

Laboratório de AWS Glue

- [Laboratório de AWS Glue](#)
- [Introdução](#)
 - [1 - Preparando os dados de origem](#)
 - [2 - Configurando sua conta para utilizar o AWS Glue](#)
 - [3 - Criando a IAM Role para os jobs do AWS Glue](#)
 - [4 - Configurando as permissões no AWS Lake Formation](#)
 - [5 - Criando novo job no AWS Glue](#)
 - [5.1 - Eliminando execuções de jobs](#)
 - [5.2 Sua vez!](#)
 - [6 - Criando novo crawler](#)

Introdução

Processos de ETL (Extract, Transform and Load) estão presentes em todos os projetos de dados. O cenário costuma ser o mesmo: fontes de dados diversas com datasets de interesse que precisam ser ingeridos, transformados e armazenados em um ou mais destinos, com formatos diferentes da origem.

Neste laboratório você será guiado na construção de um processo de ETL simplificado utilizando o serviço AWS Glue.

1 - Preparando os dados de origem

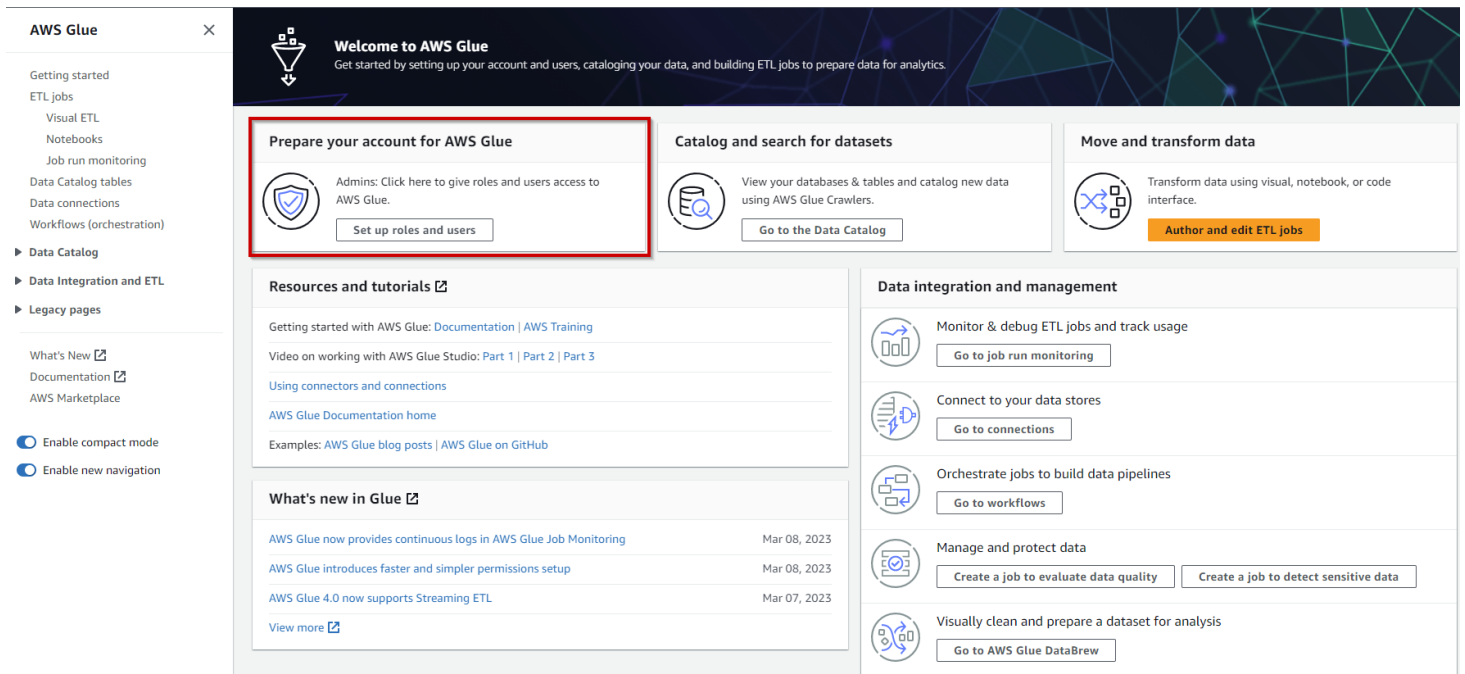
Faremos uso do arquivo *nomes.csv*, um dataset que contém os nomes mais comuns de registro de nascimento dos cartórios americanos entre os anos de 1880 e 2014. Trata-se de um arquivos CSV, com a estrutura descrita na amostra a seguir.

```
nome,sexo,total,ano  
Jennifer,F,54336,1983
```

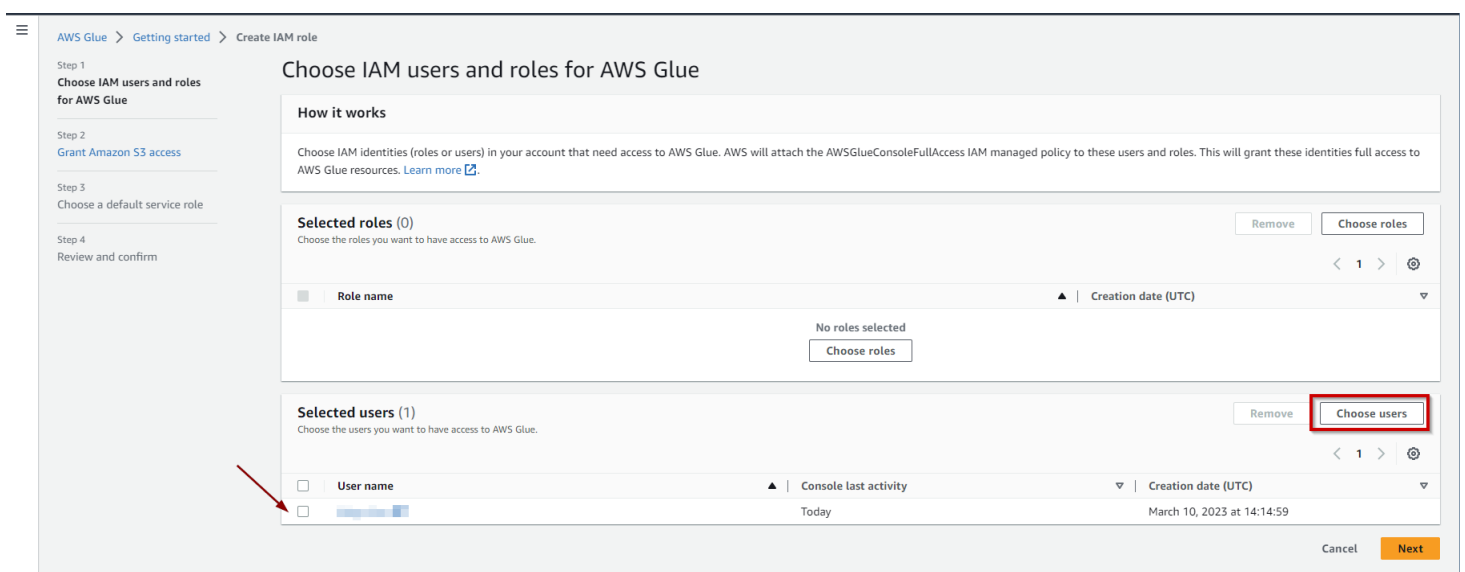
Para nosso laboratório, o arquivo deverá estar em um bucket do S3. Vamos considerar que o path do arquivo seja `s3://{BUCKET}/lab-glue/input/nomes.csv`. Lembre-se que o valor `{BUCKET}` deve ser substituído por um dos disponíveis em sua conta.

