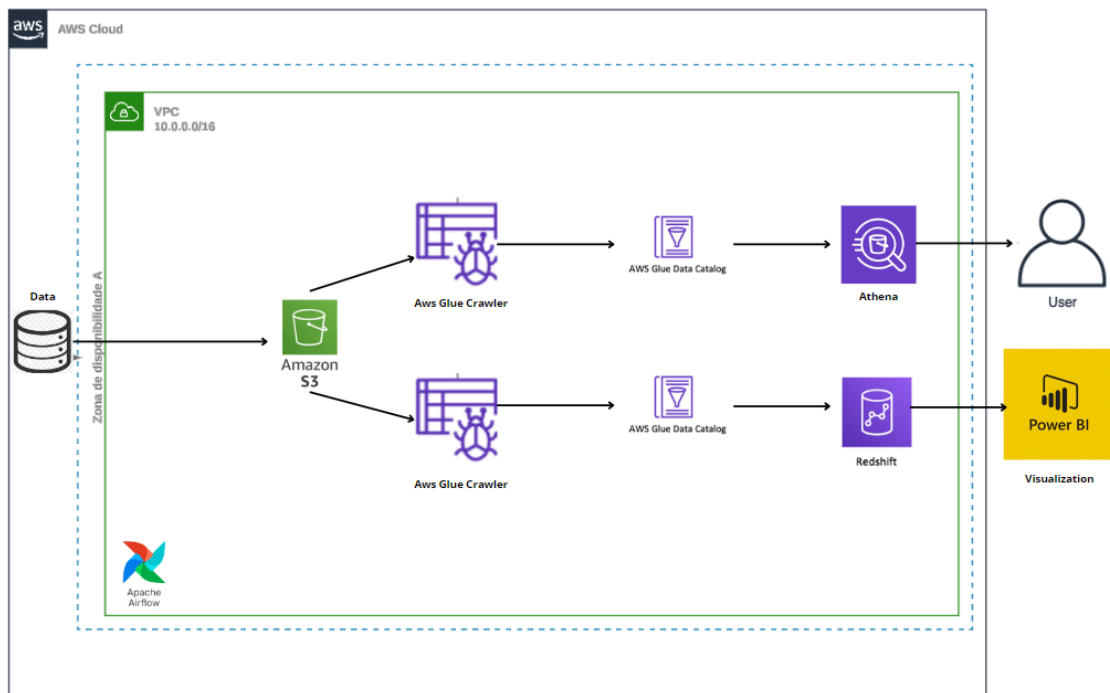


# Análise de Churn de clientes através de uma Pipeline de dados usando Apache Airflow, AWS Glue, S3, Amazon Redshift e Power BI

## Resumo

Neste projeto de engenharia de dados foi feita uma análise de churn de clientes, construí e automatizei um pipeline ETL usando o AWS Glue para carregar dados do bucket AWS S3 em um data Warehouse Amazon Redshift e depois conectar o Power BI ao cluster do Redshift para a visualização do usuário final. O AWS Glue serviu como um rastreador de dados para inferir o esquema do banco de dados depois os dados foram disponibilizados no AWS Athena para extrair os dados através de consultas SQL. O AWS Glue também serviu para carregar os dados rastreados e tratados para o cluster Redshift. O Apache Airflow foi usado para orquestrar e automatizar todo esse processo que era manual.

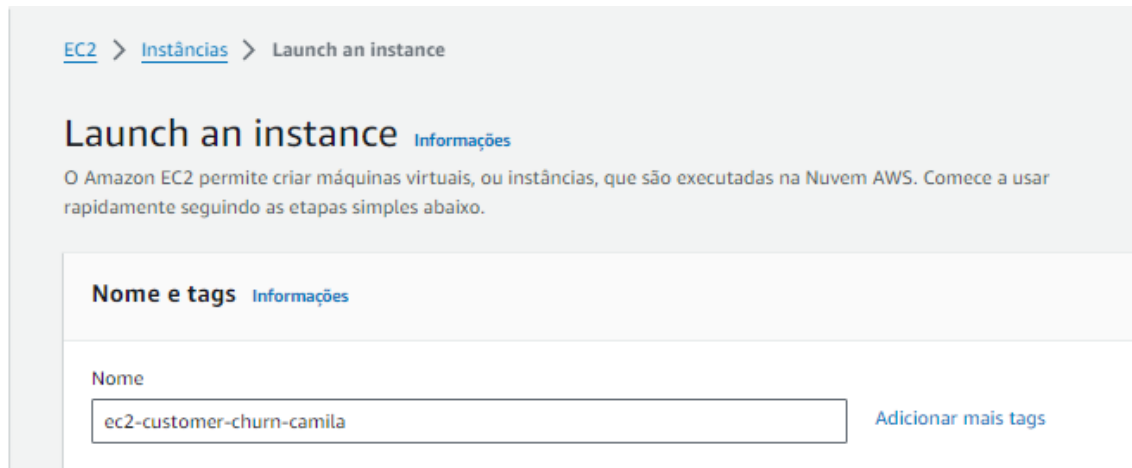
Diagrama da arquitetura na nuvem AWS:



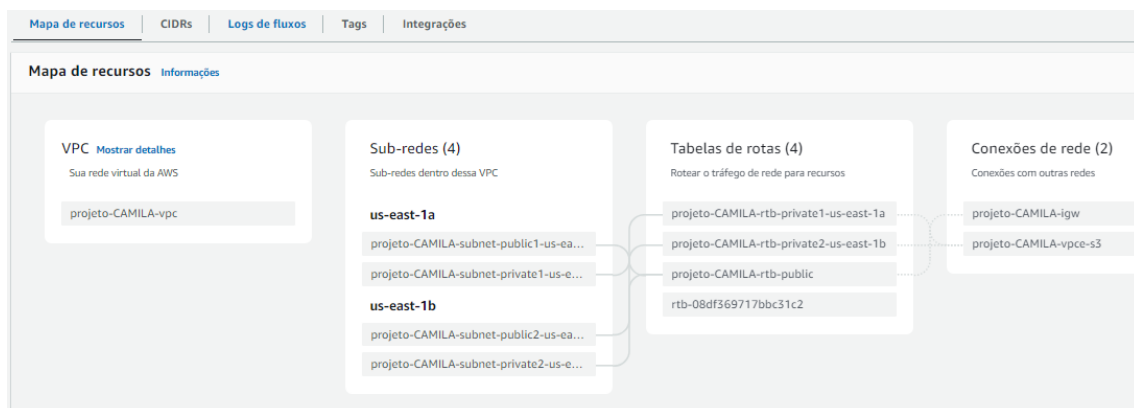
Todo o projeto foi desenvolvido do zero na nuvem AWS, a base de dados utilizada foi retirada do Kaggle, se chama “Rotatividade de clientes de telecomunicações: conjunto de dados IBM”, abaixo seguem as etapas.

Primeiro criei uma EC2, para configurações e orquestração do processo. No console da AWS, procure por EC2, depois em “Executar EC2”. A AMI escolhida foi o ubuntu, as configurações iniciais serão feitas em Linux, via o terminal da EC2. Habilitei as 3 regras de conexão de entrada SSH, HTTPS e HTTP, assim será possível

conectar via linha de comando e navegador, como a carga inicial será baixa escolhi o tipo da instância t2.medium, que possui 4Gib e 2 CPUs, isso garante que a EC2 não congelará durante o processo.



