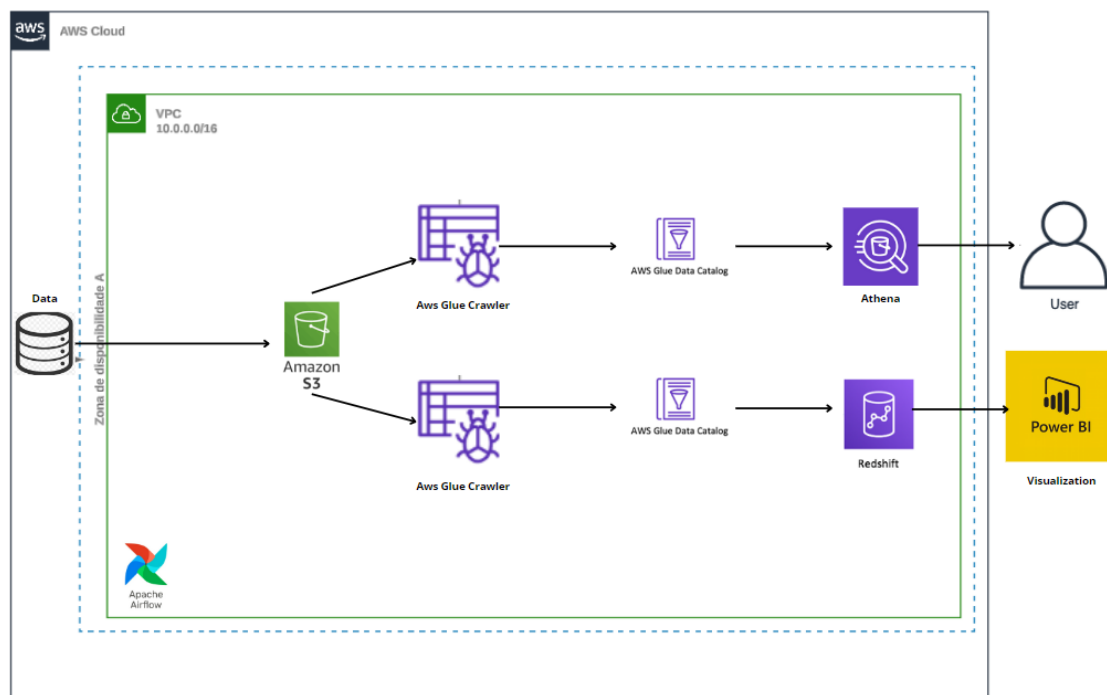


# Customer Churn Analysis through a Data Pipeline using Apache Airflow, AWS Glue, S3, Amazon Redshift and Power BI

## Summary

In this data engineering project, I performed a customer churn analysis, built, and automated an ETL pipeline using AWS Glue to load data from the AWS S3 bucket into an Amazon Redshift data warehouse and then connect Power BI to the Redshift cluster for end user view. AWS Glue served as a data crawler to infer the database schema after the data was made available in AWS Athena to extract the data through SQL queries. AWS Glue also served to upload the tracked and processed data to the Redshift cluster. Apache Airflow was used to orchestrate and automate this entire process that was previously manual.

AWS cloud architecture diagram:



The entire project was developed from scratch in the AWS cloud, a database used by Kaggle, called “Telecommunications customer turnover: IBM dataset”, below are the steps.

First, I created an EC2, for process configurations and orchestration. In the AWS console, search for EC2, then “Run EC2”. The AMI chosen was Ubuntu, the initial configurations will be made in Linux, via the EC2 terminal. I enabled the 3 incoming connection rules SSH, HTTPS and HTTP, so it will be possible to connect via command line and browser, as the initial load will be low, choosing the

t2.medium instance type, which has 4Gib and 2 CPUs, this guarantees that EC2 will not freeze during the process.

EC2 > Instâncias > Launch an instance

## Launch an instance [Informações](#)

O Amazon EC2 permite criar máquinas virtuais, ou instâncias, que são executadas na Nuvem AWS. Comece a usar rapidamente seguindo as etapas simples abaixo.

### Nome e tags [Informações](#)

Nome

[Adicionar mais tags](#)

I also generated a pair key, downloaded the pair key in csv, for security reasons and to be able to connect via SSH, in addition, a VPC was created to protect EC2.



When the instance is in “Running” status, select it and click connect. At first, I connected via the connect instance to start the settings. Click “Connect” again.

## Conectar-se à instância Informações

Conecte-se à sua instância i-00cf8d0b9a51b1c2c (ec2-customer-churn-camila) usando qualquer uma destas opções

Conexão de instância do EC2

Gerenciador de sessões

Cliente SSH

Console de série do EC2

ID da instância

i-00cf8d0b9a51b1c2c (ec2-customer-churn-camila)

Tipo de conexão

☒ Conectar-se usando o EC2 Instance Connect  
Conecte-se usando o cliente baseado em navegador do EC2 Instance Connect, com um endereço IPv4 público.

☐ Conectar-se usando o endpoint do EC2 Instance Connect  
Conecte-se usando o cliente baseado em navegador do EC2 Instance Connect, com um endereço IPv4 privado e um endpoint da VPC.

Endereço IP público

44.204.116.233

Nome de usuário

Insira o nome de usuário definido na AMI usada para iniciar a instância. Se você não definiu um nome de usuário personalizado, use o nome de usuário padrão, ubuntu.

**Observação:** na maioria dos casos, o nome de usuário padrão, ubuntu, está correto. No entanto, leia as instruções de uso da AMI para verificar se o proprietário da AMI alterou o nome de usuário da AMI padrão.

Cancelar

Conectar

A new terminal will open, I configured it with the following Linux commands:

**sudo proper update** (Updates a list of packages available for installation on the operating system using the APT (Advanced Package Tool) package manager )

**sudo apt install python3-pip** (Install the python3-pip package, which is the Python package manager for version 3.x, used to install and manage Python libraries and dependencies.)

**sudo apt install python3.10-venv** (Install the python3.10-venv package, which is required to create Python 3.10 virtual environments, allowing you to isolate Python development environments for specific projects.)

**python3 -m venv client\_churn\_camila\_venv** (Create a virtual environment called customer\_churn\_camila\_venv using the Python 3 venv module. This virtual environment is where the project's dependencies will be installed and isolated from the global system.)

**source customer\_churn\_camila\_venv /bin/ activate** (Activates the previously created virtual environment, ensuring that subsequent installations and executions occur within this isolated environment.)

**sudo pip install apache-airflow** (Install Apache Airflow, a platform for scheduling, monitoring, and managing data workflows. Here, we are using pip (Python Package Installer) to install Airflow.)

**pip install apache-airflow-providers-amazon** (Install a specific Apache Airflow provider for integration with Amazon Web Services (AWS) services such as S3, EC2, etc. This provides additional functionality for working with AWS services within Apache Airflow.)

**airflow standalone** (Start Apache Airflow in standalone mode, which means it will run on a single node without using an external database such as MySQL or PostgreSQL. This mode is useful for development or testing setups.)

After configuring the virtual environment and dependencies, Airflow should indicate that it is ready and will provide a username and password, as follows:

```
standalone | Airflow is ready
standalone | Login with username: admin password: 9xYkerwf6v3v97qE
```

My username: admin password : 9xYkerwf6v3v97qE.

After that, go back to the created EC2, select and in the “Security” tab, click on Security groups, then on Inbound Rules and Edit Inbound Rules, add the rule to access Apache Airflow via browser, this will be important to view whether the job you created is working correctly.

Editar regras de entrada [Informações](#)

As regras de entrada controlam o tráfego de entrada que tem permissão para acessar a instância.