2 - Configurando sua conta para utilizar o AWS Glue

Acesse a página inicial do serviço AWS Glue. Para que possamos utilizar o serviço com as permissões necessárias, devemos seguir o passo-a-passo disponível a partir da opção **Set up roles and users** no *card* **Prepare your account for AWS Glue**.



No primeiro passo devemos indicar quais *roles* e *usuários* terão acesso ao serviço AWS Glue. Procure pelo seu usuário em **Choose users** e o adicione à lista.



No passo seguinte, informe acesso total ao S3 para leitura e escrita.

AWS Glue > Getting started > Create IAM role

Step 1
Choose IAM users and roles for AWS Glue

Step 2
Grant Amazon S3 access

Step 3
Choose a default service role

Step 4
Review and confirm

Grant Amazon S3 access

How it works

Grant S3 access to the users and roles that you selected in Step 1. Glue will attach permissions to those identities based on the type of access you choose here. [Learn more](#)

Choose S3 locations

Targeted users and roles
[1 users and 0 roles](#)

Choose access to Amazon S3

- ☐ No additional access
Do not change permissions.
- ☐ Add access to specific Amazon S3 locations
Choose specific S3 paths that you want to grant access to.
- ☒ **Grant full access to Amazon S3**
Grant access to all S3 resources in your AWS account.

Data access permissions
Set the type of data access for the Glue users and roles.

Data access permissions

- ☐ Read only (*recommended*)
- ☒ **Read and write**

Cancel Previous **Next**

Por fim, marque a opção **Update the standard AWS Glue service role and set it as the default (recommended)** e finalize o processo.

AWS Glue > Getting started > Create IAM role

Step 1
Choose IAM users and roles for AWS Glue

Step 2
[Grant Amazon S3 access](#)

Step 3
Choose a default service role

Step 4
Review and confirm

Choose a default service role

How it works

AWS Glue uses an IAM service role to run jobs, access data, and run Data Quality tasks. We recommend that you start with the standard AWSGlueServiceRole as the default. [Learn more](#)

Choose a default AWS Glue service role

IAM role for AWS Glue

- ☒ **Update the standard AWS Glue service role and set it as the default (*recommended*)**
AWS will update the role with the IAM policies needed to run AWS Glue jobs, then set it as the default.
- ☐ Set an existing IAM role as the default
Select an IAM role that you've configured to use as an AWS Glue service role. Glue will set this role as the default, but won't add any permissions to it. [Learn more](#)

The following IAM role will be created and automatically configured for you:

- AWSGlueServiceRole

Cancel Previous **Next**

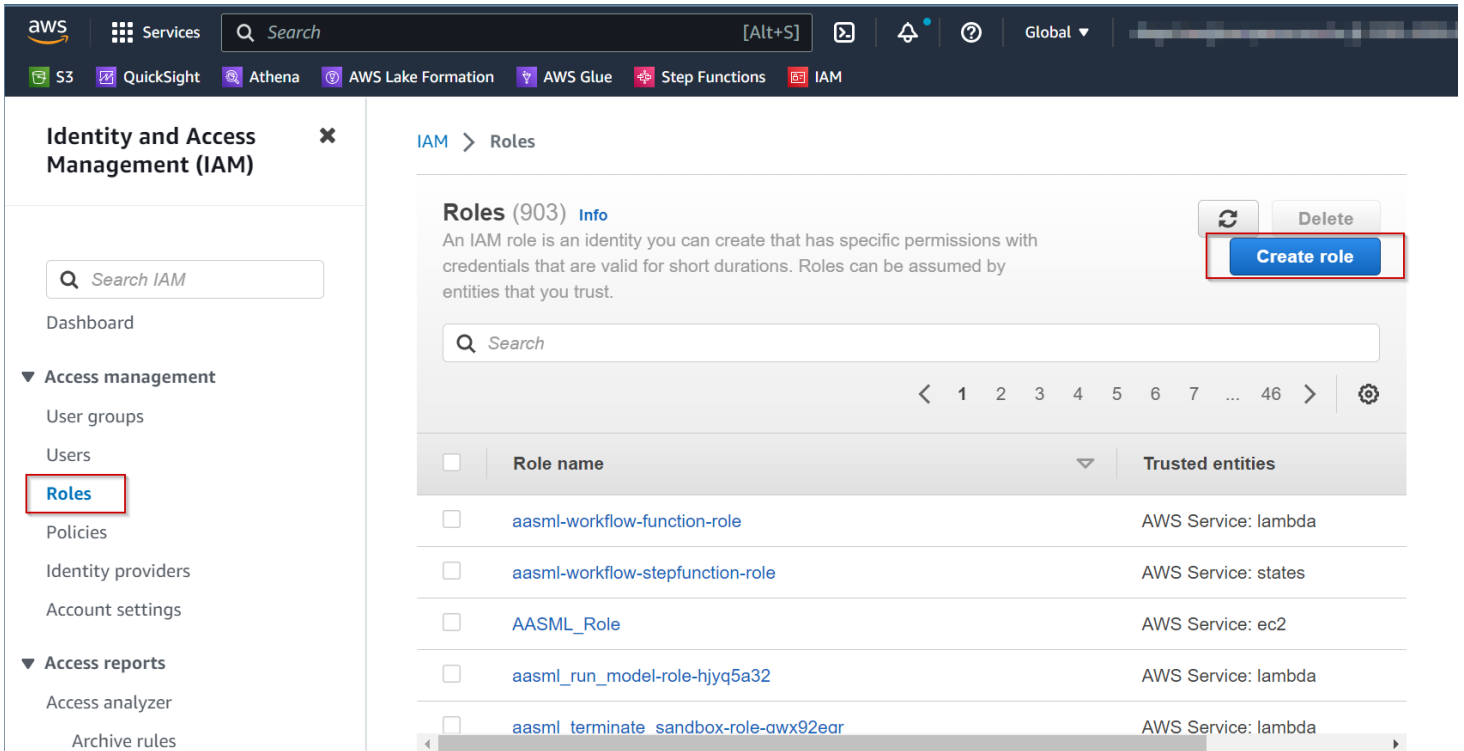
3 - Criando a IAM Role para os jobs do AWS Glue

Você deve estar lembrado que *Roles* são credenciais temporárias assumidas por serviços e aplicações para realizar operações em favor do usuário.

