192
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=54433
23/05/03 10:36:41 INFO mapreduce.ImportJobBase: Transferred 53.1572 KB in 19.3736 seconds (2.7438 KB/sec)
23/05/03 10:36:41 INFO mapreduce.ImportJobBase: Retrieved 1000 records.
[cloudera@quickstart ~]$
```

3. Tratamiento Pig

Carga csv -> obtiene columnas id_cliente y precio -> elimina filas con precio en blanco -> almacena en blq2-tarea1/tabla_ventas

3.1. Crear fichero script .pig (LOAD-Tratamientos-STORE)

```
ventas = LOAD 'blq2-tarea1/compras.csv'
USING PigStorage(',') AS (
id:int,
id_cliente:int,
producto:chararray,
precio:float
);
ventas_id_y_precio = FOREACH ventas GENERATE id_cliente, precio;
ventas_id_y_precio_limpio = FILTER ventas_id_y_precio BY ( precio is
not null);

# DUMP ventas_id_y_precio_limpio;
STORE ventas_id_y_precio_limpio INTO
'/user/cloudera/blq2-tarea1/tabla_ventas' USING PigStorage(';');
```

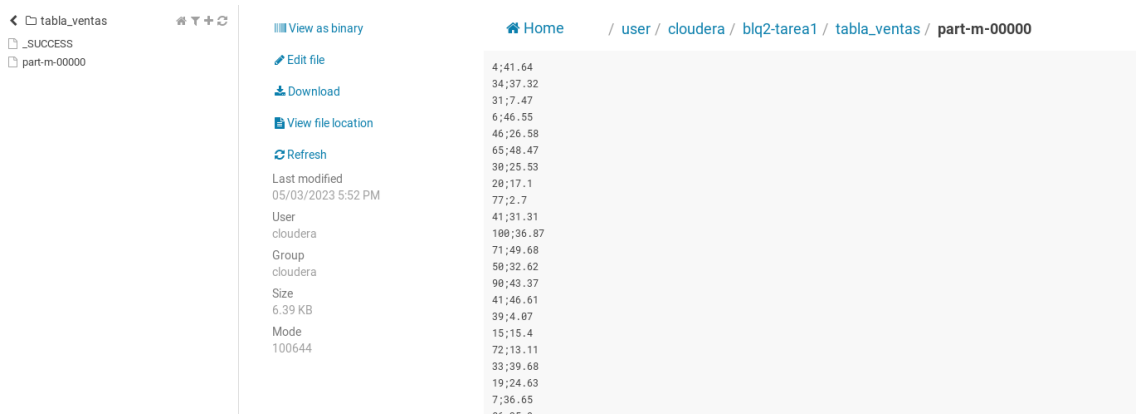
script.pig

```
ventas = LOAD 'blq2-tarea1/compras.csv'
USING PigStorage(',') AS (
id:int,
id_cliente:int,
producto:chararray,
precio:float
);
ventas_id_y_precio = FOREACH ventas GENERATE id_cliente, precio;
ventas_id_y_precio_limpio = FILTER ventas_id_y_precio BY ( precio is
not null);

# DUMP ventas_id_y_precio_limpio;
STORE ventas_id_y_precio_limpio INTO
'/user/cloudera/blq2-tarea1/tabla_ventas' USING PigStorage(';');
```

Final del comando de ejecución del script.

```
Successfully stored 758 records (6539 bytes) in: "/user/cloudera/blq2-tarea1/tabla_ventas"

Counters:
Total records written : 758
Total bytes written : 6539
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1681898044415_0029


