**UNIVERSIDAD EAFIT**

**ST0263 Tópicos Especiales en Telemática**

**2023-2**

Documentación y Evidencias Laboratorios Unidad 3 – Big Data

**Nombre :** David González Tamayo **- email :** dgonzalez2@eafit.edu.co

LABORATORIO 3-3: Implementación de un data warehouse con AWS REDSHIFT y REDSHIFT SPECTRUM

Para la creación de este laboratorio se basó con la guía dada por el profesor Edwin Nelson Montoya Munera. Para más detalles dar click en el siguiente link:

* <https://docs.aws.amazon.com/redshift/latest/gsg/new-user.html>
* <https://docs.aws.amazon.com/redshift/latest/gsg/console.html>
* <https://docs.aws.amazon.com/redshift/latest/gsg/data-loading.html>

Fuentes de datos: datasets de:

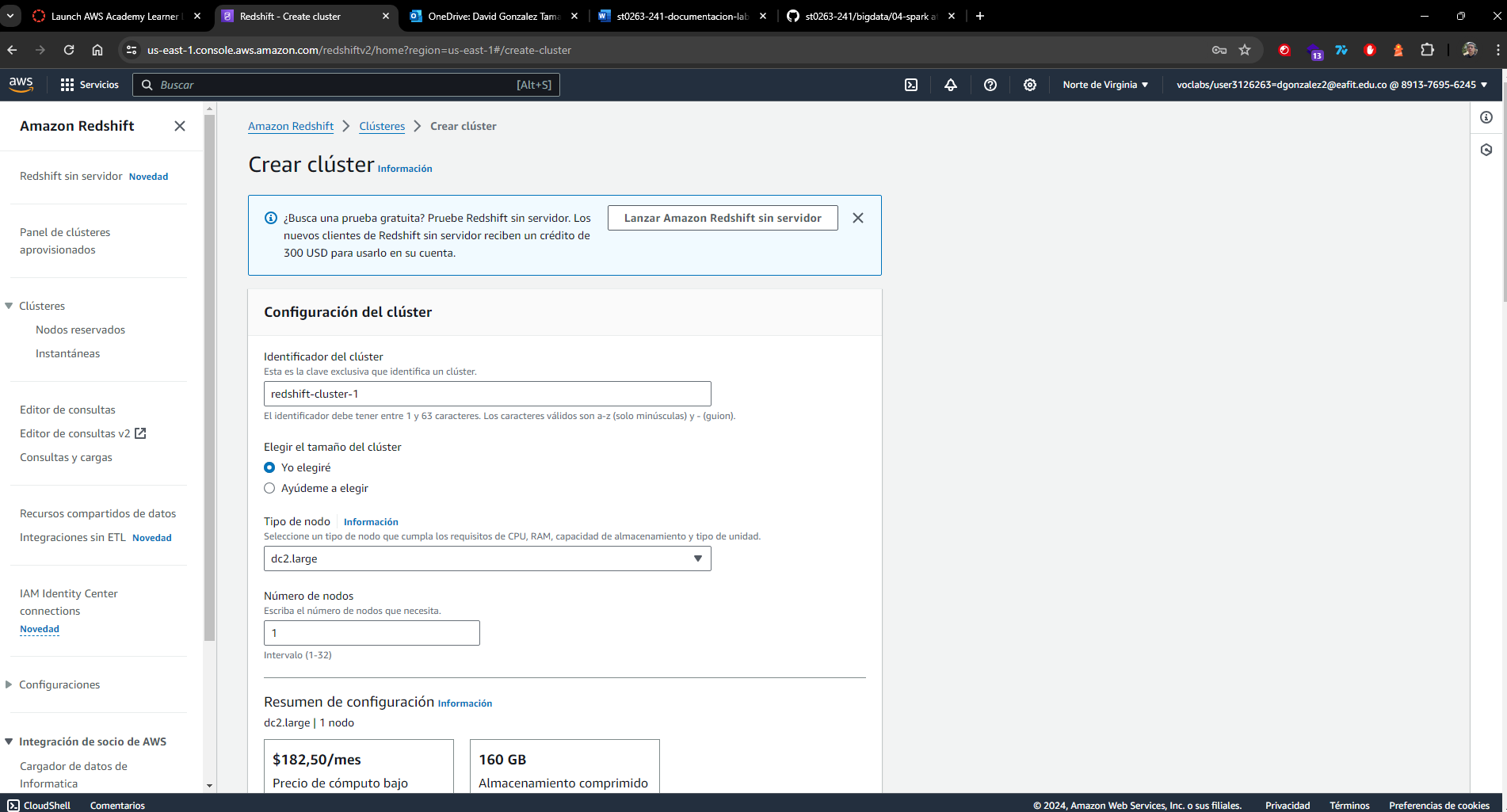
* <https://github.com/st0263eafit/st0263-241.git>