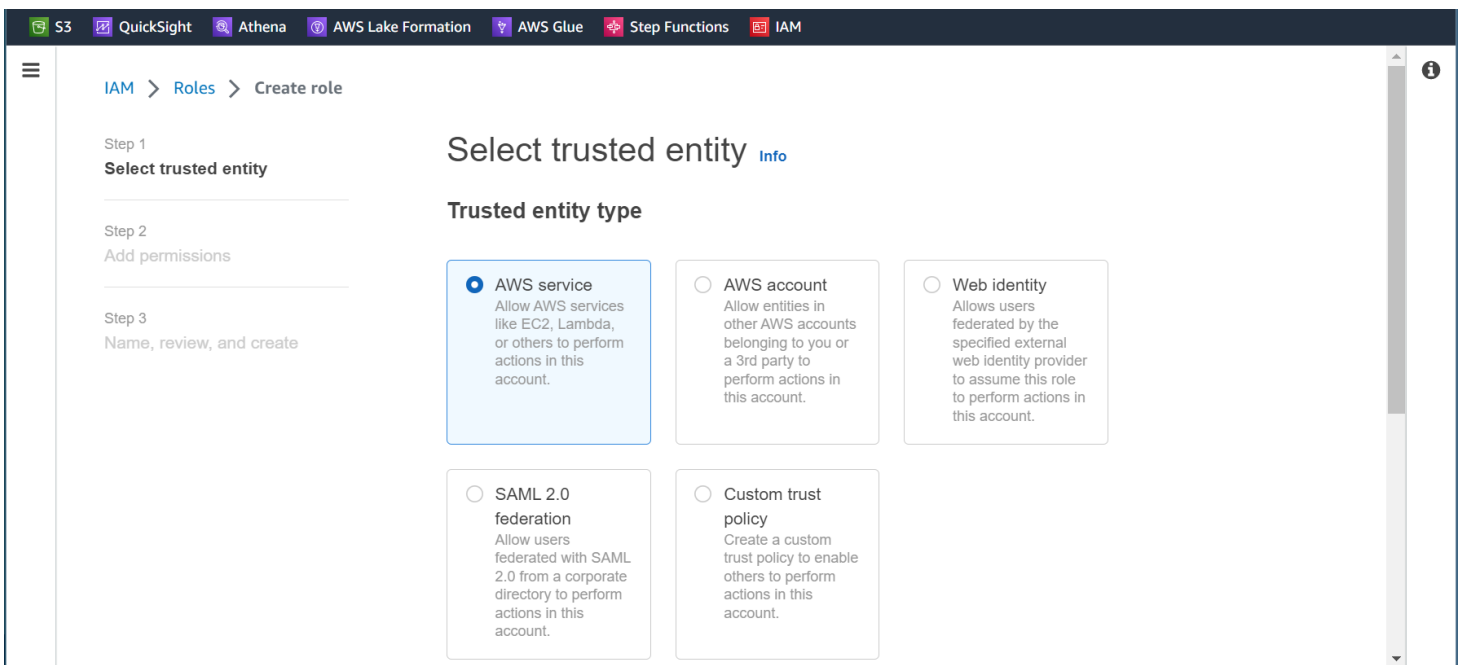
Logo, criaremos uma nova *role* chamada *AWSGlueServiceRole-Lab4*, associada a *policies* geridas pela AWS (*AmazonS3FullAccess*, *AWSLakeFormationDataAdmin*, *AWSGlueConsoleFullAccess* e *CloudWatchFullAccess*). Tais *policies* irão permitir acesso ao serviço do Glue ao **S3**, bem como outras ações, como executar códigos via *Notebooks*. Observe que estamos utilizando *policies* permissivas, o que vai de encontro ao princípio de privilégio mínimo que deve-se seguir em projetos reais. O objetivo aqui é simplificar o processo, apenas.

Vamos aos passos:

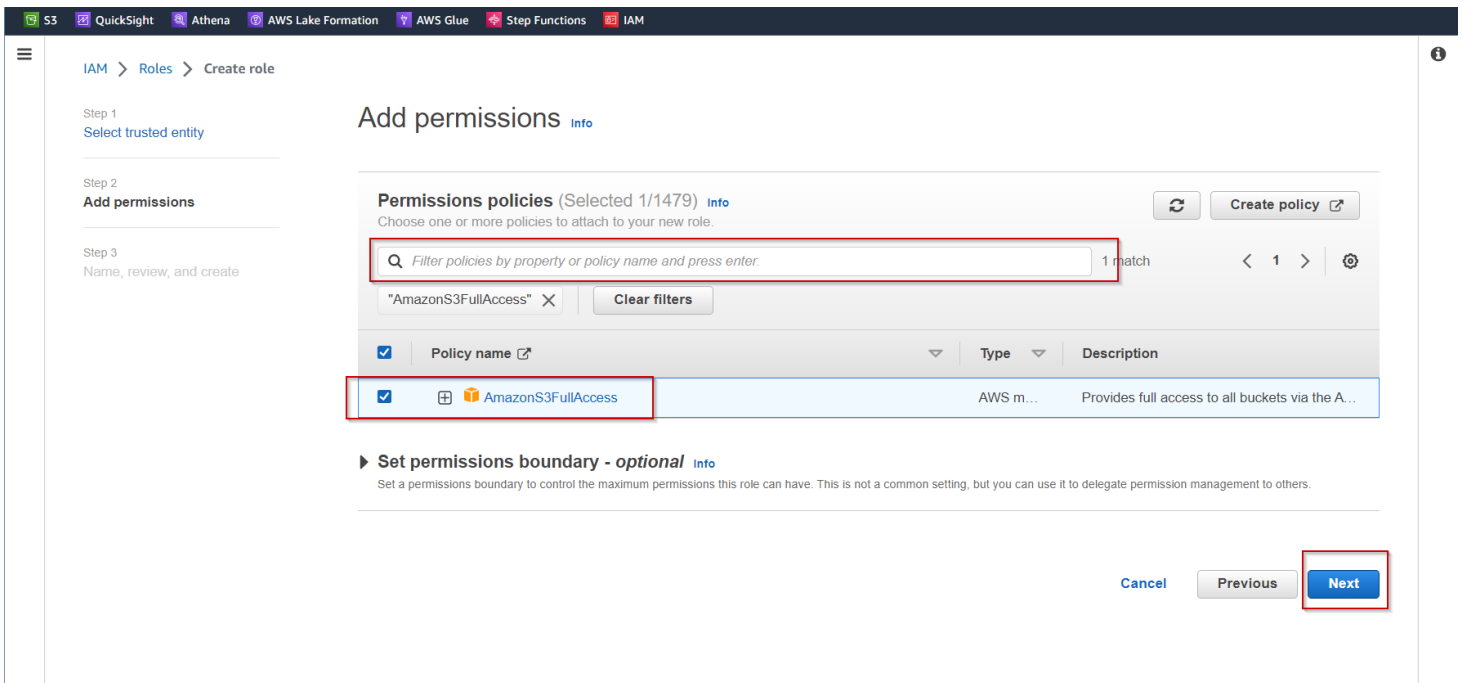
- No console, acesse a página do serviço **Identity and Access Management (IAM)** e clique no menu **Roles** à esquerda. Na sequência, clique no botão **Create Role**.



- Na primeira etapa, **Select trusted entity**, escolha **AWS Service** e para **Use Case**, informe **Glue**. Clique em **Next**.



- Na etapa **Add permissions**, pesquise por **AmazonS3FullAccess**, selecione a mesma da lista. Repita o processo para adicionar as demais *políticas* necessárias: **AWSLakeFormationDataAdmin_**, **AWSGlueConsoleFullAccess** e **CloudWatchFullAccess**. Em seguida, clique em **Next**.



- Na última etapa, informe em **Role name** o valor *AWSGlueServiceRole-Lab4* e, para finalizar, clique em **Create Role**.

4 - Configurando as permissões no AWS Lake Formation

AWS Lake Formation é um serviço que facilita a criação e gerenciamento de *data lakes*. Nos iremos utilizá-lo para criar o banco de dados no qual nosso *crawler* irá adicionar automaticamente uma tabela a partir dos dados armazenados no S3.

Após acessar o serviço **AWS Lake Formation** no console, clique na opção **Databases**, no menu à esquerda. Na sequência, clique no botão **Create Database**. O nome do novo banco deverá ser *glue-lab*. Observe que estamos criando um banco de dados no catálogo do Glue e não um banco de dados das características dos SGBD Relacionais.

