**Amazon Redshift, Glue Catalog, Athena & Amazon QuickSight demo**

**Participant’s Guide**

Version 1.1

*Authored by*

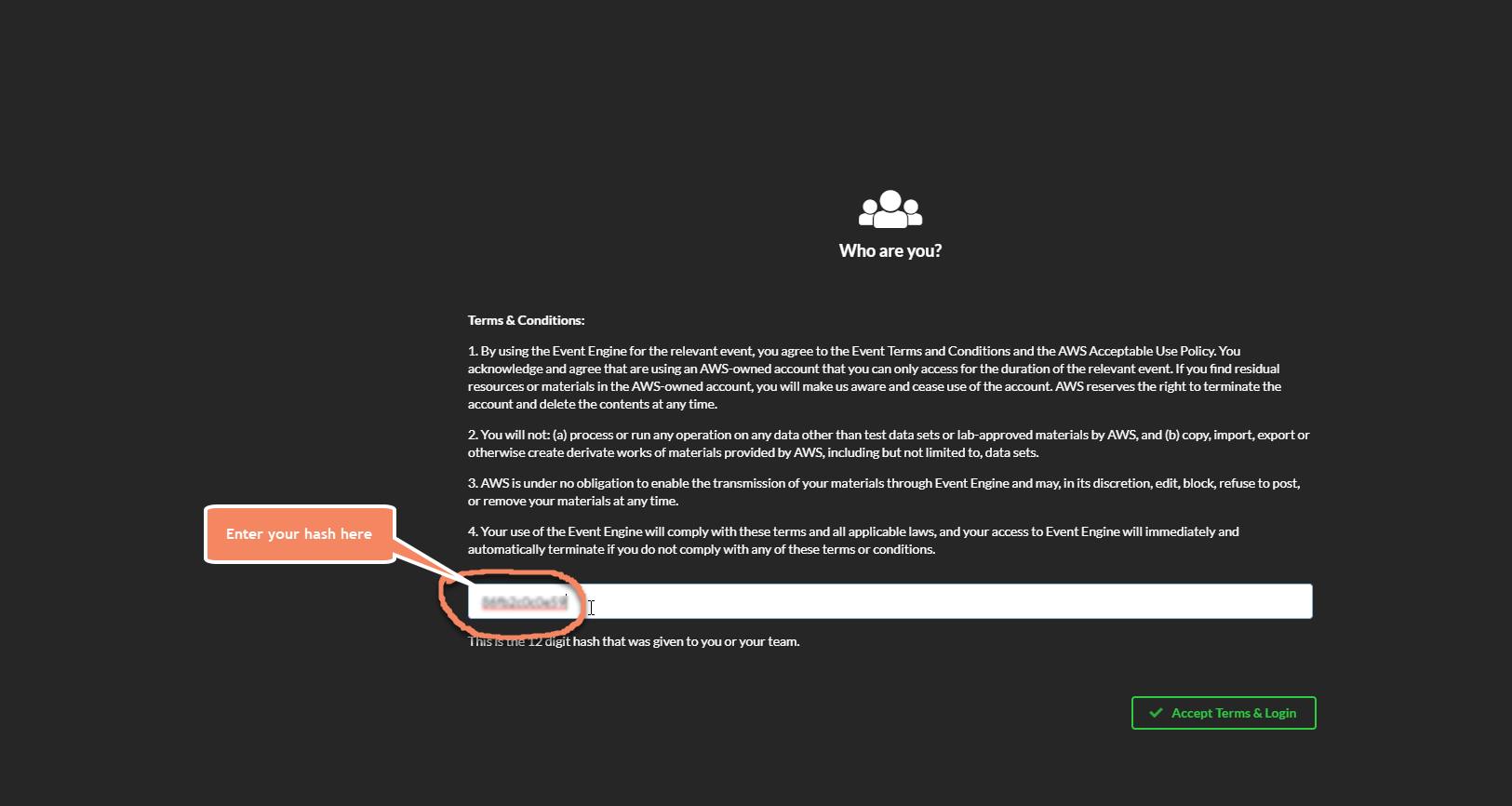
*Saunak Chandra, Partner Solutions Architect,* saunak@amazon.com

# In this lab you will create an end-to-end ELT pipeline using AWS resources to ingest and visualize [NYC taxi](https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page) sample data. This lab provides you with resources generated using Cloudformation template and requires very minimal setup. As you walk through the lab you will gradually get introduced to Amazon Redshift, AWS Glue, AWS Step Function, Ahena and QuickSight as your building blocks for ETL pipeline.

# **Log in to your lab AWS account:**

The lab environment is created using AWS Event Engine. The dashboard to login into the AWS console is <https://dashboard.eventengine.run/dashboard>.

You will be given a 12 character hashkey. Enter the hashkey to enter the login environment.



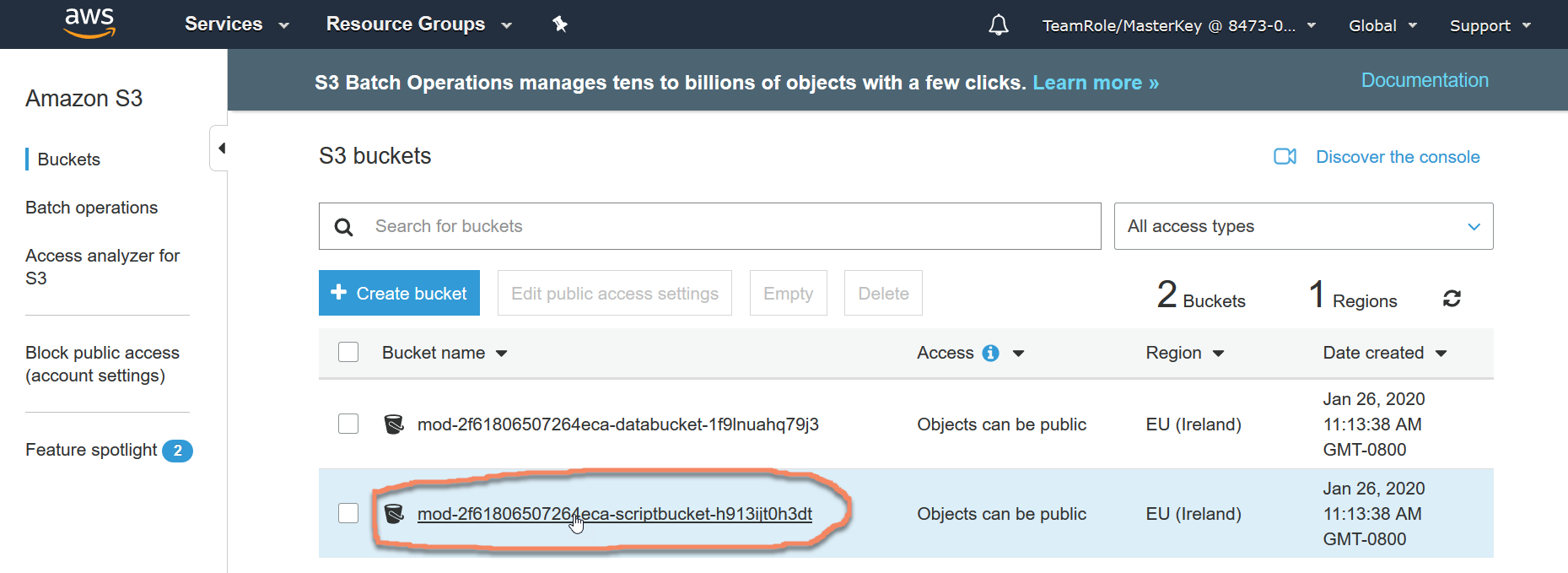
Accept Term and Login takes you to Team Dashboard page. Click **AWS Console** and in the next page AWS Console Login click **Open AWS Console**. Now you are in your team AWS account to start this lab.