Também gerei uma key pair, fiz o download da key pair em csv, por questões de segurança e para poder conectar via SSH, além disso, uma VPC foi criada como proteção a EC2.



Quando a instancia estiver com o status “Executando”, a selecione e clique em conectar. Neste primeiro momento, conectei via instance connect para iniciar as configurações. Clique em “Conectar” novamente.

## 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

Vai abrir um novo terminal, configurei com os seguintes comandos Linux:

**sudo apt update** (Atualiza a lista de pacotes disponíveis para instalação no sistema operacional usando o gerenciador de pacotes APT (Advanced Package Tool))

**sudo apt install python3-pip** (Instala o pacote python3-pip, que é o gerenciador de pacotes do Python para a versão 3.x, usados para instalar e gerenciar bibliotecas e dependências Python.)

**sudo apt install python3.10-venv** (Instala o pacote python3.10-venv, que é necessário para criar ambientes virtuais Python 3.10, permitindo isolar ambientes de desenvolvimento Python para projetos específicos.)

**python3 -m venv customer\_churn\_camila\_venv** (Cria um ambiente virtual chamado customer\_churn\_camila\_venv usando o módulo venv do Python 3. Esse ambiente virtual é onde as dependências do projeto serão instaladas e isoladas do sistema global.)

**source customer\_churn\_camila\_venv/bin/activate** (Ativa o ambiente virtual criado anteriormente, garantindo que as instalações e execuções subsequentes ocorram dentro desse ambiente isolado.)

**sudo pip install apache-airflow** (Instala o Apache Airflow, uma plataforma para programar, monitorar e gerenciar fluxos de trabalho de dados. Aqui, estamos usando o pip (Python Package Installer) para instalar o Airflow.)

**pip install apache-airflow-providers-amazon** (Instala um provedor específico do Apache Airflow para integração com serviços da Amazon Web Services (AWS), como S3, EC2, etc. Isso fornece funcionalidades adicionais para trabalhar com serviços da AWS dentro do Apache Airflow.)

**airflow standalone** (Inicia o Apache Airflow no modo standalone, o que significa que ele será executado em um único nó sem o uso de um banco de dados externo, como MySQL ou PostgreSQL. Este modo é útil para configurações de desenvolvimento ou testes.)

Após as configurações do ambiente virtual e das dependências, o Airflow deve indicar que está pronto e te informará uma username e senha, conforme abaixo:

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

Meu username: admin password: 9xYkerwf6v3v97qE.

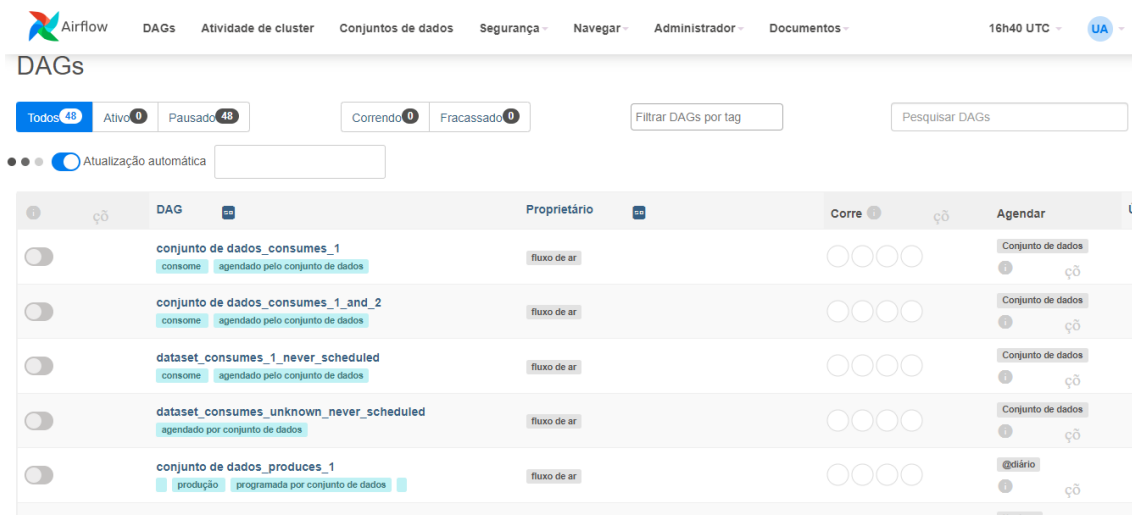
Após isso, volte na EC2 criada, a selecione e na tab “Segurança”, clique em Grupos de segurança, depois em Regras de Entrada e em Editar Regras de entrada, adicione a regra para acessar o Apache Airflow via navegador, isso será importante para visualizar se o job que criei está funcionando corretamente.

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...	<input type="text"/> <input type="button" value="Excluir"/>
sgr-007ef2de33d7df5a7	SSH	TCP	22	Persona...	<input type="text"/> <input type="button" value="Excluir"/>
sgr-0b7c50379232e920f	HTTP	TCP	80	Persona...	<input type="text"/> <input type="button" value="Excluir"/>
-	TCP personalizado	TCP	8080	Qualiqu...	<input type="text"/> <input type="button" value="Excluir"/>

Adicione a regra, tipo TCP Personalizado, intervalo de portas 8080 e Origem Anywhere IPv4. A porta 8080 é uma porta de comunicação de rede. Em muitos casos, é usada como porta padrão para servidores web locais ou de desenvolvimento. Quando você executa o Apache Airflow com o comando **airflow standalone**, por padrão ele usa a porta 8080 para fornecer uma interface web onde é possível monitorar e gerenciar fluxos de trabalho. Então, quando inicia o Airflow com **airflow standalone**, é possível acessar sua interface web digitando **http://localhost:8080** no navegador. Nosso localhost é o nosso Endereço IP público da EC2. Após faça login no Airflow pelo navegador com as credenciais que ele havia criado.