2023-05-03 10:52:43,238 [main] WARN  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Encountered Warning FIELD_DISCARDED_TYPE_CONVERSION_FAILED 194 time(s).
2023-05-03 10:52:43,238 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
[cloudera@quickstart ~]$
```

Resultado en HDFS



4. Tratamiento Hive

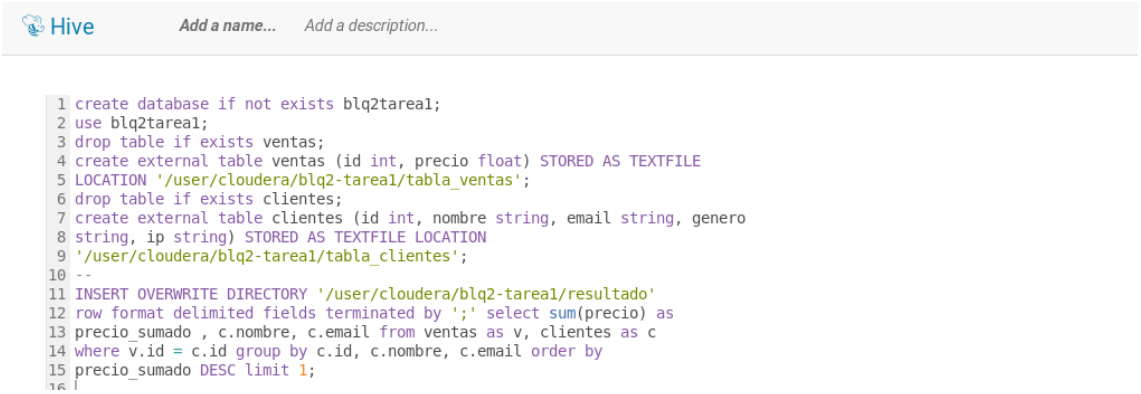
Origen de información:

/user/cloudera/blq2-tarea1/tabla_clientes

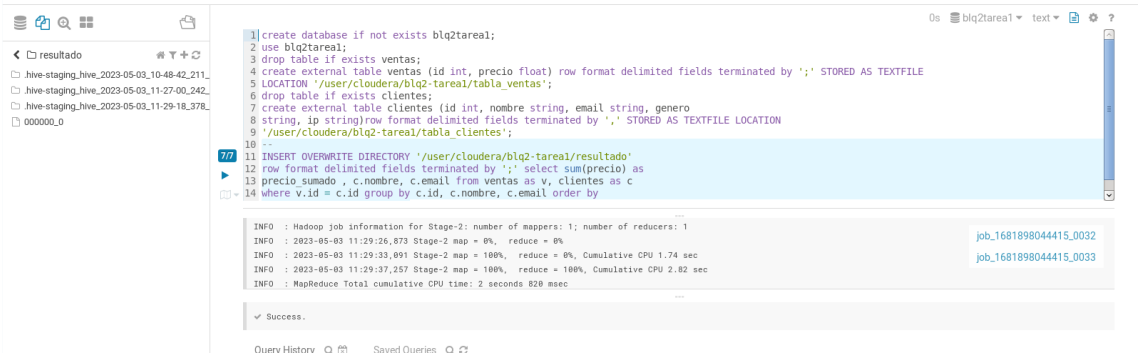
/user/cloudera/blq2-tarea1/tabla_ventas

4.1. Script hive

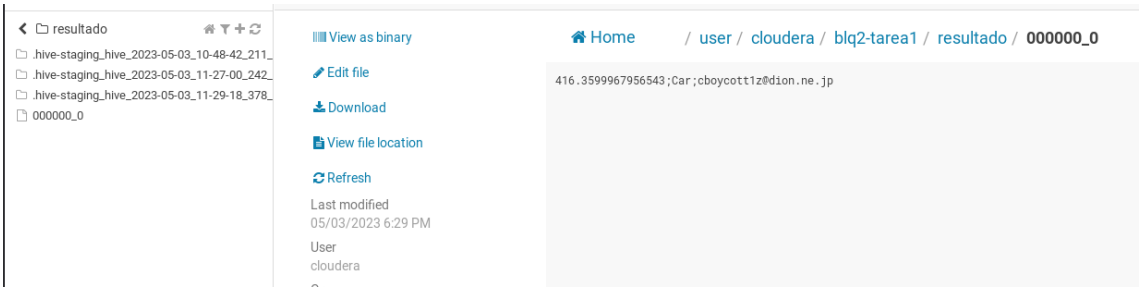
```
create database if not exists blq2tarea1;
use blq2tarea1;
drop table if exists ventas;
create external table ventas (id int, precio float) STORED AS TEXTFILE
LOCATION '/user/cloudera/blq2-tarea1/tabla_ventas';
drop table if exists clientes;
create external table clientes (id int, nombre string, email string, genero
string, ip string) STORED AS TEXTFILE LOCATION
'/user/cloudera/blq2-tarea1/tabla_clientes';
--
INSERT OVERWRITE DIRECTORY '/user/cloudera/blq2-tarea1/resultado'
row format delimited fields terminated by ';' select sum(precio) as
precio_sumado , c.nombre, c.email from ventas as v, clientes as c
where v.id = c.id group by c.id, c.nombre, c.email order by
precio_sumado DESC limit 1;
```



Ejecución en HUE del Script



Resultado en HDFS

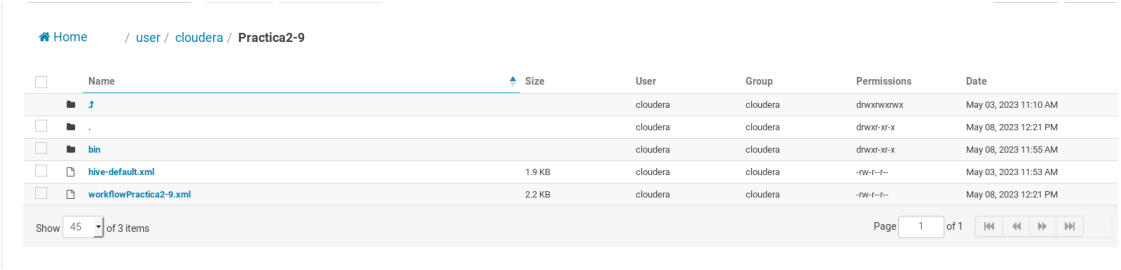


5. Unir todo con Oozie

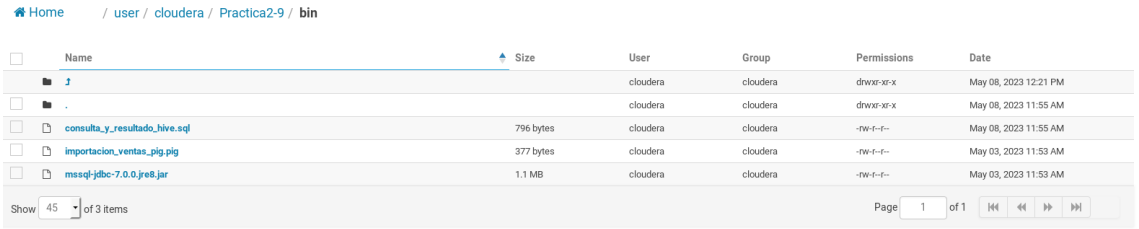
5.1. Crear una carpeta en LOCAL y en HDFS (se replicará la información)

5.2. Unir todos los scripts anteriores en subcarpeta bin

Será necesario crear un directorio en HDFS que llamaremos “Practica2-9” y pondremos el “hive-default.xml” a la misma altura que el xml.Este archivo le deberemos de cambiar el nombre de hive-site.xml con el nombre hive-dafault.xml.Este fichero se encuentra en en “/etc/hive/conf.dist/hive-site.xml”



Dentro del directorio /bin tendremos los scripts que creamos anteriormente.



5.3. Definir un workflow

