Actividad:	RedShift						
Ciclo:	IABD	Modulo:	SBD	Curso:		Agrupación:	1
Alumno/a:						Grupo:	
CIPFP www.fpmislata.com							

ETL con Spark

Servicios AWS implicados:

- AWS S3
- AWS Glue Studio
- AWS Glue Data Catalog
- AWS Glue Job
- AWS Glue Crawler
- AWS Glue Workflow

La práctica consiste en hacer una ETL con AWS Glue utilizando código pyspark.

Los datos de partida son dos ficheros en formato csv con datos de clientes y ventas.

Cargaremos los datos en un bucket S3 que también tendrá una carpeta para el script de pyspark y otra para los resultados de salida.

Con estos datos queremos un fichero con las ventas totales por cliente en formato json.

Los campos de los ficheros son:

- **customers**: **Enlace**
  - {CUSTOMERID, CUSTOMERNAME, EMAIL, CITY, COUNTRY, TERRITORY, CONTACTFIRSTNAME, CONTACTLASTNAME}
- **sales**: **Enlace**
  - {ORDERNUMBER, QUANTITYORDERED, PRICEEACH, ORDERLINENUMBER, SALES, ORDERDATE, STATUS, QTR_ID, MONTH_ID, YEAR_ID, PRODUCTLINE, MSRP, PRODUCTCODE, DEALSIZE, CUSTOMERID}

Actividad:	RedShift						
Ciclo:	IABD	Modulo:	SBD	Curso:		Agrupación:	1
Alumno/a:						Grupo:	
CIPFP www.fpmislata.com							

Pasos a seguir:

- Crear un bucket (lago de datos) S3 con la siguiente estructura de carpetas y ficheros:
  - datos
    - clientes
      - customers.csv
    - ventas
      - sales.csv
  - salida
  - scripts
- Crear un crawler con AWS Glue que rastree la carpeta datos y los introduzca en una BD (ventas)
- Crear un job con Glue de Spark Script Editor
  - El script debe:
    - Guardarse en la carpeta scripts del bucket
    - Debe seleccionar las ventas totales por cliente y guardarlas en un fichero en formato json en la carpeta de salida
- Crear un flujo de trabajo que lo haga todo.
  - Añadir Workflow
  - Añadir Trigger
    - Inicio Rastreador
  - Añadir Nodo
    - Seleccionar Crawler
  - Añadir Trigger
    - Type: Evento
    - Start after ANY watched event
  - Añadir Job/Crwaler
    - Seleccionar Crawler
    - Evento:Exito

Ejemplo: https://aws-dojo.com/ws29/labs/create-glue-workflow/

Ejemplo de código:

```
import sys
from datetime import datetime

from pyspark.sql import SparkSession
from pyspark.sql.functions import *
spark = SparkSession\
    .builder\
    .appName("SparkETL")\
    .getOrCreate()

spark.catalog.setCurrentDatabase("ventas")
df = spark.sql("select * from clientes")
#df.show()

df = df.select("customername","email")
#df.show()

df.write.format("json").mode("overwrite").save("s3://lagodatos/salida/"
```