**UNIVERSIDAD EAFIT**

**MAESTRÍA EN CIENCIA DE DATOS Y ANALÍTICA**

**ST1612 SISTEMAS INTENSIVOS EN DATOS**

**PROFESOR: EDWIN MONTOYA**

ALUMNOS:

Cindy Paola Guerra Medina

Carlos Alberto Murillo Martínez

Luz Stella Flórez Salazar

**Taller 4 – WPT Lab**

Fecha de entrega: 11 de octubre de 2021.

* **Arquitectura de referencia**

Gráfico, Diagrama, Gráfico de cajas y bigotes

Descripción generada automáticamente

La sección de Datasource lo realizamos de fuentes estrructuradas y almacenamos en S3, la ingestión se realiza a través de aws cli y un script que corre en batch, catalogación en AWS Glue procesamiento en EMR y el consumo no lo pudimos realizar en Qucksight porque no teníamos permisos de acceso al servicio de AWS.

1. archivo pdf donde describa el proceso de implementación del caso (datalake, EMR y notebooks)

Lo primero que hicimos para definir la estructura del datalake, fue entrar en la consola de AWS Educate y seleccionamos el servicio S3, en este creamos una serie de carpetas que contienen la base de datos; para el ejercicio seleccionamos la siguiente estructura:

S3:

| - raw

| | covid\_colombia

| | |casos

| | | | casos\_covid.csv

| | |vacunación

| | | | vacunación.csv

Imagen de ejemplo:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Ilustración 1 - Configuración Datalake

* **CARGA DE DATOS**

Para facilitar el proceso de carga de datos, creamos un bash script que realiza el proceso de carga con un doble click (Figura 3). Para que este script funcione es necesario instalar AWS Cli y configurar el archivo credentials en el equipo, como se ilustra en la Figura 2.

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Ilustración 2 - Credentials

Texto

Descripción generada automáticamente

Ilustración 3 - bash script load data

Una vez cargados los datos podemos pasar a catalogar la información utilizando el servicio de AWS Glue; la información catalogada genera metadatos que luego pueden ser consultados en Lenguaje SQL o por otros servicios de AWS.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración 4 - Crawlers

Se crean los crawlers, los cuales toman la información de casos y generan tablas que son consultadas en Athena, para realizar las consultas o validaciones:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración 5 - Athena

El siguiente paso que realizamos fue crear un clúster con la siguiente configuración en EMR:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración 6 – EMR

* **SCRIPT DE CREACIÓN DEL EMR**

**Todos los pasos mencionados pueden ser resumidos en una línea de comandos que se anexa en el script bash load\_data.sh**

Gráfico, Texto

Descripción generada automáticamente con confianza media

En la última línea completa sería esta instrucción:

aws emr create-cluster --applications Name=Hadoop Name=Hive Name=Hue Name=JupyterHub Name=JupyterEnterpriseGateway Name=Zeppelin Name=Tez Name=Spark Name=Livy Name=HCatalog --ec2-attributes '{"KeyName":"pckeypair","InstanceProfile":"EMR\_EC2\_DefaultRole","SubnetId":"subnet-057ea1b814b9e1482","EmrManagedSlaveSecurityGroup":"sg-0cebe96bbb159af9d","EmrManagedMasterSecurityGroup":"sg-0e1111570aa5f7fa9"}' --release-label emr-6.4.0 --log-uri 's3n://aws-logs-027146100852-us-east-1/elasticmapreduce/' --instance-groups '[{"InstanceCount":1,"BidPrice":"OnDemandPrice","EbsConfiguration":{"EbsBlockDeviceConfigs":[{"VolumeSpecification":{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"InstanceGroupType":"MASTER","InstanceType":"m4.xlarge","Name":"Master - 1"},{"InstanceCount":2,"BidPrice":"OnDemandPrice","EbsConfiguration":{"EbsBlockDeviceConfigs":[{"VolumeSpecification":{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"InstanceGroupType":"CORE","InstanceType":"m4.xlarge","Name":"Core - 2"}]' --configurations '[{"Classification":"jupyter-s3-conf","Properties":{"s3.persistence.bucket":"cmurill5notebooks","s3.persistence.enabled":"true"}},{"Classification":"hive-site","Properties":{"hive.metastore.client.factory.class":"com.amazonaws.glue.catalog.metastore.AWSGlueDataCatalogHiveClientFactory"}},{"Classification":"spark-hive-site","Properties":{"hive.metastore.client.factory.class":"com.amazonaws.glue.catalog.metastore.AWSGlueDataCatalogHiveClientFactory"}}]' --auto-scaling-role EMR\_AutoScaling\_DefaultRole --ebs-root-volume-size 10 --service-role EMR\_DefaultRole --enable-debugging --auto-termination-policy '{"IdleTimeout":3600}' --name 'cmurill5clustercli' --scale-down-behavior TERMINATE\_AT\_TASK\_COMPLETION --region us-east-1