AWS Lake Formation X

Dashboard

▼ Data catalog

Databases

Tables

Data filters

Data sharing [New](#)

Settings

▼ Register and ingest

Data lake locations

Blueprints

Crawlers [🔗](#)

Jobs [🔗](#)

▼ Permissions

Administrative roles and tasks

LF-Tags

LF-tag permissions

Data lake permissions

Data locations

External data filtering

Database details

Create a database in the AWS Glue Data Catalog.

☒ Database
Create a database in my account.

☐ Resource link
Create a resource link to a shared database.

Name
glue-lab

Location - optional
Choose an Amazon S3 path for this database, which eliminates the need to grant data location permissions on catalog table paths that are this location's children
e.g.: s3://bucket/prefix/ [Browse](#)

Description - optional
Enter a description

Descriptions can be up to 2048 characters long.

Default permissions for newly created tables
This setting maintains existing AWS Glue Data Catalog behavior. You can still set individual permissions, which will take effect when you revoke the Super permission from IAMAllowedPrincipals. See [Changing Default Settings for Your Data Lake](#).

☐ Use only IAM access control for new tables in this database

Cancel [Create database](#)

Agora precisamos que você adicione seu usuário IAM como administrador do *data lake*. Para tal, acesse a opção **Administrative roles and tasks** no menu à esquerda. Na tela que se apresenta, clique em **Choose administrators**.

AWS Lake Formation X

Dashboard

▼ Data catalog

Databases

Tables

Data filters

Data sharing [New](#)

Settings

▼ Register and ingest

Data lake locations

Blueprints

Crawlers [🔗](#)

Jobs [🔗](#)

▼ Permissions

Administrative roles and tasks

LF-Tags

LF-tag permissions

Data lake permissions

Data locations

External data filtering

AWS Lake Formation > Administrative roles and tasks

How it works

- 1 Set administrative roles**
Decide who should be the administrators for your data lake, and optionally who can create new databases.
[Choose administrators](#)
- 2 Define LF-tag ontology**
In order to create and manage catalog and data access permissions, define a set of LF-Tags that will help you quickly decide all types of access needs.
[Manage LF-Tags](#)
- 3 Delegate LF-tag permissions - optional**
Lastly, you can decide who should see the LF-tag ontology and LF-tag catalog resources (databases, tables, columns) in order to control data access.
[Manage LF-tag permissions](#)

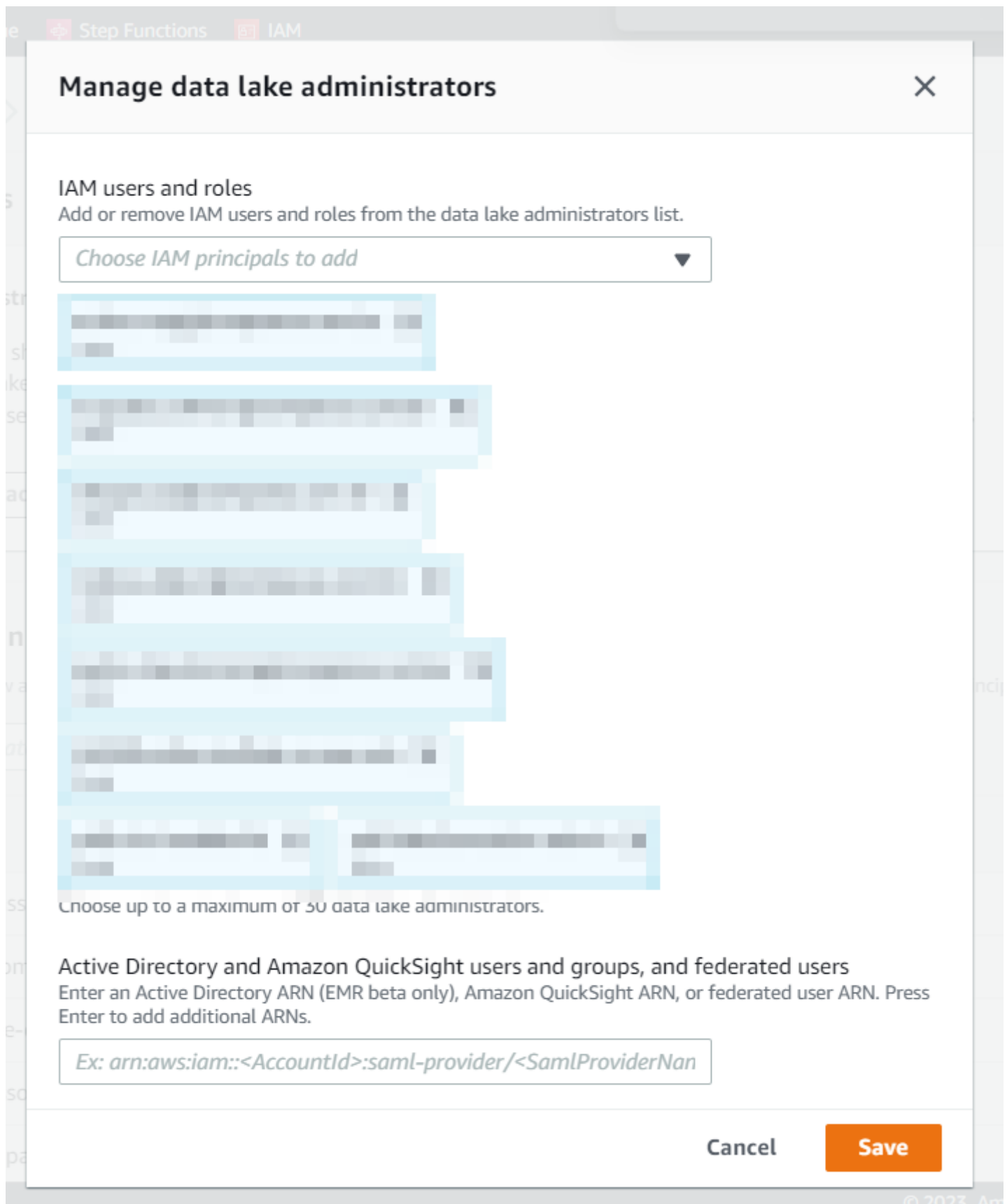
Data lake administrators (0/8)

Administrators can view all metadata in the AWS Glue Data Catalog. They can also grant and revoke permissions on data resources to principals, including themselves.

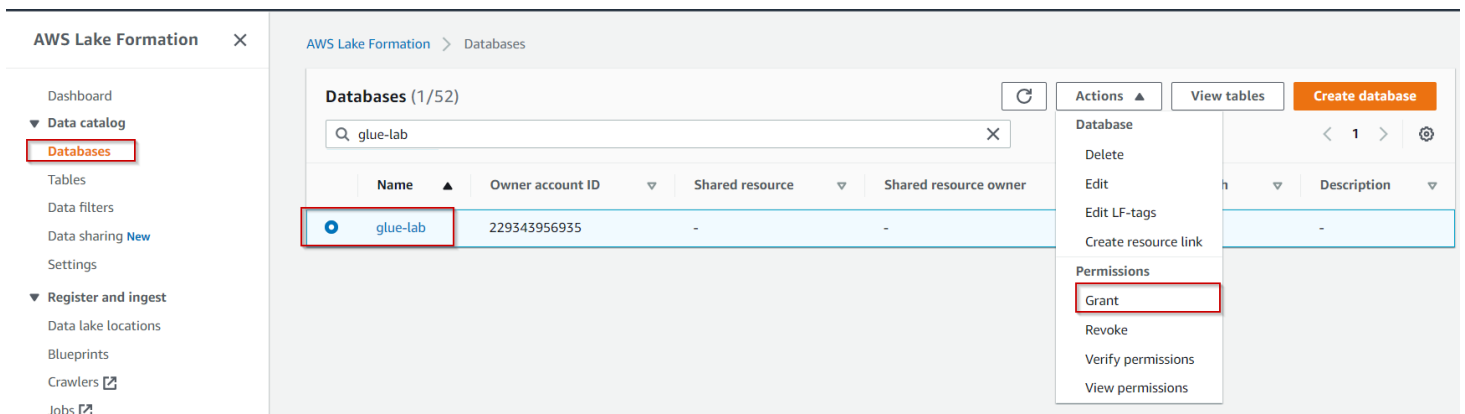
[🔄](#) [Choose administrators](#)

