Amazon Web Services
Data Engineering Immersion Day

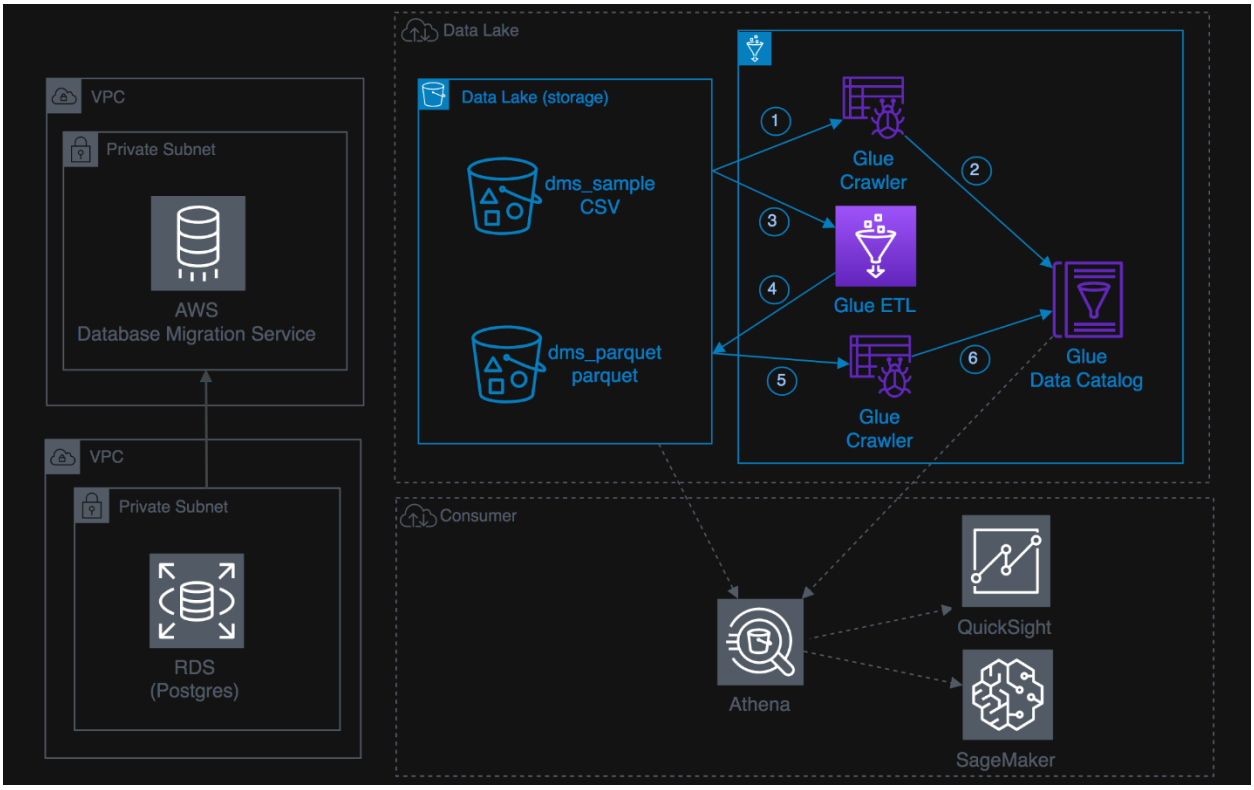
Lab 2. ETL with AWS Glue

August 2020

Table of Contents

Introduction

This lab will give you an understanding of the AWS Glue – a fully managed data catalog and ETL service



Prerequisites

1. Completed Lab 1. Hydrating the Data Lake with DMS

Tasks Completed in this Lab:

In this lab you will be completing the following tasks. You can choose to complete only **Part-(A)** to move to next lab where tables can be queried using Amazon Athena and Visualize with Amazon Quciksight

1. PART-(A): Data Validation and ETL
2. PART- (B): Glue Job Bookmark Functionality(Optional)
3. PART- (C): Glue Workflows(Optional)