# **Pre-requisites:**

In this lab a Python Shell job is the heart of all the ELT being done in Glue. This Python scripts and other SQL scripts are required to be placed in your lab specific s3 bucket.

The very first thing you will do after logging in into your team AWS account is to navigate to S3 service.

Navigation: **Services > S3**



You will see 2 buckets already created for you- one is scriptbucket and another is databucket.

Next you will replace files in each of these buckets. Follow below steps.

Note: If you are not familiar working with s3 console please ask the instructor how to perform create folder, upload files etc.

* First download necessary files from below locations into your local machine.
  + python <https://github.com/saunakc/redshift-dataengineering-lab/tree/master/python>
  + sql <https://github.com/saunakc/redshift-dataengineering-lab/tree/master/sql>
  + data <https://github.com/saunakc/redshift-dataengineering-lab/tree/master/data>
* Upload these downloaded files into your lab specific s3 bucket in proper location as shown below-
  + [python](https://github.com/saunakc/redshift-dataengineering-lab/tree/master/python) 🡪 mod{randomstring}-**scriptbucket**-{anotherrandomstring}/python
  + [sql](https://github.com/saunakc/redshift-dataengineering-lab/tree/master/sql) 🡪 mod{randomstring}-**scriptbucket**-{anotherrandomstring}/sql
  + [data](https://github.com/saunakc/redshift-dataengineering-lab/tree/master/data) 🡪 mod{randomstring}-**databucket**-{anotherrandomstring}/csv

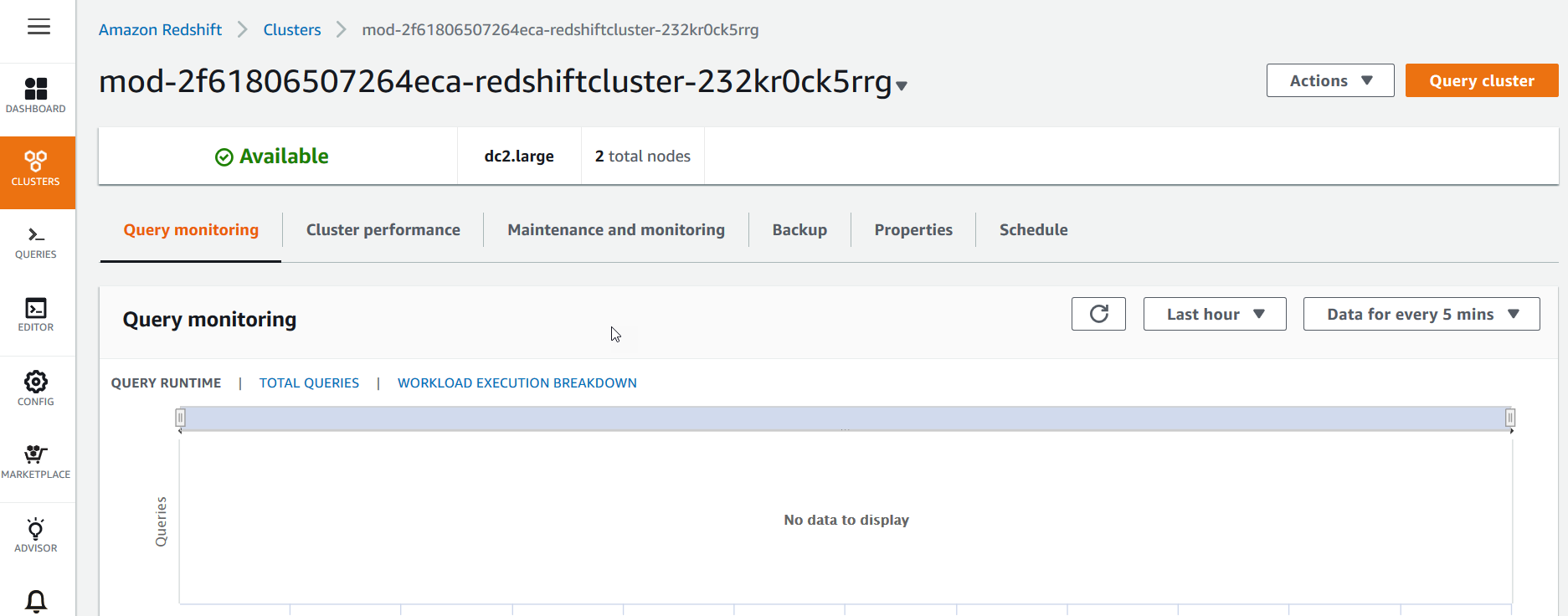
# **Navigate to your Amazon Redshift cluster:**

Navigation: **Services > Amazon Redshift > Clusters**.

For this lab a 2 node dc2.large Redshift cluster has already been crated for you.

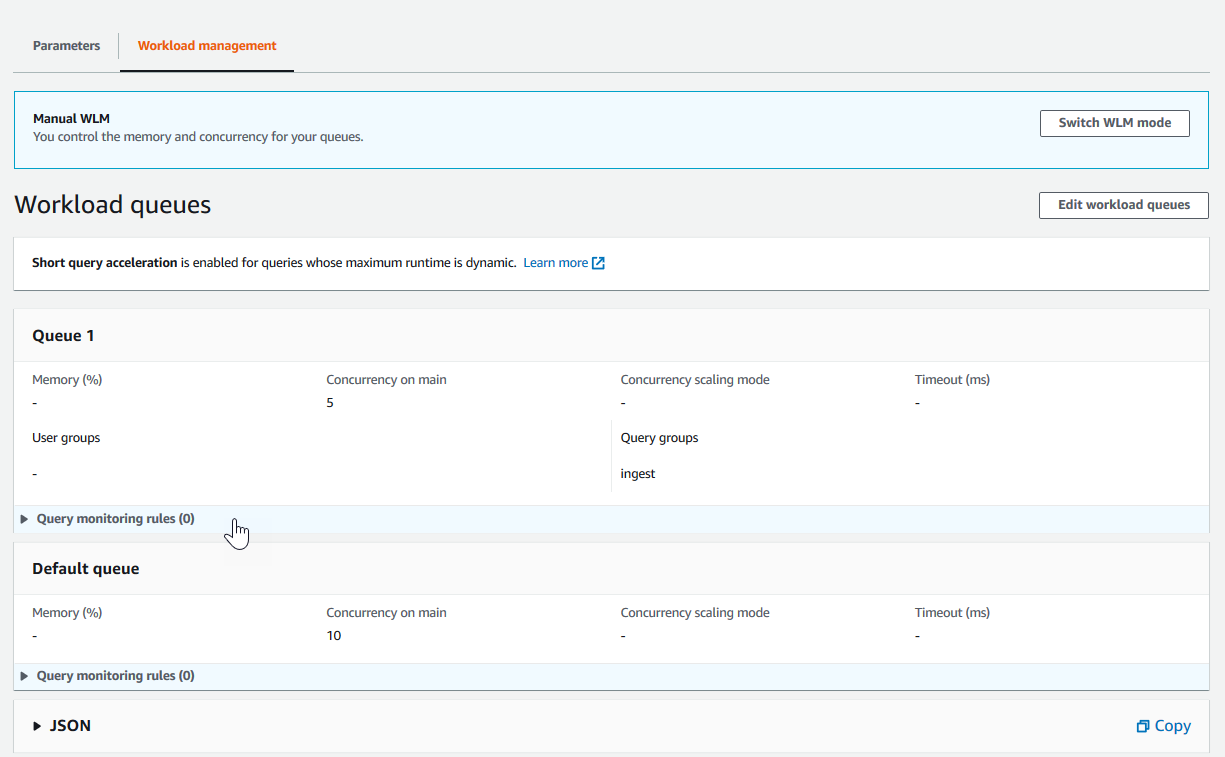
Review and familiarize yourself with the new Redshift console. Specifically check out the **Dashboard**, **Editor** and **Config** icons from the left pane.

Under Dashboard click on the single cluster



Click on the cluster and then **Properties** tab.

