

UNIVERSIDAD EAFIT
INGENIERÍA DE SISTEMAS

STO0263 TÓPICOS ESPECIALES EN TELEMÁTICA, 2020-2
GRUPOS 001 y 002

PROYECTO 3 – BIG DATA / SPARK

Integrantes: José Jaime Ramírez Mejía

Fuente de datos: covid19 - en colombia. Tomado de
<https://www.datos.gov.co/Salud-y-Proteccion-Social/Casos-positivos-de-COVID-19-en-Colombia/gt2j-8ykr/data>

Creamos un cluster en EMR de AWS, se creó también un bucket en S3 para el almacenamiento de los datos, de forma tal que los reciba desde la fuente de datos, posteriormente pasarlos al cluster y luego enviarlos procesados al bucket nuevamente.

GitHub del proyecto: <https://github.com/jojarame/proyectobigData.git>

Paso a paso:

1. Primero instanciamos el cluster (con todos los permisos de seguridad, abriendo los puertos necesarios y las llaves que teníamos configuradas desde antes) y paralelamente creamos el bucket en S3. Se llaman “Mi cluster” y “bucketpruebabigdata” respectivamente.
2. Posteriormente procedemos a cargar un archivo de forma manual en el bucket.

Se ha realizado la carga correctamente
Consulte los detalles a continuación.

Cargar: estado

La información que aparece a continuación ya no estará disponible una vez que abandone la página.

Resumen		
Destino s3://bucketpruebabigdata	Realizado correctamente 1 archivo, 61.0 B (100.00%)	Con errores 0 archivos, 0 B (0%)

3. Entramos a Putty, ingresamos a EMR, iniciamos spark y hacemos un conteo de palabras del archivo que se encuentra en el bucket para verificar que todas las conexiones se encuentren funcionando correctamente.


```
palabras - Notepad
File Edit Format View Help
uno
dos
tres
cuatro
cinco
seis
siete
ocho
nueve
diez
```

4. Primero haremos una prueba para ver si efectivamente podemos traer los datos desde la fuente

```
[hadoop@ip-172-31-4-218 ~]$ curl -l -o datos-covid.csv https://www.datos.gov.co/api/views/gt2j-8ykr/rows.csv?accessType=DOWNLOAD
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 21.6M    0 21.6M    0     0  6083k      0 --:--:--  0:00:03 --:--:-- 6083k

[hadoop@ip-172-31-4-218 ~]$ aws s3 cp datos-covid.csv s3://bucketpruebabigdata/
upload: ./datos-covid.csv to s3://bucketpruebabigdata/datos-covid.csv
```

5. Verificamos que los pasos que hicimos en el punto anterior reflejen la información en el bucket de S3

Objetos (2)
Los objetos son las entidades fundamentales que se almacenan en Amazon S3. Para que otras personas obtengan acceso a los objetos, tendrá que concederles permisos de forma explícita. [Más información](#)

Eliminar Acciones ▼ Crear carpeta Cargar

< 1 >

<input type="checkbox"/>	Nombre ▲	Tipo ▼	Última modificación ▼	Tamaño ▼	Clase de almacenamiento ▼
<input type="checkbox"/>	datos-covid.csv	csv	2 Dec 2020 3:50:55 PM -05	217.0 MB	Estándar
<input type="checkbox"/>	palabras.txt	txt	2 Dec 2020 11:35:03 AM -05	61.0 B	Estándar

Al evidenciar que efectivamente funciona podemos continuar con la creación del cronjob.