The Lab is also available - https://aws-dataengineering-day.workshop.aws/

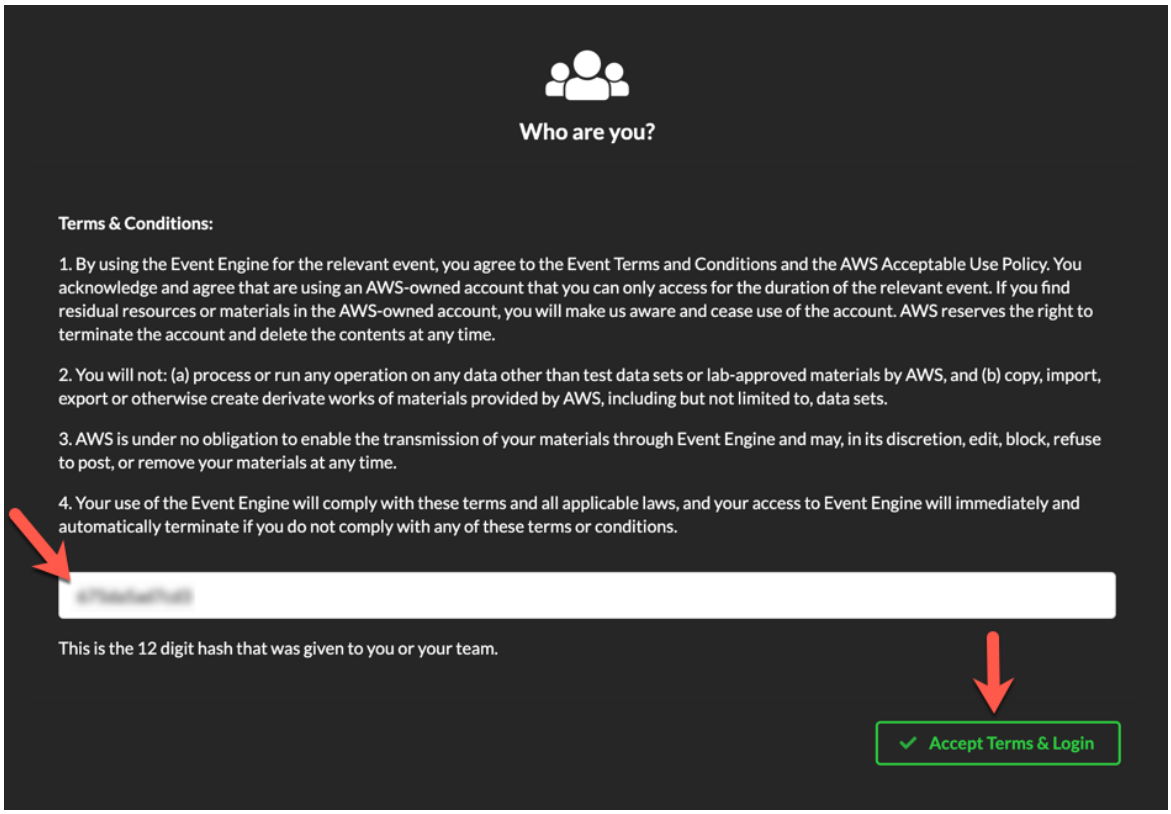
Get Started Using the Lab Environment

Please skip this section if you are running the lab on your own AWS account.

Today, you are attending a formal event and you will have been sent your access details beforehand. If in the future you might want to perform these labs in your own AWS environment by yourself, you can follow instructions on GitHub - https://github.com/aws-samples/data-engineering-for-aws-immersion-day.

A 12-character access code (or 'hash') is the access code that grants you permission to use a dedicated AWS account for the purposes of this workshop.

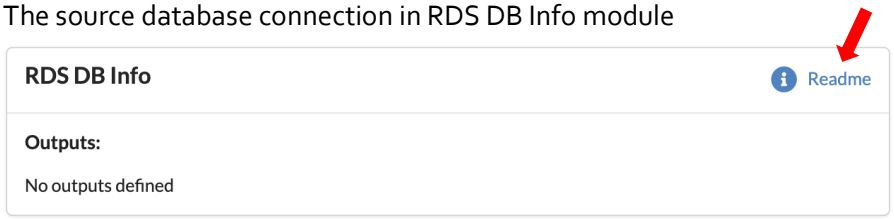
1. Go to https://dashboard.eventengine.run/, enter the access code and click Proceed:



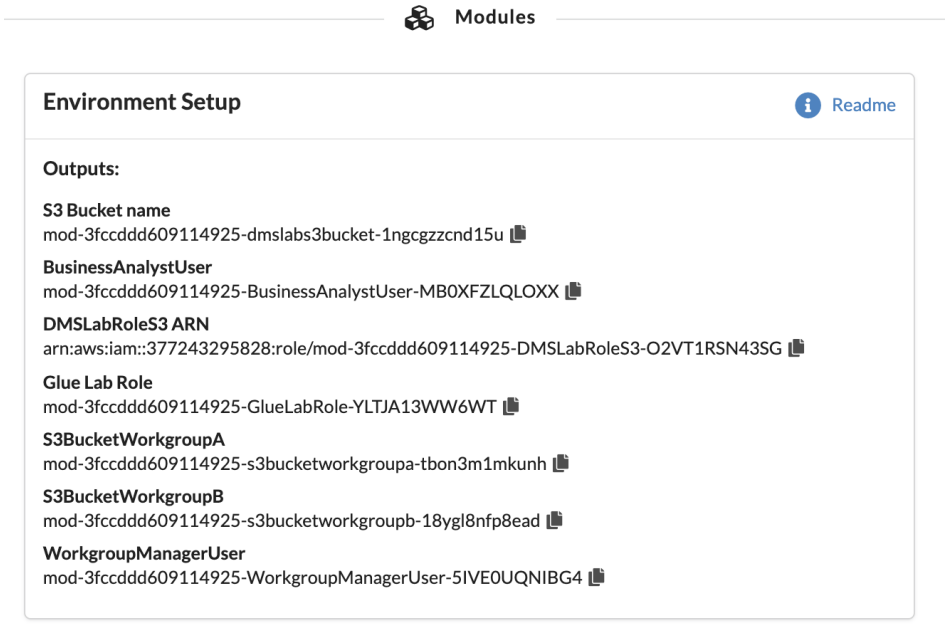
2. On the Team Dashboard web page you will see a set of parameters that you will need during the labs. Best to save them to a text file locally, alternatively you can always go to this page to review them. Replace the parameters with the corresponding values from here where indicated in subsequent labs:

Because you're at a formal event, some AWS resources have been pre-deployed for your convenience, for example:

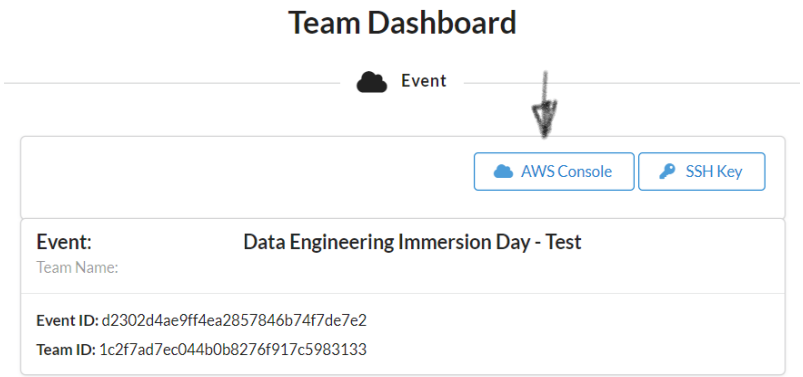
- The source database connection in RDS DB Info module



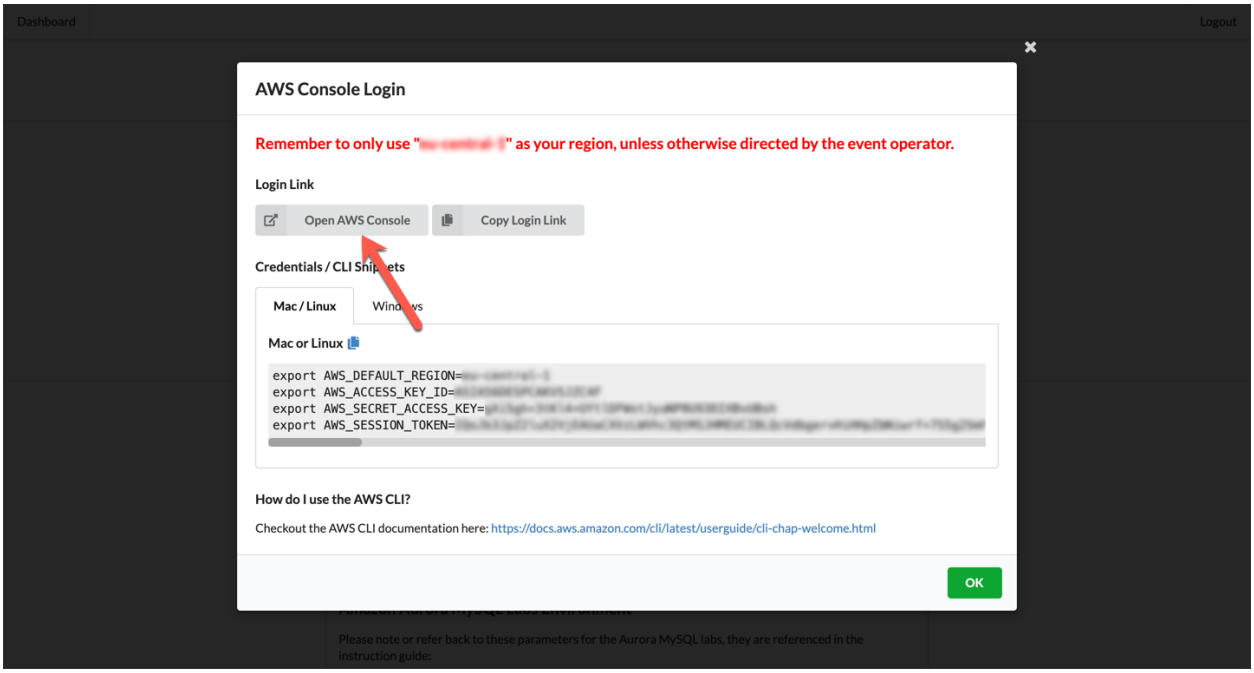
- S3 Bucket, IAM role for the Glue lab etc



3. On the Team Dashboard, please click AWS Console to log into the AWS Management Console:



4. Click Open Console. For the purposes of this workshop, you will not need to use command line and API access credentials:

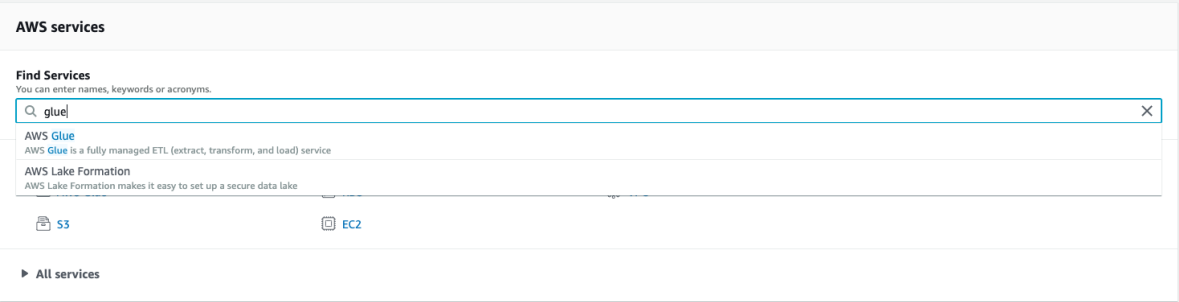


Once you have completed these steps, you can continue with the rest of this lab.

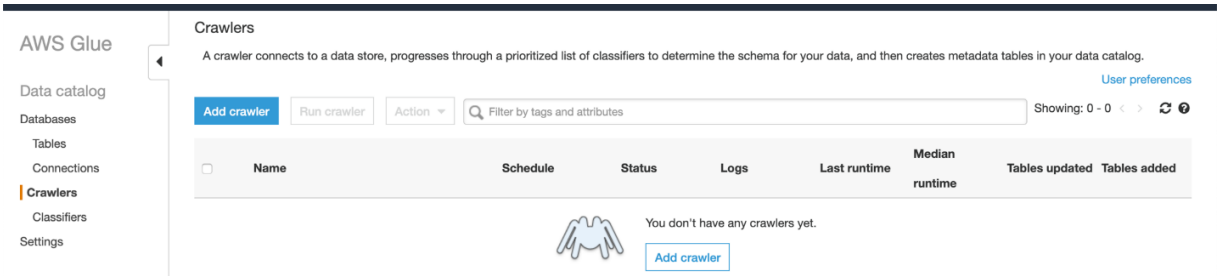
PART A: Data Validation and ETL

Create Glue Crawler for initial full load data

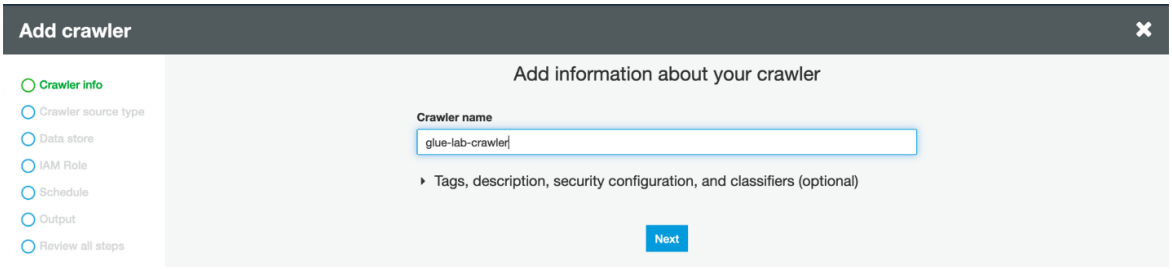
1. Navigate to the AWS Glue service



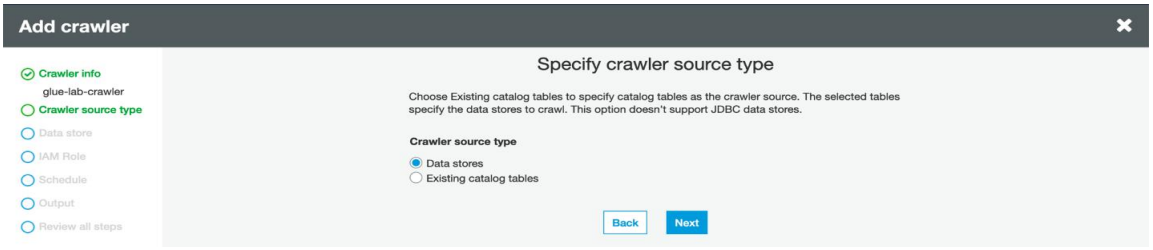
2. On the AWS Glue menu, select **Crawlers**.



3. Click **Add crawler**.
4. Enter **glue-lab-crawler** as the crawler name for initial data load.
5. Optionally, enter the description. This should also be descriptive and easily recognized and Click **Next**.