* **PROCESO DE CREACIÓN DE ETL CON GLUE**

Una de las tablas de la base de datos en Athena tiene un formato donde cada columna es un departamento, para solucionar esto creamos un nuevo formato con la columna departamento. Utilizando glue creamos un job:

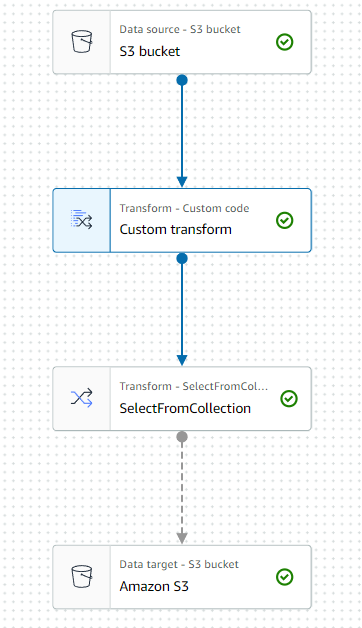


Ilustración 7- proceso ETL

Dentro del job creamos un paso de transformación manual en el cual pivoteamos la tabla como se observa a continuación:

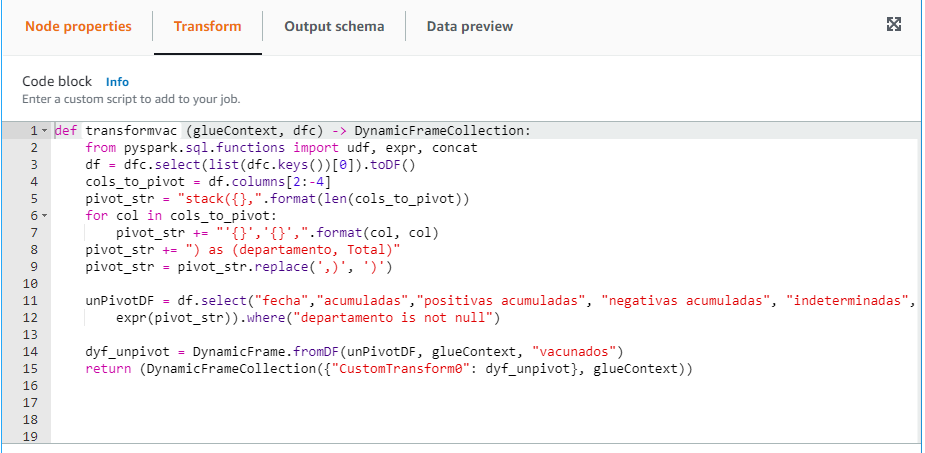
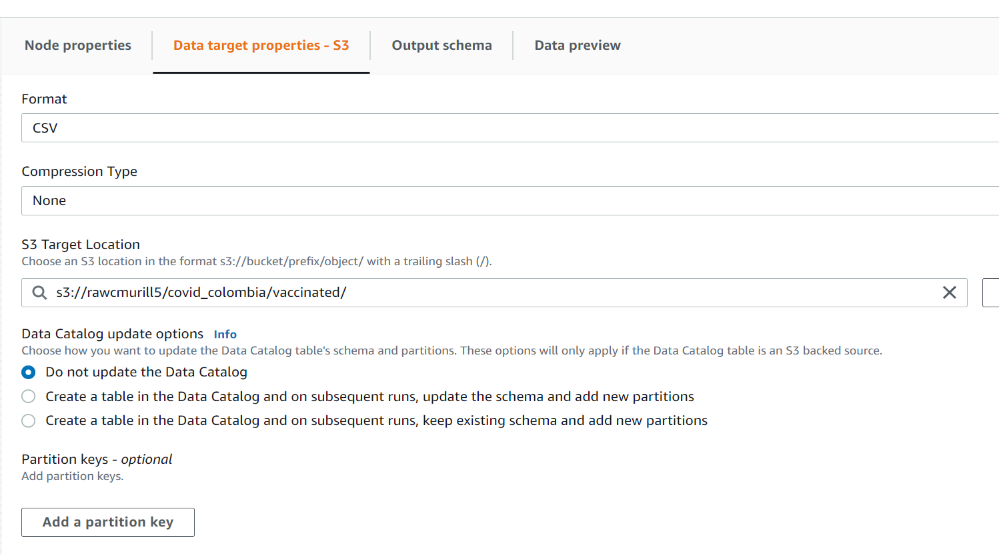