Under **Database configurations** click on the parameter group and review the WLM configuration.



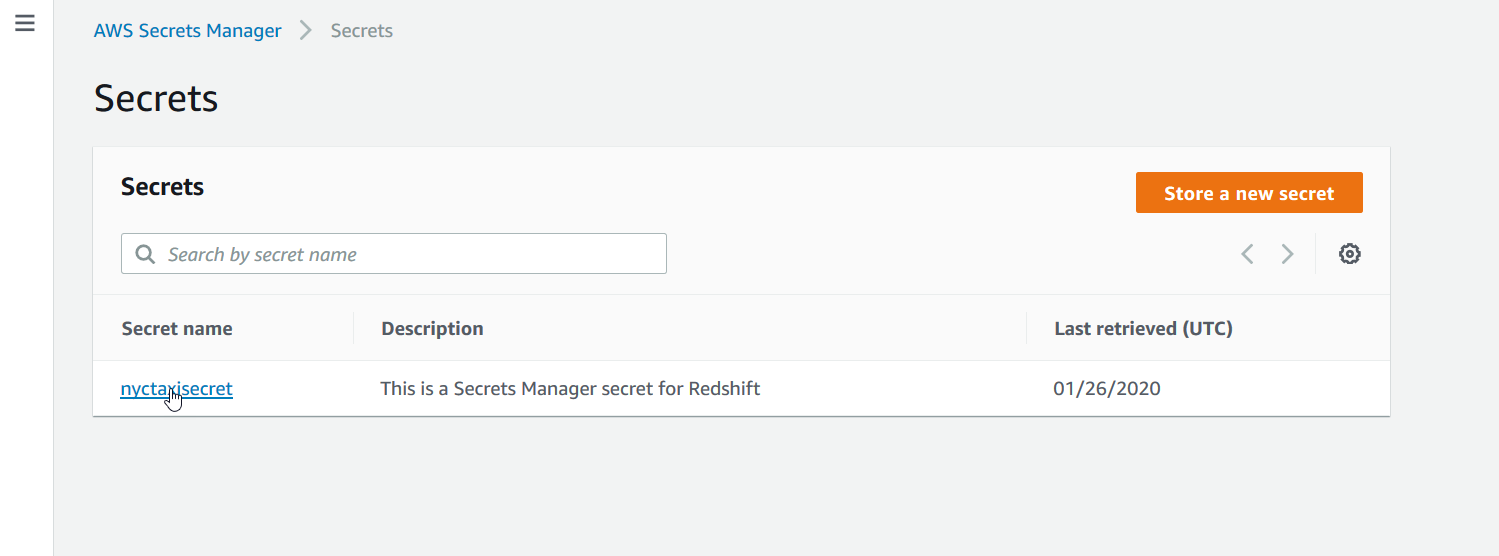
**Summary of the WLM:**

* Manual WLM with a total of 2 queues
* Queue 1 has concurrency of 5 and query group = “ingest”
* The 2nd queue is the Default queue with a concurrency of 10.

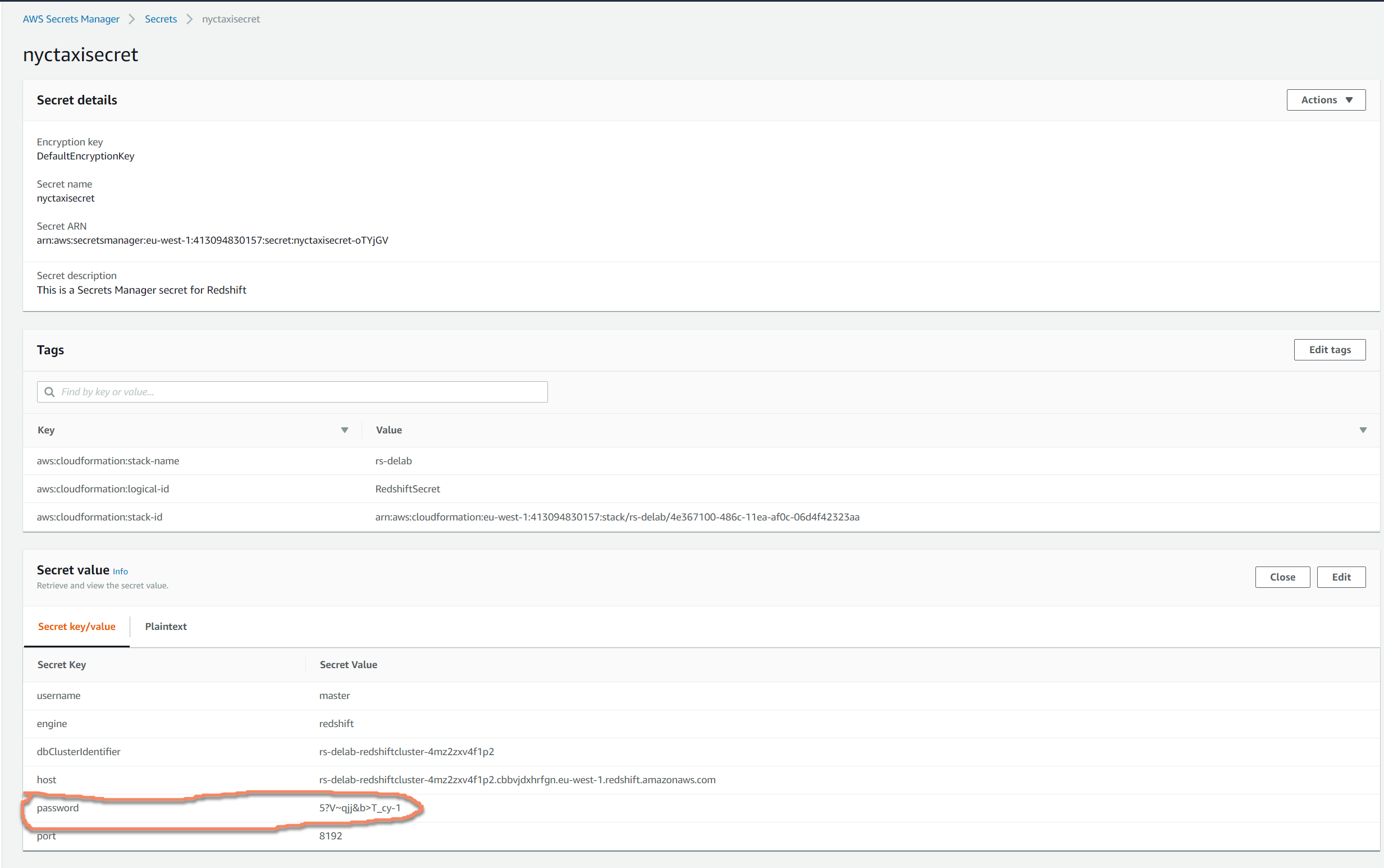
## Retrieve Redshift password

The login info for the Redshift cluster is to be stored in AWS service **Secrets Manager**.

Navigation: **Services > Secrets Manager**.



Retrieve the secret value after clicking **nyctaxisecret** .

