

UNIVERSIDAD EAFIT
MAESTRÍA EN CIENCIA DE DATOS Y ANALÍTICA
ST1800 ALMACENAMIENTO Y RECUPERACIÓN DE INFORMACIÓN, 2023-2
PROFESOR: EDWIN MONTOYA – emontoya@eafit.edu.co

Fecha de entrega: 4-septiembre-2023

Trabajo 1 – Despliegue de un Lakehouse que integre datos en S3 (datalake) y el data warehouse RedShift con procesamiento en Hadoop/Spark de AWS EMR, Glue, Athena y Redshift (Spectrum y ML)

1. Los objetivos de este trabajo son:

General:

Diseñar e implementar un ecosistema de almacenamiento y procesamiento de datos relacionados con CAMBIO CLIMATICO o CALENTAMIENTO GLOBAL de Medellín, Colombia y el Mundo. Para esto debe Diseñar e Implementar un Datalake para almacenar los datos, debe Catalogarlos (con glue), debe poderlos consultar con SQL (Athena y Hive), cargar datos desde S3 a Redshift y consultar datos en S3 desde Redshift Spectrum y hacer un Análisis Exploratorio de Datos con Spark (jupyter/pyspark). Para esto utilizará alguna de las plataformas de procesamiento SQL o Spark utilizadas en este proyecto. Además, comprobará la visión integrada de Lakehouse al poder consultar los datos de manera indistinta desde Athena, Redshift, Hive, SparkSQL.

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 o Azure), siempre y cuando tenga los accesos suficientes para desarrollar el caso de estudio del proyecto1.
2. Desarrollar un caso de estudio que aplique las etapas de un ciclo de vida de un proyecto de Big Data Analytics contemplando al menos los siguientes pasos:
 - a. Fuentes de datos: Identificación y descarga relacionados con Cambio Climático o Calentamiento Global y datasets relacionados.
 - b. Ingesta de los datos como archivos / datasets desde URLs o APIs
 - i. Ingesta y Almacenamiento de datos en el Data Lake en la zona 'raw'
 - c. Diseñar un Data Lake como Almacenamiento de los datos de diferentes orígenes, 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
- 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 la zona 'trusted' utilizando Glue y pyspark.
- 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 modelado (deseable multidimensional) de datos / tablas para ser almacenadas en RedShift o ser accedidas desde S3 a través de RedShift Spectrum.
- h. 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, athena y Redshift 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.

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 EMR⁵ (Elastic Map Reduce) quien con otras tecnologías conforman un ecosistema completo de Big Data. Amazon tiene una implementación robusta y

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](#)

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:

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, postgres) 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, principalmente basados en S3, Glue y Athena.

- Data Warehouse Moderno: Redshift y su ecosistema. Permite almacenar grandes conjuntos de datos, principalmente estructurados, optimizado para procesamiento SQL con incorporación de modelos de ML.
- Lakehouses que integra Datalakes con Datawarehouses, integrando catalogos y exponiendo una vista unificada a la organización de los datos de analítica con diferentes opciones de procesamiento (SQL, Spark, etc).
- 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 (nota: pueden algunos ser de versiones anteriores, lo cual implicará un proceso de actualización):

(algunos videos se van a actualizar a nuevas versiones en 2023-agosto)

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://youtu.be/MyXSwxN5Zdk
Video: parte 2 – creación de clusters EMR https://youtu.be/3sao-qJG34Y
Video: Creación de notebooks administrados sin servidor en EMR. https://web.microsoftstream.com/video/dc8f039b-302e-4348-90f7-f5d8ee2a34b7
Demo AWS EMR Hive 20211119 https://youtu.be/zsic9XS1R-A
lab aws glue athena 20211119 https://youtu.be/2WliTIK1ips

Ver datos y ejemplos de pyspark:

<https://github.com/st1800eafit/st1800-232.git>

3. Datalake

Contexto:

La mayoría de las empresas que han entrado a la era de la analítica moderna, están migrando sus data warehouse clásicos o legacy 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 cambio climático o calentamiento global, 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.

Deberá hacer una búsqueda organizada de datasets sobre cambio climático o calentamiento global en el ámbito local, nacional o internacional. Existen fuentes como datos abiertos en Colombia (datos.gov.co o medata.gov.co), proyectos específicos con datos como SIATA, o muchos otros proyectos a nivel nacional o internacional, también buscar datos en kaggle o diferentes sitios de indexación de datasets.

Algunos de los siguientes son potenciales ejemplos de datos:

- [global-warming data on data.world | 13 datasets available](#)
- [Datasets | GlobalChange.gov](#)
- [Climate Change Data - Awesome Datasets - DataHub - Frictionless Data](#)
- <https://github.com/sn2606/Climate-Change-Datasets>
- [Cambio climático | Data \(bancomundial.org\)](#)
- [Search | Kaggle](#)
- [SIATA | Área Metropolitana del Valle de Aburrá](#)
 - Otros datasets: <https://datosabiertos.metropol.gov.co>

- [Datos Abiertos Bogotá \(bogota.gov.co\)](http://bogota.gov.co)
- [DATOS ABIERTOS DE PRONÓSTICOS DEL TIEMPO Y ALERTAS NACIONALES - IDEAM \(pronosticosyalertas.gov.co\)](http://pronosticosyalertas.gov.co)

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'.

4. Entregables:

1. Archivo pdf donde describa el proceso de implementación del caso (datalake, data warehouse, lakehouse, EMR y notebooks)
2. Datalake desplegado en aws academy: recolección de fuentes, buckets s3, servicios Glue de catalogación y ETL, servicio Athena de consulta de datos en el datalake a través de las tablas Glue. Servicio de consulta desde EMR/Hive. Servicio de consulta desde Redshift Spectrum.
 - a. Datos de trabajo almacenados en S3 y compartidos con el profesor o públicos a todo el mundo.
3. Notebooks almacenados en AWS S3 y compartidos con el profesor.
4. Todos los códigos fuentes deben ser enviados al profesor mediante un repositorio github o mediante un adjunto en entregas (buzón).
 - a. Scripts/pantallazos Glue, SQL, Athena, Hive, RedShift.
 - b. Scripts/pantallazos de creación de la infraestructura AWS, principalmente EMR en el repo github.

Criterios de evaluación:

- Documento del proceso del caso en pdf, 20% (utilizar pantallazos de algunos de los procesos que ejecuta a lo largo del proyecto)
- Modelo organizado, documentado de la implementación de zonas en el lago de datos, y todos los datos almacenados en AWS S3, 30%
 - Generación las tablas SQL con AWS Glue (Rastreador o Crawler) del caso de “Datos ficticios de una empresa para modelado multidimensional”, para su posterior consulta/verificación con AWS Athena (ajuntar pruebas de las consultas en el informe).
 - Generación de al menos un proceso ETL con AWS Glue, para paso entre zonas.
- EMR funcional corriendo en Amazon, 30%
 - ejecutando los ejemplos que desarrolle en notebooks.
 - accediendo los datos desde dichos notebooks a datos en S3.
 - Accediendo a los datos almacenados en S3/Glue desde Hive.
- Redshift, 20%
 - Manejo de tablas nativas
 - Consultas de tablas con datos en S3.

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)
```