6. Procedemos a crear el script y mediante un cronjob hacer que se haga de forma automática en el cluster. Para esto debemos ingresar al master y crearemos un script llamado poblars3.sh

```
[hadoop@ip-172-31-4-218 ~]$ vim poblars3.sh
```

Ponemos los pasos requeridos dentro del script

```
hadoop@ip-172-31-4-218:~  
curl -L -o datos-covid.csv https://www.datos.gov.co/api/views/gt2j-8ykr/rows.csv?accessType=DOWNLOAD  
aws s3 cp datos-covid.csv s3://bucketpruebabigdata/datos-covid.csv
```

"poblars3.sh" 2L, 168C 2,66 All

Y salimos nuevamente a la consola (ESC, :wq)

Ejecutamos el script para verificar

```
[hadoop@ip-172-31-4-218 ~]$ sh poblars3.sh
```

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current
			Dload	Upload	Total	Spent	Left
100	34.7M	0	34.7M	0	0	5564k	0
--:--:--	--:--:--	--:--:--	0:00:06	--:--:--	--:--:--	5844k	

Verificamos en S3 que se hayan cargado los datos correctamente

Objetos (2)

Los objetos son las entidades fundamentales que se almacenan en Amazon S3. Para que otras personas obtengan acceso a los objetos, tendrá que concederles permisos de forma explícita. [Más información](#)

🔄

Eliminar

Acciones ▼

Crear carpeta

Cargar

🔍

Buscar objetos por prefijo

<

1

>

⚙️

<input type="checkbox"/>	Nombre ▲	Tipo ▼	Última modificación ▼	Tamaño ▼	Clase de almacenamiento ▼
<input type="checkbox"/>	 datos-covid.csv	csv	2 Dec 2020 4:06:13 PM -05	217.0 MB	Estándar

7. Una vez tenemos el script funcionando procedemos a la creacion del cronjob, para esto hacemos lo siguiente

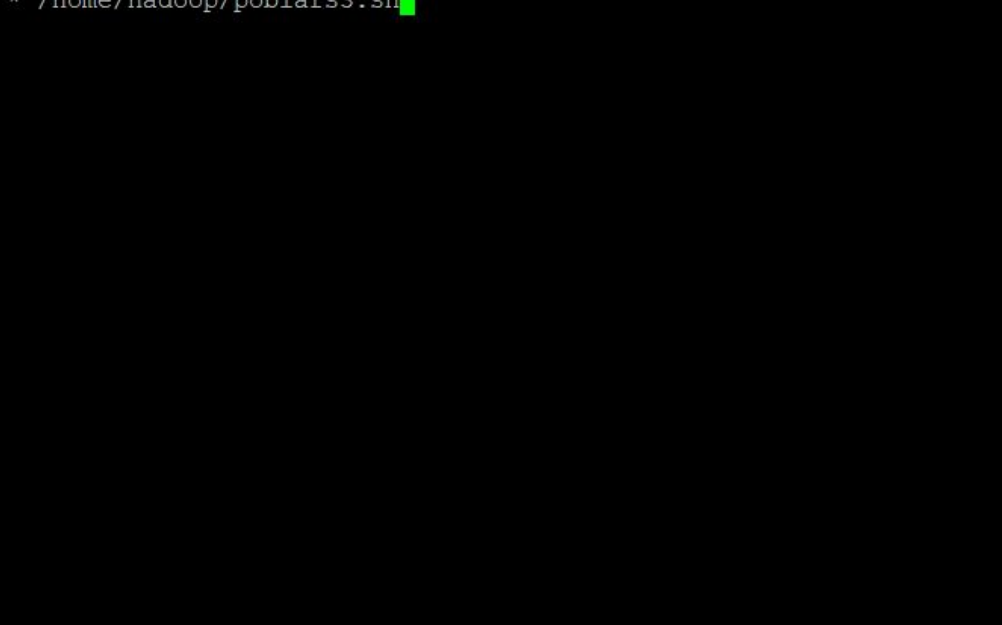
Verificamos la ruta en la que se encuentra nuestro script

```
[hadoop@ip-172-31-4-218 ~]$ pwd
/home/hadoop
```

Abrimos crontab para crear nuestra tarea programada

```
[hadoop@ip-172-31-4-218 ~]$ crontab -e
```

Ingresamos los argumentos que requerimos de acuerdo a la periodicidad con la que deseamos se ejecute el script. En este caso está programado para realizarse a las 3 am de cada día. Para nuestro caso, queremos que se actualice diariamente, el cronjob queda `@daily /home/hadoop/poblars3.sh`

A screenshot of a terminal window. The title bar at the top shows a file icon, the text "hadoop@ip-172-31-4-218:~", and standard window control buttons (minimize, maximize, close). The terminal content shows a shell prompt "0 3 * * * /home/hadoop/poblars3.sh" followed by a green cursor. On the left side of the terminal, there is a vertical column of blue and red symbols. At the bottom left, the text "-- INSERT --" is visible. On the right side, there is a vertical scrollbar.