ID da regra do grupo de segurança	Tipo <a href="#">Informações</a>	Protocolo <a href="#">Informações</a>	Intervalo de portas <a href="#">Informações</a>	Origem <a href="#">Informações</a>	Descrição - opcional <a href="#">Informações</a>	
sgr-067bdd7bca2868021	HTTPS	TCP	443	Persona...	Q	Excluir
					0.0.0.0/0 X	
sgr-007ef2de33d7df5a7	SSH	TCP	22	Persona...	Q	Excluir
					0.0.0.0/0 X	
sgr-0b7c50379232e920f	HTTP	TCP	80	Persona...	Q	Excluir
					0.0.0.0/0 X	
-	TCP personalizado	TCP	8080	Qualqu...	Q	Excluir
					0.0.0.0/0 X	

[Adicionar regra](#)

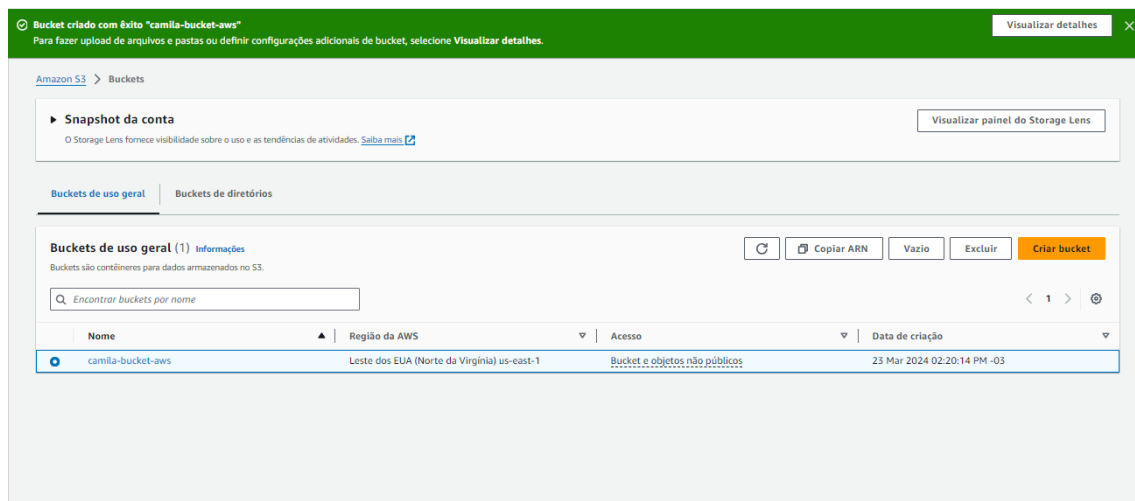
Add the rule, Custom TCP type, port range 8080, and Source Anywhere IPv4. Port 8080 is a network communication port. In many cases, it is used as the default port for local or development web servers. When you run Apache Airflow with the **airflow standalone command**, by default it uses port 8080 to provide a web interface where you can monitor and manage workflows. So, when you start Airflow with **airflow standalone**, you can access its web interface by typing **http://localhost:8080** in the browser. Our localhost is our EC2 public IP address. After logging into Airflow through the browser with the credentials he had created.

DAG	Proprietário	Corre	Agendar
conjunto de dados_consumes_1 consome agendado pelo conjunto de dados	fluxo de ar	0000	Conjunto de dados
conjunto de dados_consumes_1_and_2 consome agendado pelo conjunto de dados	fluxo de ar	0000	Conjunto de dados
dataset_consumes_1_never_scheduled consome agendado pelo conjunto de dados	fluxo de ar	0000	Conjunto de dados
dataset_consumes_unknown_never_scheduled agendado por conjunto de dados	fluxo de ar	0000	Conjunto de dados
conjunto de dados_produces_1 produção programada por conjunto de dados	fluxo de ar	0000	@diário

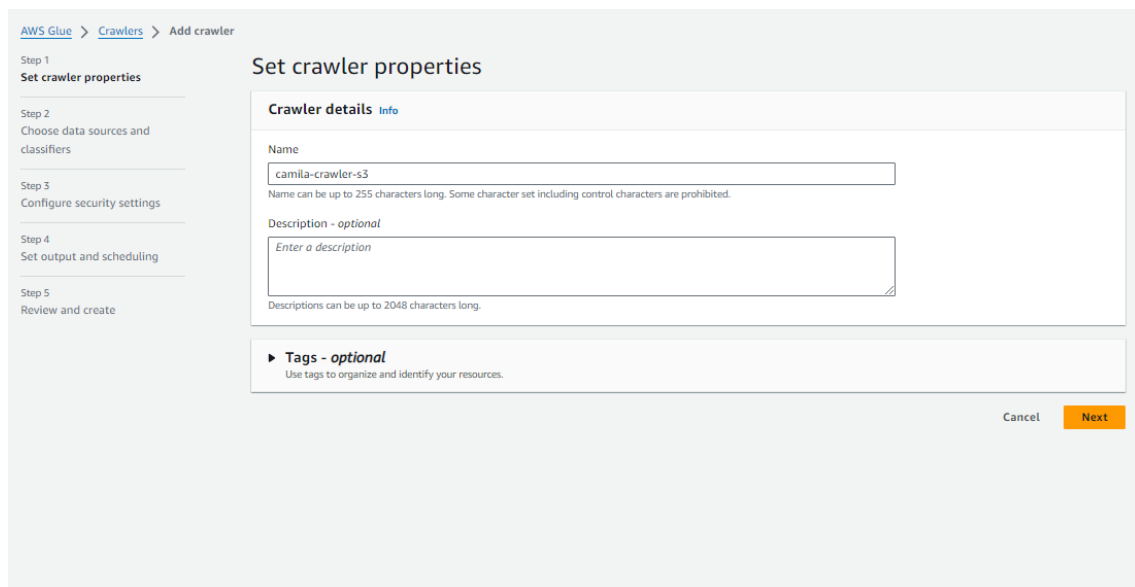
Then I set up remote access via SSH through VScode, because it's more user-friendly to use VIM or the EC2 terminal. In the footer on the left, “Open a remote window”, connect to the Host, inform your EC2 and connect to the virtual environment created previously.



S3 bucket, it is like a cloud storage folder where you can store any type of data, such as files, documents, images, videos, database backups, among others. I stored the data in the S3 bucket.



Then I created AWS Glue, which provides an easy way to create, manage, and run ETL (Extract, Transform, Load) pipelines to process and transform large volumes of data. I used the resource called AWS Glue Crawler, it automates the discovery and classification of data in various data sources, such as Amazon S3, related and non-relational databases, so it will also be used with Athena and Redshift. We investigate using AWS Glue, then in “Data catalog”, “Crawlers” we will process the data that will be stored in S3.



As my table schema was not created to do zero.

## Choose data sources and classifiers

**Data source configuration**

Is your data already mapped to Glue tables?

☒ Not yet  
Select one or more data sources to be crawled.

☐ Yes  
Select existing tables from your Glue Data Catalog.

**Data sources (1)** [Info](#)

EditRemoveAdd a data source

The list of data sources to be scanned by the crawler.

Type	Data source	Parameters
<input type="radio"/> S3	s3://camila-bucket-aws	Recrawl all

► **Custom classifiers - optional**

A classifier checks whether a given file is in a format the crawler can handle. If it is, the classifier creates a schema in the form of a StructType object that matches that data format.

CancelPreviousNext

I added the data source, which will be the contents of the S3 bucket.

**Add data source** ×

**Data source**

Choose the source of data to be crawled.

S3 ▼

**Network connection - optional**

Optionally include a Network connection to use with this S3 target. Note that each crawler is limited to one Network connection so any other S3 targets will also use the same connection (or none, if left blank).

▼ ↺

Clear selectionAdd new connection ↗

**Location of S3 data**

☒ In this account

☐ In a different account

**S3 path**

Browse for or enter an existing S3 path.

✕

View ↗

Browse S3

All folders and files contained in the S3 path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

**Subsequent crawler runs**

This field is a global field that affects all S3 data sources.

☒ Crawl all sub-folders  
Crawl all folders again with every subsequent crawl.

☐ Crawl new sub-folders only  
Only Amazon S3 folders that were added since the last crawl will be crawled. If the schemas are compatible, new partitions will be added to existing tables.

☐ Crawl based on events  
Rely on Amazon S3 events to control what folders to crawl.

CancelAdd an S3 data source

It will be necessary to inform the S3 path and I set it to “Crawl all subfolders” because I want the bucket data to be updated with each new load.

The AWS Glue Crawler determines that the IAM role must already be created to operate the process, so it adds the administrator user permission to manage Glue.

### Configure security settings

**IAM role** [Info](#)

Existing IAM role  

aws\_glue\_administrative\_role ▼ ↺

View ↗

Create new IAM role Update chosen IAM role

Only IAM roles created by the AWS Glue console and have the prefix "AWSGlueServiceRole-" can be updated.