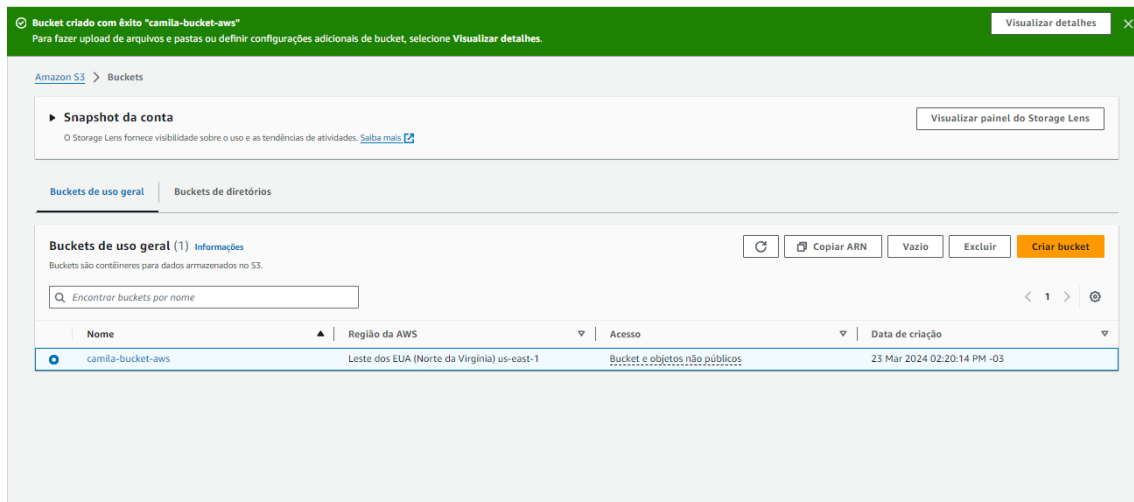
The image shows the Airflow web interface. At the top, there's a navigation bar with links: DAGs, Atividade de cluster, Conjuntos de dados, Segurança, Navegar, Administrador, and Documentos. The current time is 16h40 UTC. Below the navigation bar, the 'DAGs' section is active. It includes filters for 'Todos' (48), 'Ativo' (0), 'Pausado' (48), 'Correndo' (0), and 'Fracassado' (0). There's also a search bar for 'Pesquisar DAGs' and a toggle for 'Atualização automática'. The main table lists DAGs with columns: DAG, Proprietário, Corre, and Agendar. The table contains five rows of DAGs, each with a toggle switch, a name, a description, a owner, a status, and an action menu.

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

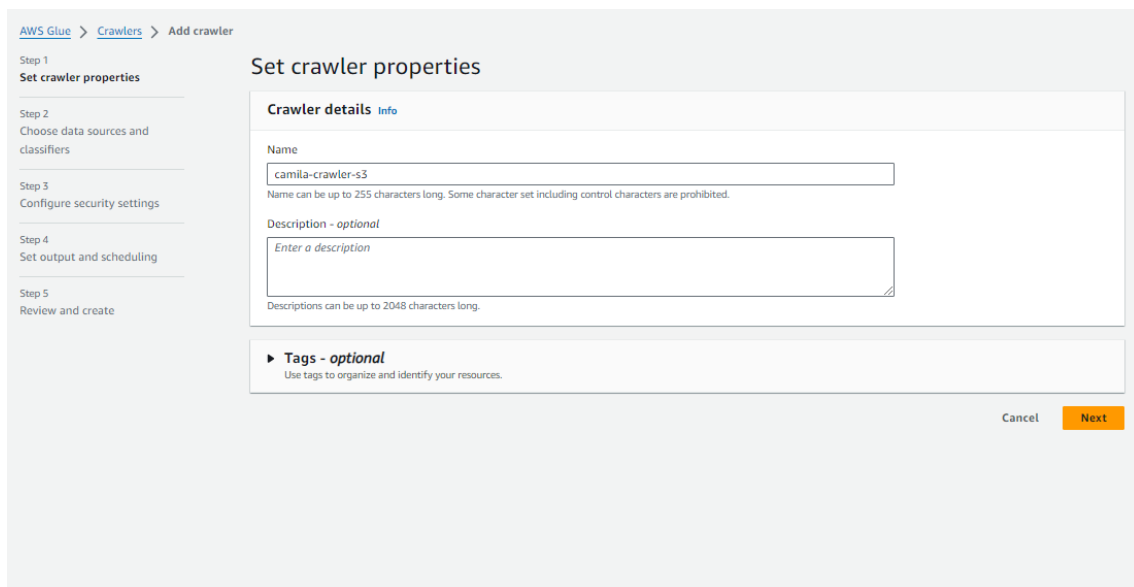
Depois configurei o acesso remoto via SSH pelo VScode, porque é mais amigável que usar o VIM ou o terminal da EC2. No rodapé, à esquerda, “Open a Remote Window”, conecta com o Host, informe a sua EC2 e conecte ao ambiente virtual criado anteriormente.



Depois criei um bucket S3, ele é como uma pasta de armazenamento na nuvem onde é possível armazenar qualquer tipo de dados, como arquivos, documentos, imagens, vídeos, backups de banco de dados, entre outros. Armazenei os dados no bucket S3.



Depois criei o AWS Glue, que fornece uma maneira fácil de criar, gerenciar e executar pipelines de ETL (Extract, Transform, Load) para processar e transformar grandes volumes de dados. Utilizei o recurso chamado Crawler do AWS Glue, ele automatiza a descoberta e classificação de dados em diversas fontes de dados, como Amazon S3, bancos de dados relacionais e não relacionais, por isso, ele também será utilizado com o Athena e o Redshift. Pesquise por AWS Glue, depois em “Data catalog”, “Crawlers” nós vamos processar os dados que serão armazenados no S3.



Como o schema da minha tabela não estava criado tive que fazer 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

Adicionei a fonte de dados, que será o conteúdo do bucket S3.

**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.  

🔍 s3://camila-bucket-aws ×

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

Será necessário informar qual o path do S3, e defini como “Crawl all subfolders” porque quero que os dados do bucket sejam atualizados a cada nova carga.

O Crawler do AWS Glue, determina que a função IAM já deva estar criada para operar o processo, então adicionei a permissão ao usuário administrador para administrar o 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

Adicionei a database que já havia adicionado ao 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 ▼

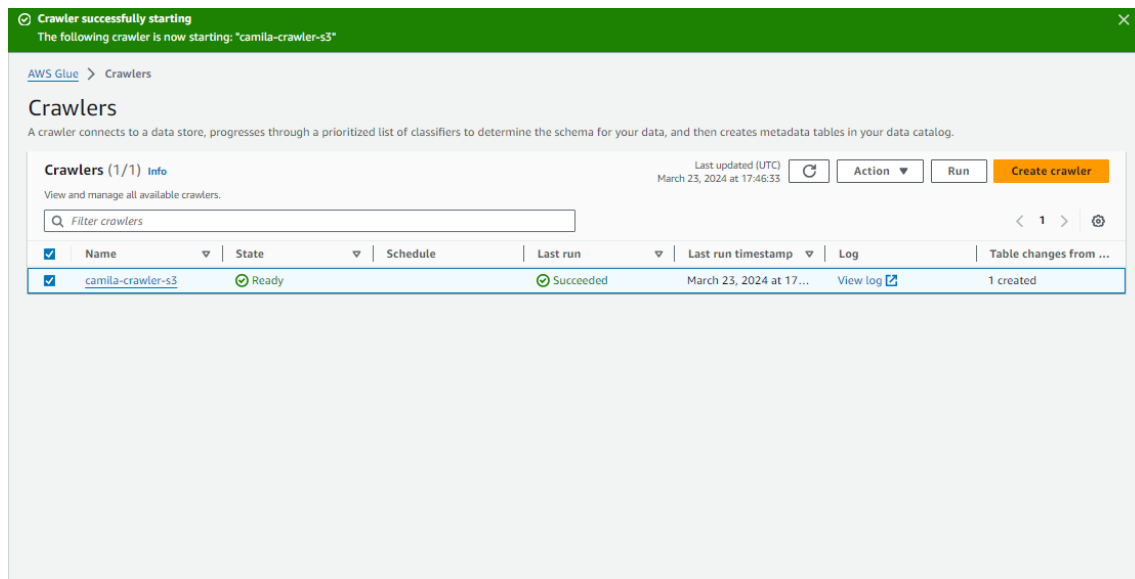
Cancel

Previous

Next

Mantive as demais configurações padrão e revisei e depois criei, depois dê “Run” espere até ficar com o status “Ready”.