Y salimos nuevamente a la consola (ESC, :wq)

8. A continuación ingresamos a Spark, para esto iremos a un Notebook de Jupyter, accesible desde EMR, podemos ver el enlace al lado izquierdo de la pantalla donde dice “Blocs de notas”. Allí crearemos un nuevo cuaderno, lo vinculamos a S3 y a nuestro cluster. Si en algún momento finalizamos el cluster, este se daña o por cualquier otro motivo deje de ser accesible, desde la configuración del cuaderno podemos vincularlo a un cluster nuevo y nos ahorraremos una cantidad de trabajo considerable. A continuación veremos algunas de las operaciones que hacemos a través de spark, lo que nos permite procesar los datos que tenemos en RAW para obtener resultados.

```

[1]: from pyspark.sql import SparkSession
import pyspark.sql.functions as f

spark = SparkSession.builder.getOrCreate()

Starting Spark application
ID      YARN Application ID  Kind  State  Spark UI  Driver log  Current session?
0  application_1606962205491_0003  pyspark  idle  Link  Link  ✓

SparkSession available as 'spark'.

[2]: Df = spark.read.options(header='True', inferSchema='True', delimiter=',') .csv ("s3://bucketpruebabigdata/RAW/datos-covid.csv")
Df.take(1)

▶ Spark Job Progress

[Row(fecha reporte web='6/3/2020 0:00:00', ID de caso=1, Fecha de notificación='2/3/2020 0:00:00', Código DIVIPOLA departamento=11, Nombre departa
tamento='BOGOTA', Código DIVIPOLA municipio=11001, Nombre municipio='BOGOTA', Edad=19, Unidad de medida de edad=1, Sexo='F', Tipo de contagio='I
mportado', Ubicación del caso='Casa', Estado='Leve', Código ISO del país=380, Nombre del país='ITALIA', Recuperado='Recuperado', Fecha de inicio
de síntomas='27/2/2020 0:00:00', Fecha de muerte=None, Fecha de diagnóstico='6/3/2020 0:00:00', Fecha de recuperación='13/3/2020 0:00:00', Tipo
de recuperación='PCR', Pertenencia étnica=6, Nombre del grupo étnico=None)]

[25]: Df.groupBy("Nombre departamento").count().select("Nombre departamento",f.col("count").alias("cantidad")).show()

```

- Ahora debemos entrar a Amazon Glue el cual nos permite generar los esquemas y las tablas con las cuales haremos el análisis de nuestros datos. Desde aquí debemos hacer que nuestros esquemas coincidan con lo que ingresamos en spark en el paso anterior.

Tablas Una tabla es la definición de metadatos que representa sus datos, incluido el esquema. Una tabla se puede usar como un origen o un destino de una definición de trabajo.

Añadir tablas	Acción	<input type="text" value="Filtrar por atributos o buscar por palabra clave"/>	Guardar vista	Mostrando: 1 - 4	Recargar	Configuración	Ayuda
<input type="checkbox"/> Nombre	Base de datos	Ubicación	Clasificación	Última actualización	Obsoleto		
<input type="checkbox"/> casos-departamento-fecha	covid	s3://bucketpruebabigdata/REFI...	csv	3 diciembre 2020 8:55 p. m. U...			
<input type="checkbox"/> casos-edad-fecha	covid	s3://bucketpruebabigdata/REFI...	csv	3 diciembre 2020 8:55 p. m. U...			
<input type="checkbox"/> casos-sexo-fecha	covid	s3://bucketpruebabigdata/REFI...	csv	3 diciembre 2020 8:55 p. m. U...			

- Entramos a Amazon Athena, este servicio hace posible la consulta de datos en S3 haciendo uso de SQL, para esto debemos conectarnos a nuestra fuente de datos (bucket) y escribir algunas consultas según queramos visualizar los datos.