**Lake Formation configuration - optional**  
Allow the crawler to use Lake Formation credentials for crawling the data source. [Learn more.](#) [↗](#)

☐ Use Lake Formation credentials for crawling S3 data source  
Checking this box will allow the crawler to use Lake Formation credentials for crawling the data source. If the data source is registered in another account, you must provide the registered account ID. Otherwise, the crawler will crawl only those data sources associated to the account. Only applicable to S3, Glue Catalog, Iceberg, and Hudi data sources.

**► Security configuration - optional**  
Enable at-rest encryption with a security configuration.

Cancel Previous **Next**

I added a database that I had already added to the bucket.

### Set output and scheduling

**Output configuration** [Info](#)

Target database  

customer-churn-s3-glue-database ▼ ↺

Clear selection Add database ↗

Table name prefix - optional  

Type a prefix added to table names

Maximum table threshold - optional  
This field sets the maximum number of tables the crawler is allowed to generate. In the event that this number is surpassed, the crawl will fail with an error. If not set, the crawler will automatically generate the number of tables depending on the data schema.  

Type a number greater than 0

**► Advanced options**

**Crawler schedule**  
You can define a time-based schedule for your crawlers and jobs in AWS Glue. The definition of these schedules uses the Unix-like cron [↗](#) syntax. [Learn more](#) [↗](#).

Frequency  

On demand ▼

Cancel Previous **Next**

I kept the other default settings and reviewed and then created, then press “Run” and wait until it reaches “Ready” status.



Crawler successfully starting

The following crawler is now starting: "camila-crawler-s3"

AWS Glue

>

Crawlers

Crawlers

A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

Crawlers (1/1) Info

Last updated (UTC)  
March 23, 2024 at 17:46:33

Refresh

Action

Run

Create crawler

View and manage all available crawlers.

Filter crawlers

<

1

>

<input checked="" type="checkbox"/>	Name	State	Schedule	Last run	Last run timestamp	Log	Table changes from ...
<input checked="" type="checkbox"/>	<a href="#">camila-crawler-s3</a>	Ready		Succeeded	March 23, 2024 at 17:...	<a href="#">View log</a>	1 created

Now we will use the Crawler created to connect with AWS Athena. AWS Athena is an interactive data query service provided by AWS that allows you to analyze directly in Amazon S3 using standard SQL. When loading data into a database or traditional data store, Athena allows you to run SQL queries directly against files stored in S3, so you don't need to set up and manage a database infrastructure. The queries I made to create a report for the end user must be stored in another S3 Bucket, which is why I created another bucket. After creating the S3, I can adjust the Athena settings.

Amazon S3 > Buckets > Criar bucket

## Criar bucket [Informações](#)

Buckets são contêineres para dados armazenados no S3.

### Configuração geral

Região da AWS

Leste dos EUA (Norte da Virgínia) us-east-1 ▼

Tipo de bucket [Informações](#)

☒ **Propósito geral**

Recomendados para a maioria dos casos de uso e padrões de acesso. Os buckets de uso geral são do tipo original do S3. Eles permitem uma combinação de classes de armazenamento que armazenam objetos de maneira redundante em várias zonas de disponibilidade.

☐ **Diretório - Novo**

Recomendados para casos de uso de baixa latência. Esses buckets usam somente a classe de armazenamento do S3 Express One Zone, que fornece processamento mais rápido de dados em uma única zona de disponibilidade.

Nome do bucket [Informações](#)

glue-query-bucket-camila

O nome do bucket deve ser exclusivo no namespace global e seguir as regras de nomenclatura do bucket. [Veja as regras para nomenclatura de buckets](#)

Copiar configurações do bucket existente - *opcional*

Somente as configurações de bucket na configuração a seguir são copiadas.

**Escolher bucket**

Formato: s3://bucket/prefix

After “Settings”, in Athena I selected this created bucket.

Amazon Athena > Editor de consultas

Editor | Consultas recentes | Consultas salvas | [Configurações](#)

Grupo de trabalho primary ▼

### Configurações de criptografia e resultados de consulta [Gerenciar](#)

Localização e criptografia do resultado da consulta

Localização do resultado da consulta s3://glue-query-bucket-camila/ <a href="#">🔗</a>	Criptografar resultados da consulta -	Proprietário esperado do bucket -	Atribuir controle total sobre os resultados da consulta ao proprietário do bucket Desligado
--	--	--------------------------------------	--

And in the “Editor” tab we can create and execute our queries. It looks like PostgreSQL or MySQL, in this case would be for users to perform queries on a structured database, so Athena was connected via the AWS Glue Crawler.

After the analysis and the end-user dashboard, I used AWS Redshift. It is designed to process large volumes of data and perform complex analysis in real time. Redshift is based on a columnar relational database model and has been optimized to provide high performance in analytical queries, aggregations, and processing large data sets. Redshift easily integrates with other AWS tools and services, such as AWS Glue for data ETL (Extract, Transform, Load), AWS Data Pipeline for orchestrating data workflows, and Amazon S3 for data storage. Therefore, Crawler will also be integrated with Redshift.

In the console, search for Amazon Redshift, then Clusters and finally create the cluster. Due to load size and cost optimization, I decided to choose the ra3.xlplus node type and an availability zone.

[Amazon Redshift](#) > [Clusters](#) > [Crie o cluster](#)

## Criar cluster [Informações](#)

Procurando um teste gratuito? Experimente o Redshift sem servidor. Os clientes novos do Redshift sem servidor recebem um crédito de USD 300 para usar em suas contas.

[Iniciar o Redshift sem servidor](#)

### Configuração do cluster

**Identificador do cluster**  
Essa é a chave exclusiva que identifica um cluster.

O identificador deve ter de 1 a 63 caracteres. Os caracteres válidos são a-z (somente em minúsculas) e - (hifen).

**Escolher o tamanho do cluster**

☒ Eu vou escolher

☐ Ajude-me a escolher

**Tipo de nó** [Informações](#)  
Escolha um tipo de nó que atenda aos requisitos de CPU, RAM, capacidade de armazenamento e tipo de unidade.

**Configuração AZ** [Informações](#)  
Escolha se deseja implantar o cluster do Redshift em outra zona de disponibilidade.

☒ **Single-AZ**  
Compute resources are deployed in a single Availability Zone. O cluster é padrão para usar a **Atualfaixa**

☐ **Multi-AZ - novo**  
Os recursos computacionais são implantados em duas zonas de disponibilidade. O cluster é padrão para usar a **Antecedente**faixa

About database settings, choose admin username and set password.

### Configurações do banco de dados

**Nome do usuário administrador**  
Insira um ID de login para o usuário administrador da sua instância de banco de dados.

awsusercamilacustomerchurn

O nome deve ter de 1 a 128 caracteres alfanuméricos e não pode ser uma [palavra reservada](#).

---

**Senha do administrador**  
Selecione uma opção para gerenciar a senha de administrador.

☐ Gerenciar credenciais de administrador no AWS Secrets Manager [Informações](#)  
AWS manages a KMS key that encrypts your data.

☐ Gere uma senha  
O Amazon Redshift gera uma senha de administrador.

☒ Adicione manualmente a senha de administrador  
Insira manualmente a senha de administrador.

**Senha do usuário administrador**

\*\*\*\*\*

Deve ter de 8 a 64 caracteres. Deve conter pelo menos uma letra maiúscula, uma letra minúscula e um número. Pode ser qualquer caractere ASCII imprimível, exceto "/", "" ou "@".

☐ Mostrar senha

Redshift has already created an IAM profile with full access, as shown below.

### Permissões do cluster

**ⓘ**  Crie um perfil do IAM como padrão para esse cluster que tenha a política [AmazonRedshiftAllCommandsFullAccess](#) anexada. Essa política contém permissões para executar comandos SQL para COPY, UNLOAD e para consultar dados com o Amazon Redshift. A política também concede permissões para executar instruções SELECT para serviços relacionados, como o Amazon S3, o Amazon CloudWatch Logs, o Amazon SageMaker e o AWS Glue.