Agora usarei o Crawler criado para conectar com o AWS Athena. O AWS Athena é um serviço de consulta interativa fornecido pela AWS que permite analisar dados diretamente no Amazon S3 usando SQL padrão. Em vez de carregar os dados em um banco de dados ou armazenamento de dados tradicional, o Athena permite executar consultas SQL diretamente nos arquivos armazenados no S3, assim não será necessário configurar e gerenciar uma infraestrutura de banco de dados. As queries que fiz para criar um relatório para o usuário final, devem ficar armazenadas em um outro Bucket S3, por isso criei outro bucket. Depois de criar o S3, posso ajustar as configurações do Athena.

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

Depois em “Configurações”, no Athena selecionei este bucket criado.

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
--	--	--------------------------------------	--

E na tab “Editor” podemos criar e executar nossas queries. Parece com o PostgreSQL ou o MySQL, neste caso seria para os usuários realizar consultas em um banco estruturado, assim o Athena foi conectado via o Crawler do AWS Glue.

Após para análises e dashboard para o usuário final, utilizei o AWS Redshift. Ele é projetado para processar grandes volumes de dados e realizar análises complexas em tempo real. O Redshift é baseado em um modelo de banco de dados relacional colunar e foi otimizado para fornecer alto desempenho em consultas analíticas, agregações e processamento de grandes conjuntos de dados. O Redshift se integra facilmente com outras ferramentas e serviços da AWS, como o AWS Glue para ETL (Extract, Transform, Load) de dados, o AWS Data Pipeline para

orquestração de fluxos de trabalho de dados e o Amazon S3 para armazenamento de dados. Por isso, o Crawler também será integrado ao Redshift.

No console, pesquise por Amazon Redshift, depois em Clusters e por último criar o cluster. Devido ao tamanho da carga e otimização dos custos decidi escolher tipo de nó ra3.xlplus e uma zona de disponibilidade.

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.

redshift-cluster-1-customer-churn-camila

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.

ra3.xlplus

**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**

Sobre as configurações do banco de dados, escolhi o nome do usuário administrador e defini a senha.

### 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

O Redshift já cria um perfil do IAM com total acesso, como mostra abaixo.

