4. Spark

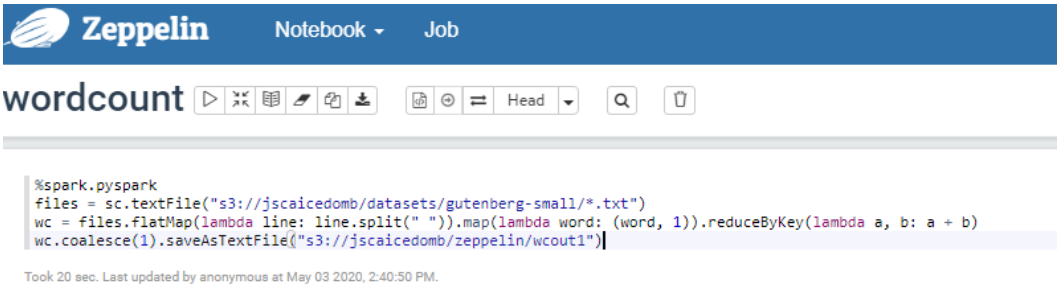
miércoles, 29 de abril de 2020 15:59

Creo una carpeta notebooks en mi bucket de s3 para los notebooks.

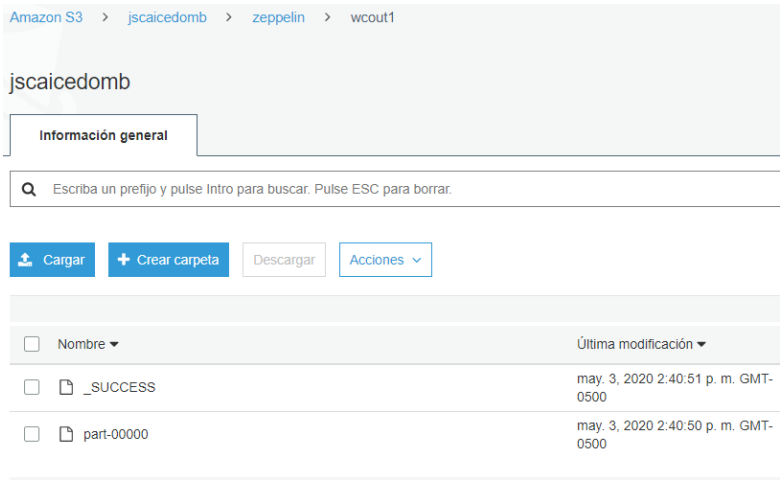
Realizo el tutorial de zeppelin en el clúster

Wordcount en zeppelin

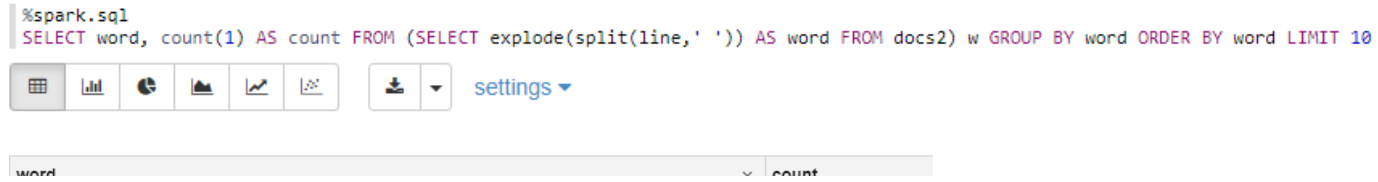
Creamos un notebook y realizamos el wordcount con datos en s3 y lo guardamos en s3



