UNIVERSIDAD EAFIT
ST1612 SISTEMAS INTENSIVOS EN DATOS, 2021-2
PROFESOR: EDWIN MONTOYA – emontoya@eafit.edu.co

LAB 2/3 – Despliegue de una solución Hadoop/Spark en AWS EMR (procesamiento) + S3 (datalake)

1. **Los objetivos de este trabajo son:**

General:

Diseñar e implementar un ecosistema de almacenamiento y procesamiento de datos relacionados con el Covid-19 en Colombia y el Mundo (contagios y vacunación). Para esto debe Diseñar e Implementar un Datalake para almacenar los datos, debe Catalogarlos (con glue), debe poderlos consultar con SQL (Athena y Hive) y hacer un Análisis Exploratorio de Datos con Spark (jupyter/pyspark). Para esto mínimo utilizará los servicios de cluster EMR, de almacenamiento S3, de catalogación Glue y de consulta SQL con Athena.

Específicos:

1. Conocer en mayor profundidad las arquitecturas de una solución big data de un proveedor de nube, en este caso se seleccionará dentro del curso Amazon AWS, sin embargo, si el equipo desea utilizar y trabajar otra nube no hay problema (GCP de google), siempre y cuando tenga los accesos suficientes para desarrollar el caso de estudio.

2. Desarrollar un caso de estudio que aplique varias etapas del ciclo de vida de un proyecto de Big Data Analytics contemplando al menos los siguientes pasos:
    a. Identificación, descarga e ingesta de fuentes de datos (datasets, APIs, BD, etc) relacionados con el covid-19 de diferentes tipos.
    b. Diseñar un Data Lake como Almacenamiento de los datos de diferentes origenes, tipos y estructuras
        i. Adoptar una arquitectura de referencia para datalakes
        ii. definir las zonas que contendrá el datalake.
        iii. definir la estructura de directorios óptimo
    c. Ingesta y Almacenamiento de datos en el Data Lake en la zona ‘raw’
    d. Realizar catalogación en ‘raw’ y otro en ‘trusted’ de datos utilizando AWS Glue

e. Realizar uno o dos procesos ETL para preparar los datos hacia las zonas ‘trusted y ‘refined utilizando Glue y pyspark. (uno de los ETL será la implementación de un diseño dimensional para facilitar la visualización de datos)
f. Realizar consultas básicas SQL mediante AWS Athena y Hive de los datos almacenados en el datalake mediante la catalogación realizada por AWS Glue y EMR.
g. Realizar el diseño e implementación de un ecosistema Hadoop/Spark basado en AWS EMR que permita:
    i. Aprender a desplegar clústeres EMR
    ii. Aprender a almacenar datos temporales en HDFS
    iii. Aprender a Catalogar y acceder datos estructurados vía Hive y su correspondiente integración con S3, glue y athena desde EMR.
    iv. Aprender a instalar un ambiente de procesamiento de datos basado en Spark con pyspark utilizando jupyter-notebooks accediendo los datos crudos en S3 o datos catalogados en Hive o Glue mediante SparkSQL.

3. Crear un modelo dimensional que facilite la visualización de datos y responsa a preguntas de negocio.

2. Introducción a Hadoop, Spark y AWS EMR

Los marcos de trabajo Hadoop[1],[2] y Spark[3],[4] son 2 marcos de trabajo y software para soporte verdadero de tecnologías big data. Permitiendo almacenar y procesar grandes cantidades de datos para analítica. Amazon AWS tiene una implementación de esta tecnología llamada EMR[5] (Elastic Map Reduce) quien con otras tecnologías conforman un ecosistema completo de Big Data. Amazon tiene una implementación robusta y administrada que permite en cuestión de minutos u horas crear un Clúster con varios nodos y muchos de las aplicaciones y servicios para almacenamiento y procesamiento masivo. Destacamos a continuación los principales servicios que implementa EMR:

---

[1] Apache Hadoop - Wikipedia, la enciclopedia libre

[2] Apache Hadoop

[3] Apache Spark - Wikipedia, la enciclopedia libre

[4] Apache Spark™ - Unified Analytics Engine for Big Data

[5] Análisis big data | Ejecución marcos Hadoop | Amazon EMR

HDFS: Almacenamiento masivo de datos en el clúster (para diferentes proyectos, no será conveniente dejar de forma permanente datos en el sistema de archivos HDFS, ya que en muchas aplicaciones se crean y destruyen clústeres EMR. Normalmente se emplea para almacenamiento y archivos temporales para el procesamiento).