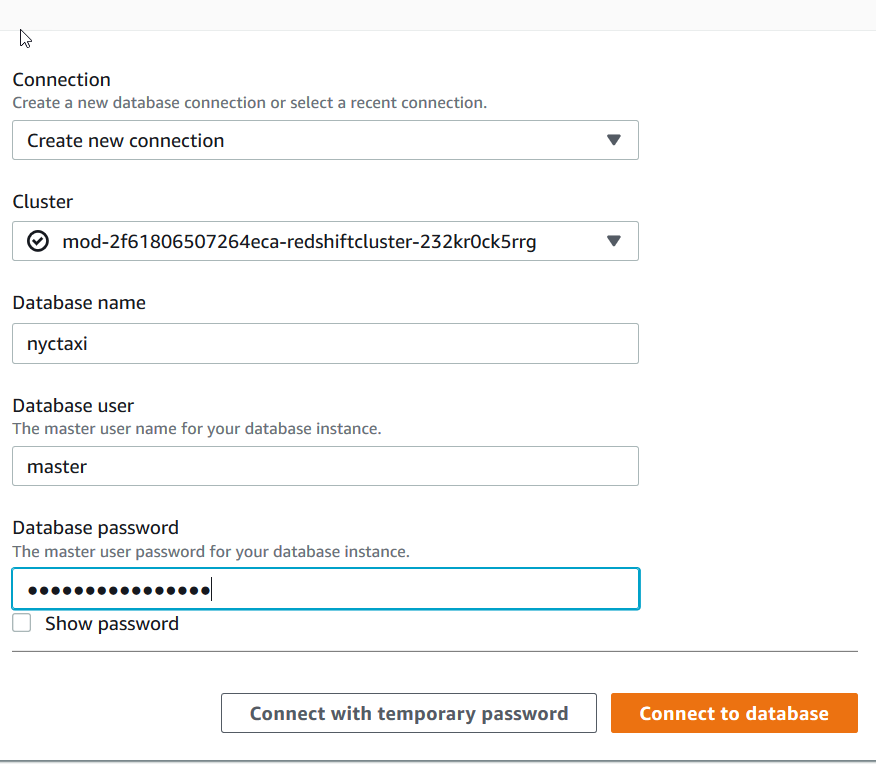


## Connect to the Redshift cluster

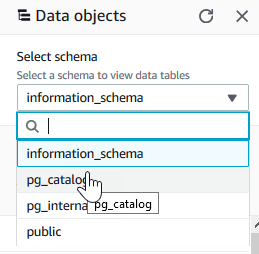
Navigate to **Services > Redshift**

Click on **Editor** from the left pane. In the **Connect to database** prompt enter the details for your cluster and connect to database.

* Database name: nyctaxi
* Database user: master
* Database password: (You should have retrieved from the Secrets Manager described under section Retrieve Redshift password)



Under **Select schema** click on drop down. There will be only 4 schema at this point.



# **Create ELT pipeline:**

We will create Glue job as the building block for ELT pipeline. The pipeline will be created using State machine to orchestrate between jobs in an ELT flow.

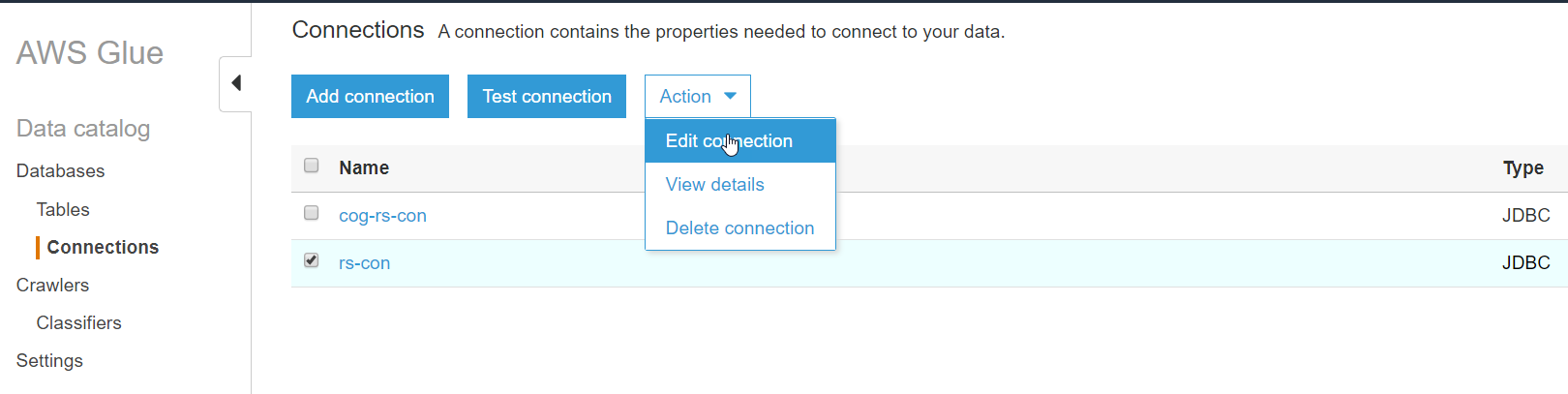
Note: NO code writing is required. The necessary code is included in the lab package. However you can use this code to build your real world pipeline.

## Glue Job

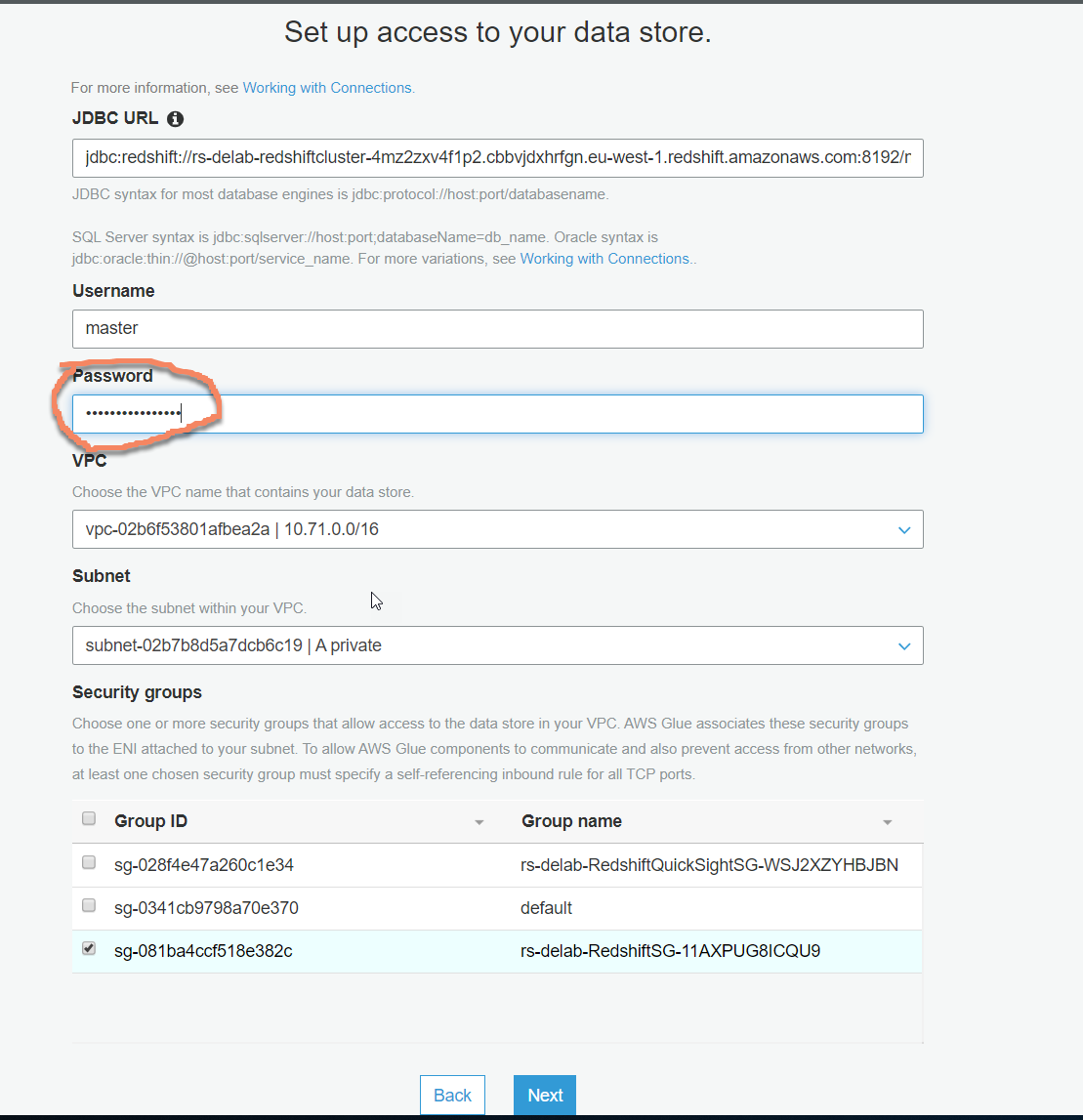
Navigate to AWS Glue service by **Services > AWS Glue**

Click on **Databases > Connections**

A connection name “rs-con” should exist. Select the connection and **Action > Edit connection**



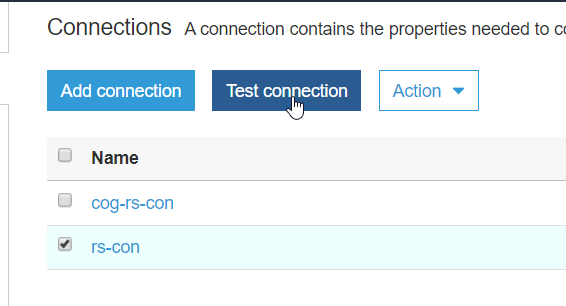
Click Next until you reach the **Set up access to your data store**. window.