New query 1 New query 3 New query 4 **New query 5** New query 6 New query 7 +

```
1 SELECT * FROM "covid"."casos-departamento-fecha";
```

Run query Save as Create ▾ (Run time: 2.41 seconds, Data scanned: 214.79 KB) Format query Clear

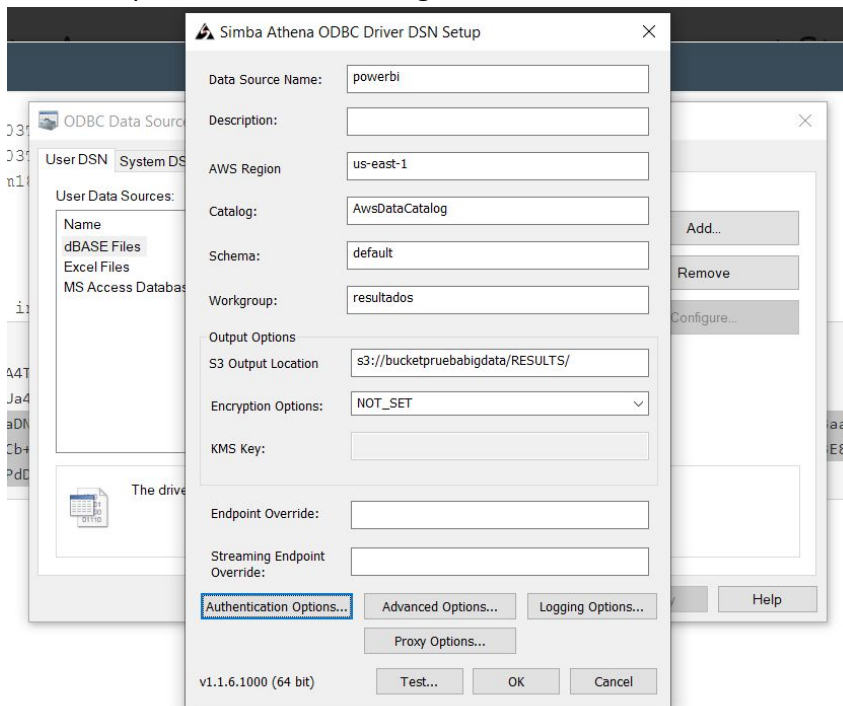
Use Ctrl + Enter to run query, Ctrl + Space to autocomplete Athena engine version 1 [Release versions](#)

Results

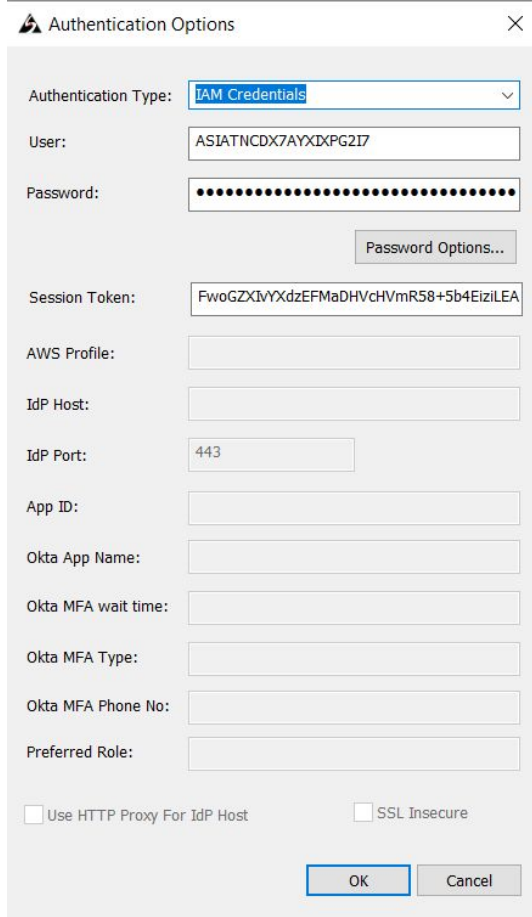
	nombre departamento ▾	fecha reporte web ▾	cantidad ▾
1	NARIÑO	6/11/2020 0:00:00	198
2	ANTIOQUIA	22/8/2020 0:00:00	1304
3	CAUCA	6/9/2020 0:00:00	256