### Permissões do cluster

**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.

Funções do IAM

Status

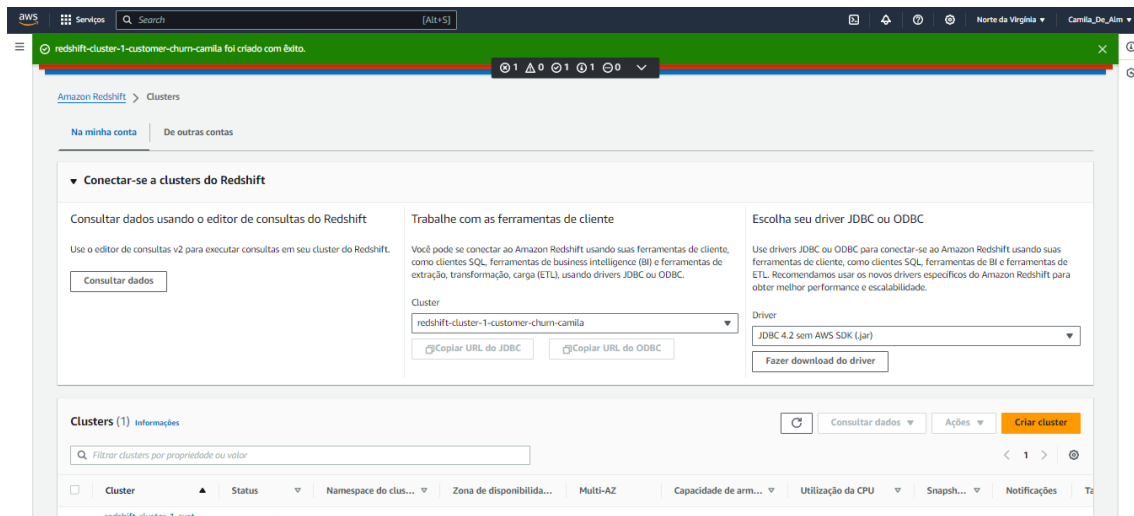
Tipo de função

Não há recursos

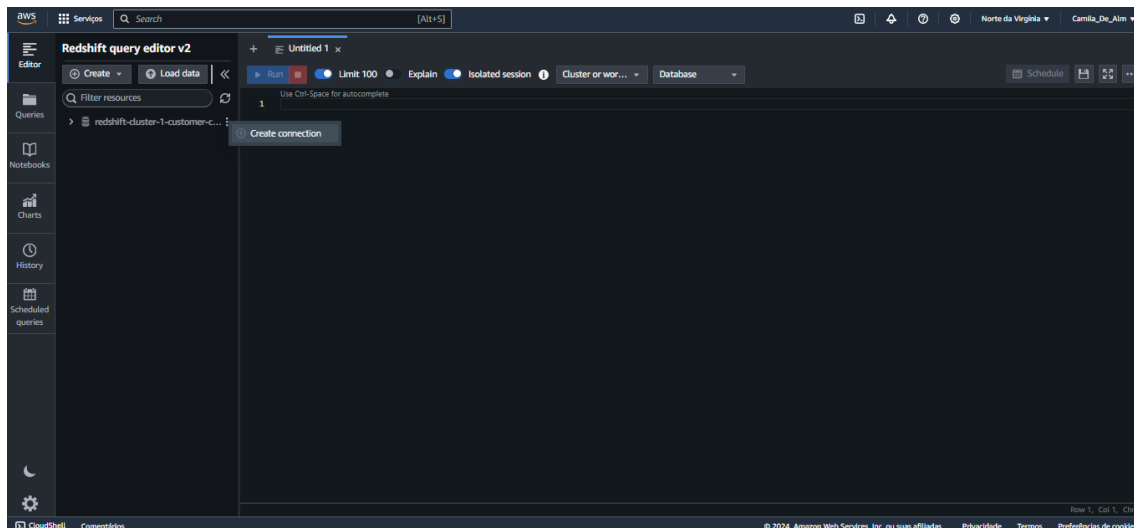
Nenhuma função do IAM associada

Associar função do IAM

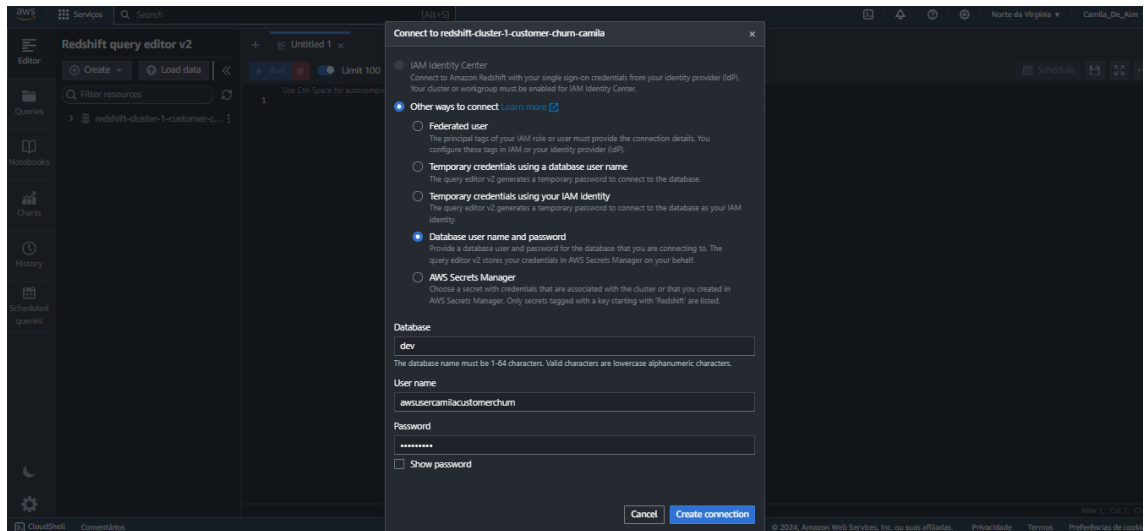
As demais configurações permaneceram as mesmas, como eu já havia criado uma VPC, bastou selecionar.



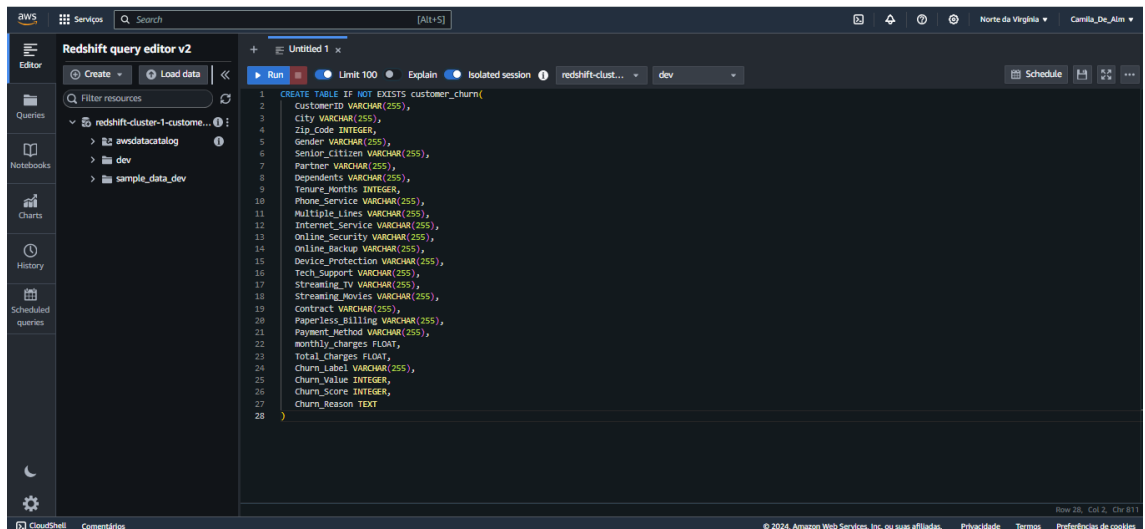
Depois à direita clique em Query Editor V2, vai abrir o editor abaixo, nos três pontos clique em “Create connection”



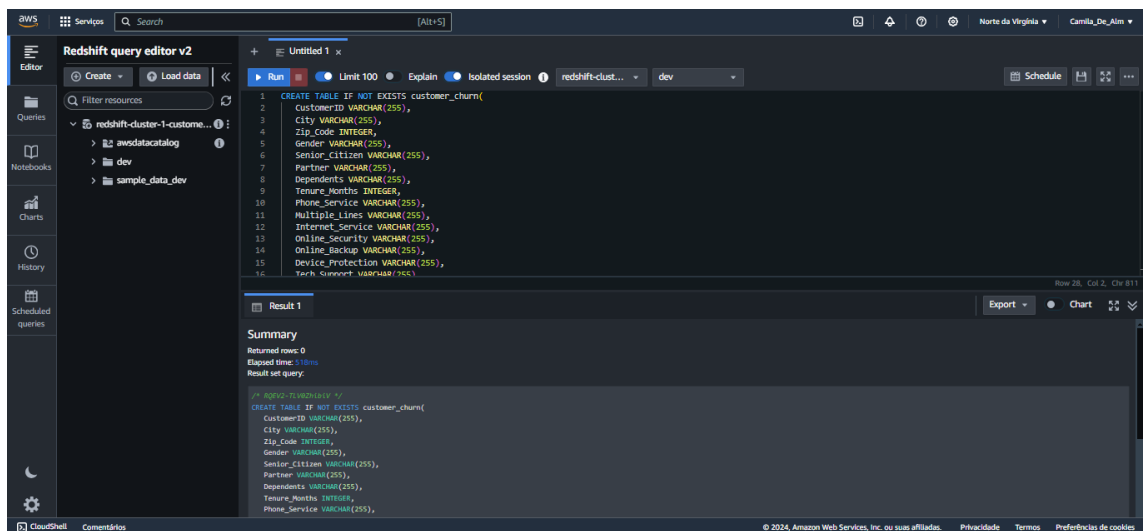
Agora devemos selecionar a opção Database user name and password e colocar o Username e senha que definimos na criação do Cluster. O nome do database foi definido pelo próprio editor do Redshift, mas pode ser alterado.



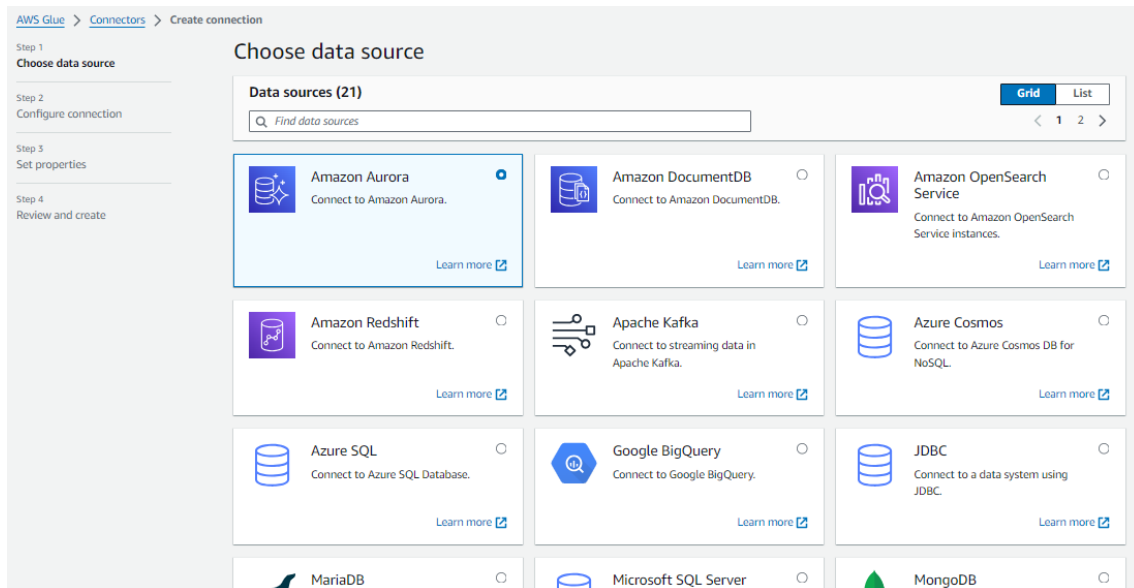
Após criei o Schema da nossa tabela, como já conhecia a base de dados do Kaggle, bastou formatar em formato SQL, conforme abaixo.