Here you need to enter the connection password. Find the Redshift password as described under section “Retrieve Redshift password”.

Click Next and then Finish.

Come back to the Glue connection screen if you are not already and select the “rs-con” connection and hit “Test connection”.

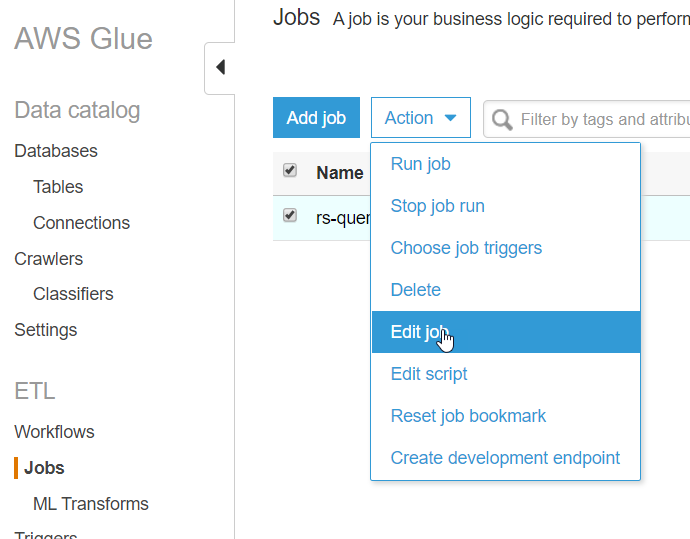


From the IAM Role dropdown select the only IAM role and hit Test connection.

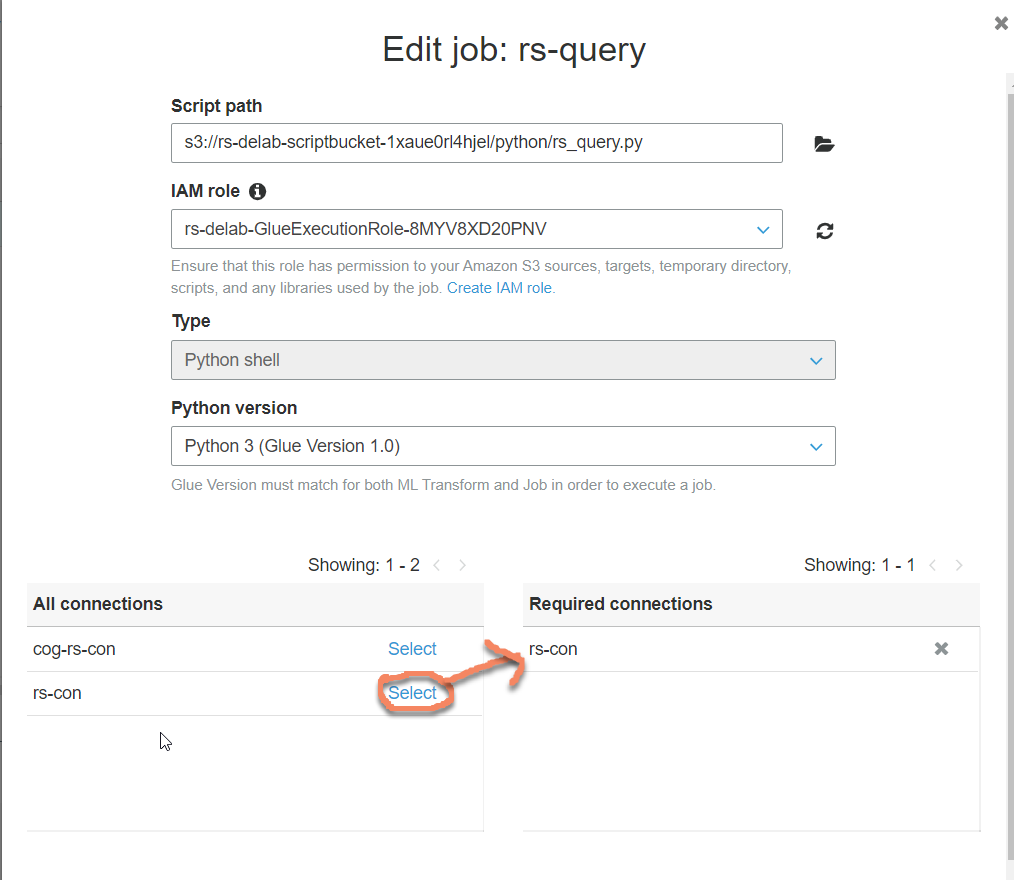
Note: It will take a while to test the connection. You will get a confirmation saying “**rs-con** connected successfully to your instance.”.

Once connection is established you will update the Glue Job to include the connection information.

From the left pane **ETL > Jobs** Select the job“rs-query”and then **Action > Edit job.**



In this screen select the “rs-con” connection from All connections so that it appears under Required connections. Hit save.



# **Redshift schema setup**

We will setup the schema and create the database tables from Redshift Editor.

Navigate **Services > Redshift**. From the left pane click **Editor**. If you have already connected to the editor in the section **Navigate to your Amazon Redshift cluster** then you will be automatically connected to the database. Otherwise follow the steps in the section **Navigate to your Amazon Redshift cluster**.

In the Query editor click on the “+” sign to open a new tab. Paste below scripts.

CREATE SCHEMA taxischema;

CREATE EXTERNAL SCHEMA spectrum\_schema FROM DATA CATALOG database 'default' region '{0}' iam\_role '{1}';