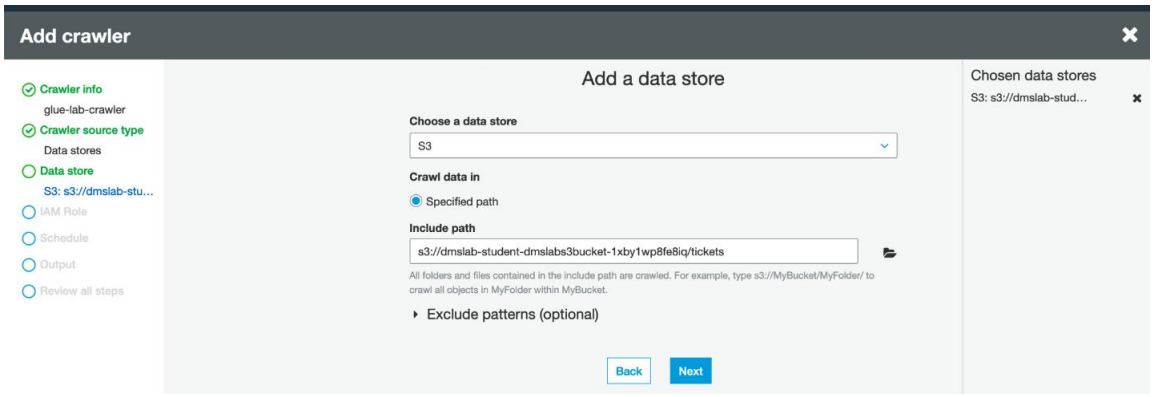
6. Choose **Crawler Source Type** as **Data Stores** and **Click Next**



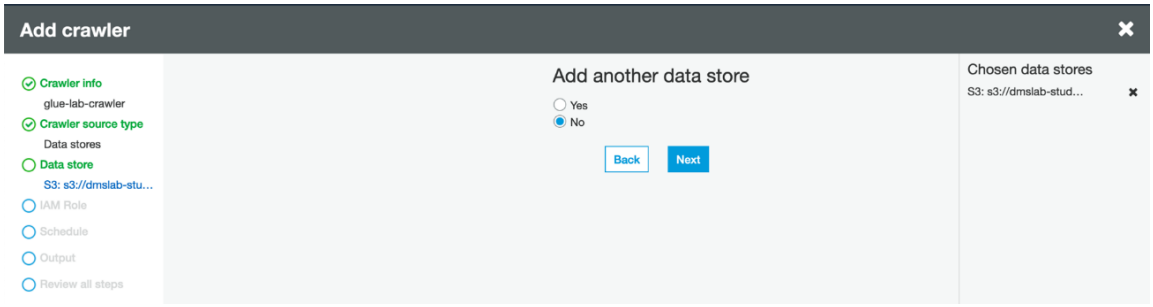
7. On the **Add a data store** page, make the following selections:

a. For Choose a data store, click the drop-down box and select **S3**.
b. For Crawl data in, select **Specified path in my account**.
c. For Include path, browse to the target folder for your DMS initial export from Lab 1, e.g., **s3://dmslab-student-dmslabs3bucket-woti4bf73cw3/tickets**

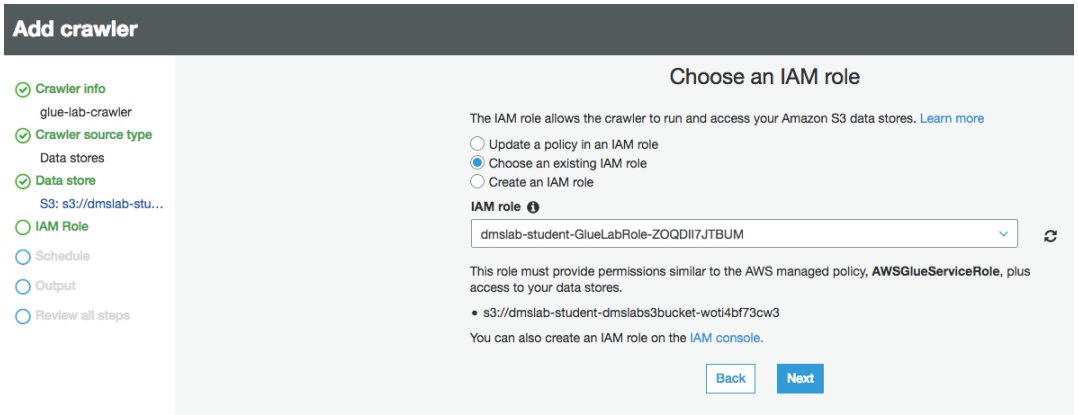
8. Click **Next**.