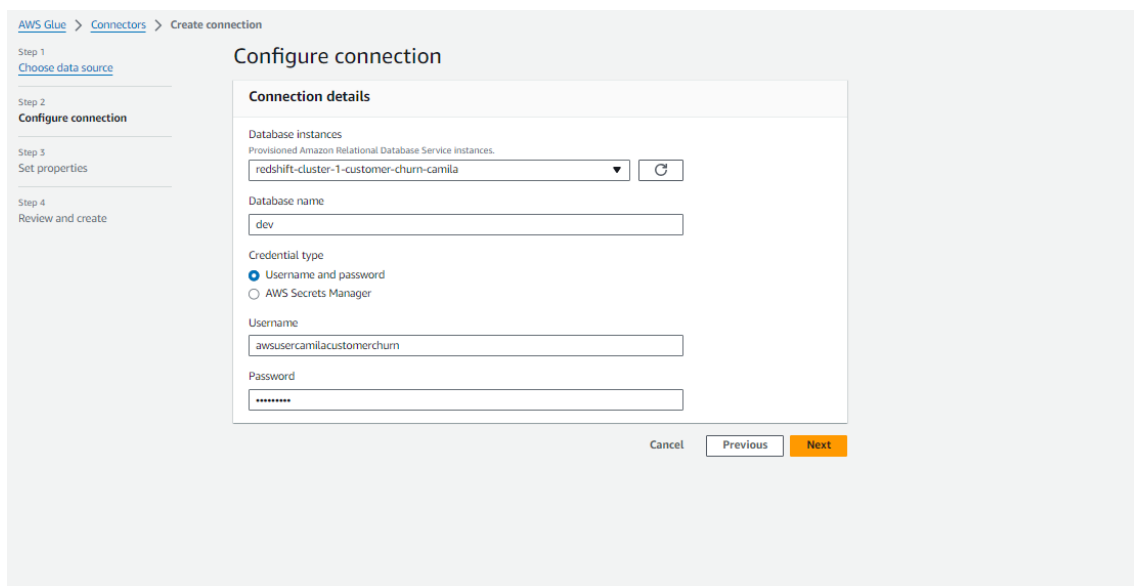
Executei o comando.



Agora precisamos conectar o Redshift ao Glue. Então volte ao Glue em Data connections à direita, clique em “Create connection” podemos conectar com diversas fontes de dados, mas escolha o Redshift.



Informei o cluster criado e as credenciais.



Revisei e criei.

[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-  
f466966ef657

Database name  
dev

Credentials type  
Username and password


Step 3: Set properties Edit

Connection properties

Name  
Redshift connection

Description  
-

O tipo de conector é o JDBC que atua como uma ponte entre o aplicativo e o banco de dados, fornecendo os meios para estabelecer uma conexão com o banco de dados, enviar consultas SQL, receber resultados e manipular dados e é o conector padrão do 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

Driver class name  
-

Username  
awsusercamilacustomerchurn

Subnet  
subnet-04725b7ab34c9126c

Description  
-

Last modified  
2024-03-23 15:49:41.471000

Connection URL  
jdbc:redshift://redshift-cluster-1-customer-churn-  
camila.cnv8ptoe3bs1.us-east-  
1.redshift.amazonaws.com:5439/dev

Driver path  
-

Require SSL connection  
-

Security groups  
sg-0dbddfd5f38c01e01

Created on  
2024-03-23 15:49:41.471000

Class name  
-



Agora teremos que conectar o Crawler (do AWS Glue) ao Redshift, vá em Crawler e adicione um novo para o Redshift, conforme abaixo.

The screenshot shows the 'Set crawler properties' page 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. This section has a 'Name' field with the value 'glue-redshift-crawler-camila' and a 'Description' field with the placeholder 'Enter a description'. Below these fields, there is a 'Tags' section with the label 'Tags - optional' and a note 'Use tags to organize and identify your resources.' At the bottom right of the form, there are 'Cancel' and 'Next' buttons.

Cliquei em “Next”, depois adicionei uma nova database e a conexão do JDBC que é a do Redshift, informei o path.

The screenshot shows the 'Add data source' dialog box. It has a title bar with a close button. The main content area is divided into several sections. The 'Data source' section has a dropdown menu with 'JDBC' selected. The 'Connection' section has a dropdown menu with 'Redshift connection' selected and a refresh button. Below this are two buttons: 'Clear selection' and 'Add new connection'. The 'Include path' section has a text input field with the value 'dev/public/customer\_churn'. Below this is a paragraph of text explaining how to use the percent (%) character for schemas and tables. The 'Additional metadata - optional' section has a dropdown menu. At the bottom, there is a checkbox labeled 'Exclude files matching pattern' and two buttons: 'Cancel' and 'Add a JDBC data source'.

O IAM role pode ser o mesmo criado para o 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

Em “Target database” teremos que criar um para o Redshift. Criei em add database e depois selecionei.

[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](#)

Agora temos dois Crawlers no AWS Glue, um para fazer integração com o Redshift e outro com o 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

Ainda no AWS Glue, precisamos configurar o job ETL, em AWS Glue, à direita em ETL Jobs. Abrirá a tela abaixo, nomeei o job e em Visual, configurei cada parte do ETL, basta clicar em cada item e definir as configurações à direita.

As configurações para o S3 bucket, estão à direita. Nomeei e poderia indicar a localização do S3, mas como havia criado uma “Data Catalog table” direcionei para esta tabela.

The screenshot shows the AWS Glue console interface for a job named 's3\_upload\_to\_redshift\_gluejob'. The 'Data source properties - S3' panel is open, showing the configuration for the 'Data source - S3 bucket Amazon S3'. The 'Name' field is set to 'Amazon S3'. The 'S3 source type' is set to 'Data Catalog table'. The 'Database' is set to 'customer-churn-s3-glue-database'. The 'Table' is set to 'camila\_bucket\_aws'. The 'Data preview' section shows a table with columns: customerid, count, country, state, city, zip code. The 'Output schema' section shows the schema for the data source.

Transformação da base de dados, o “Change Schema”, apresenta as configurações à direita. Aqui fiz os ajustes nos dados brutos que estavam no S3, dropei e ajustei os nomes e os tipos das colunas. O Redshift não tem float ele tem o double, atentar para esta diferença.

The screenshot shows the AWS Glue console interface for a job named 's3\_upload\_to\_redshift\_gluejob'. The 'Transform' panel is open, showing the configuration for the 'Transform - Change Schema' job. The 'Name' field is set to 'Change Schema'. The 'Node parents' section shows the 'Amazon S3' data source. The 'Change Schema (Apply mapping)' section shows a table mapping for the 'customerid' column. The 'Source key' is 'customerid', the 'Target key' is 'customerid', and the 'Data type' is 'string'. The 'Drop' checkbox is checked. The 'Output schema' section shows the schema for the data source.