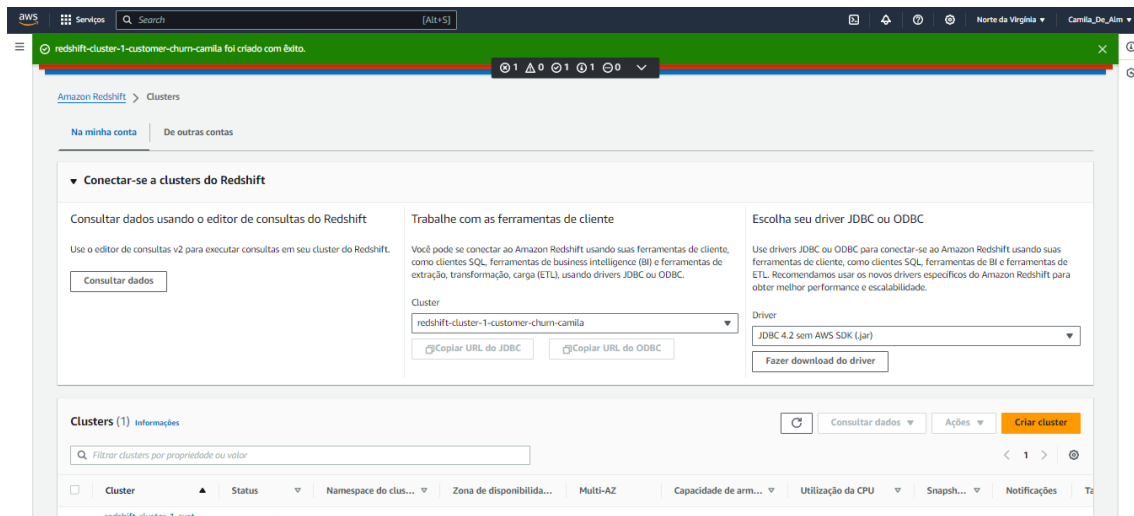
**Funções do IAM associadas (0)** [Informações](#) Definir padrão ▼ Gerenciar funções do IAM ▼

Crie, associe ou remova uma função do IAM. É possível associar até 50 funções do IAM. Você também pode escolher uma função do IAM e defini-la como padrão para o cluster.

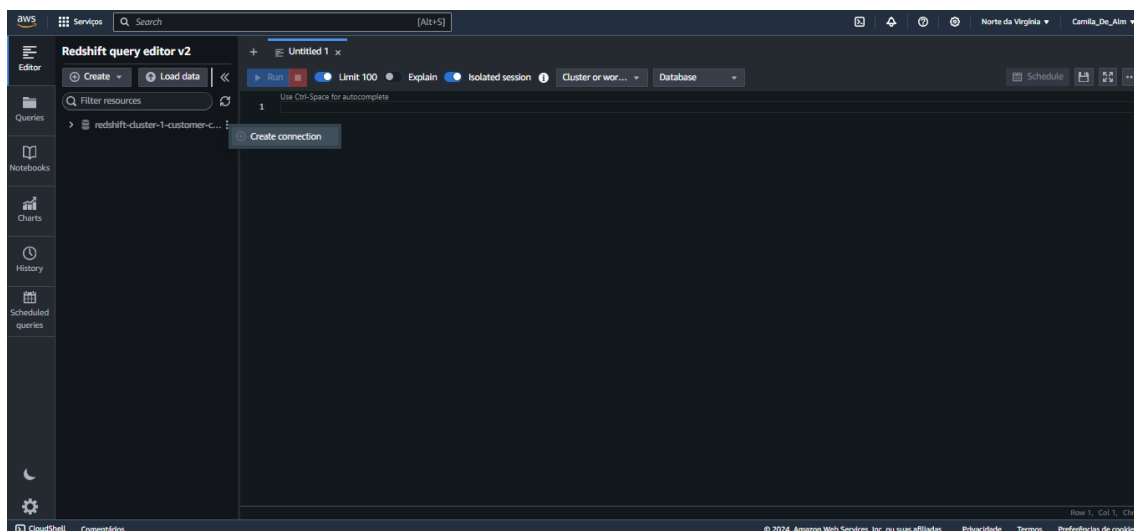
< 1 >

<input type="checkbox"/>	Funções do IAM <a href="#">↗</a>	Status ▼	Tipo de função ▼
Não há recursos Nenhuma função do IAM associada			
<div>Associar função do IAM</div>			

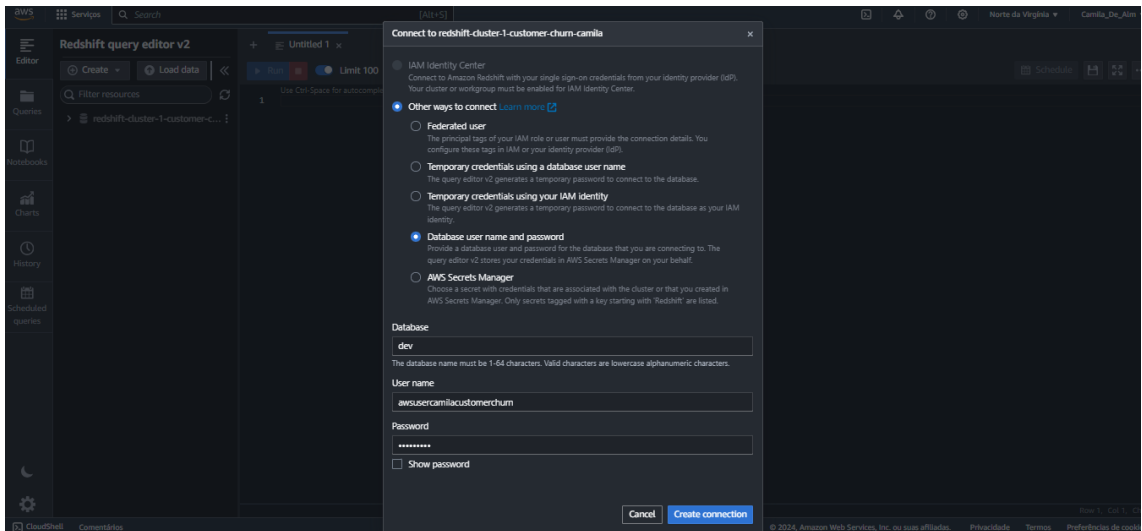
The other settings remain the same, as I had already created a VPC, I just had to select it.



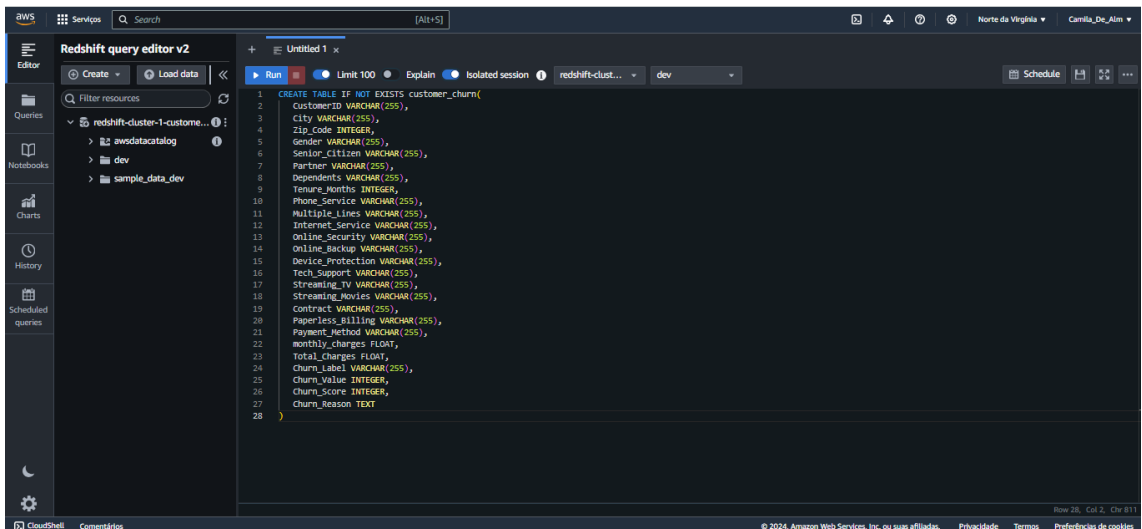
Then on the right click on Query Editor V2, open the editor below, in the three dots click on “Create connection.”