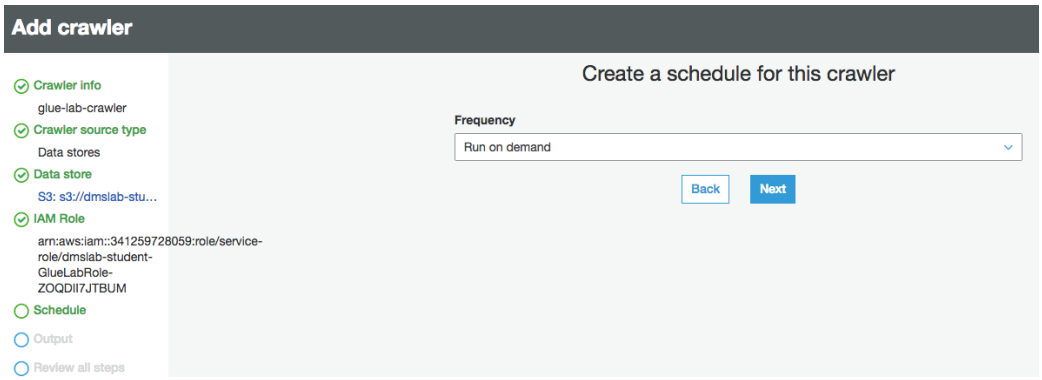
9. On the **Add another data store page,** select **No**. and Click **Next**.



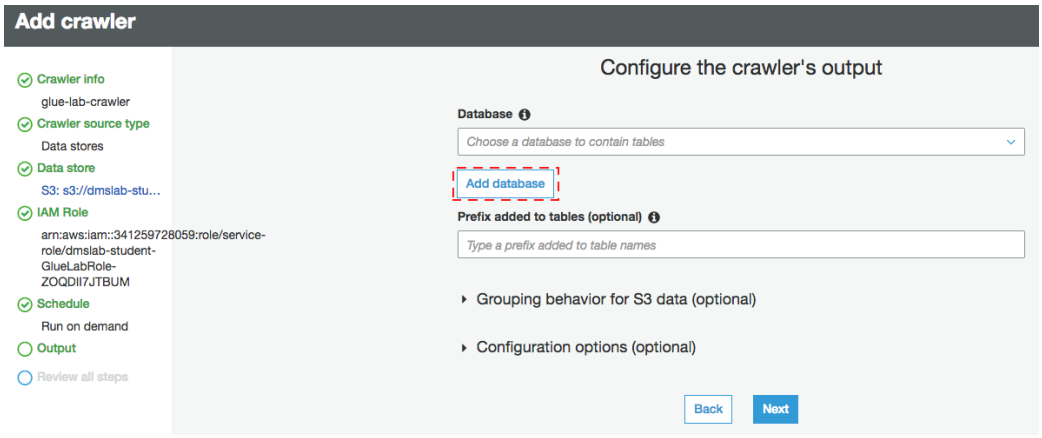
10. On the **Choose an IAM role** page, make the following selections:
    a. Select **Choose an existing IAM role**.
    b. **For IAM role,** select **<stackname>-GlueLabRole-<RandomString>** pre-created for you. For example "dmslab-student-GlueLabRole-ZOQDII7JTBUM"
11. Click **Next**.



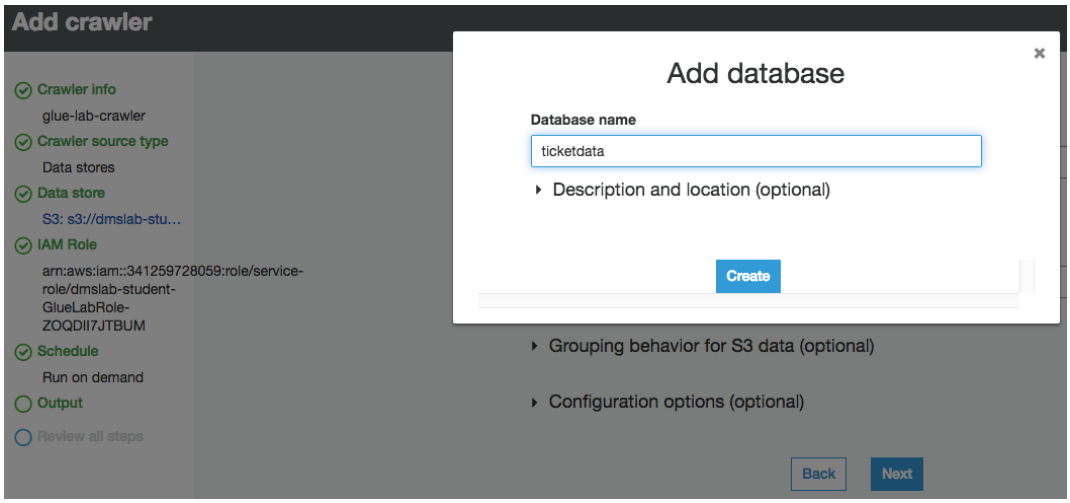
12. On the Create a schedule for this crawler page, for Frequency, select **Run on demand** and Click **Next**.