O carregamento no Redshift, abaixo as configurações do job. Poderia ter feito uma conexão direta, mas como já havia criada a tabela, só direcionei para ela. Defini que a tabela truncasse a cada nova carga de dados, assim os dados estarão sempre atualizados com os últimos dados disponíveis, mas também seria possível somente usar o Append, merge ou até recriar do zero a tabela.

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 parents  
Choose which nodes will provide inputs for this one.  
Choose one or more parent node  
Change Schema  
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	-

## Resultado

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

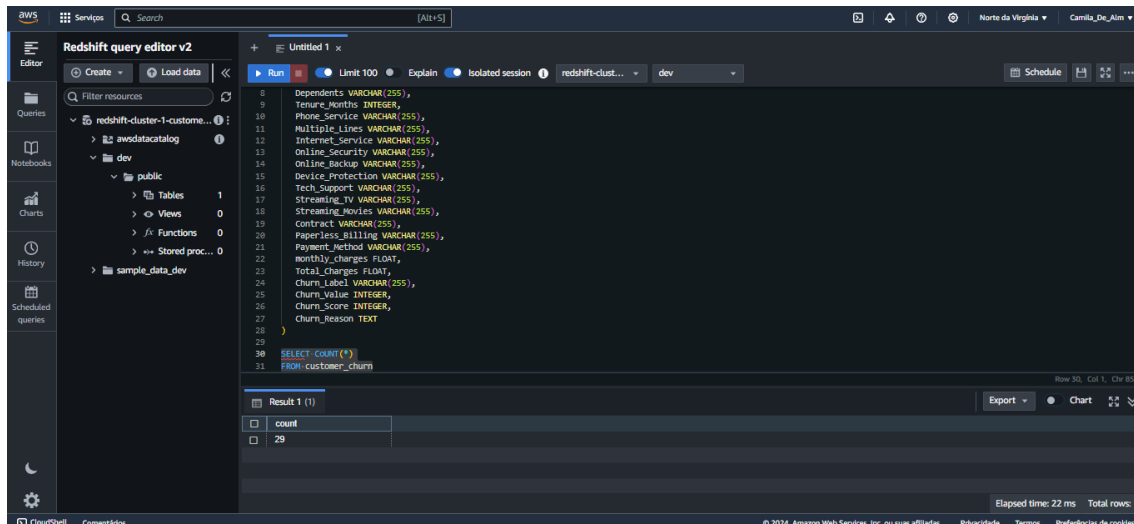
Visual Script Job details Runs Data quality - updated Schedules Version Control

Data source - S3 bucket Amazon S3

Transform - Change Schema

Data target - Amazon Redshift

Executei manualmente o job para ver se estava funcionando na tab Runs. Para ter certeza, volte no editor do Redshift e execute o comando Count, a tabela original estava vazia, somente tinha o schema, mas agora já apresenta alguns valores.



Podemos executar manualmente ao clicar em Run, ou automatizar com o Airflow, por isso fizemos a conexão do Airflow no VScode, nesta conexão executei o seguinte script:

**from airflow import DAG** # Importa a classe **DAG** do módulo **airflow**. A classe **DAG** é usada para definir e descrever um fluxo de trabalho em Airflow.

**from datetime import timedelta, datetime**

**from airflow.operators.python import PythonOperator** #Importa a classe PythonOperator do módulo operators.python do Airflow. O PythonOperator é usado para executar funções Python como tarefas em um fluxo de trabalho.

**from airflow.providers.amazon.aws.hooks.base\_aws import AwsGenericHook** #Importa a classe AwsGenericHook do módulo base\_aws do pacote providers.amazon.aws.hooks do Airflow. Essa classe é usada para estabelecer conexões com serviços da AWS.

**import time**

**from airflow.providers.amazon.aws.sensors.glue import GlueJobSensor** #Essa classe é usada para esperar até que um trabalho do AWS Glue seja concluído antes de continuar o fluxo de trabalho.

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

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

#Define a região em que eu estava us-west-2

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

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

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

**JobName=job\_name,**

**)**

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

**time.sleep(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

```

#Define um dicionário default\_args que contém os argumentos padrão para o DAG, como o proprietário, a data de início, as configurações de e-mail em caso de falha, 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)

}

```

#Inicia a definição do DAG com o nome 'my\_dag' e os argumentos padrão definidos anteriormente.

```

with DAG('my_dag',

        default_args=default_args,

        schedule_interval = '@weekly',

        catchup=False) as dag:

    glue_job_trigger = PythonOperator(

        task_id='tsk_glue_job_trigger',

        python_callable=glue_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,

    )

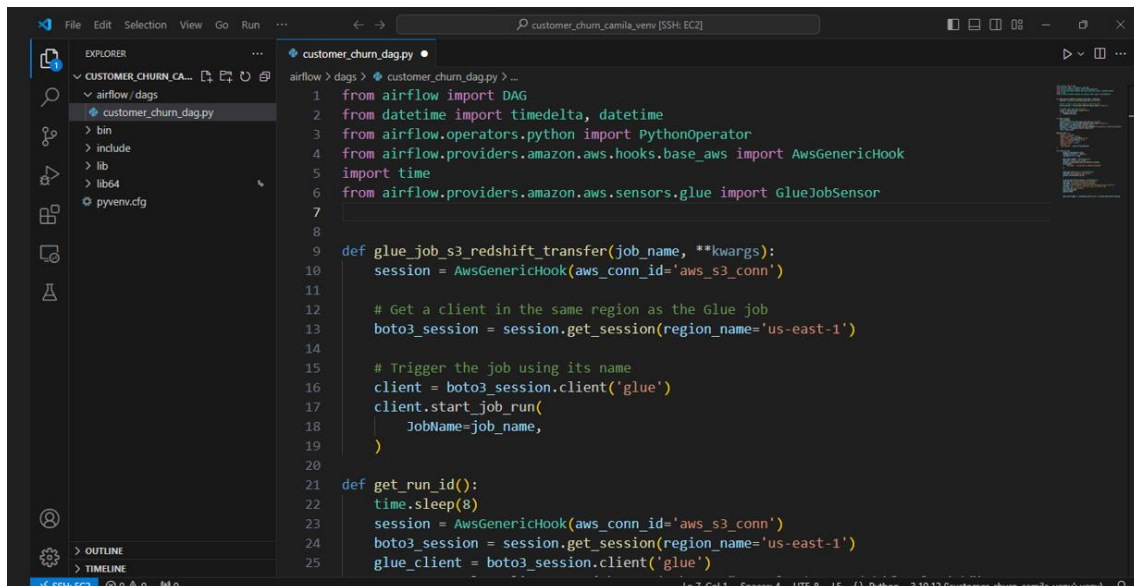
```

```

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, # prints glue job logs in airflow logs
    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')