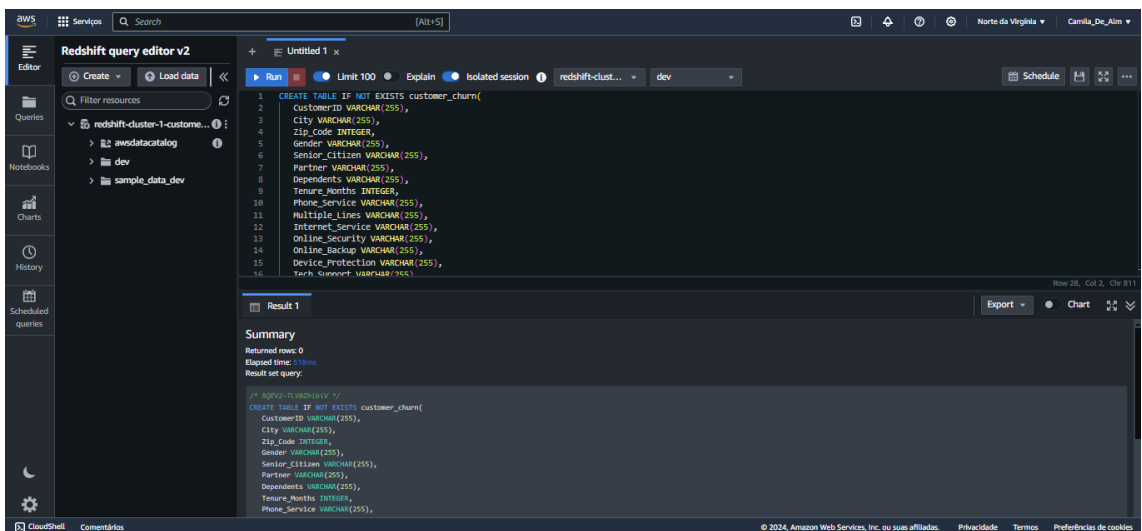
Now you must select the Database username and password option and enter the username and password that we defined when creating the Cluster. The database name was defined by the Redshift editor itself but can be changed.



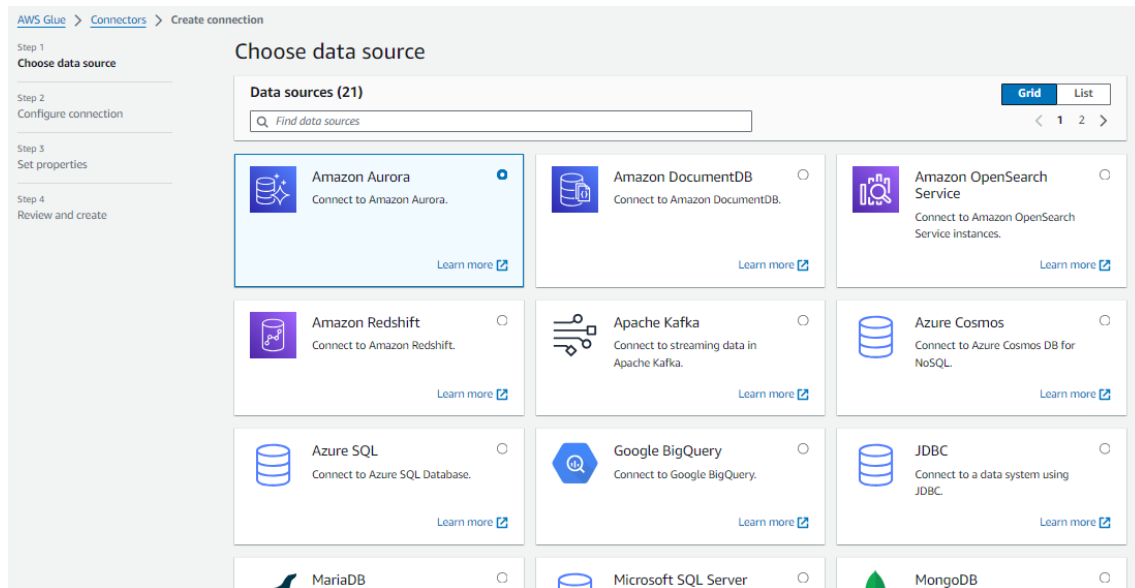
After creating the Schema of our table, as I already knew the Kaggle database, it was enough to format it in SQL format, as shown below.



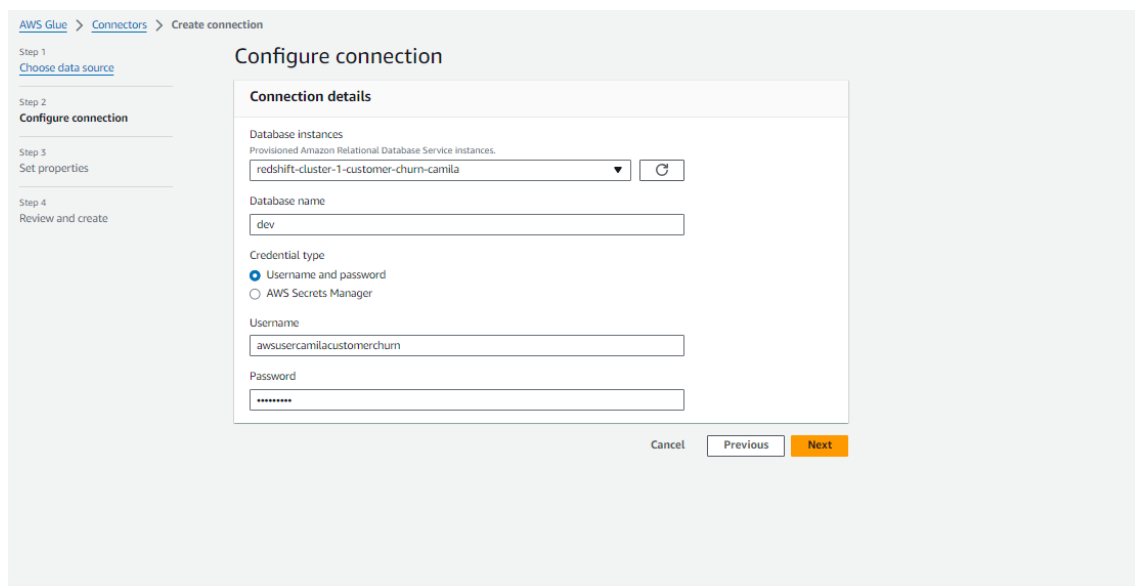
I executed the command.



Now we need to connect Redshift to Glue. Then go back to Paste in Data Connections on the right, click on “Create connection” we can connect with different data sources, but I chose Redshift.



I provided the cluster created and the credentials.



I revised and created.

[AWS Glue](#) > [Connectors](#) > Create connection

Step 1

[Choose data source](#)

Step 2

[Configure connection](#)

Step 3

[Set properties](#)


Step 4

Review and create

## Review and create

Step 1: Choose data source Edit

Data source

 Name  
Amazon Redshift

Step 2: Configure connection Edit

Connection details


Redshift cluster redshift-cluster-1-customer-churn-camila.cnv8ptoe3bs1.us-east-1.redshift.amazonaws.com	Redshift instance arn:aws:redshift:us-east-1:891377286262:namespace:5b32e3f9-47d2-4c55-81b0-f466966ef57
Database name dev	Credentials type Username and password

Step 3: Set properties Edit

Connection properties

Name	Description
Redshift connection	-

Connector type is JDBC which acts as a bridge between the application and the database, provides the means to establish a connection to the database, send SQL queries, receive results, and manipulate data, and is the default connector for Redshift.

 **"Redshift connection" connection successfully created.**  
To begin using your connection you must create a job.

[AWS Glue](#) > [Connectors](#) > Redshift connection

## Redshift connection

Edit Delete Create job

### Connection details Info

Connector type JDBC	Connection URL jdbc:redshift://redshift-cluster-1-customer-churn-camila.cnv8ptoe3bs1.us-east-1.redshift.amazonaws.com:5439/dev
Driver class name -	Driver path -
Username awsusercamilacustomerchurn	Require SSL connection -
Subnet subnet-04725b7ab34c9126c	Security groups sg-0dbddfd5f38c01e01
Description -	Created on 2024-03-23 15:49:41.471000
Last modified 2024-03-23 15:49:41.471000	Class name -



Now we need to connect Crawler (from AWS Glue) to Redshift, go to Crawler and add a new one for Redshift, as below.