Integración con S3: Dada la temporalidad de muchas aplicaciones de HDFS, es recomendable tener acceso desde EMR a los archivos, tablas, etc almacenados en S3, de esta forma se puede utilizar EMR para procesar, S3 para almacenamiento permanente y HDFS para almacenamiento temporal.

HIVE: Es un motor de consultas SQL, que tiene su propio gestor de catálogos (Meta Store) para la creación de bases de datos y tablas. También se puede integrar con el catálogo AWS Glue y tener tablas externas almacenadas en AWS S3. Realizar funciones similares a AWS Athena. EMR también implementa el servicio SQOOP que permite realizar procesos de EL (Extract-Load) desde bases de datos relacionales convencionales (Oracle, MySQL, postgress) hacia HDFS/Hive o viceversa.

HBASE: soporte para base de datos NoSQL columnar. Puede tener su almacenamiento en S3.

SPARK: Quizás una de las principales aplicaciones y usos de EMR es el soporte para clústeres de procesamiento en Apache Spark, permite además soporte para el desarrollo y ejecución de notebooks en Python, R y Scala a través del servicio: Jupyter y Zepellin. EMR ofrece un servicio de notebooks jupyter sin servidor (serverless) que permite lanzar trabajos en un clúster EMR. Además, estos notebooks son almacenados en AWS S3 para su independencia de la creación o destrucción de clústeres EMR. En las versiones EMR-6.x soporta la nueva versión de Spark 3.x.

Otros servicios de EMR: Ganglia, Zookeeper, Oozie, TensorFlow, Tez, Presto, Hue,Livy,Flink,Pig,PrestoSQL, MXNET, Phoenix, HCatalog.

Complementario a EMR, AWS tiene muchos otros componentes que son requeridos para implementar una solución analítica, entre las que destacamos:

- Datalake: Amazon tiene un conjunto de servicios que permiten implementar lagos de datos, siendo el principal S3, Glue y Athena.
- ML/DL: Amazon presenta en Sagemaker, todo un ecosistema de procesamiento administrado, que permite diseñar y ejecutar gran cantidad de modelos de Machine Learning y Deep Learning basado tanto en modelos entrenados por científicos de datos, hasta el uso de la gran variedad de servicios cognitivos de

amazon. Sagemaker provee la Infraestructura de procesamiento y el ambiente de desarrollo y entrenamiento de modelos basados en una variedad de lenguajes, entre los que destacamos Python, R entre otros.

Como links de ayuda para desarrollar este caso, se comparten varios videos pregrabados hace unos semestres (las versiones y procesos han cambiado, tener en cuenta esto). Igualmente se recomienda leer las presentaciones de hadoop y spark en los contenidos de Interactiva Virtual:

Video: Amazon-lab-01-aws-CLI-windows-install-20191014	https://web.microsoftstream.com/video/636cdaa0-3cd9-4503-99c1-5dc9f23db80b
Video: Amazon-lab-02-aws-S3-20191014	https://web.microsoftstream.com/video/4c4969dc-9eba-455a-9c48-af3b3875d5db
Video: parte 1 – creación de clusters EMR	https://web.microsoftstream.com/video/03427647-0b8c-4e0d-a4dc-95e73a064a2d
Video: parte 2 – creación de clusters EMR	https://web.microsoftstream.com/video/8e3bbed2-3596-4f7a-865c-66ec241c5647
Video: Creación de notebooks administrados sin servidor en EMR.	https://web.microsoftstream.com/video/dc8f039b-302e-4348-90f7-f5d8ee2a34b7

Ver datos y ejemplos de pyspark:

https://github.com/ST1612eafit/st1612_20212.git

3. Datalake

Contexto:

La mayoría de las empresas que han entrado a la era de la analítica moderna, están migrando sus data warehouse hacia los lagos de datos (datalakes) o están implementando desde cero dicho proyecto. En el lago de datos, se va a almacenar todos los datos de analítica de una empresa. Sin embargo, estos conjuntos de datos deben ser organizados en *zonas*, definido bien el esquema de directorios, catalogados, transformados y dejados listos para ser accedido por programas analíticos de ML o motores de consulta SQL para su posterior análisis.