11. Para poder visualizar y analizar los datos haremos uso de Microsoft PowerBI, para esto se requiere instalar dicho software en el equipo personal o de trabajo y establecer una conexión entre dicho equipo y S3. Instalaremos primero PowerBI, se puede descargar desde Microsoft Store. Luego descargar Simba Athena ODBC Driver, el cual hace posible la conexión entre AWS y PowerBI.

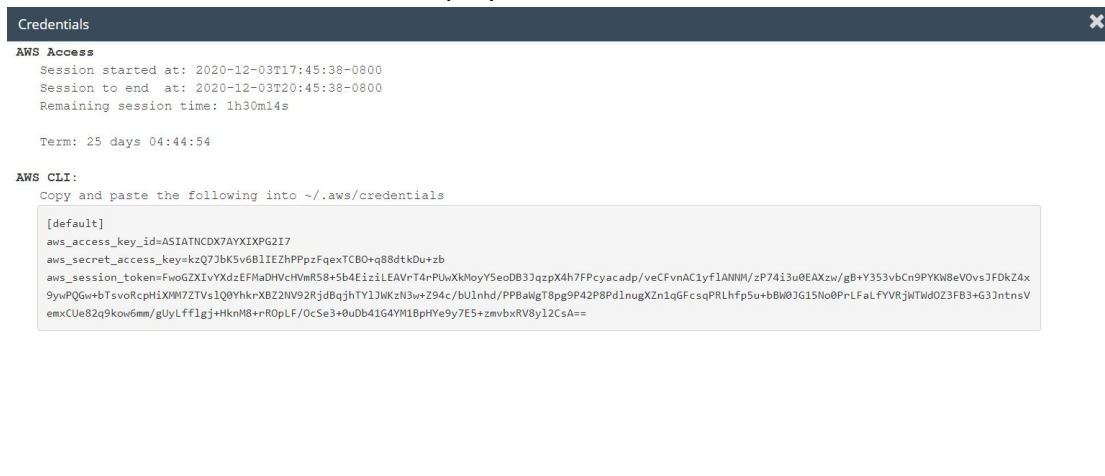
Una vez instalados ambos programas debemos ejecutar Simba, creamos un nuevo Data Source. Es de suma importancia poner la región en la que se encuentra el bucket y la ruta donde se guardarán los datos procesados. En la imagen podemos ver un ejemplo de como quedarían dichas configuraciones.



Luego dar clic en Authentication Options... e ingresar las credenciales de AWS.



Para encontrar las credenciales basta con ir al Workbench y dar clic en Account Details, luego en el botón show y veremos nuestras credenciales, debemos tener presente que el token se vence cada cierto tiempo y debemos renovar la conexión a través de Simba