The screenshot shows the 'Set crawler properties' dialog in the AWS Glue console. On the left, a sidebar lists five steps: Step 1 (Set crawler properties), Step 2 (Choose data sources and classifiers), Step 3 (Configure security settings), Step 4 (Set output and scheduling), and Step 5 (Review and create). The main area is titled 'Set crawler properties' and contains a 'Crawler details' section with an 'info' icon. This section has a 'Name' field with the value 'glue-redshift-crawler-camila' and a 'Description - optional' field with the placeholder 'Enter a description'. Below these fields, a note states 'Descriptions can be up to 2048 characters long.' At the bottom of the details section is a 'Tags - optional' section with a note 'Use tags to organize and identify your resources.' At the bottom right of the dialog are 'Cancel' and 'Next' buttons.

I clicked on “Next”, then added a new database and the JDBC connection, which is Redshift, and entered the path.

The screenshot shows the 'Add data source' dialog. It has a title bar with a close button. The 'Data source' section has a dropdown menu set to 'JDBC'. The 'Connection' section has a dropdown menu set to 'Redshift connection' and a refresh button. Below these are 'Clear selection' and 'Add new connection' buttons. The 'Include path' section has a text input field containing 'dev/public/customer\_churn'. Below this is a detailed explanation of the path format: 'You can substitute the percent (%) character for a schema or table. For databases that support schemas, enter MyDatabase/MySchema/% to match all tables in MySchema within MyDatabase. Oracle Database and MySQL don't support schema in the path; instead, enter MyDatabase/%. For Oracle database without SSL, MyDatabase can be either the system identifier (SID) or the service name (SERVICE\_NAME). For Oracle database with SSL, MyDatabase must be the service name (SERVICE\_NAME).' The 'Additional metadata - optional' section has a dropdown menu. At the bottom, there is an unchecked checkbox for 'Exclude files matching pattern' and two buttons: 'Cancel' and 'Add a JDBC data source'.

The IAM role could be the same as the one created for Athena.

### Configure security settings

**IAM role** [Info](#)

Existing IAM role

aws\_glue\_administrative\_role ▼ ↻ View ↗

Create new IAM role Update chosen IAM role

Only IAM roles created by the AWS Glue console and have the prefix "AWSGlueServiceRole-" can be updated.

► **Security configuration - optional**

Enable at-rest encryption with a security configuration.

Cancel Previous Next

In “Target database” we will have to create one for Redshift. I created it in add database and then selected it.

[AWS Glue](#) > [Crawlers](#) > Add crawler

Step 1  
[Set crawler properties](#)

Step 2  
[Choose data sources and classifiers](#)

Step 3  
[Configure security settings](#)

Step 4  
**Set output and scheduling**

Step 5  
[Review and create](#)

### Set output and scheduling

**Output configuration** [Info](#)

Target database

customer-churn-glue-redshift-database-camila ▼ ↻

Clear selection Add database ↗

Table name prefix - optional

Type a prefix added to table names

► **Advanced options**

**Crawler schedule**

You can define a time-based schedule for your crawlers and jobs in AWS Glue. The definition of these schedules uses the Unix-like cron [syntax](#). [Learn more](#) ↗

Frequency

On demand ▼

Cancel Previous Next

[AWS Glue](#) > [Databases](#) > [Add database](#)

## Create a database

Create a database in the AWS Glue Data Catalog.

### Database details

**Name**

Database name is required, in lowercase characters, and no longer than 255 characters.

**Description - optional**

Descriptions can be up to 2048 characters long.

### Database settings

**Location - optional**

Set the URI location for use by clients of the Data Catalog.

[Cancel](#) [Create database](#)

We now have two Crawlers on AWS Glue, one to integrate with Redshift and the other with Athena.

**Crawler successfully starting**

The following crawler is now starting: "glue-redshift-crawler-camila"

[Cola AWS](#) > [Rastreadores](#)

### Rastreadores

Um rastreador se conecta a um armazenamento de dados, avança por uma lista priorizada de classificadores para determinar o esquema dos seus dados e, em seguida, cria tabelas de metadados no seu catálogo de dados.

**Rastreadores (0)** [Informações](#)

Visualize e gerencie todos os rastreadores disponíveis.

Última atualização (UTC) 23 de março de 2024 às 19:34:09 [Atualizar](#) [Ação](#) [Correr](#) [Criar rastreador](#)

<input type="checkbox"/>	Nome	Estado	Agendar	Última corrida	Data e hora da última execução	Registro	Alterar
<input type="checkbox"/>	<a href="#">camila-crawler-s3</a>	Preparar		Sucesso	23 de março de 2024 às 17:37:14	<a href="#">Ver registro</a>	1 criado
<input type="checkbox"/>	<a href="#">cola-redshift-crawler-camila</a>	Preparar		Sucesso	23 de março de 2024 às 19:25:52	<a href="#">Ver registro</a>	1 criado

Still in AWS Glue, we need to configure the ETL job, in AWS Glue, on the right under ETL Jobs. Open the screen below, I named the job and in Visual, I configured each part of the ETL, just click on each item and configure the settings on the right.

In the S3 bucket settings, they are on the right. I named it and could indicate the location of S3, but as I had created a “Data Catalog table” I directed it to this table.

The screenshot shows the AWS Glue console interface for a job named `s3_upload_to_redshift_gluejob`. The right-hand pane displays the **Data source properties - S3** configuration. The **Name** is set to `Amazon S3`. Under **S3 source type**, the **Data Catalog table** option is selected. The **Database** is set to `customer-churn-s3-glue-database` and the **Table** is set to `camila_bucket_aws`. The **Partition predicate** is optional and currently empty. The main canvas shows a workflow with three nodes: **Data source - S3 bucket Amazon S3**, **Transform - Change Schema**, and **Data source - Amazon Redshift**. The **Data preview** section at the bottom shows a table with 29 rows and 6 columns: `customerid`, `count`, `country`, `state`, `city`, and `zip code`.

Database transformation, the “Change Schema”, displays the settings on the right. Here we made adjustments to the raw data that was in S3, dropped and adjusted the column names and types. Redshift doesn't have a float, it has double , pay attention to this difference.

The screenshot shows the AWS Glue console interface for the same job, but with the **Transform** configuration pane open. The **Name** is set to `Change Schema`. The **Node parents** section shows `Amazon S3` as the parent node. The **Change Schema (Apply mapping)** table is displayed, showing the mapping of source keys to target keys and data types. The mapping table is as follows:

Source key	Target key	Data type	Drop
customerid	customerid	string	<input type="checkbox"/>
count	count	long	<input checked="" type="checkbox"/>
country	country	string	<input checked="" type="checkbox"/>
state	state	string	<input checked="" type="checkbox"/>
city	city	string	<input type="checkbox"/>
zip code	zip_code	long	<input type="checkbox"/>
lat long	lat long	string	<input checked="" type="checkbox"/>
latitude	latitude	string	<input checked="" type="checkbox"/>
longitude	longitude	string	<input checked="" type="checkbox"/>

Loading in Redshift, below job settings. I could have made a direct connection, but as I had already created the table, I just directed it to it. I defined that the table would truncate with each new data load, so the data would always be updated with the latest available data, but it would also be possible to just use Append , merge or even recreate the table from scratch.

Successfully updated job  
Successfully updated job s3\_upload\_to\_redshift\_gluejob. To run the job choose the Run Job button.

s3\_upload\_to\_redshift\_gluejob

Last modified on 23/03/2024, 17:33:09

Visual Script Job details Runs Data quality - updated Schedules Version Control

Change Schema

Data target - Amazon Redshift

Data target properties - Amazon Redshift

Include partitions  
Choose which nodes will provide inputs for this one.  
Choose one or more parent node  
Change Schema X  
ApplyMapping - Transform

Redshift access type  
☐ Direct data connection - recommended  
☒ Glue Data Catalog tables

Database  
Search AWS Glue Catalog databases  
customer-churn-glue-redshift-database-camila

Table  
Search AWS Glue Catalog tables created from Amazon Redshift  
dev\_public\_customer\_churn

Handling of data and target table  
☐ APPEND (insert) to target table  
AWS Glue will append data to existing columns of the table and discard any extra columns.  
☐ MERGE data into target table  
AWS Glue will either update or append data to the table based on a set of conditions.  
☒ TRUNCATE target table  
Same as Append, except AWS Glue will first clear the contents of the table.  
☐ DROP and recreate target table  
AWS Glue will delete and recreate the table with the schema from the source data.