Datos específicos de: tickit

* <https://docs.aws.amazon.com/redshift/latest/gsg/samples/tickitdb.zip>

Lo primero que se hace es crear el cluster de AWS REDSHIFT:

Con la siguiente configuración:



Le asignamos estas identificaciones:

Username: awsuser

Password: St1800232.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Seleccionamos los roles adecuados los cuales son:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Posteriormente se crea el clúster, abrimos el ‘redshift-cluster-1’ y nos saldrá lo siguiente:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Luego le damos a ‘Datos de consulta’ y seleccionamos de tipo v2:

Interfaz de usuario gráfica, Texto, Correo electrónico

Descripción generada automáticamente

Con el cluster una vez ya creado, nos autenticamos con los usuarios y contraseña mencionadas anteriormente:

Captura de pantalla de computadora

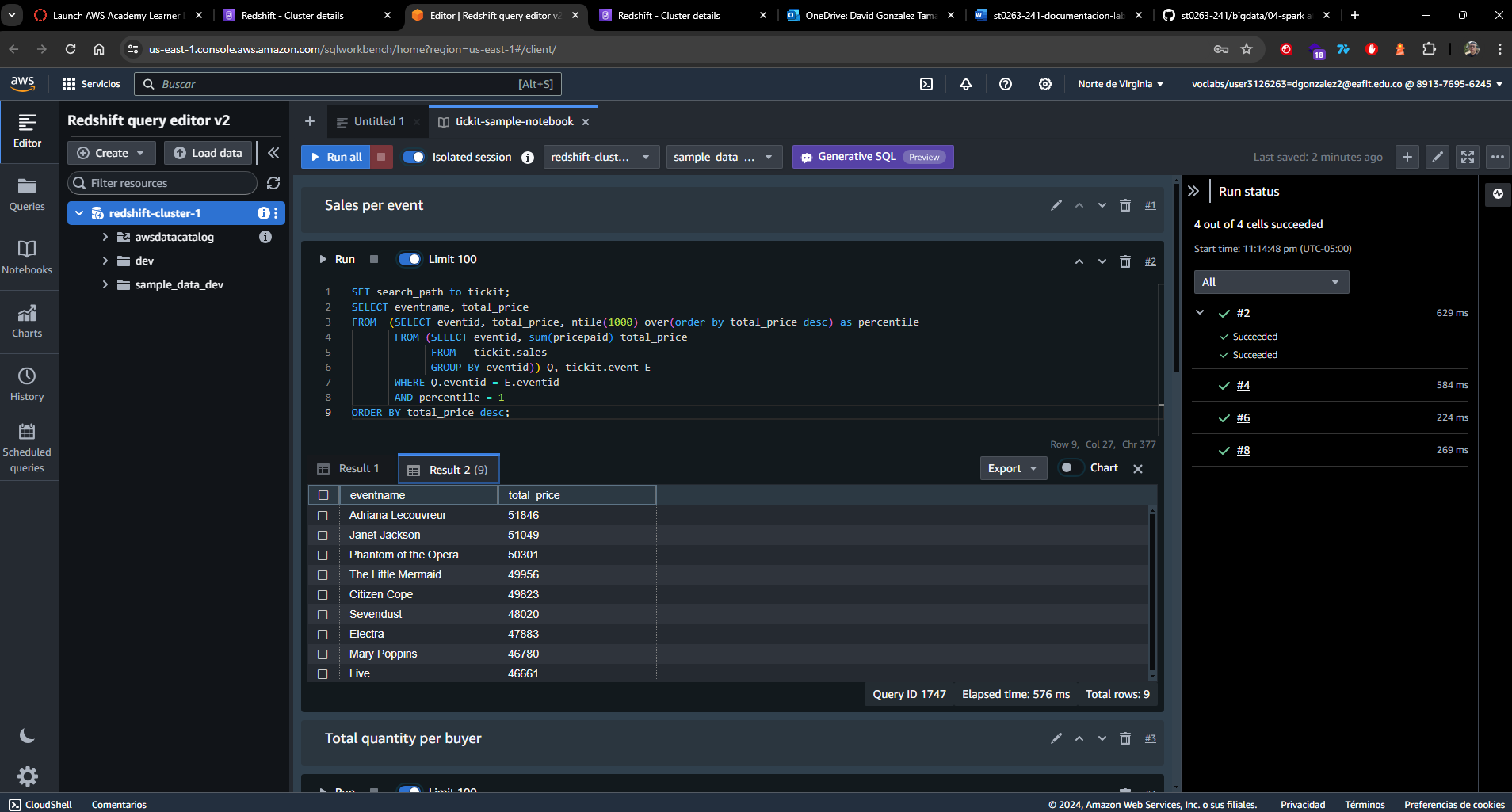
Descripción generada automáticamente

Luego creamos la base de datos con el tickit, lo cual se presiona en la parte de izquierda en la carpeta y le damos ‘create’.

Captura de pantalla de computadora

Descripción generada automáticamente

Ahora podemos ejecutar algunas de las consultas SQL proporcionadas por GENERATIVE SQL.



Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

Para la siguiente parte del laboratorio utilizaremos redshift spectrum, primero vamos a crear el rol necesario para ejecutar el spectrum:

Nos ubicamos en la pagina de inicio de la consola de AWS y seleccionamos ‘IAM’.

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

Luego de acceder al ‘IAM’ buscamos el apartado de ‘Roles’.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Una vez entrado seleccionamos ‘Crear rol’.

Una captura de pantalla de una computadora

Descripción generada automáticamente

Seleccionamos este tipo de entidad de confianza y a su vez en el caso de uso ‘Redshift’.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Seleccionamos estos permisos específicos:

* AmazonS3ReadOnlyAccess
* AWSGlueConsoleFullAccess
* AmazonAthenaFullAccess

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Word, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Le asignamos el siguiente nombre de rol ‘myspectrum\_role’