Verificamos en s3



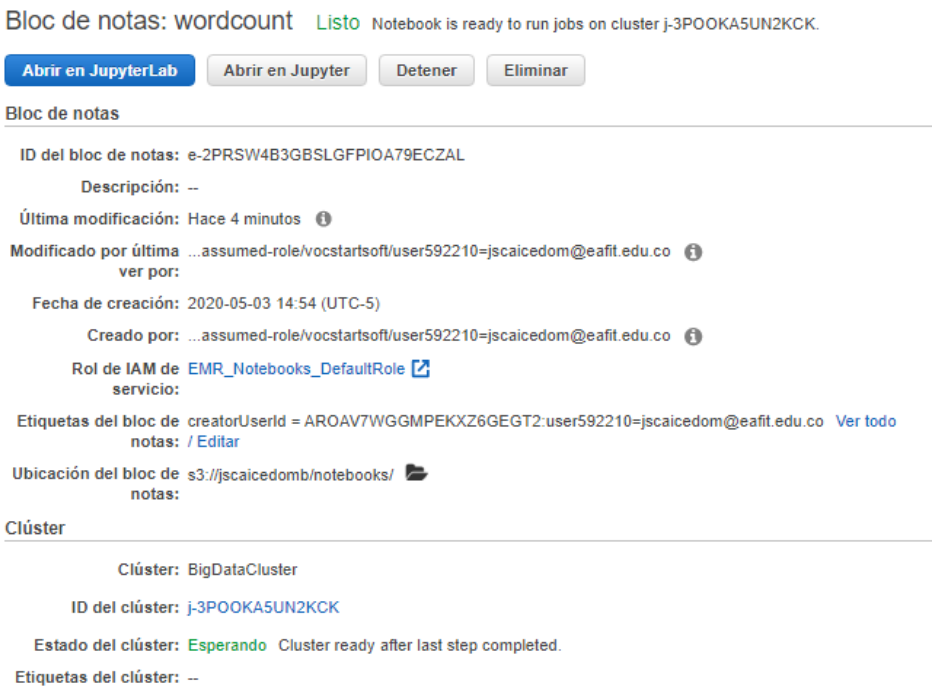
Wordcount con sql en zeppelin



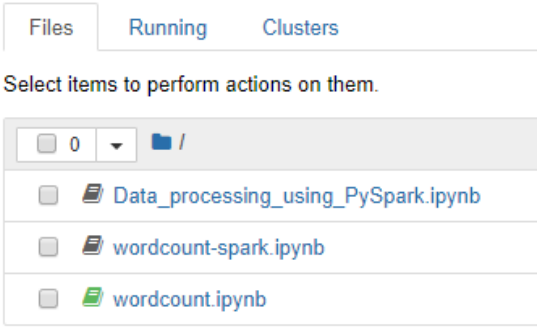
	27298
"	8
""ARTICLE	1
""But	1
""FROM	1
""KANSAS	1
""Nothing	1
""One	1

EMR NOTEBOOK

Creo un clúster y un notebook asociado a él con ubicación en la carpeta que creé para los notebooks



Lo abro y subo los ejemplos



Wordcount_spark.ipynb

Archivo completo

Data_processing_using_PySpark.ipynb

Archivo completo

Primero configuramos python 3

```
In [1]: %%configure -f
{ "conf":{
"spark.pyspark.python": "python3",
"spark.pyspark.virtualenv.enabled": "true",
"spark.pyspark.virtualenv.type":"native",
"spark.pyspark.virtualenv.bin.path":"/usr/bin/virtualenv"
}}
```

Current session configs: {'conf': {'spark.pyspark.python': 'python3', 'spark.pyspark.virtualenv.enabled': 'true', 'spark.pyspark.virtualenv.type': 'native', 'spark.pyspark.virtualenv.bin.path': '/usr/bin/virtualenv'}, 'kind': 'pyspark'}

ID	YARN Application ID	Kind	State	Spark UI	Driver log	Current session?
0	application_1588531763854_0002	pyspark	idle	Link	Link	

Después de iniciar la sesión de spark instalamos unas librerías que necesitaremos después:

```
sc.install_pypi_package("Pyarrow==0.14.1")
sc.install_pypi_package("Pandas")
```

```
In [2]: #import SparkSession
from pyspark.sql import SparkSession
```

Starting Spark application

ID	YARN Application ID	Kind	State	Spark UI	Driver log	Current session?
3	application_1588531763854_0005	pyspark	idle	Link	Link	✓

SparkSession available as 'spark'.

```
In [3]: #create spar session object
spark=SparkSession.builder.appName('data_processing').getOrCreate()
```

```
In [4]: sc.install_pypi_package("Pyarrow")
```

```
Collecting Pyarrow
  Using cached pyarrow-0.17.0-cp36-cp36m-manylinux2014_x86_64.whl (63.8 MB)
Requirement already satisfied: numpy>=1.14 in /usr/local/lib64/python3.6/site-packages (from Pyarrow) (1.14.5)
Installing collected packages: Pyarrow
Successfully installed Pyarrow-0.17.0
```

```
In [5]: sc.install_pypi_package("Pandas")
```

```
Collecting Pandas
  Using cached pandas-1.0.3-cp36-cp36m-manylinux1_x86_64.whl (10.0 MB)
Collecting python-dateutil>=2.6.1
  Using cached python_dateutil-2.8.1-py2.py3-none-any.whl (227 kB)
```

Luego cargamos el archivo csv desde s3 y ejecutamos todo

```
In [6]: # Load csv Dataset
df=spark.read.csv('s3://jscaicedomb/datasets/spark/sample_data.csv',inferSchema=True,header=True)
```

▸ Spark Job Progress

```
In [7]: #columns of dataframe
df.columns
```

```
['ratings', 'age', 'experience', 'family', 'mobile']
```

El archivo completo está en github, no hubo problemas corriéndolo.