Data preview Output schema

Schema AVAILABLE Infer schema from session

Key	Data type	Partition
customerid	string	-
city	string	-

## Result

Successfully started job  
Successfully started job s3\_upload\_to\_redshift\_gluejob. Navigate to Run details for more details.

s3\_upload\_to\_redshift\_gluejob

Last modified on 23/03/2024, 17:33:09

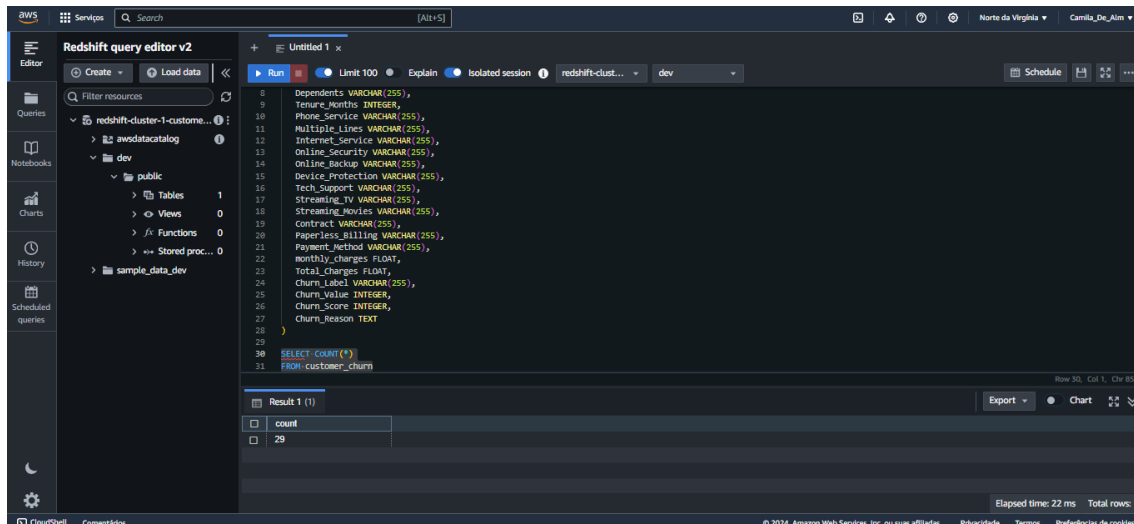
Visual Script Job details Runs Data quality - updated Schedules Version Control

Data source - S3 bucket Amazon S3

Transform - Change Schema

Data target - Amazon Redshift

I manually ran the job to see if it was running in the Runs tab. To be sure, go back to the Redshift editor and run the Count command, the original table was empty, it only had the schema, but now it already has some values.



We can run it manually by clicking Run, or automate it with Airflow, that's why we connected Airflow to VScode, in this connection I ran the following script:

**airflow import DAG** # Import the **DAG** class from the **airflow** module. The **DAG** class is used to define and describe a workflow in Airflow.

**from datetime import timedelta , datetime**

**airflow.operators.python import Python Operator** #Imports the PythonOperator class from the Airflow operators.python module. The PythonOperator is used to execute Python functions as tasks in a workflow.

**from airflow.providers.amazon.aws.hooks.base\_aws import AWSGenericHook** #Imports the AWSGenericHook class from the base\_aws module of the Airflow providers.amazon.aws.hooks package. This class is used to establish connections to AWS services.

**import time**

**from airflow.providers.amazon.aws.sensors.glue import Sensor GlueJob** #This class is used to wait until an AWS Glue job completes before continuing the workflow.

**def cola\_job\_s3\_redshift\_transfer( job\_name , \*\* kwargs ):**

**session = AwsGenericHook ( aws\_conn\_id ='aws\_s3\_conn')**

**#Set the region I was in us-west-2**

**boto3\_session = session.get\_session (region\_name ='us-west-2')**

**client = boto3\_session.client(' cola ')**

**client.start\_job\_run (**

**task\_name = task\_name,**

**)**

**get\_run\_id() definition:**

**sleeptime (8)**

**session = AwsGenericHook ( aws\_conn\_id ='aws\_s3\_conn')**

**boto3\_session = session.get\_session (region\_name ='us-west-2')**

```

glue_client = boto3_session.client(' glue ')
response = glue_client.get_job_runs(JobName="s3_upload_to_redshift_gluejob")
job_run_id = response[" JobRuns "][0]["Id"]
return job_run_id

```

#Defines a default\_args dictionary that contains the default arguments for the DAG, such as the owner, start data, email settings on failure, etc.

```

default_args = {
'owner': 'airflow',
' depends_on_past ': False,
'start_date': datetime (2023, 8, 1),
'email': ['myemail@domain.com'],
' email_on_failure ': False,
' email_on_retry ': False,
' retries ': 2,
' retry_delay ': timedelta ( seconds =15)
}

```

# Starts the DAG definition with the name ' my\_dag ' and the previously defined default arguments.

```

with DAG(' my_day ',
default_args = default_args,
scheduling_interval = '@weekly ',
ketchup =False) as dag :

```

```

glue_job_trigger = PythonOperator (
task_id = ' tsk_glue_job_trigger ',
python_callable = cola_job_s3_redshift_transfer,
op_kwargs ={
'job_name': 's3_upload_to_redshift_gluejob'
},
)

grab_glue_job_run_id = PythonOperator (
task_id = ' tsk_grab_glue_job_run_id ',
python_callable = get_run_id ,
)

```

```

is_glue_job_finish_running = GlueJobSensor (

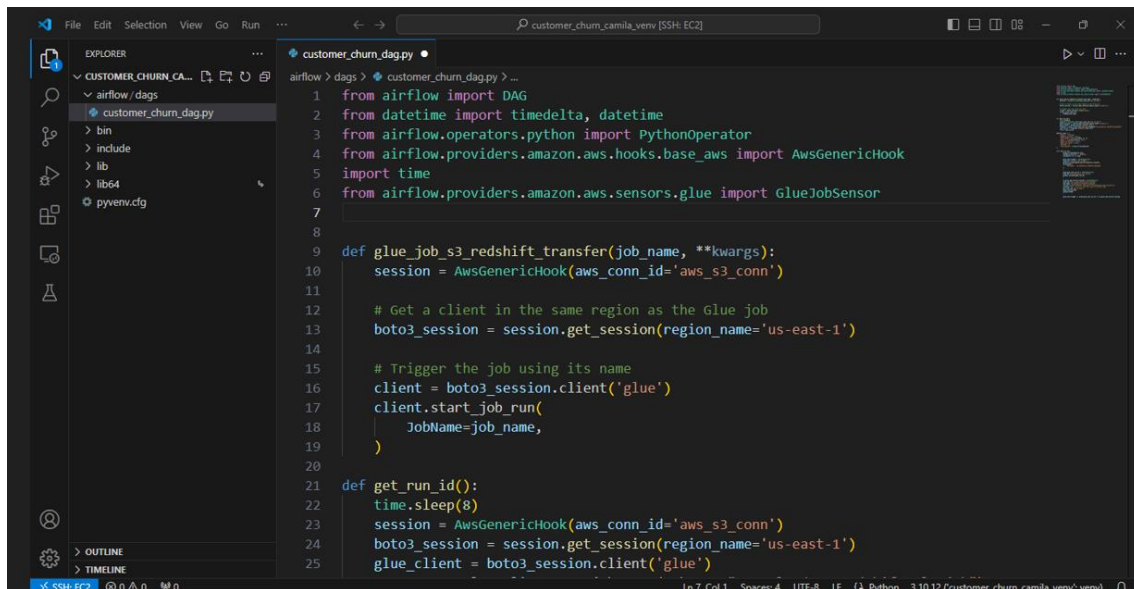
```

```

task_id="tsk_is_glue_job_finish_running",
job_name='s3_upload_to_redshift_gluejob',
run_id='{{task_instance.xcom_pull("tsk_grab_glue_job_run_id")}}',
verbose = True , # print pastes job records into airflow records
aws_conn_id='aws_s3_conn',
poke_interval=60,
timeout = 3600,
)