CREATE TABLE taxischema.nyc\_greentaxi(

vendorid varchar(10),

lpep\_pickup\_datetime timestamp,

lpep\_dropoff\_datetime timestamp,

store\_and\_fwd\_flag char(1),

ratecodeid int,

pulocationid int,

dolocationid int,

passenger\_count int,

trip\_distance decimal(8,2),

fare\_amount decimal(8,2),

extra decimal(8,2),

mta\_tax decimal(8,2),

tip\_amount decimal(8,2),

tolls\_amount decimal(8,2),

ehail\_fee varchar(100),

improvement\_surcharge decimal(8,2),

total\_amount decimal(8,2),

payment\_type varchar(10),

trip\_type varchar(10),

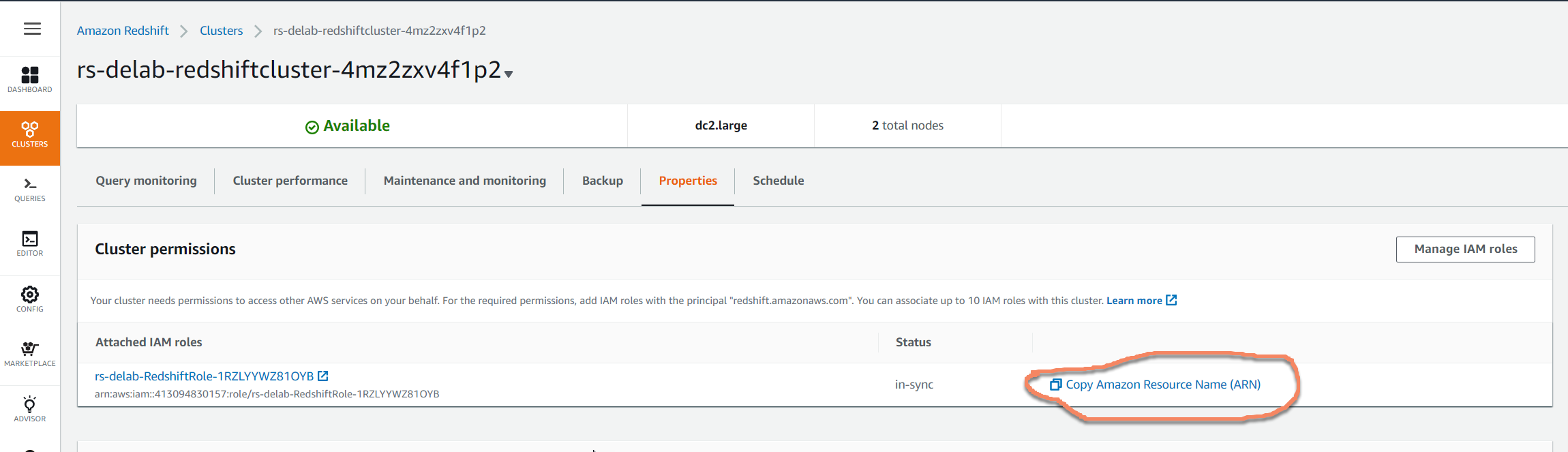
congestion\_surcharge decimal(8,2))

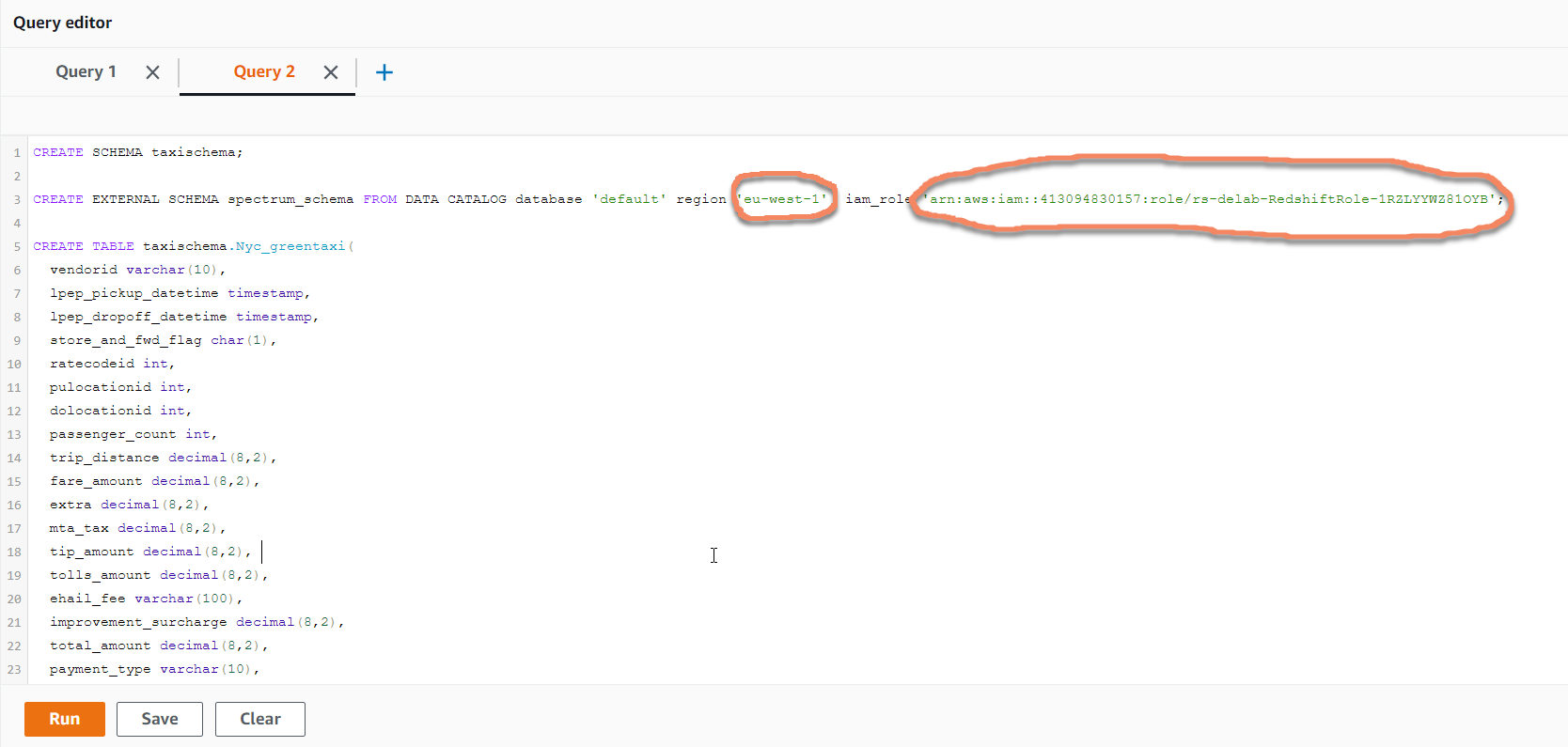
;

Replace the {0} value by the region you are in e.g us-east-1.

Replace the {1} value by the IAM arn that is attached to your Redshift cluster.

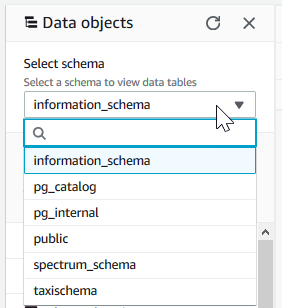
You can find your IAM arn by navigating to Redshift CLUSTERS > Properties tab





Execute each statement one after another by selecting the statement and hit Run.

Once the statements are executed check the schema that got created in Redshift.



# ELT Execution

Now the State machine needs to be executed to orchestrate 1) ingestion into Redshift i.e **EL**(T) and 2) Unload the data into data lake i.e (EL)**T**

Navigate to **Step Functions** page and click on theState machineand hit Start execution. Leave the JSON input as is. Monitor the progress of the job under Visual workflow.

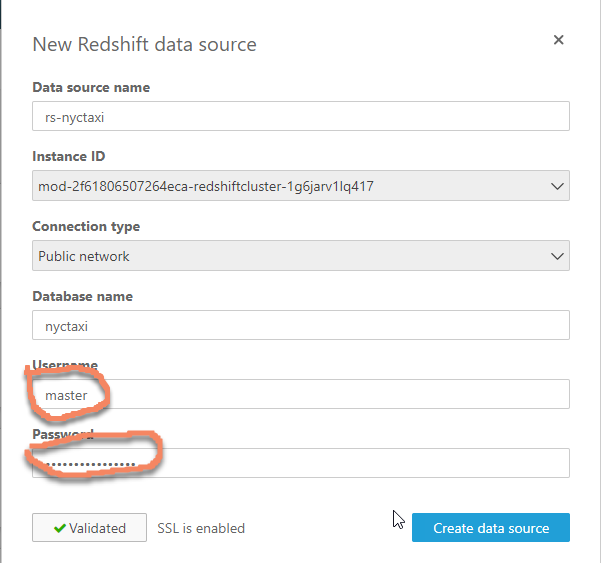
# 

# Reporting

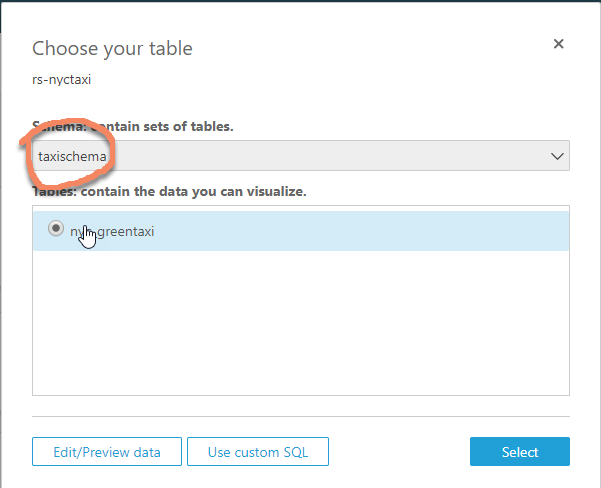
The above step brings us to for the first time to look at the data we just loaded. Open Quicksight from the Services options and then select the region i.e eu-west-1.

