CREATING AND UTILIZING A CLOUD COMPUTING ENVIRONMENT

**Introduction**

**Objective:** To understand the fundamentals of S3 buckets in AWS and how to load data into Snowflake.

**Goals:**

1. Create an AWS S3 Bucket and successfully upload data into it from a CSV file.
2. Create a database in Snowflake that could import the data from AWS.
3. Create a stage in Snowflake that imports the data from AWS and successfully use SQL commands to query the data.

**Why I Chose AWS and Snowflake:**

I chose a project focused on AWS and Snowflake to understand the basics of cloud environments and how that relates to storing data. The purpose of this project is to showcase how one can utilize Amazon S3 buckets to store data and then migrate that data into Snowflake for querying and analytical purposes. I am hoping to demonstrate how using cloud environments is much better for database storage and creating data pipelines than using software such as MySQL.

**What data is being used with this project?**

I am using a CSV file downloaded directly from the CMS website regarding nursing homes in America. The file contains almost 100 columns of information including anything from ratings, penalties, location, etc. The goal will be to successfully create an S3 bucket in AWS using the CSV file, and then successfully load it into Snowflake for analysis purposes.

**Benefits of completing this project:**

Cloud computing and cloud data storage with repositories such as data warehouses, data lakes, and data marts are becoming highly desired for many companies. With companies growing faster than ever and billions of data records becoming available each day, its becoming more important for companies to keep up with the influx of data and keeping it organized and readily available. My hope in completing this project is that will show how quickly and easily data can be stored in AWS and how a few simple lines of code can be used to illustrate the data in an environment such as Snowflake for analytics purposes.

**Methodology**

**Step 1: Create an AWS S3 Bucket for the Nursing Home data from the CSV file:**

The first step in the process was to create what is known as an “AWS S3 Bucket” to store the data from the Nursing Home CSV file that I downloaded.

Graphical user interface, text, application, email

Description automatically generated

I created this bucket above by choosing “Create Bucket” from the Amazon S3 console and answering some security questions regarding Object Ownership, Public Access, Version control, and encryption.

Graphical user interface, text, application, email

Description automatically generated

Once that was done, I uploaded the data into the bucket by choosing “Upload” in the top right section of the screen. This gave me a basic screen where I can just drag and drop the CSV file into the page, and it would upload the file to my bucket.

Next, I would be able to click on the “nursinghomedata” link (shown above) and then the “NH\_ProviderInfo\_Jan2023.csv” link. This takes me to the following screen which gives me the S3 URI that I’ll need to create the stage in Snowflake.

Graphical user interface, text, application

Description automatically generated

**Step 2: Create a database in Snowflake along with a Stage to import the S3 Bucket:**

The next step in the process was to create a database in Snowflake which would operate as our computing platform and environment for importing data into the database. This first step was fairly easy and only required the following steps:

1. Choose “Data”- “Databases” from the Home page.
2. Click “ + Database” in the top right.
3. Name the Database

Next, the option to create a Schema would be available to store the different tables or stages associated with the Database. The purpose of a schema in Snowflake is to organize database objects (such as views, tables, stages, etc.) into separate namespaces to provide a logical grouping of data objects. In this case, I chose to use Snowflakes default “Public” schema because this database won’t be very complex and won’t require any kind of access control.

From here, my next step was to open a blank worksheet and create a Stage from scratch as shown below.

Graphical user interface, text

Description automatically generated

The SQL commands I used were:

* “CREATE OR REPLACE STAGE” – used to create the stage
* “url=” - setting the url from AWS to pull data from.
* “file\_format – cleaning up the formatting the data that’s pulled into Snowflake
* CREDENTIALS: credentials needed to access S3 bucket

A picture containing chart

Description automatically generated

In the above step, I am listing out the different files that are available on the S3 bucket to confirm the URL is working properly and communicating with Snowflake properly.

Graphical user interface, text

Description automatically generated

In this step, I am creating the table for Nursing Home data. The table is created by including the database, schema, and table, along with the different attributes and their data types.

Text

Description automatically generated

In the above step, I am migrating the data from AWS to Snowflake via the stage “@my\_s3\_stage”. The lines of code include the following:

* COPY INTO – Copying into a specific table created in Snowflake (database, schema, table)
* FROM – Copying from a specific location. In this case, it is from specific columns of the CSV file from the s3 stage. For instance, $1 represents column #1, or in this case, the “Federal Provider Number.”
* FILE FORMAT- Type = CSV file, Field\_Delimiter is a comma, and Skip\_Header = 1 says to recognize that the top row are only header values.
* ON\_ERROR- This represents what Snowflake should do in the event an particular piece of data cannot be imported. For example, some rows may have data that cannot be read or are ‘N/A’ values. We can use the command “CONTINUE” to skip over these pieces of data.

Text

Description automatically generated

In this last step, I am writing a simple select \* statement and a simple SELECT, FROM, WHERE statement to confirm the data populated correctly and ensure everything looks good. Here is the result of the WHERE statement:

Graphical user interface, text, application, email

Description automatically generated

**Conclusion**

There were a few keys points with this project as it relates to loading data into Snowflake from AWS.

1. All 15,108 rows of data did not successfully populate into the database. I only managed to load about 8,100 rows. This is largely because many rows of data needed to be cleaned before I imported them into the S3 bucket. To give an example, I noticed there were many Federal Provider Numbers listed like “2.80E+192” which I could have addressed before loading the data into AWS.
2. It’s difficult to keep track of specific columns that you want to pull in from S3 buckets. For this project, I would’ve loved to continue and do a full analysis of the available data, but there were so many columns to keep track of, especially since I was only interested in 11 specific columns. In the future, it may be easier to prep a CSV file ahead of time that has a build of exactly what I want to load in Snowflake.
3. Automating this process can make pulling reports incredibly easy in the future. There are so many times that I need to pull different excel sheets in together with V-lookups and other formulas just to get the data I need. By building databases and tables in AWS and Snowflake, however, obtaining these reports will be much easier and save so much time.