Captura de pantalla de computadora

Descripción generada automáticamente

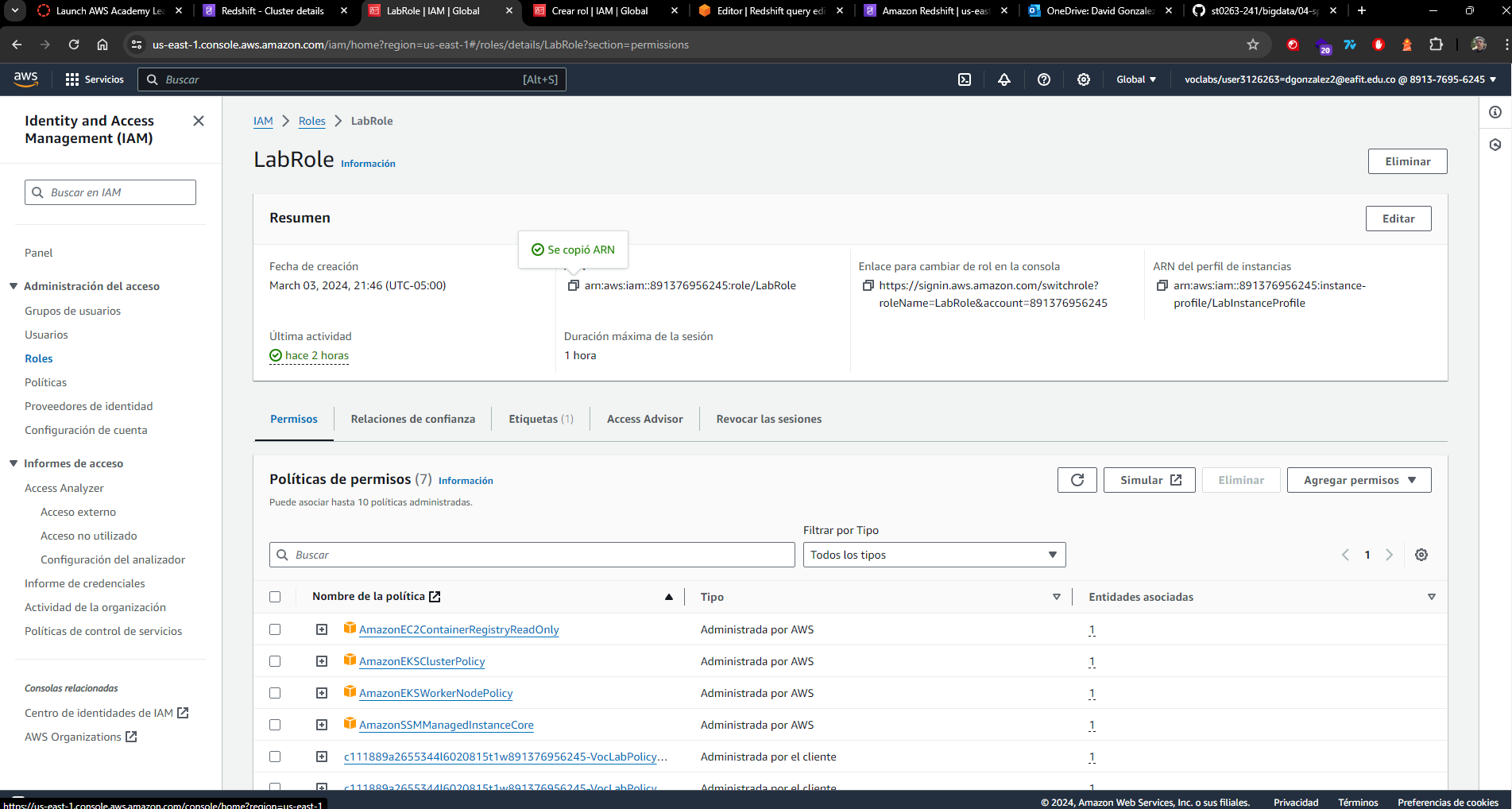
Una captura de pantalla de una computadora

Descripción generada automáticamente

Le damos crear rol y nos sale dicho error:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico, Sitio web

Descripción generada automáticamente



Con el labrole se crea el schema:

create external schema myspectrum\_schema

from data catalog

database 'myspectrum\_db'

iam\_role 'arn:aws:iam::891376956245:role/LabRole'

create external database if not exists;