```
<workflow-app name='practica_guiada' xmlns="uri:oozie:workflow:0.1">
	<start to="forking" />
	<fork name="forking">
		<path start="importacion_clientes_sqoop" />
		<path start="importacion_ventas_pig" />
	</fork>
	<action name="importacion_clientes_sqoop">
		<sqoop xmlns="uri:oozie:sqoop-action:0.2">
			<job-tracker>${jobTracker}</job-tracker>
			<name-node>${nameNode}</name-node>
			<prepare>
				<delete
path="${nameNode}/user/cloudera/blq2-tarea1/tabla_clientes" />
			</prepare>
			<configuration>
				<property>

<name>mapred.job.queue.name</name>
					<value>${queueName}</value>
				</property>
			</configuration>
			<command>import --connect
jdbc:mysql://localhost/practica_oozie --username root --password cloudera
--table clientes --target-dir /user/cloudera/blq2-tarea1/tabla_clientes -m
1</command>
		</sqoop>
		<ok to="joining" />
		<error to="kill" />
	</action>
	<action name="importacion_ventas_pig">
		<pig>
			<job-tracker>${jobTracker}</job-tracker>
			<name-node>${nameNode}</name-node>
			<prepare>
				<delete
path="${nameNode}/user/cloudera/blq2-tarea1/tabla_ventas" />
			</prepare>
			<configuration>
				<property>