```

glue\_job\_trigger >> grab\_glue\_job\_run\_id >> is\_glue\_job\_finish\_running



```

1 from airflow import DAG
2 from datetime import timedelta, datetime
3 from airflow.operators.python import PythonOperator
4 from airflow.providers.amazon.aws.hooks.base_aws import AwsGenericHook
5 import time
6 from airflow.providers.amazon.aws.sensors.glue import GlueJobSensor
7
8
9 def glue_job_s3_redshift_transfer(job_name, **kwargs):
10     session = AwsGenericHook(aws_conn_id='aws_s3_conn')
11
12     # Get a client in the same region as the Glue job
13     boto3_session = session.get_session(region_name='us-east-1')
14
15     # Trigger the job using its name
16     client = boto3_session.client('glue')
17     client.start_job_run(
18         JobName=job_name,
19     )
20
21 def get_run_id():
22     time.sleep(8)
23     session = AwsGenericHook(aws_conn_id='aws_s3_conn')
24     boto3_session = session.get_session(region_name='us-east-1')
25     glue_client = boto3_session.client('glue')

```

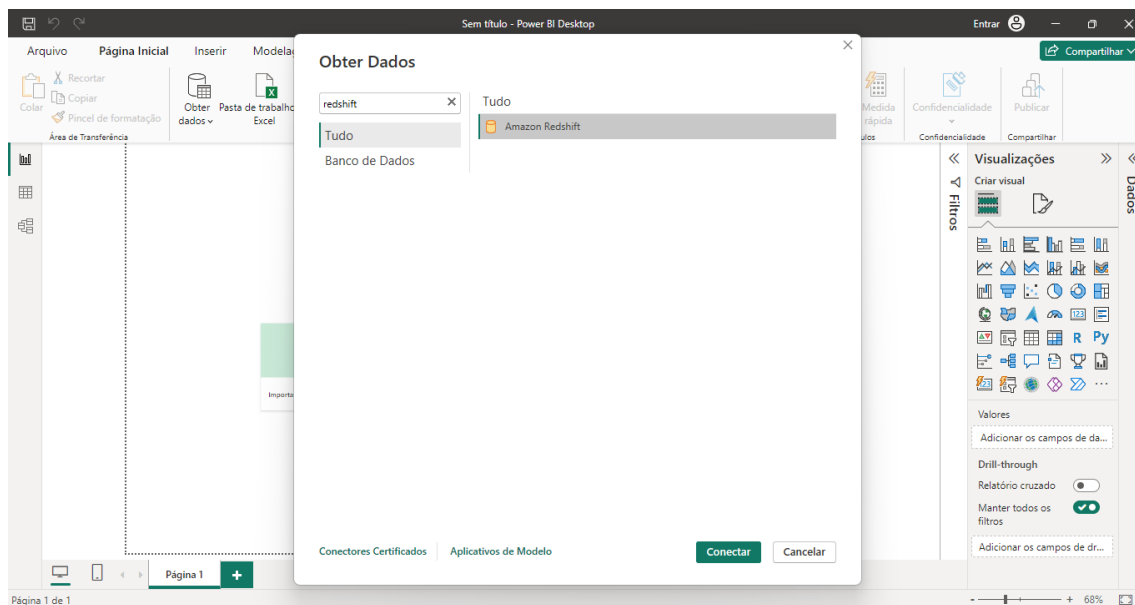
Afterwards, we must connect Airflow via the Root User's Access Keys to automate the job. This must be done via the Airflow browser for both S3 and Redshift and via the EC2 terminal, accessing the virtual environment again and entering Access Keys. After running again, the job in Crawler

After checking that the status is “Running” in Crawler and Success in Airflow,

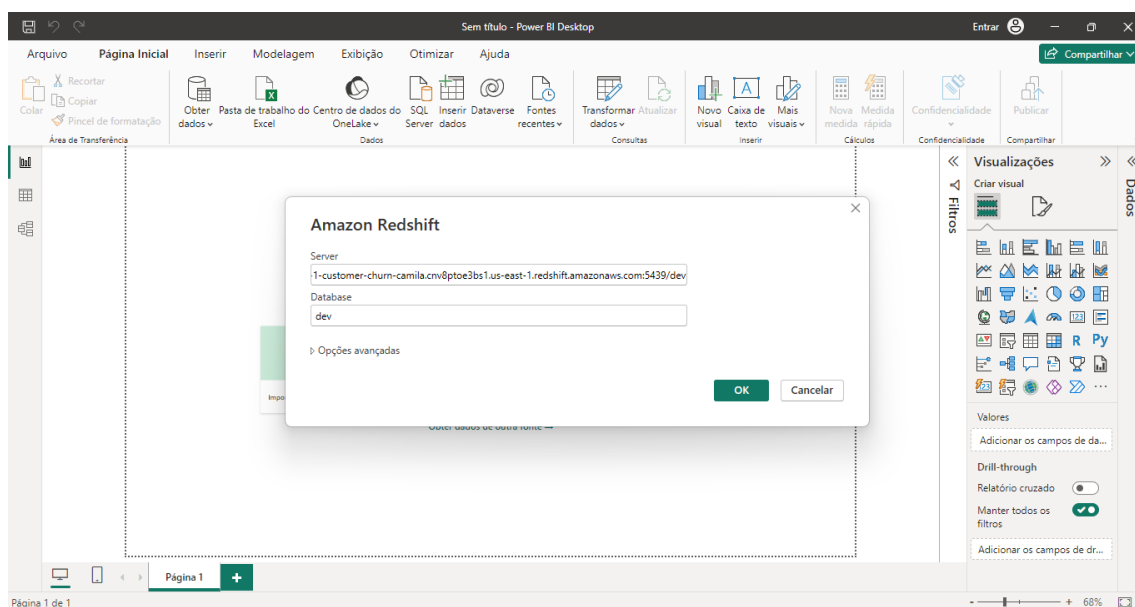


Finally, we will present data visualization in Microsoft Power BI. Connect PowerBI with Redshift don't forget to enable the cluster for connections outside the VPC.

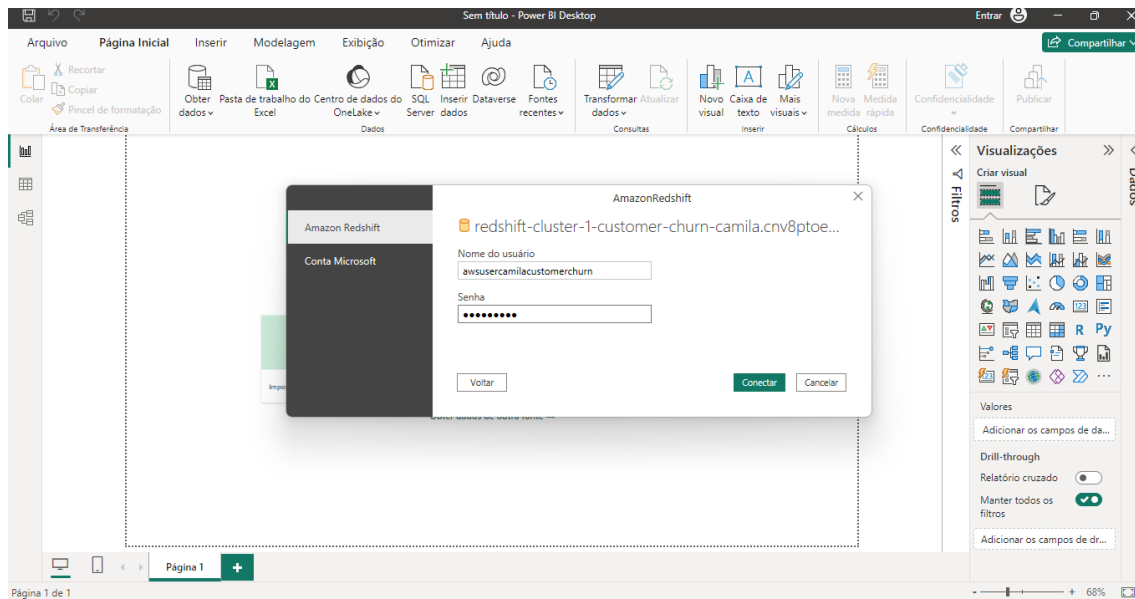




It will ask for the server and the database, the server is the endpoint of the created cluster, and the database is dev, a name that AWS itself suggested.



Enter the cluster credentials.



Afterwards, dashboards will be created, with this table that will be updated in real time with each new load received in the S3 Bucket. Below is a visualization I created.