Ilustración 8- JOB para el ETL

Exportamos los datos a s3 y creamos un nuevo crawler para el proceso



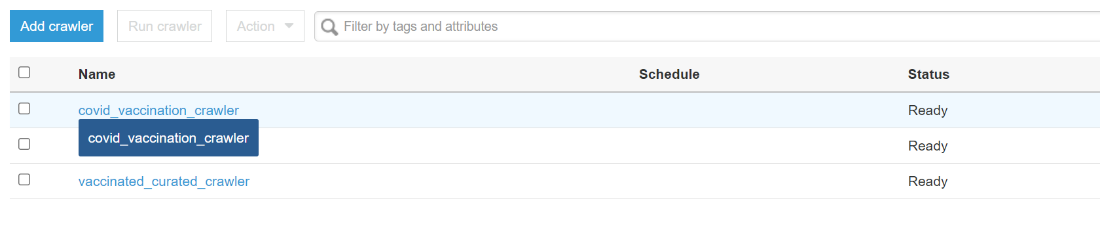


Ilustración 9- crawler para el ETL

Para continuar se crea Covid vaccination crawler, para generar la tabla ya pivoteada, definimos llamarla vaccinated:

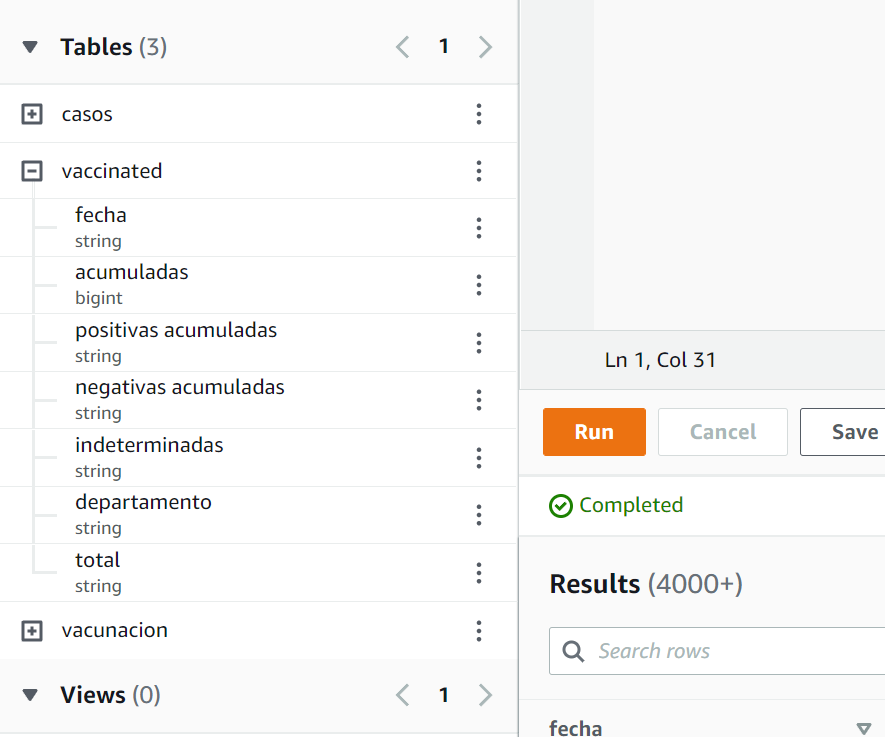


Ilustración 10- tabla pivoteada el ETL

. **CODIGO PYTHON ETL GLUE**

Para poder compartir este proceso con otros usuarios podemos descargar el script en un archivo que se llama custom\_pivot.py, en parte es autogenerado por Glue y en los nodos especiales utilizamos código Python:

Texto

Descripción generada automáticamente

Ilustración 11- código Python ETL

**REPOSITORIO DE INFORMACIÓN**

<https://github.com/cabymetal/sitemas_intensivos_datos>