## Building an ETL Pipeline in AWS

## 0.0 Description and Objective

We connected to an external data source for this project, loading it into a MySQL instance and then visualizing the data through the Tableau desktop version. The data set we used was the 2019 IRS submissions from the IRS 990 database.

Here are the steps to follow:

* Set-up the infrastructure and create the database
* Load the data in S3 and test the Glue Crawler
* Create the connections and AWS ETL job using Glue
* Create the visualization

The background of this dataset can be found in the following documentation:

* <https://docs.opendata.aws/irs-990/readme.html>
* <https://aws.amazon.com/opendata/public-datasets/>

Objective:

Generated a CSV file from public data source (IRS 990) data and load it into a database that we created (S3 bucket), which was propagated to a MySQL database for connection to Tableau.

## 1.0 Set up the Infrastructure and Create the Database

* 1. First, created a database (‘database-3’) in Amazon RDS

Graphical user interface, application

Description automatically generated

* 1. Once created, test the connection using our database credentials and connection string from MySQL workbench[[1]](#footnote-2)

Graphical user interface, text, application, chat or text message

Description automatically generated

## 2.0 Load the data in S3 bucket

Now that we have set up the database connection, we created an S3 bucket called ‘**irs-990-lab4’**. Then, Configure the bucket so that it is publicly accessible.

Graphical user interface, text, application

Description automatically generated

## Create the connections and AWS ETL job using Glue

* 1. Now that we have created our S3 bucket, we set up an AWS Glue Crawler and connected it to the AWS database we created in MySQL. This Crawler runs a check on the S3 bucket we made and extracts and loads the file from IRS into your MySQL instance.
     1. Note: we created a Glue Role with the following privileges:

|  |
| --- |
| AmazonRDSFullAccess |
| [AmazonRDSDirectoryServiceAccess](https://console.aws.amazon.com/iam/home?region=us-east-1#/policies/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2Fservice-role%2FAmazonRDSDirectoryServiceAccess) |
| [AWSGlueServiceRole](https://console.aws.amazon.com/iam/home?region=us-east-1#/policies/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2Fservice-role%2FAWSGlueServiceRole) |
| [AWSGlueServiceNotebookRole](https://console.aws.amazon.com/iam/home?region=us-east-1#/policies/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2Fservice-role%2FAWSGlueServiceNotebookRole) |
| [AdministratorAccess](https://console.aws.amazon.com/iam/home?region=us-east-1#/policies/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2FAdministratorAccess) |
| [AmazonRDSDataFullAccess](https://console.aws.amazon.com/iam/home?region=us-east-1#/policies/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2FAmazonRDSDataFullAccess) |
| [AWSGlueConsoleFullAccess](https://console.aws.amazon.com/iam/home?region=us-east-1#/policies/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2FAWSGlueConsoleFullAccess) |
| [AWSGlueConsoleSageMakerNotebookFullAccess](https://console.aws.amazon.com/iam/home?region=us-east-1#/policies/arn%3Aaws%3Aiam%3A%3Aaws%3Apolicy%2FAWSGlueConsoleSageMakerNotebookFullAccess) |
| AmazonS3FullAccess |

* + 1. Before connecting to mysql instance, the inbound rules of security should be updated. If we do not add all tcp type and choose ‘anywhere, 0.0.0.0/0‘ it will cause below error.

Graphical user interface, text, application

Description automatically generated

For avoiding this error, we need follow below instructions.

RDS 🡪 database-3 🡪 VPC security group 🡪 inbound rules tab 🡪 Edit inbound rules 🡪

Graphical user interface, application

Description automatically generated

Also , we need to add endpoint to avoid the other error .

Graphical user interface, text, application

Description automatically generated

Add ENDPOINTS instructions:

VPC 🡪 endpoints 🡪create endpoints 🡪

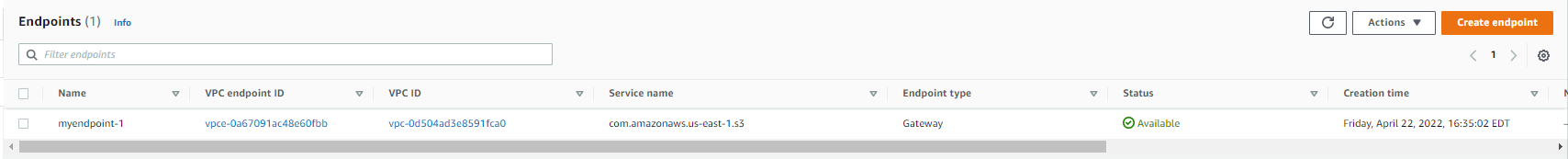
Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated



After we added endpoint and updated the security type, we successfully connected to database.

Text, application

Description automatically generated

|  |  |
| --- | --- |
|  |  |

Then, Create a Data Store called ‘irs-990’ and link it to our database using the jdbc connection string

|  |  |
| --- | --- |
| **Location** | jdbc:mysql://database-3.cv8ht0hlsqz8.us-east-1.rds.amazonaws.com:3306/irs-990-2 |

Graphical user interface, text, application, email, website

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

* 1. Create a Classifier to specify the structure of the file set being created and use the database name to create a schema in our own AWS RDS MySQL instance

Click right and create schema 🡪 update name that same as AWS database name

Graphical user interface, application

Description automatically generated

* + 1. *Name: irs-990-classifier-for-public-dataset*

Graphical user interface, text, application, email

Description automatically generated

* 1. Create the Crawler under AWS Glue left-hand menu items. The Crawler you will create will crawl the public IRS 990 s3 bucket with the following path:
     1. s3://irs-990-lab4/index\_2019.csv
     2. *Name: irs-990-crawl-public-dataset*

*Graphical user interface

Description automatically generated with medium confidence*

* 1. Once we have created the Crawler, run the Crawler to the specified location in ourS3 bucket (‘irs-990’). The Crawler pulls the file into S3 location. Note: this may take several minutes to execute.
  2. Once we have crawled the irs-form-990 database, check to see that the database and table was successfully created by navigating to ‘Tables’ under the Glue service screen.

Graphical user interface, text, application

Description automatically generated

* 1. Once confirmed, now we can create a ‘Job’ to move the data from this location (your S3 bucket), to the MySQL database. we can use the Spark/Python Shell to autogenerate the script using the Glue interface.
  2. *Name: glue-to-rds-jobGraphical user interface, text, application, email

     Description automatically generated*

Graphical user interface, text, application

Description automatically generated

Once we have created the Job, run it and check that the database was properly updated with the table. Our result looks like this:

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

## Create Connection and Visualization

Connect to our Tableau Desktop instance. We will connect to the IRS 990 database we created in our AWS Database Instances.

### Connect Tableau Desktop to AWS RDS

To create a connection to ourAWS RDS MySQL instance, we will follow a very similar process to connecting the MySQL Workbench. Graphical user interface, text, application, email

Description automatically generated

### Creating visualizations

Chart, bar chart

Description automatically generated

1. For the RDS MySQL configuration, only MySQL v5 is compatible with [AWS Glue.](https://forums.aws.amazon.com/thread.jspa?threadID=306814) [↑](#footnote-ref-2)