Name	Type

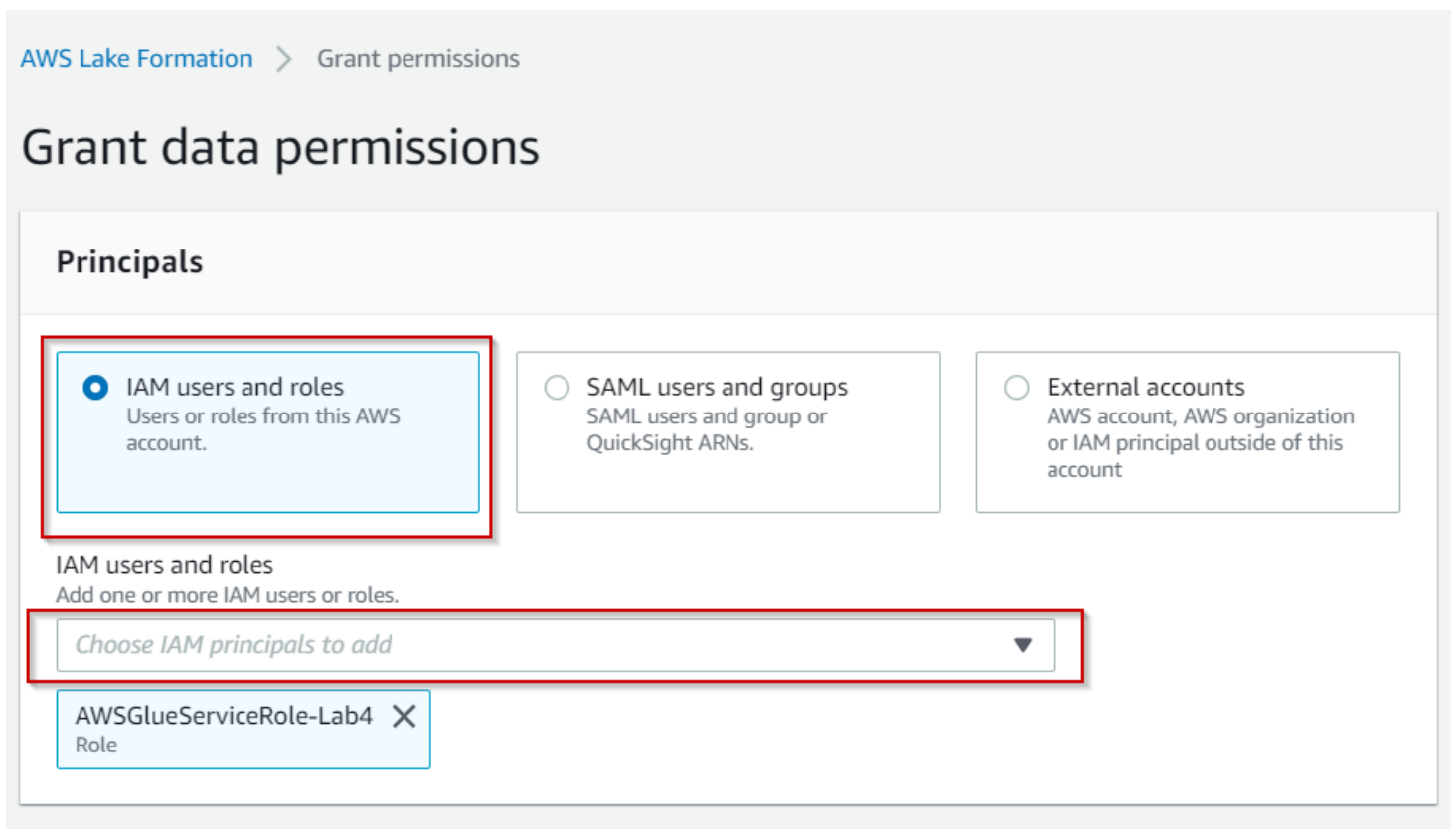
Procure pelo seu usuário em **IAM User and roles**. Adicione-o à lista e clique em **Save**.



Retorne ao menu **Databases**, busque pelo banco *glue-lab* criado anteriormente. Selecione-o e vá em **Actions**, escolhendo a opção **Grant**



Vamos conceder privilégios para a role do IAM criada anteriormente (*AWSGlueServiceRole-Lab4*). Para tal, escolha a opção **IAM users and roles** na seção **Principals**. Selecione a role a partir da lista apresentada.



Na seção **LF-Tags or catalog resources**, escolha **Named data catalog resources**, procurando pela base *glue-lab* em **Databases**.

LF-Tags or catalog resources

- ☐ Resources matched by LF-Tags (recommended)
Manage permissions indirectly for resources or data matched by a specific set of LF-Tags.

- ☒ Named data catalog resources
Manage permissions for specific databases or tables, in addition to fine-grained data access.

Databases

Select one or more databases.

Choose databases

Load more

glue-lab
229343956935

Tables - optional

Select one or more tables.

Choose tables

Load more

Data filters - optional

Select one or more data filters.

Choose data filters

Load more

Create new

[Manage data filters](#)

E, finalmente, na seção **Database permissions**, em **Database permissions**, escolha as opções **Create table**, **Alter**, **Drop** e **Describe**. Para finalizar, clique em **Grant**.

Database permissions

Database permissions

Choose specific access permissions to grant.

- ☒ Create table ☒ Alter ☒ Drop
☒ Describe

☐ Super

This permission is the union of all the individual permissions to the left, and supersedes them.

Grantable permissions

Choose the permission that may be granted to others.

- ☐ Create table ☐ Alter ☐ Drop
☐ Describe

☐ Super

This permission allows the principal to grant any of the permissions to the left, and supersedes those grantable permissions.