<name>mapred.job.queue.name</name>
					<value>root.default</value>
				</property>
			</configuration>
```

```
                        <script>
                                bin/importacion_ventas_pig.pig
                        </script>
                </pig>
                <ok to="joining" />
                <error to="kill" />
        </action>
        <join name="joining" to="consulta_y_resultado_hive" />
        <action name="consulta_y_resultado_hive">
                <hive xmlns="uri:oozie:hive-action:0.2">
                        <job-tracker>${jobTracker}</job-tracker>
                        <name-node>${nameNode}</name-node>
                        <configuration>
                                <property>

<name>mapred.job.queue.name</name>
                                        <value>${queueName}</value>
                                </property>
                    <property>
                        <name>ozzie.hive.default</name>

<value>${nameNode}/user/cloudera/Practica2-9/hive-default.xml</value>
                    </property>
                        </configuration>
                        <script>
                                bin/consulta_y_resultado_hive.sql
                        </script>
                </hive>
                <ok to='end' />
                <error to='kill' />
        </action>
        <kill name='kill'>
                <message>
                        Se rompio, mensaje de error
[${wf:errorMessage(wf:lastErrorNode())}]
                </message>
        </kill>
        <end name='end' />
</workflow-app>
```

5.4. Definir jobPractica2-9.properties

```
nameNode=hdfs://localhost:8020

jobTracker=localhost:8032

queueName=default

oozie.use.system.libpath=true

oozie.wf.application.path=${nameNode}/user/${user.name}/Practica2-9/workflowPractica2-9.xml
```

5.5. Lanzar Oozie

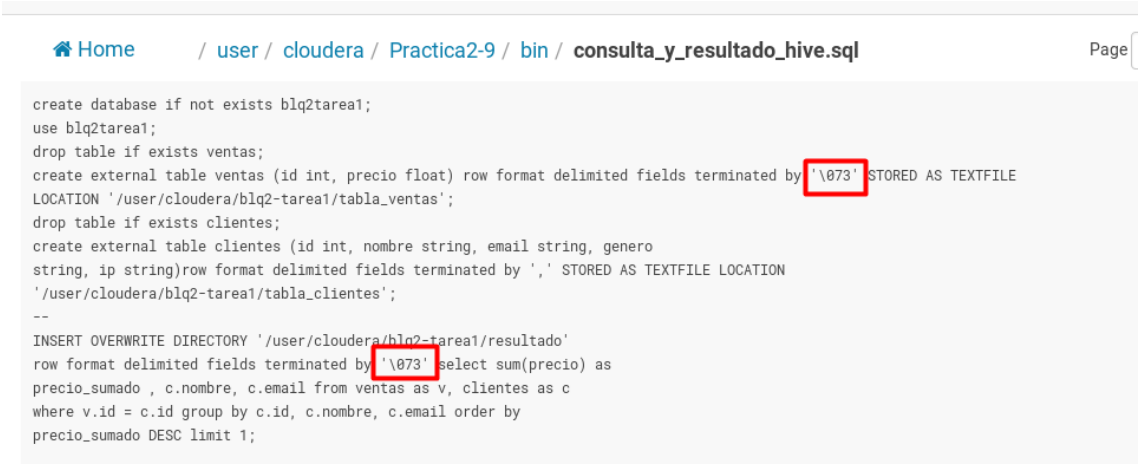
```
export OOZIE_URL="http://localhost:11000/oozie"