```

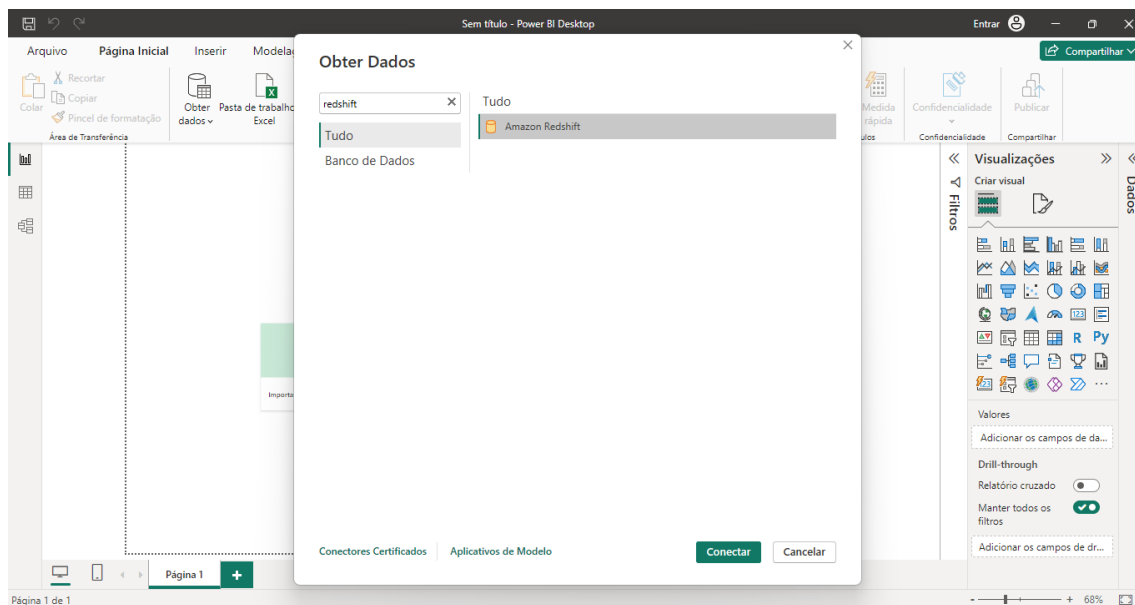
Após devemos conectar o Airflow via Access Keys do Usuário Raiz para automatizar os jobs acima, isso deve ser feito via navegador no Airflow tanto para o S3 quanto para o Redshift e via terminal da EC2, acessando novamente o ambiente virtual e informando Access Keys. Após execute novamente, o job no Crawler

Após verifique se o status está como “Running” no Crawler e Success no Airflow,

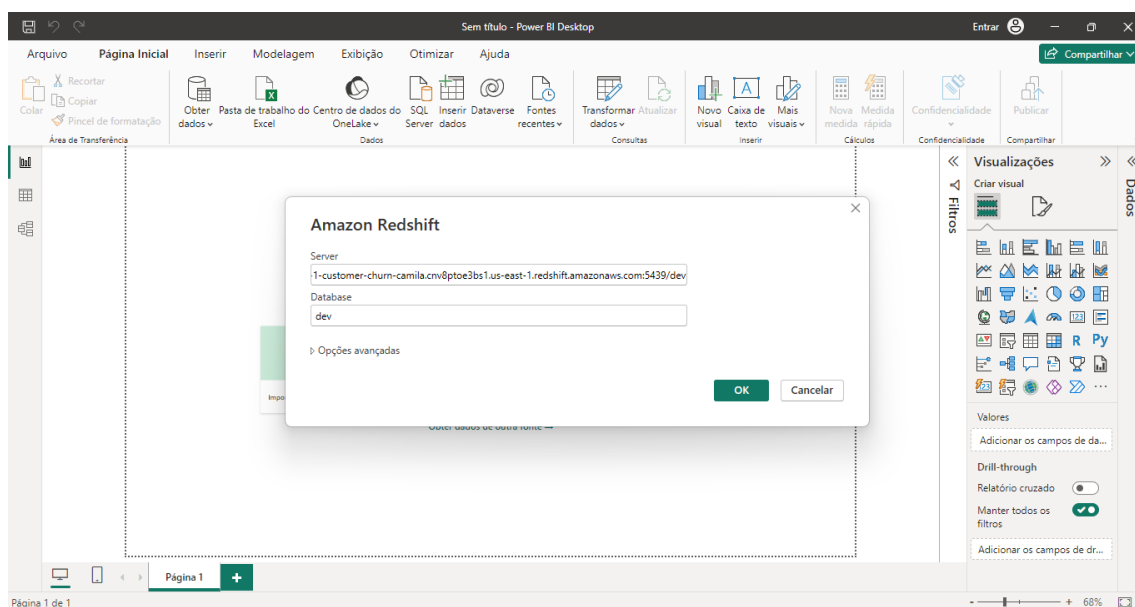


Por último apresentaremos a visualização dos dados no Microsoft Power BI. Faça conexão no PowerBI com o Redshift, não esqueça de habilitar o cluster para conexões fora da VPC.

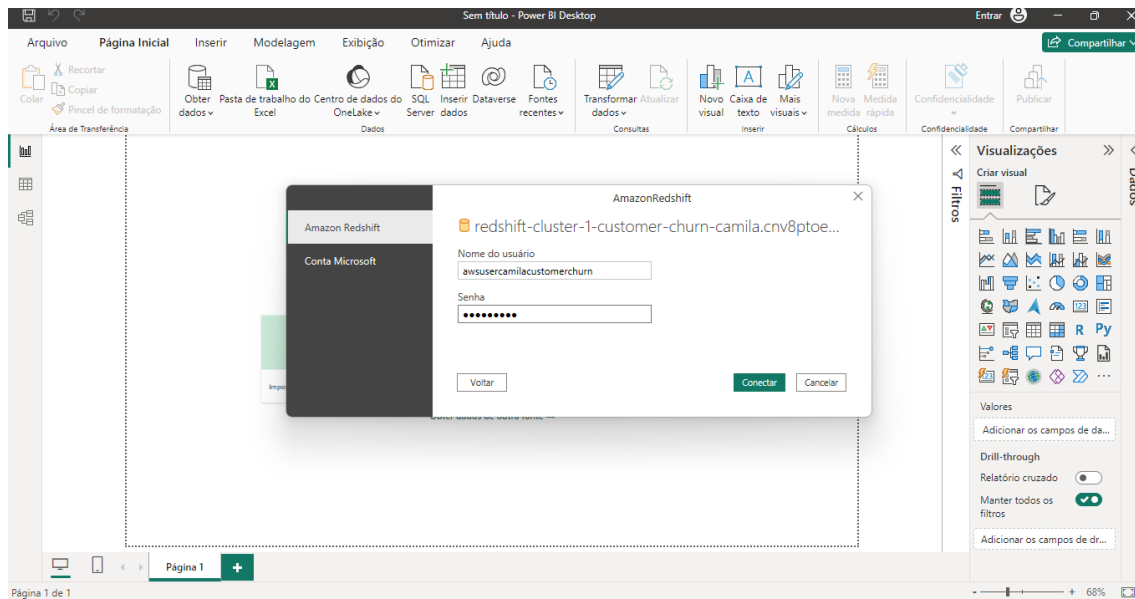




Vai pedir o server e a database, o server é o endpoint do cluster criado e a database é dev , nome que a própria AWS sugeriu.



Informe as credenciais do cluster.



Após será possível criar visualizações, com a tabela que será atualizada em tempo real a cada nova carga recebida no Bucket S3. Abaixo uma visualização que criei.