Para el desarrollo de este trabajo se identificarán varias fuentes de datos sobre covid-19, las cuales uds deberán almacenar y transformar para quedar listos en procesamiento posterior.

Tendremos varios conjuntos de datos, que representarán diferentes tipos (estructurados, no estructurados y semi-estructurados, así como potencialmente en diferentes formatos de archivos.

La siguiente es la lista y descripción de los datasets mínimos con los que diseñará el datalake, pero puede encontrar otras fuentes de datos adicionales sobre el covid-19. El objetivo es descargar todas las fuentes y copiarlas al datalake, otra cosa son la catalogación y transformación que debe ser pocas fuentes.

Para los datos estructurados o semi-estructurados, deberá catalogarlos con AWS Glue para ser posteriormente consultados por AWS Athena.

Igualmente, en AWS, para datos estructurados o semi-estructurados deberá realizar al menos una transformación o ETL para ser llevados de la zona 'raw' a la zona 'trusted'.

Lista de fuentes de datos:

Datos de casos de covid en Colombia	Datos estructurados	https://www.datos.gov.co/Salud-y-Protecci-n-Social/Casos-positivos-de-COVID-19-en-Colombia/gt2j-8ykr/data
Datos de covid19 – casos reportados en Colombia	Datos estructurados	Casos positivos de COVID-19 en Colombia | Datos Abiertos Colombia
Datos de vacunación covid19 en Colombia	Datos estructurados	https://www.ins.gov.co https://www.datos.gov.co

4. Procesamiento principal:

El procesamiento principal de este lab, estará centrado en:

- Transformación de modelo de datos (ETL) desde el origen de los datos en SQL y PySpark de los datos de covid19 hacia un esquema dimensional que facilite la visualización futura de datos.
- De forma opcional, realizar un módulo de visualización en la plataforma de su mayor dominio o interés. Podría exportar los datos dimensionales para ser cargados en https://webpivottable.com

5. **Entregables:**

1. archivo pdf donde describa el proceso de implementación del caso (datalake, EMR y notebooks)
2. datalake desplegado en aws educate: recolección de fuentes, buckets s3, servicios Glue de catalogación y ETL, servicio Athena de consulta de datos en el data lake a través de las tablas Glue. Servicio de consulta desde EMR/Hive.
    a. Datos de trabajo almacenados en S3 y compartidos con el profesor o públicos a todo el mundo. Estos datos ejemplo también deberán estar en un repo github.
3. Notebooks almacenados en AWS S3 y compartidos con el profesor. Estos notebooks ejemplo también deberán estar en un repo github.
4. Scripts Glue, SQL, Athena, Hive en el repo github.
5. Scripts de creación de la infraestructura AWS, principalmente EMR en el repo github.

Criterios de evaluación:

- Documento del proceso del caso en pdf, 20%
- Modelo organizado, documentado de la implementación de zonas en el lago de datos y todos los datos almacenados en AWS S3, 40%
    - Generación las tablas SQL con AWS Glue (Rastreador o Crawler) del caso de Covid-19, para su posterior consulta/verificación con AWS Athena.
    - Generación de al menos un proceso ETL con AWS Glue, para paso entre zonas y creación del modelo dimensional en SQL
- EMR funcional corriendo en Amazon, 40%
    - ejecutando los ejemplos que desarrolle en notebooks.
    - accediendo los datos desde dichos notebooks a datos en S3.
    - Documentación de creación del Clúster EMR
    - Accediendo a los datos almacenados en S3/Glue desde Hive con HUE
    - Creación del modelo dimensional en jupyter notebooks pyspark.

Notas adicionales:

- puede realizar la ingesta de datos a través de la web de s3, aws cli, o algún robot. ver mejores prácticas en: Data Lake Governance Best Practices - DZone Big Data
- Ingesta:
    - Con fuentes de datos como archivos o datasets, donde se pueda, deberán implementar un robot o proceso, que periódicamente descargue los archivos y los almacene en el datalake (aws s3) zona raw.

ej:

script-download.sh

curl - http:// (traer datos de internet)

aws s3 cp … (copiar al datalake)