Guardamos tipo csv y tipo parquet:

```
In [31]: # saving file (csv)
```

▸ Spark Job Progress

```
In [33]: #target directory
write_uri='s3://jscaicedomb/df_csv'
```

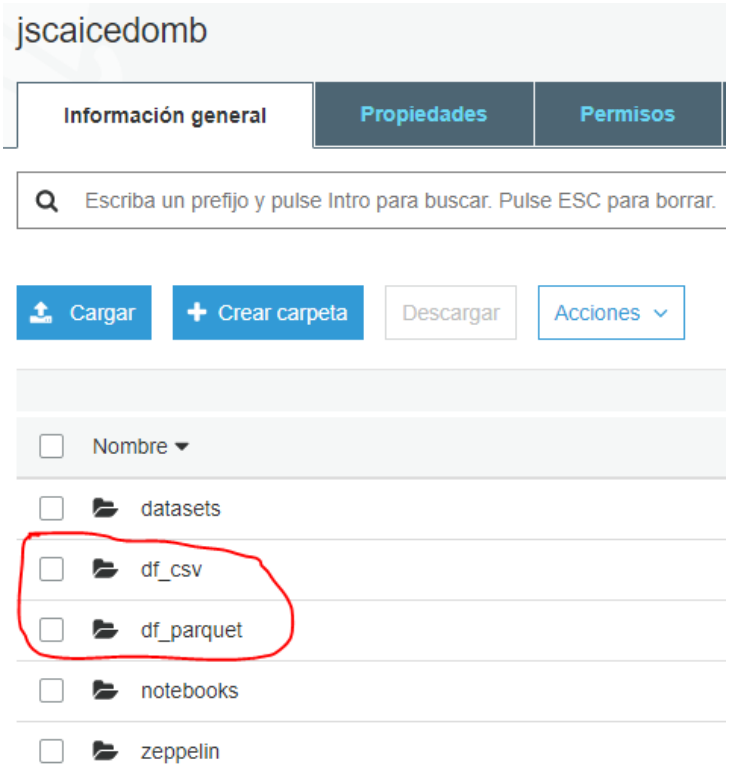
```
In [34]: #save the dataframe as single csv
         df.coalesce(1).write.format("csv").option("header","true").save(write_uri)

In [35]: # parquet

In [38]: #target location
         parquet_uri='s3://jscaicedomb/df_parquet'

In [39]: #save the data into parquet format
         df.write.format('parquet').save(parquet_uri)
```

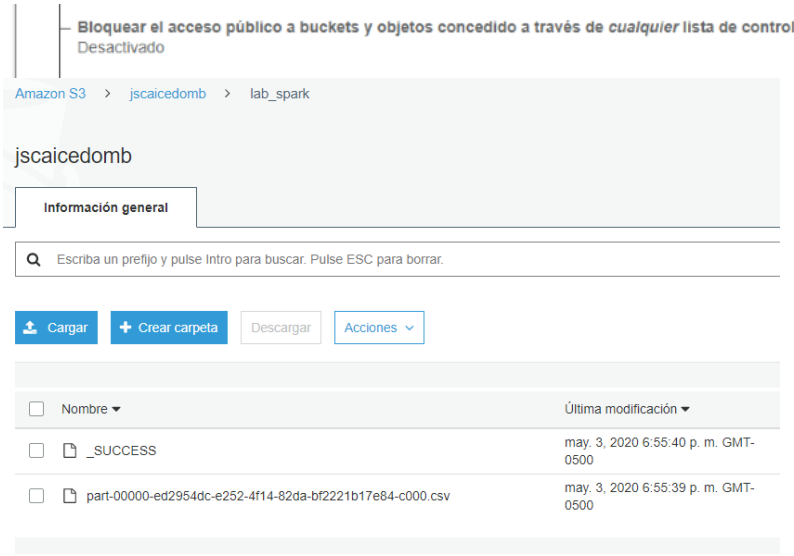
Verificamos que estén en s3 (jscaicedomb)



LAB EVALUABLE de SPARK

ARCHIVO COMPLETO

Bucket público en s3


Bloquear el acceso público a buckets y objetos concedido a través de cualquier lista de control de acceso (ACL)
Desactivado
Amazon S3 > jscaicedomb > lab_spark
jscaicedomb
Información general
Escriba un prefijo y pulse Intro para buscar. Pulse ESC para borrar.
Cargar
Crear carpeta
Descargar
Acciones
Nombre
Última modificación
_SUCCESS
may. 3, 2020 6:55:40 p. m. GMT-0500
part-00000-ed2954dc-e252-4f14-82da-bf2221b17e84-c000.csv
may. 3, 2020 6:55:39 p. m. GMT-0500