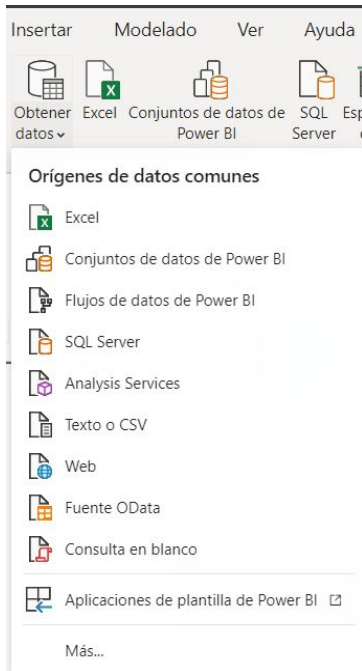
```
Credentials
AWS Access
Session started at: 2020-12-03T17:45:38-0800
Session to end at: 2020-12-03T20:45:38-0800
Remaining session time: 1h30m14s

Term: 25 days 04:44:54

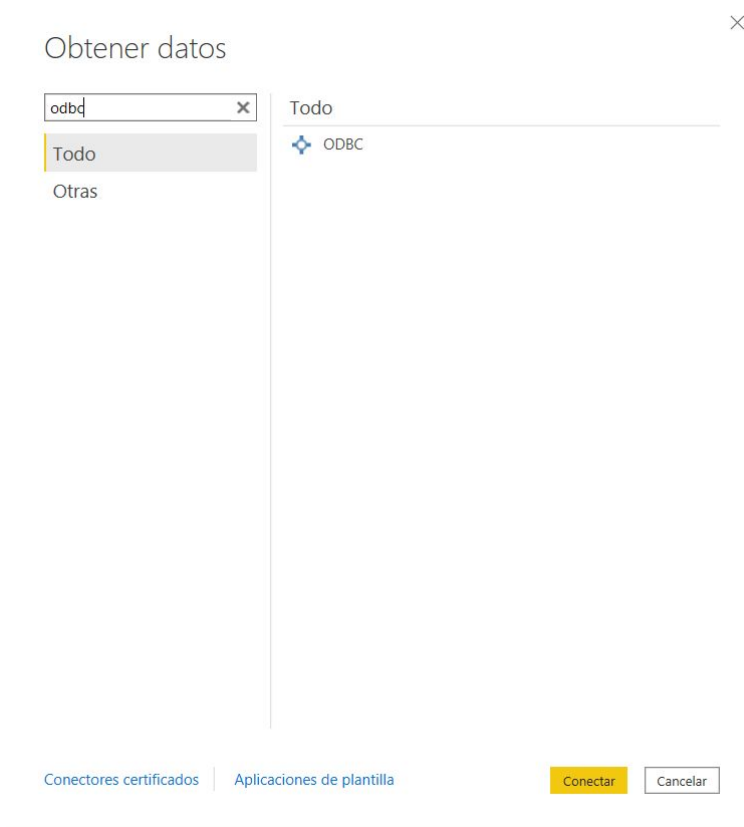
AWS CLI:
Copy and paste the following into ~/.aws/credentials

[default]
aws_access_key_id=ASIATNCDX7AYXDXPG2I7
aws_secret_access_key=kzQ73bK5v6B1IEZhPPpZfQexTCB0+q88dtkDu+zb
aws_session_token=FwoGZXIvYXZEFMaDHVcHVmR58+Sb4EiziLEAVrT4rPUwXkHoyY5eoDB3JzqX4h7FPcyacadv/veCFvnAC1yflA1NNM/zP74i3u0EAXzw/gB+Y353vbCn9PYKW8eVOvsJFDkZ4x
9yvpQ0w+btSvRcpHiXMM7ZTVs1Q0YhkrXBZ2NV92RjdBqjhtY1JWkzN3u+Z94c/bUlnhd/PPBaWgT8pg9P42P8PdlnugXZn1qGfcsqPRLHfp5u+bBw0JG15No0PrLFaLfyYVRjWtWd0Z3FB3+G3JntnsV
emxCue82q9kow6mm/gUyLff1gJ+HknM8+rR0pLF/OcSe3+0uDb41G4YM1BpHYe9y7E5+zmVbxRV8y12CsA=
```


12. Ya tenemos todo listo para visualizar y analizar los datos desde PowerBI. Abrimos el programa y damos clic en la barra superior en el botón Obtener datos-Más...



13. Escribimos ODBC en el buscador, seleccionamos en el lado derecho y clic en conectar



14. Seleccionamos la fuente que creamos en Simba y hacemos clic en Aceptar

De ODBC

Nombre de origen de datos (DSN)

dBASE Files
dBASE Files
Excel Files
MS Access Database
powerbi
Simba Athena

Aceptar

15. Seleccionamos todas las tablas que deseemos analizar y hacemos clic en cargar y una vez traiga toda la información podremos hacer todos los gráficos y análisis que las consultas creadas en spark nos permitan

📁 ODBC (dsn=powerbi) [1]

📁 AwsDataCatalog [2]

📁 covid [3]

☐ 📄 casos-departamento-fecha

☐ 📄 casos-edad-fecha

☐ 📄 casos-sexo-fecha

▷ 📁 sampledb