13. On the Configure the crawler's output page, click **Add database** to create a new database for our Glue Catalogue.

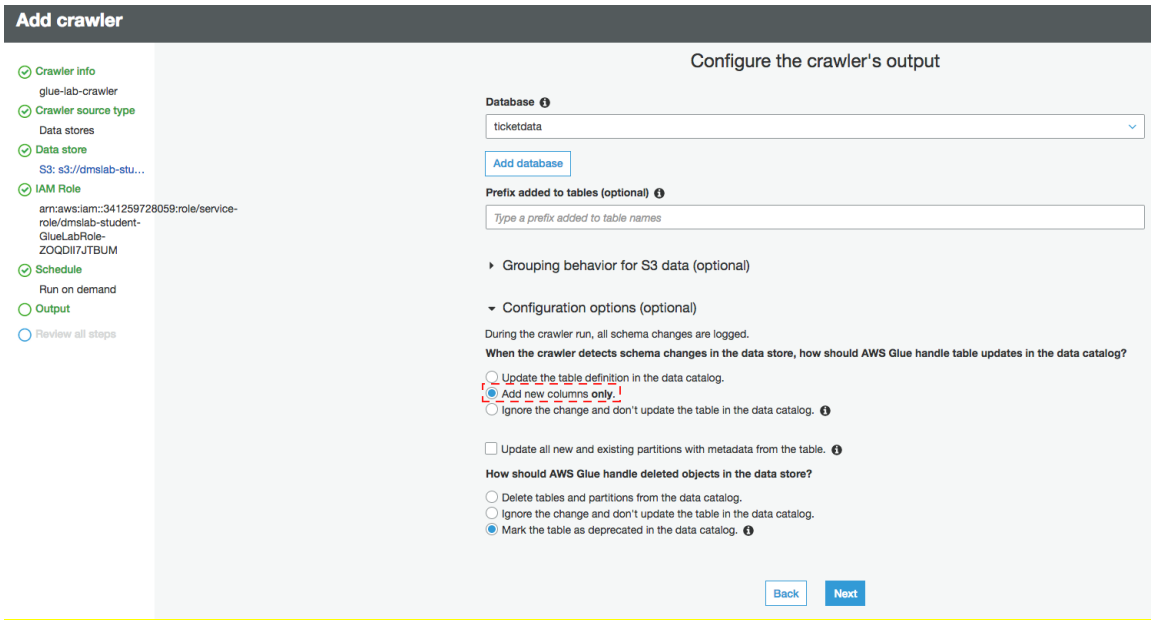


14. Enter **ticketdata** as your database name and click **create**

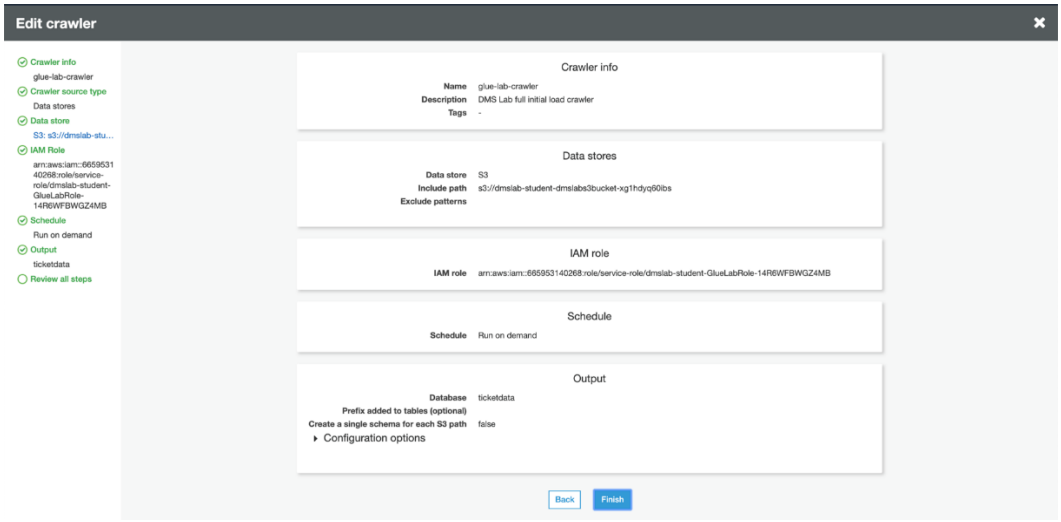


15. For **Prefix added to tables (optional),** leave the field empty.

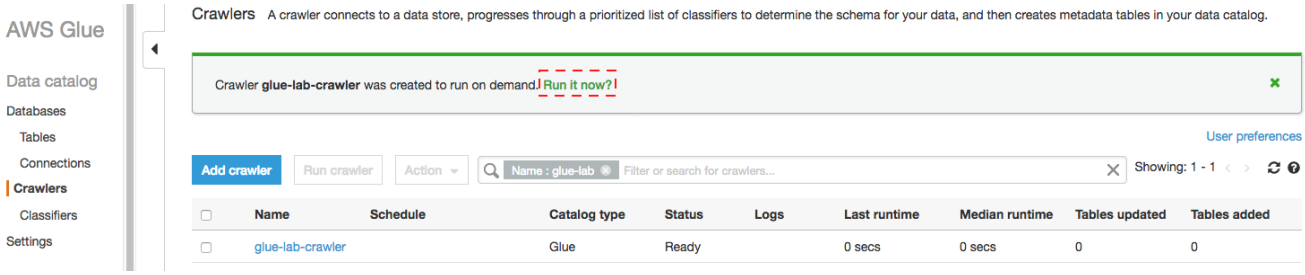
16. For **Configuration options (optional),** select **Add new columns only** and keep the remaining default configuration options and Click **Next**.