**Note**: If this is your first time in QuickSight you will need to sign up for Standard License. All you need is to provide your email id and everything else is taken care of.

Click on **New Dataset > Redshift (auto discovered)**. In the next screen select the instance Id from the drop down and fill in the Username and password for the Redshift cluster. Click on **Validated** button to make sure connection is successful. Hit **Create data source.**

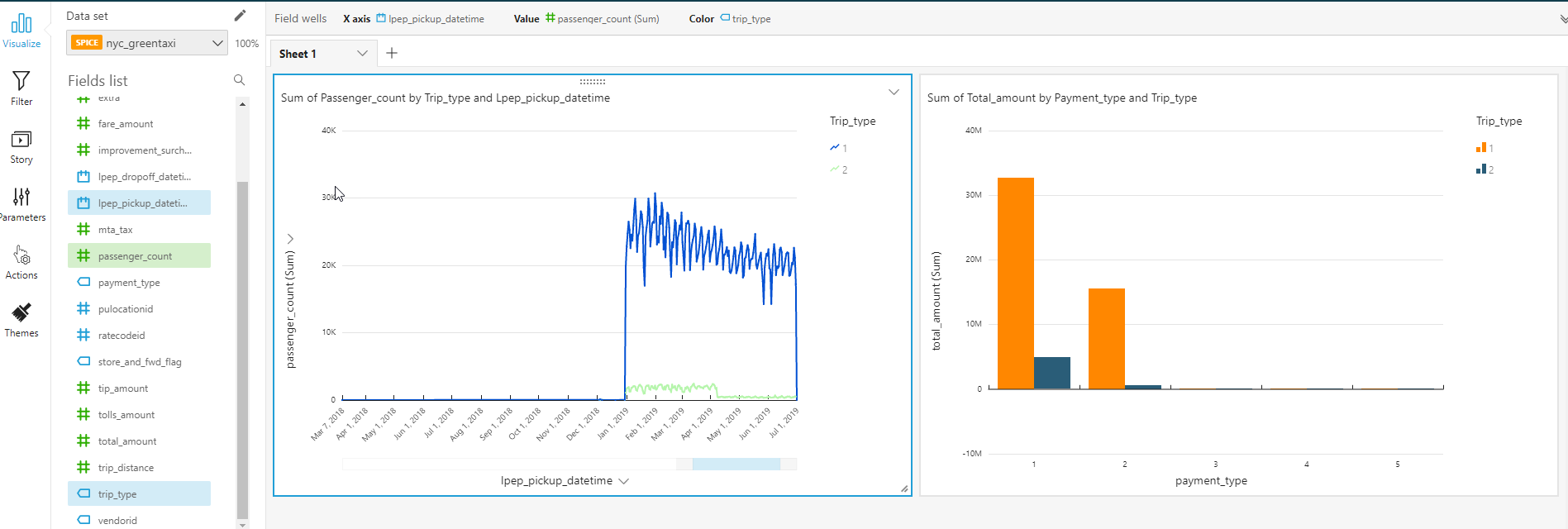


In the next screen select “taxischema” schema and the “nyc\_greentaxi” table.



In the following screens select **Preview data > SPICE** as Data source and then **Save & visualize**.

Let’s do some visualization of the data in Redshift.



Put your analytical hat on and keep analyzing.

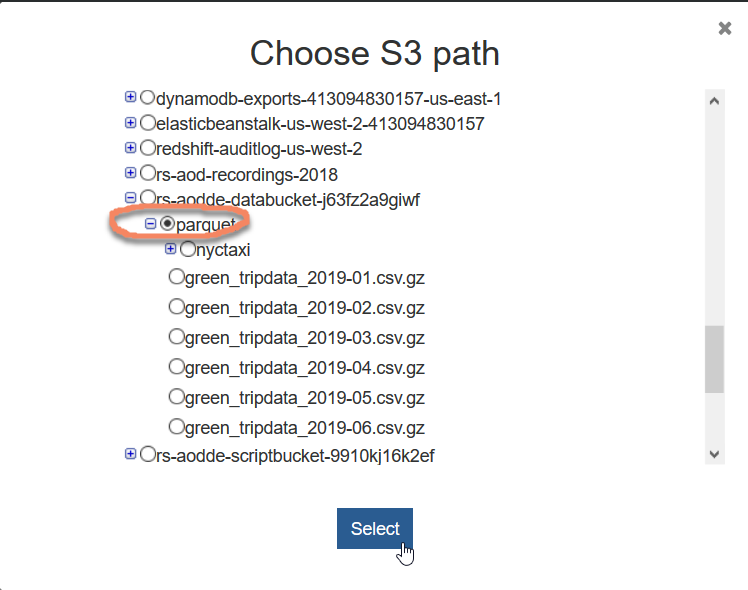
# **Build on the data lake**

The above viz provides very little insight on different dimensions of the data for example trip\_type and payment\_type are encoded as integer value.

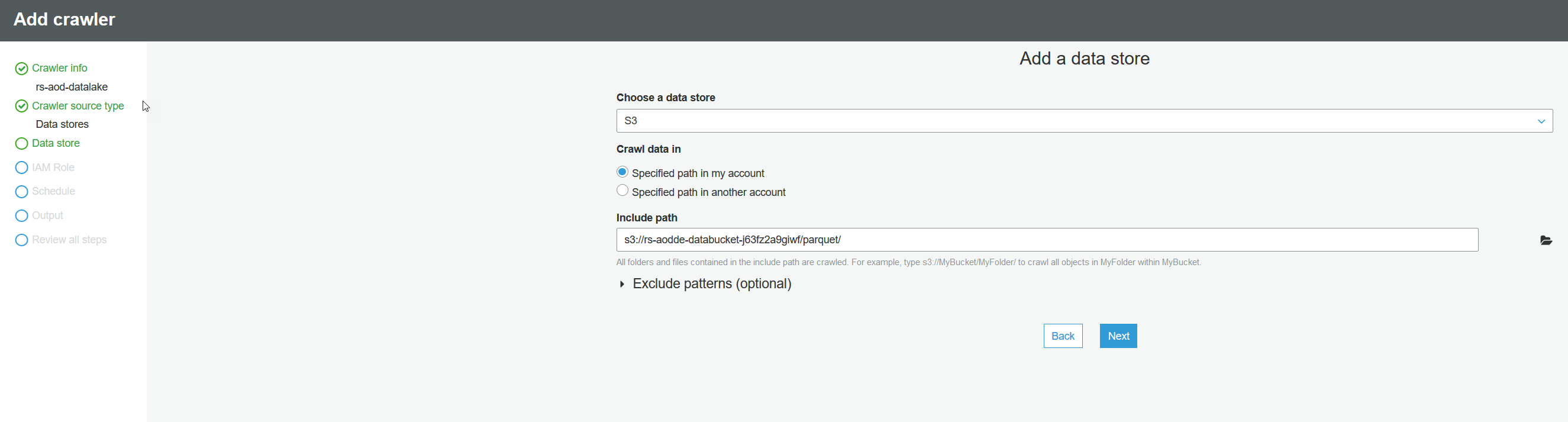
Next we want to create transformation and visualize the transformed data using QuickSight.