Captura de pantalla de computadora

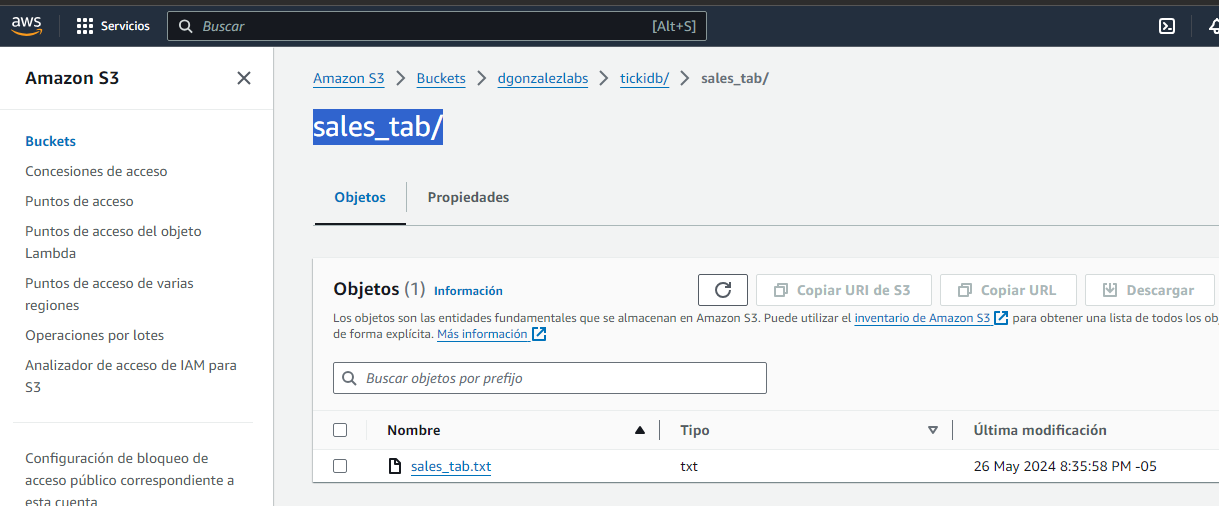
Descripción generada automáticamente

Crear una table con datos externos en S3:

Verificamos en cual Buckets se encuentra tickit:

**Imagen que contiene Escala de tiempo

Descripción generada automáticamente**

****

Una vez verificado en cual Buckets se encuentra tickit se debe cambiar ‘location’ con los datos asignados anteriormente.

create external table myspectrum\_schema.sales(

salesid integer,

listid integer,

sellerid integer,

buyerid integer,

eventid integer,

dateid smallint,

qtysold smallint,

pricepaid decimal(8,2),

commission decimal(8,2),

saletime timestamp)

row format delimited

fields terminated by '\t'

stored as textfile

location 's3://dgonzalezlabs/tickidb/sales\_tab/'

table properties ('numRows'='172000');

Ejecutamos el programa y nos indica lo siguiente:

**Captura de pantalla de computadora

Descripción generada automáticamente**

Se puede observar las tablas que tenemos creadas en la base de datos myspectrum\_db:

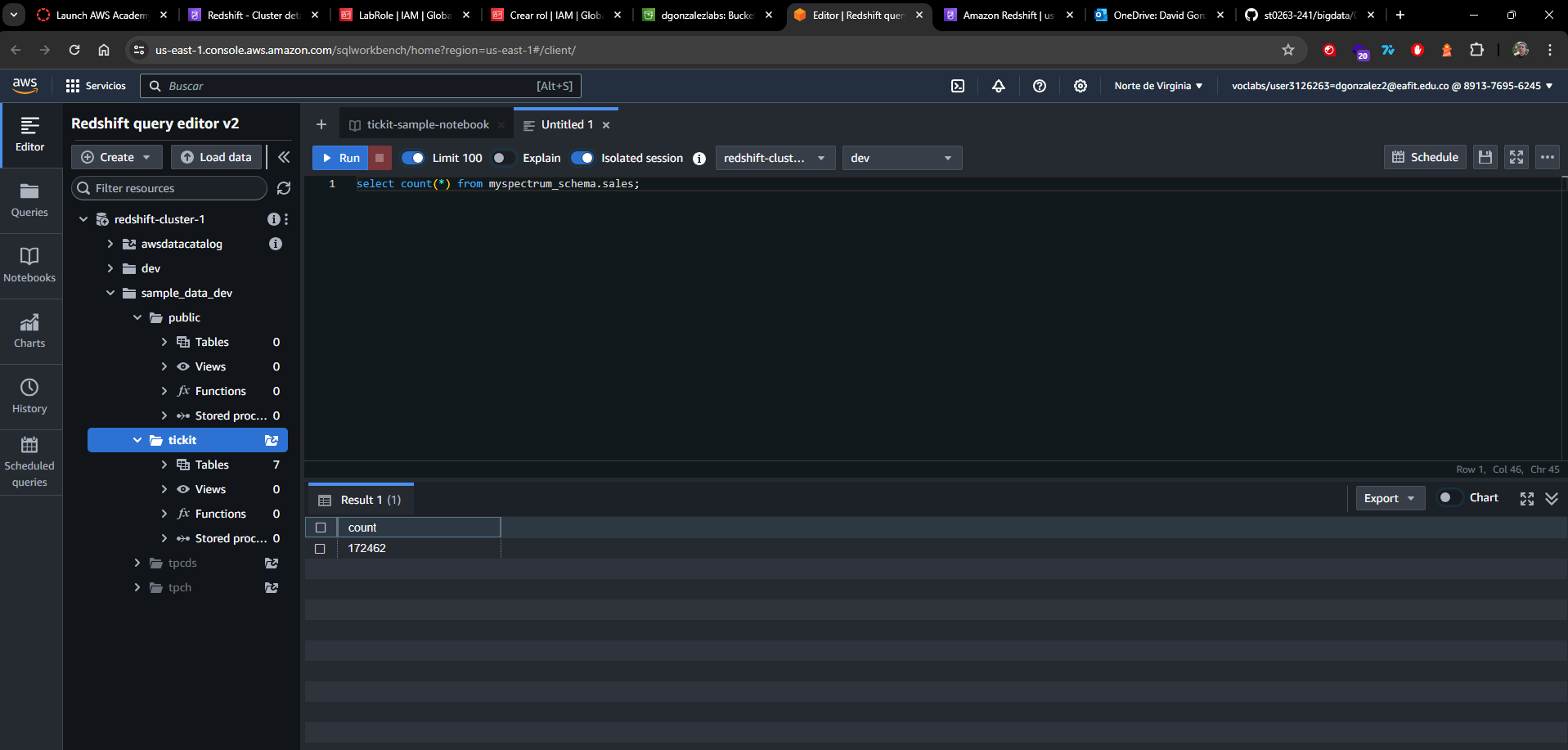
**Texto

Descripción generada automáticamente**

Ahora ejecutaremos ciertas sentencias SQL:

Consultar datos:

select count(\*) from myspectrum\_schema.sales;

****

Crear una tabla nativa en redshit para combinarla con la tabla externa en un query:

create table event2(

eventid integer not null distkey,

venueid smallint not null,

catid smallint not null,

dateid smallint not null sortkey,

eventname varchar(200),

starttime timestamp);

Captura de pantalla de computadora

Descripción generada automáticamente

Cargar datos en la table ‘event2’:

Copiamos el ARN del rol en IAM:

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Posteriormente verificamos como llamamos nuestro buckets donde esta ubicado los eventos:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Con los datos anteriores se cambia la primera línea de código con la ubicación del buckets y la segunda línea el ARN:

COPY event2 FROM 's3://dgonzalezlabs/tickidb/allevents\_pipe/allevents\_pipe.txt'

iam\_role 'arn:aws:iam::891376956245:role/LabRole'

delimiter '|' timeformat 'YYYY-MM-DD HH:MI:SS' region 'us-east-1';



Ahora podemos realizar una consulta con tablas externas y nativas:

select top 10 myspectrum\_schema.sales.eventid, sum(myspectrum\_schema.sales.pricepaid)