oozie job -config jobPractica2-9.properties -run
```

```
[cloudera@quickstart map-reduce]$ oozie job -config jobPractica2-9.properties -run
job: 0000003-230419025730046-oozie-oozi-W
[cloudera@quickstart map-reduce]$
```

Posibles problemas y soluciones

Fichero SQL

En este fichero me he encontrado el error de codificación ya que el workflow no identifica el carácter ";" y se debe de reemplazar por los caracteres "\073" que es el mismo carácter pero con la codificación correspondiente.



Fichero Workflow.xml

En este fichero debemos agregar al workflow una propiedad en la configuracion de la cola de los procesos para que no se sature y funcione correctamente.

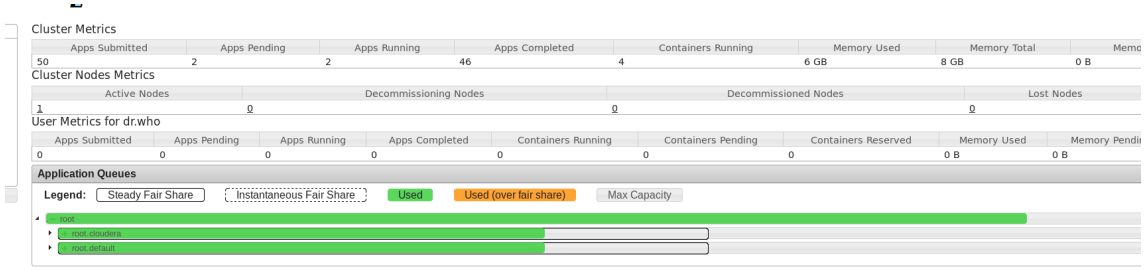
Para ello agregaremos el siguiente comando ya sea en la parte del workflow de pig como de Hive.Pero solo en una de ellas.

Lo que estamos haciendo es mandar ese job a otra cola diferente al job anterior, en este caso lo mandamos a la cola del usuario root.

```
<configuration>
        <property>
                <name>mapred.job.queue.name</name>
                <value>root.default</value>
        </property>
</configuration>
```