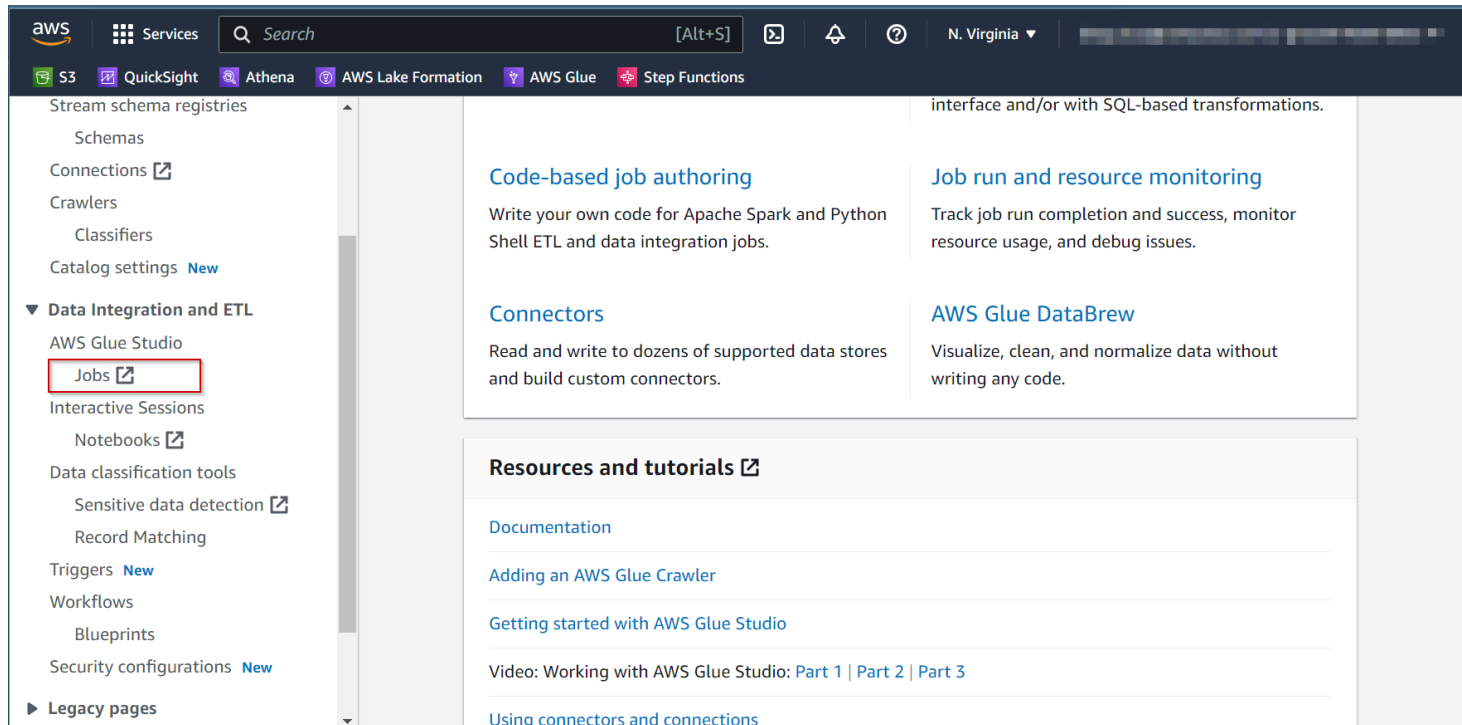
Cancel

Grant

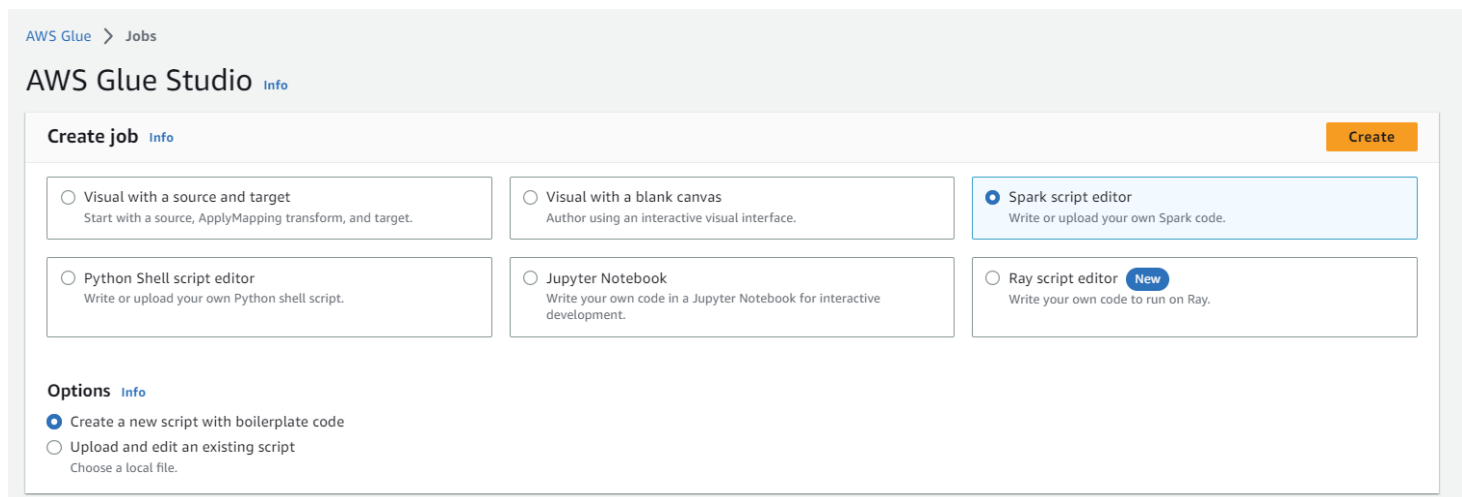
5 - Criando novo job no AWS Glue

Para realizar o processamento do arquivo *nomes.csv* iremos criar um *job* através do serviço **AWS Glue**.

Após acessar a página inicial na console, busque pela opção *Job* no menu à esquerda.



Você perceberá que existem diferentes opções para criarmos um *job*. Em nosso laboratório, faremos uso da opção **Spark script editor**, escolhendo a alternativa **Create a new script with boilerplate code**. Após, clique em **Create**.



Na sequência você estará na tela de edição do código e das configurações do seu *job*. Iniciaremos pela configuração, através da aba **Job details**.

Basic properties [Info](#)

Name

Untitled job

Description - *optional*

Descriptions can be up to 2048 characters long.

IAM Role

Role assumed by the job with permission to access your data stores. Ensure that this role has permission to your Amazon S3 sources, targets, temporary directory, scripts, and any libraries used by the job.

AWSGlueServiceRole



Type

The type of ETL job. This is set automatically based on the types of data sources you have selected.

Spark



Glue version [Info](#)

Glue 3.0 - Supports spark 3.1, Scala 2, Python 3



Language

Python 3



Worker type

Set the type of predefined worker that is allowed when a job runs.

As propriedades que iremos configurar serão:

- **Name:** Corresponde ao nome do job. Informe *job_aws_glue_lab_4*.
- **IAM Role:** Informe *AWSGlueServiceRole-Lab4*.
- **Type:** Mantenha *Spark*.
- **Glue version:** Manhanha *Glue 3*
- **Language:** Python 3
- **Worker Type:** Escolha *G 1x*, ou seja, a opção com menos vCPUs e RAM.
- A opção *Automatically scale the number of workers* deve estar desmarcada.
- **Requested number of workers:** Informe 2.
- **Number of retries:** Informe 0.
- **Job timeout (minutes):** Informe 5.

Em **Advanced properties**, informe:

- *Script filename*: Defina o nome do seu script.
- *Spark UI*: Desmarque a opção.

Agora você pode clicar em **Save**.

Você deve ter percebido que na aba **Script** há um código base para você iniciar o desenvolvimento. O código é semelhante a este:

```
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job

## @params: [JOB_NAME]
args = getResolvedOptions(sys.argv, ['JOB_NAME'])

sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args['JOB_NAME'], args)
job.commit()
```

Perceba que o código já oferece um objeto de sessão do Spark (`spark = glueContext.spark_session`) que você pode utilizar para realizar as atividades propostas na sequência.

Todo código que você construir, deverá estar entre os comandos `job.init(args['JOB_NAME'], args)` e `job.commit()` .

Vamos imaginar que o objetivo seja ler um arquivo CSV do S3, filtrar os dados pelo ano de 1934 e armazenar o resultado para PARQUET, em outro local do S3. O código a seguir realiza justamente tal tarefa.