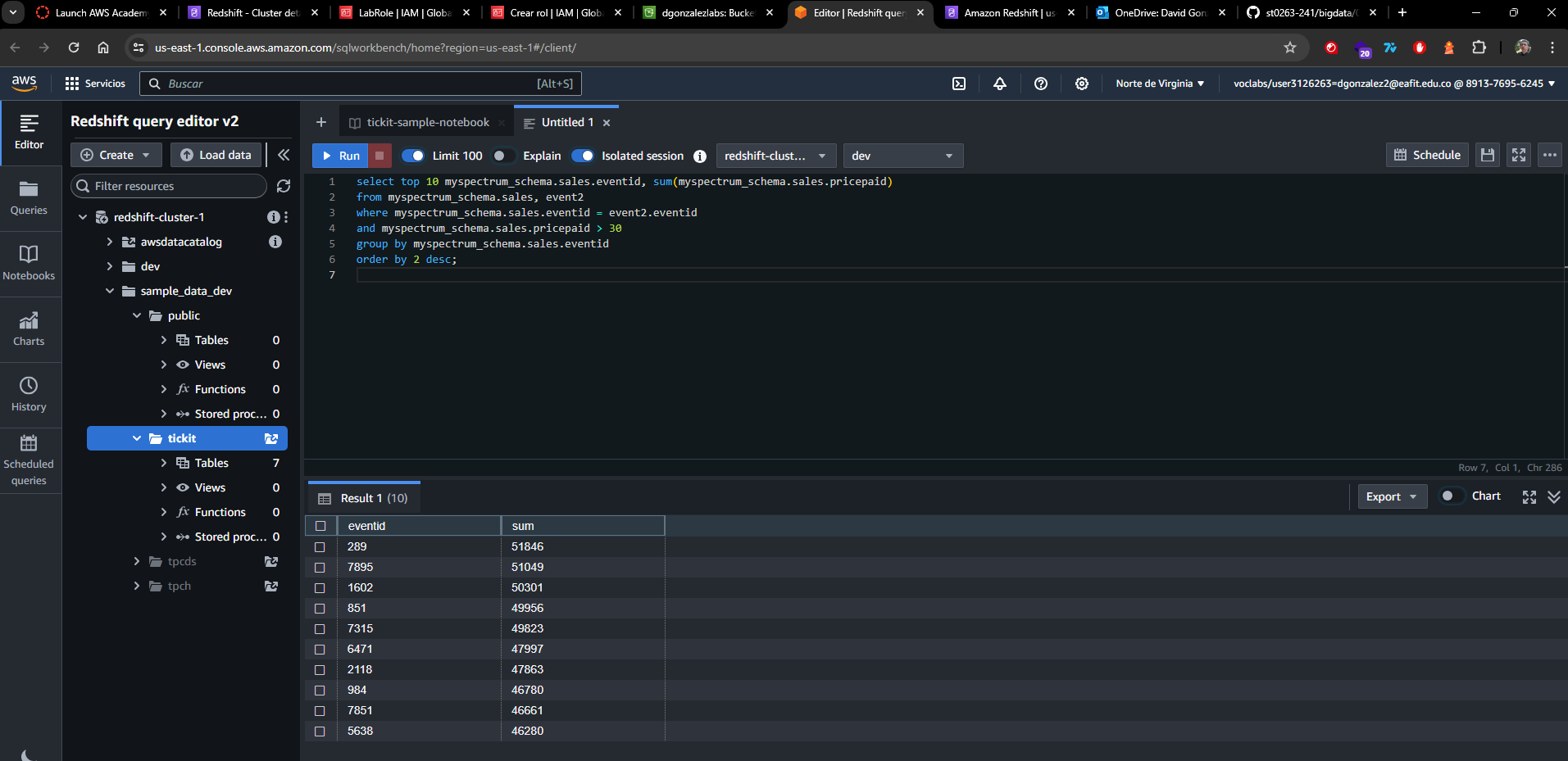
from myspectrum\_schema.sales, event2

where myspectrum\_schema.sales.eventid = event2.eventid

and myspectrum\_schema.sales.pricepaid > 30

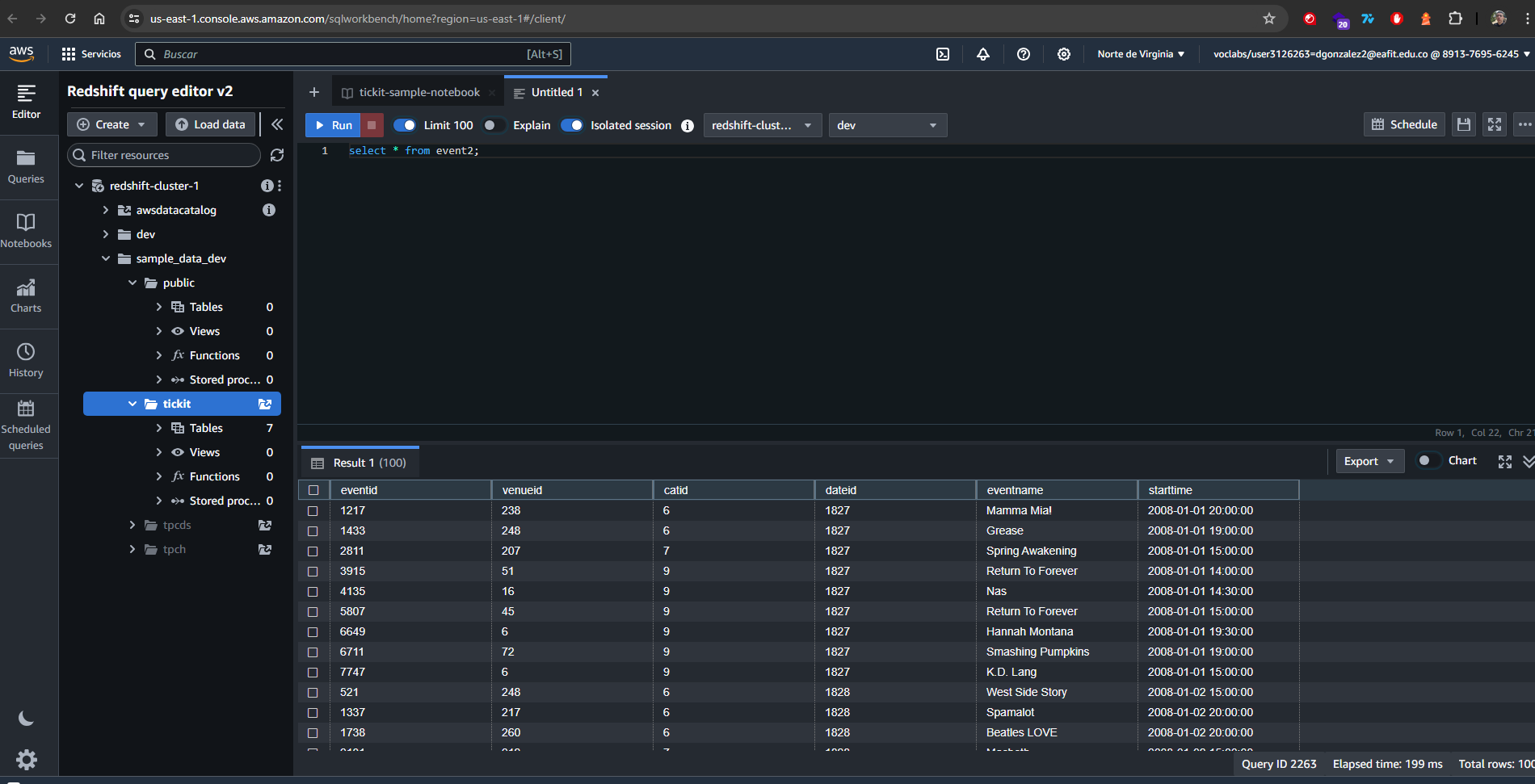
group by myspectrum\_schema.sales.eventid

order by 2 desc;



Si queremos verificar si todo esta bien, realizamos unas consultas simples para mirar que tiene la tabla event2 lo cual se indica con las siguientes peticiones:

select \* from event2;



select \* from event2 where catid = 9;