```
</action>
<action name="importacion_ventas_pig">
        <pig>
                <job-tracker>${jobTracker}</job-tracker>
                <name-node>${nameNode}</name-node>
                <prepare>
                        <delete path="${nameNode}/user/cloudera/blq2-tarea1/tabla_ventas" />
                </prepare>
                <configuration>
                        <property>
                                <name>mapred.job.queue.name</name>
                                <value>root.default</value>
                        </property>
                </configuration>
                <script>
                        bin/importacion_ventas_pig.pig
                </script>
```

Podemos observar cómo se están realizando los dos trabajos de forma paralela en el panel de control de yarm.



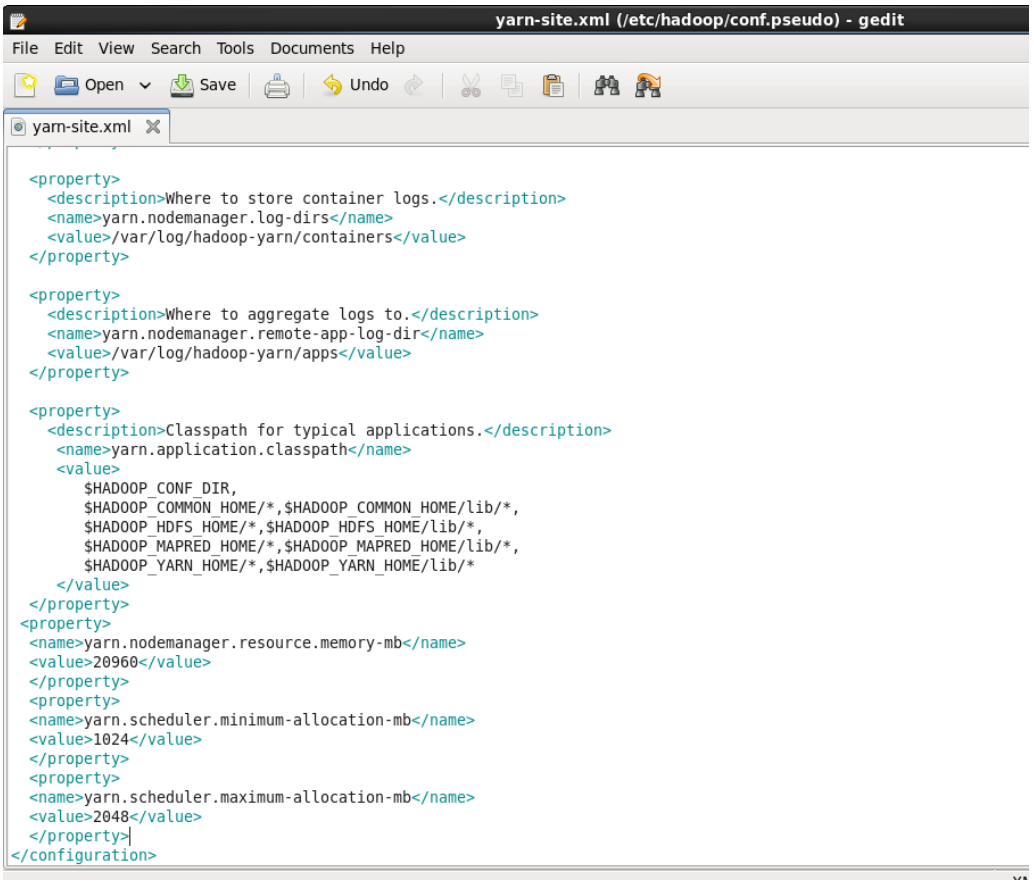
También nos sucederá que el proceso se queda procesando continuamente y esto sucede por la memoria asignada a los procesos. Para solucionar este problema tendremos que dirigirnos al fichero de configuración de yarn, para asignarle una nueva configuración.

El fichero de configuración se encuentra en :"etc/hadoop/conf.pseudo/yarn-site.xml"

Una vez localizado el fichero será necesario agregar estas líneas de comando.

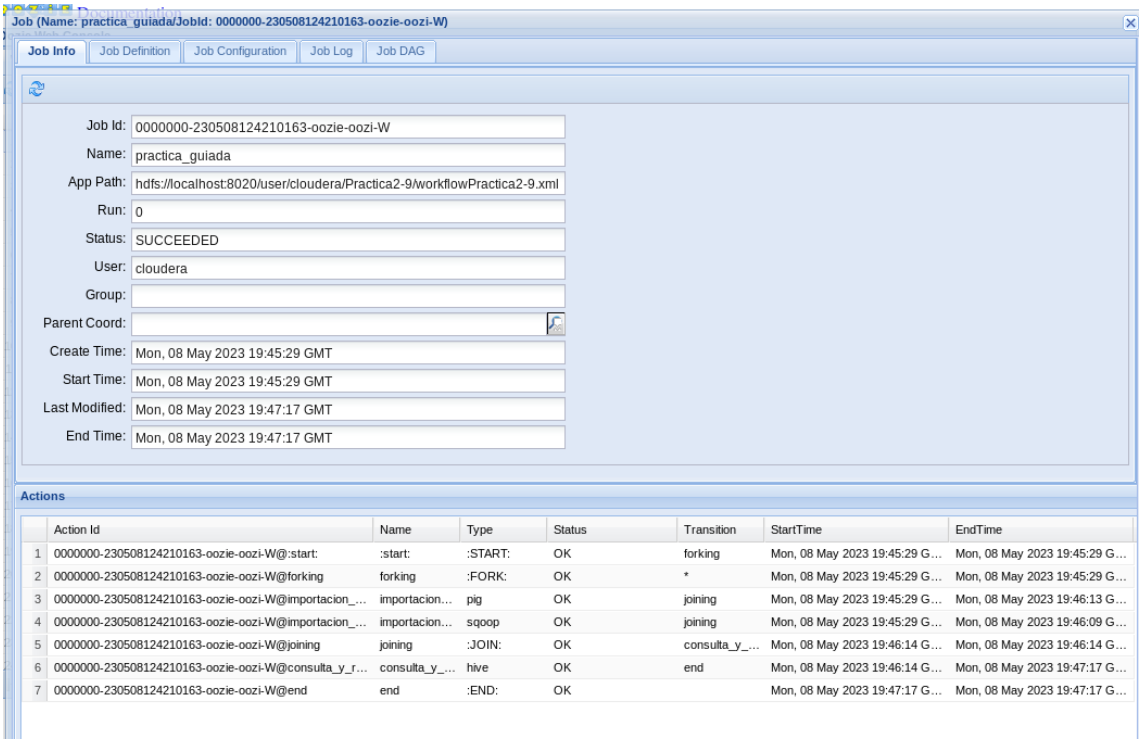
```
<property>
  <name>yarn.nodemanager.resource.memory-mb</name>
  <value>20960</value>
</property>
<property>
  <name>yarn.scheduler.minimum-allocation-mb</name>
  <value>1024</value>
</property>
<property>
```

```
 <name>yarn.scheduler.maximum-allocation-mb</name>
 <value>2048</value>
</property>
```



Resultado Final

Observamos el job y los subprocesos que genera con sus respectivos estados.



Nos dirigimos a la carpeta “blq2-tarea1/resultado/0000_0” en este archivo podemos observar el resultado final del workflow.