Para abordar parâmetros, vamos considerar a existência de 2 neste *job*:

- **S3_INPUT_PATH**: Indica nosso caminho do origem no S3.
- **S3_TARGET_PATH**: Indica nosso caminho de destino no S3.

Os parâmetros são utilizados para tornar o código flexível, genérico. Este é sempre um ponto importante a considerar. Você pode criá-los na aba *Job Details*, opção *Advanced Options*, *Job*

parameters. Eles devem iniciar com --.

laboratorio-glue

Script

Job details

Runs

Data quality New

Schedules

Version Control

Dependent JARs path

Referenced files path

Job parameters Info

Key

Value - optional

Q --S3_INPUT_PATH X

Q s3://a X

Remove

Q --S3_TARGET_PATH X

Q s3://a X

Remove

Q Enter key

Q Enter value

Remove

Add new parameter

You can add 47 more parameters.

Veja o código de exemplo:

```

import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job

## @params: [JOB_NAME]
args = getResolvedOptions(sys.argv, ['JOB_NAME', 'S3_INPUT_PATH', 'S3_TARGET_PATH'])

sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args['JOB_NAME'], args)

source_file = args['S3_INPUT_PATH']
target_path = args['S3_TARGET_PATH']

df = glueContext.create_dynamic_frame.from_options(
    "s3",
    {
        "paths": [
            source_file
        ]
    },
    "csv",
    {"withHeader": True, "separator": "|"},
)

only_1934 = df.filter(lambda row: row['anoLancamento']=='1934')

glueContext.write_dynamic_frame.from_options(
    frame = only_1934 ,
    connection_type = "s3",
    connection_options = {"path": target_path},
    format = "parquet")

job.commit()

```

No exemplo estamos utilizando *dynamic_frames*, uma abstração sobre um dataframe Spark oferecida pelo Glue. Naturalmente nós podemos alternar entre *dynamic frames* e *dataframes* conforme demonstramos no exemplo que segue.

```
#Obtendo um dataframe Spark a partir de um dataframe dinâmico do Glue
spark_df = my_dynamic_df.toDF()
```

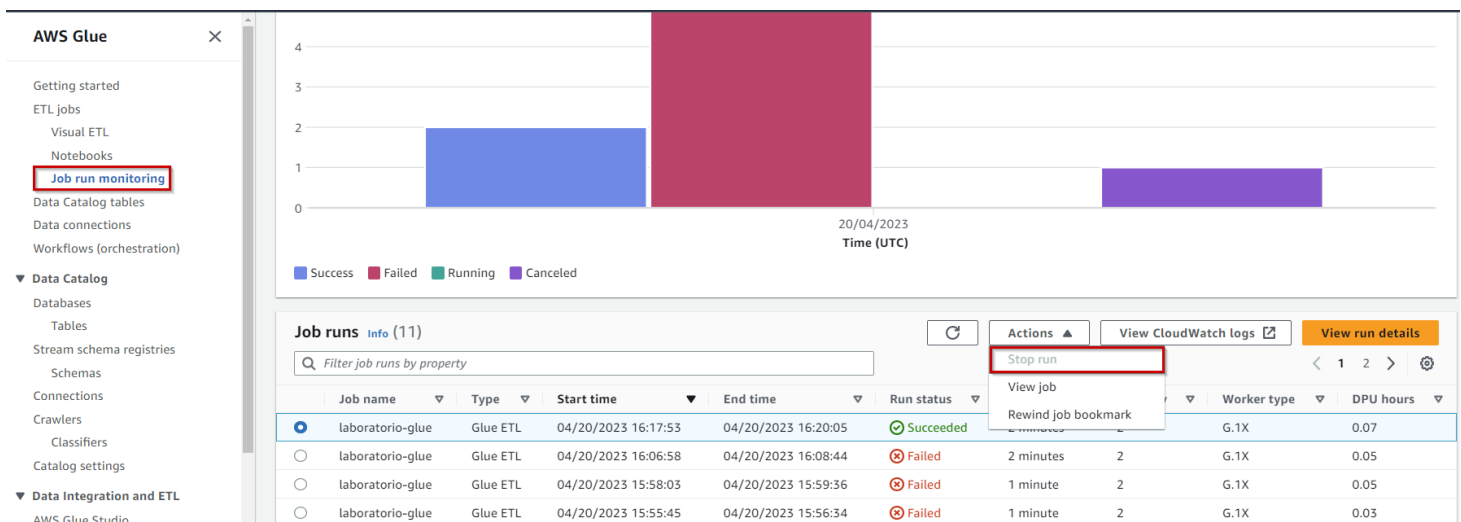
```
#Obtendo um dataframe dinâmico do Glue a partir de um dataframe Spark
dynamic_df = DynamicFrame(spark_df, glueContext)
```

Para mais informações sobre o desenvolvimento de *jobs* ETL com Glue, você pode acessar o endereço [Program AWS Glue ETL scripts in PySpark](#).

5.1 - Eliminando execuções de jobs

Após executar *jobs*, devemos nos certificar que não hajam sessões em execução desnecessárias. Para tal, acesse a opção **Job run monitoring** no menu à esquerda do Console.

Caso hajam execuções em andamento e você não precisa mais delas, solicite a finalização da mesmas. O processo é simples e compreende escolher a execução e ir até o botão **Actions**, opção **Stop run**.



5.2 Sua vez!

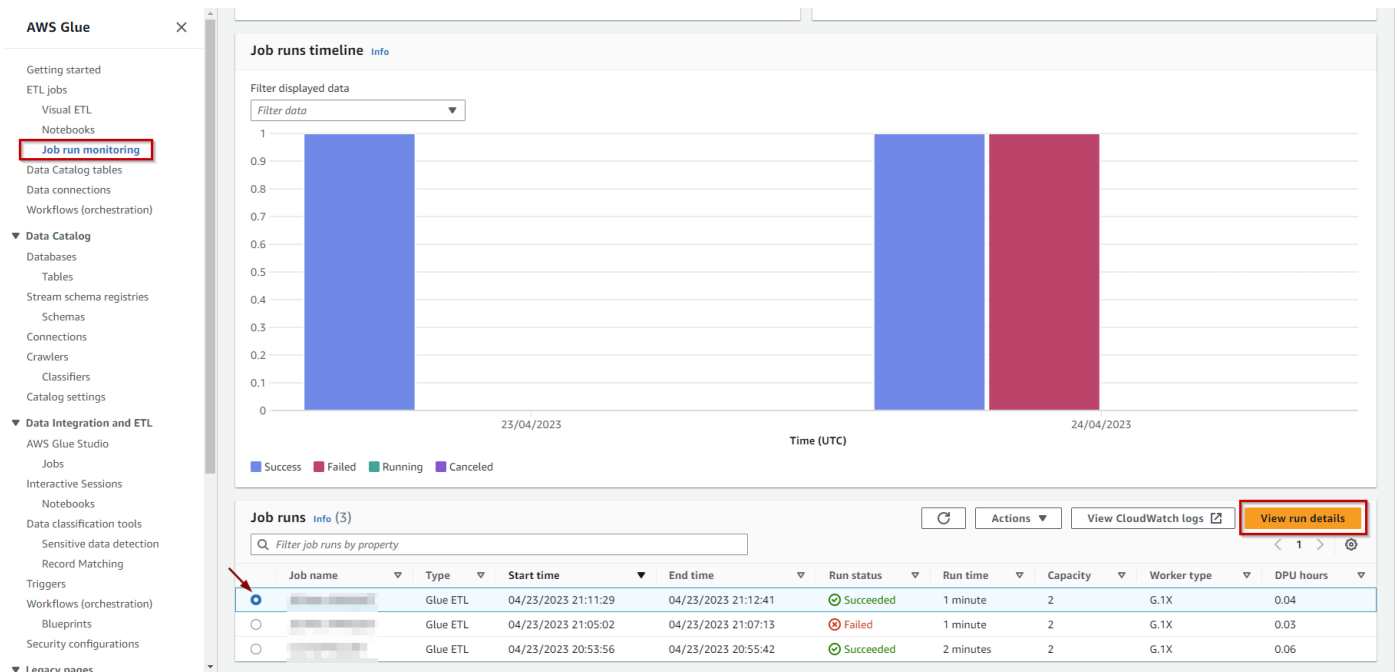
Agora vamos construir um *job* Glue nos moldes dos exemplos anteriores. Seguem os passos para você desenvolver:

- Ler o arquivo *nomes.csv* no S3 (lembre-se de realizar upload do arquivo antes).
- Imprima o schema do dataframe gerado no passo anterior.
- Escrever o código necessário para alterar a caixa dos valores da coluna *nome* para MAIÚSCULO.
- Imprimir a contagem de linhas presentes no dataframe.

- Imprimir a contagem de nomes, agrupando os dados do dataframe pelas colunas *ano* e *sexo*. Ordene os dados de modo que o *ano* mais recente apareça como primeiro registro do dataframe.
- Apresentar qual foi o nome feminino com mais registros e em que ano ocorreu.
- Apresentar qual foi o nome masculino com mais registros e em que ano ocorreu.
- Apresentar o total de registros (masculinos e femininos) para cada ano presente no dataframe. Considere apenas as primeiras 10 linhas, ordenadas pelo ano, de forma crescente.
- Escrever o conteúdo do dataframe com os valores de nome em maiúsculo no S3.
 - Atenção aos requisitos:
 - A gravação deve ocorrer no subdiretório *frequencia_registro_nomes_eua* do path `s3://<BUCKET>/lab-glue/`
 - O formato deve ser JSON
 - O particionamento deverá ser realizado pelas colunas *sexo* e *ano* (nesta ordem)

Algumas dicas:

- Os logs de execução estarão disponíveis através do menu **Job run monitoring**, opção **View run details**.



- Dentre as opções apresentadas, busque pela seção **CloudWatch continuous logs**.

CloudWatch continuous logs

Driver logs

[Driver and executor log streams](#)

```
23/04/24 00:12:30 INFO DAGScheduler: Job 1 finished: save at NativeMethodAccessorImpl.java:0, took 3.768684 s
23/04/24 00:12:30 INFO DAGScheduler: Job 1 is finished. Cancelling potential speculative or zombie tasks for this job
23/04/24 00:12:26 INFO DAGScheduler: Got job 1 (save at NativeMethodAccessorImpl.java:0) with 1 output partitions
23/04/24 00:12:24 INFO MultipartUploadOutputStream: close closed:false s3://aws-glue-assets-985421866342-us-east-1/sparkHistoryLogs
23/04/24 00:12:23 INFO DAGScheduler: Job 0 finished: json at NativeMethodAccessorImpl.java:0, took 21.907152 s
23/04/24 00:12:23 INFO DAGScheduler: Job 0 is finished. Cancelling potential speculative or zombie tasks for this job
23/04/24 00:12:16 WARN TaskSchedulerImpl: Initial job has not accepted any resources; check your cluster UI to ensure that workers
23/04/24 00:12:01 INFO DAGScheduler: Got job 0 (json at NativeMethodAccessorImpl.java:0) with 1 output partitions
23/04/24 00:11:58 INFO GlueContext: GlueMetrics configured and enabled
23/04/24 00:11:56 INFO Utils: Successfully started service 'sparkDriver' on port 45647.
```

- Você encontrará informações complementares sobre desenvolvimento de Scripts ETL com Glue na [documentação oficial do produto](#).

6 - Criando novo crawler

Crawlers são mecanismos que podemos utilizar para monitorar nosso armazenamento de dados de modo a criar/atualizar metadados no catálogo do Glue de forma automática. Na sequência iremos desenvolver um crawler para automaticamente criar uma tabela chamada **frequencia_registro_nomes_eua** a partir dos dados escritos no S3 (verifique a última atividade do notebook).

Vamos as passos para criação de nosso crawler:

- No console, acesse o serviço **AWS Glue**. Na página do serviço, escolha a opção **Crawlers** no menu à esquerda. Na sequência, clique no botão **Create**.
- No primeiro passo de criação do **Crawler**, informe *FrequenciaRegistroNomesCrawler* no campo **Name**. Clique em **Next**.

AWS Glue > Crawlers > Add new 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 crawler properties

Crawler details [Info](#)

Name
FrequenciaRegistroNomesCrawler

Name may contain letters (A-Z), numbers (0-9), hyphens (-), or underscores (_), and can be up to 255 characters long.

Description - *optional*
Enter a description

Descriptions can be up to 2048 characters long.

► **Tags - optional**
Use tags to organize and identify your resources.

Cancel **Next**

- Em **Choose data sources and classifiers**, devemos informar o caminho do S3 a ser monitorado. Para **Is your data already mapped to Glue tables?**, informe **Not yet**. E, na sequência, clique em **Add a data source**.

AWS Glue > Crawlers > Add new 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

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 (0) [Info](#) [Edit](#) [Remove](#) **Add a data source**

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

Type	Data source	Parameters
You don't have any data sources.		

[Add a data source](#)

- Na tela aberta, em **Data source**, certifique que esteja S3. Em **Location of S3 data**, informe **In this account**. Finalmente, no campo **S3 path**, informe o caminho `s3://<BUCKET>/lab-glue/frequencia_registro_nomes_eua/`, lembrando de substituir `<BUCKET>` pelo utilizado anteriormente.

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 selection](#) [Add new connection](#)

Location of S3 data

☒ In this account
☐ In a different account

S3 path
Browse for or enter an existing S3 path.

s3:// [View](#) [Browse](#)

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.

☐ Sample only a subset of files

☐ Exclude files matching pattern

[Cancel](#) [Add an S3 data source](#)

- Na etapa **Configure security settings** informe a role `AWSGlueServiceRole-Lab4` no campo **Existing IAM role**. Avance clicando em **Next**.

Configure security settings

Existing IAM role

AWSGlueServiceRole-Lab4

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 Preview

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 belongs to 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 and Glue Catalog data sources.

Security configuration - optional

Enable at-rest encryption with a security configuration.

Cancel Previous Next

- Em **Set output and scheduling**, no campo **Target database**, informe **glue-lab**. Em **Crawler schedule**, no campo **Frequency**, defina **On Demand**. Avance e finalize o processo de criação.

Set output and scheduling

Output configuration Info

Target 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

Crawler criado, agora vamos executá-lo. Na tela inicial (Crawlers), selecione **FrequenciaRegistroNomesCrawler** e clique em **Run**. A execução pode leva alguns segundos e você pode acompanhar o resultado na própria tela em que está.

Se a execução for bem sucedida, nós esperamos que uma nova tabela, de nome **frequencia_registro_nomes_eua** tenha sido criada na base **glue-lab**. Você pode vê-la por meio do **Glue Catalog** e também no **Athena**.

Para consultar os dados, você deverá conceder privilégios de **DESCRIBE** e **SELECT** no **Lake Formation**. Vamos deixar essa parte de desafio para você 😊.