You will first create a crawler to register the Parquet data unloaded from Redshift, into Glue catalog.

**AWS Glue> Crawlers > Add crawler**. Give a crawler name and hit next couple of times until you reach **Choose a data store**

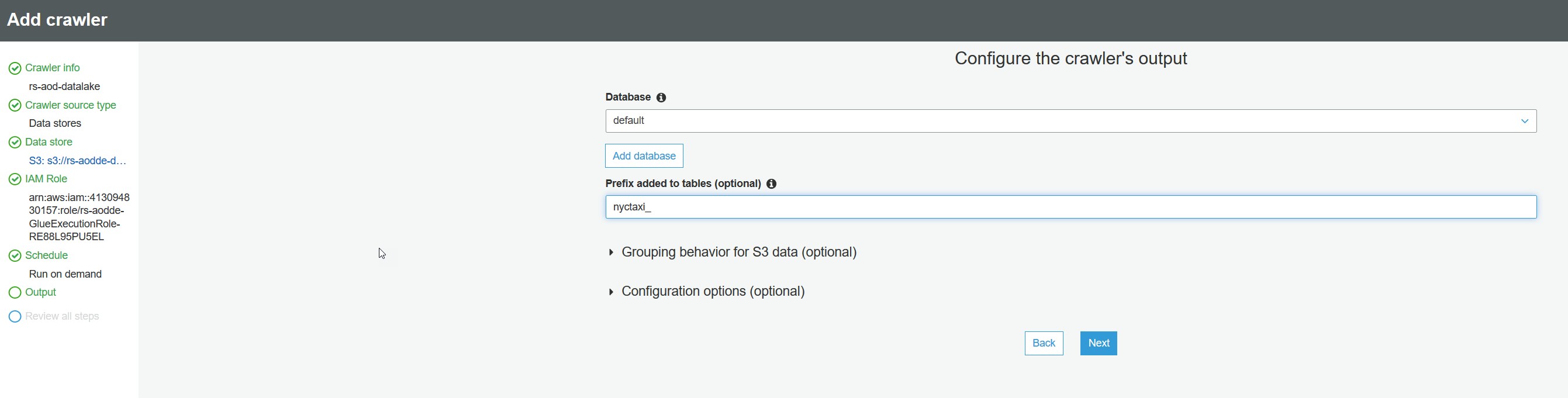


Include path add the location of the parquet folder. It will look like below-



Hit next couple of times. Choose an IAM role specific to your environment.

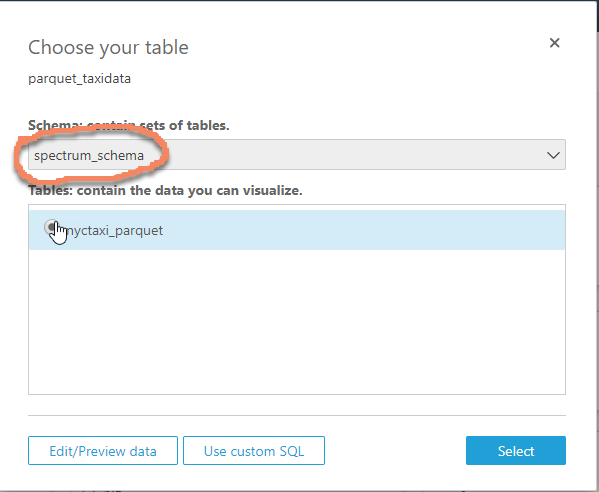
After next few screens select “**default**” Database and enter “**nyctaxi\_**” as Prefix added to tables.



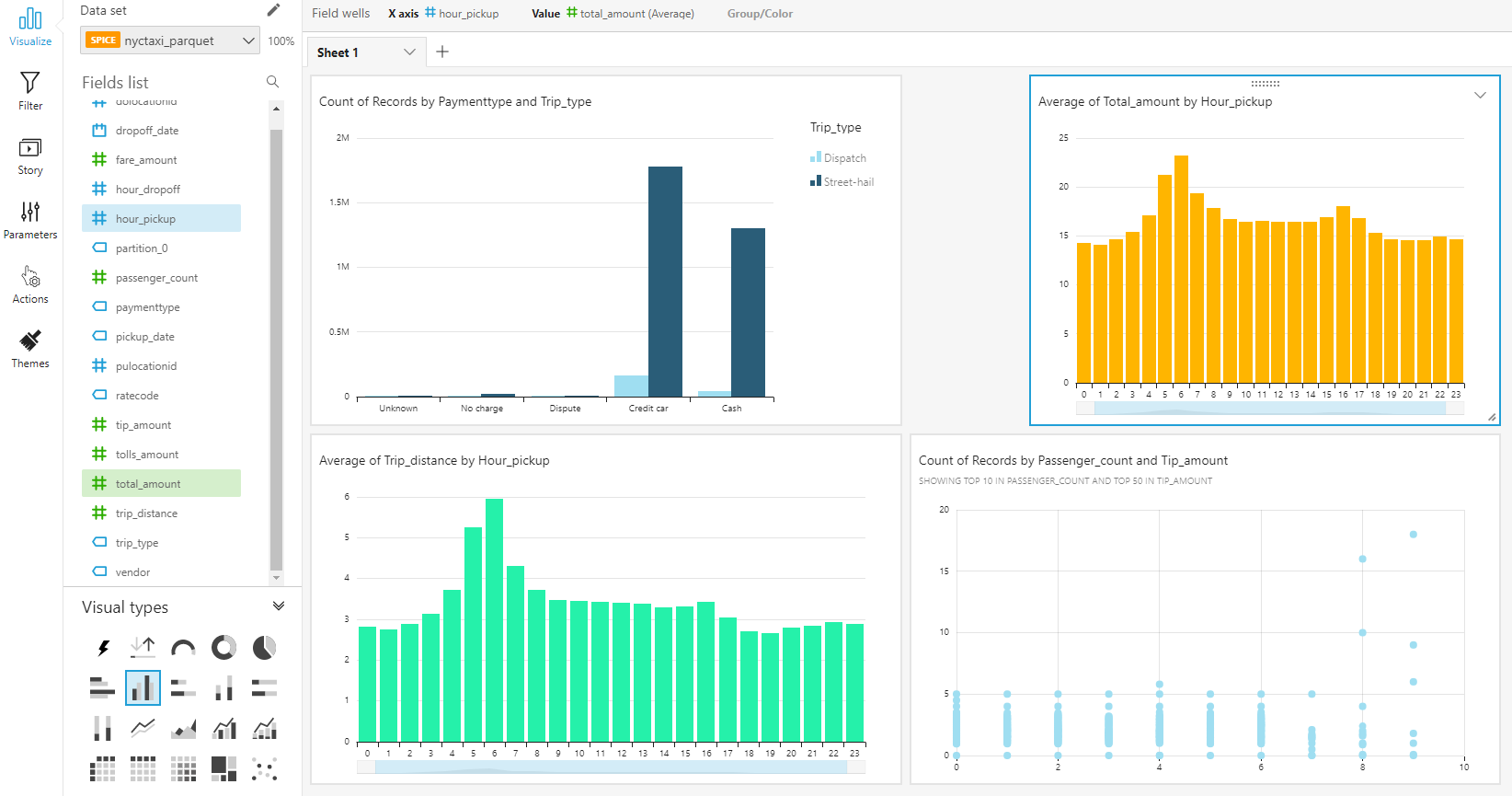
Hit Finish and click on **Run it now**.

Once finished the new table **nyctaxi\_parquet** will show up under **Databases > Tables**.

We go back to QuickSight creating a new Dataset. Create a **New Redshift data source**. Select the spectrum\_schema and nyctaxi\_parquet table and then **Edit/ Preview data**



Select **SPICE** and then **Save and visualize**.



This brings us to the end of the lab.

Please fill in the survey at the end of this lab.