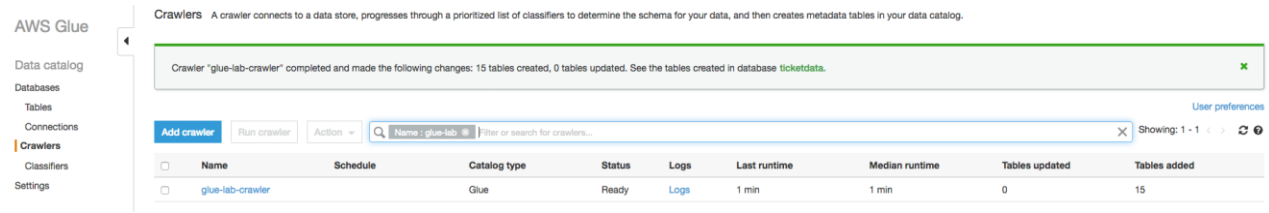
17. Review the summary page noting the Include path and Database output and Click **Finish**. The crawler is now ready to run.



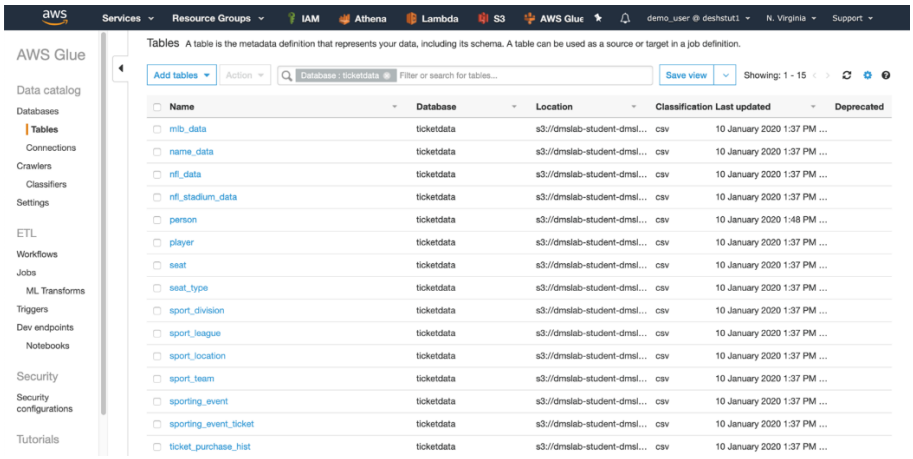
18. Click **Run it now**.



Crawler will change status from starting to stopping, wait until crawler comes back to ready state (the process will take a few minutes), you can see that it has created 15 tables.

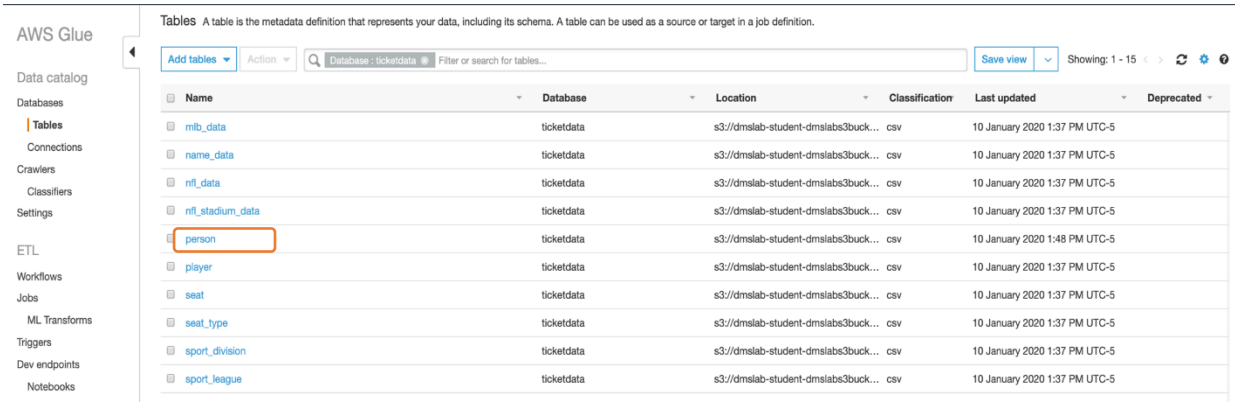


19. In the AWS Glue navigation pane, click **Databases** > **Tables**. You can also click the **ticketdata** database to browse the tables.



Data Validation Exercise

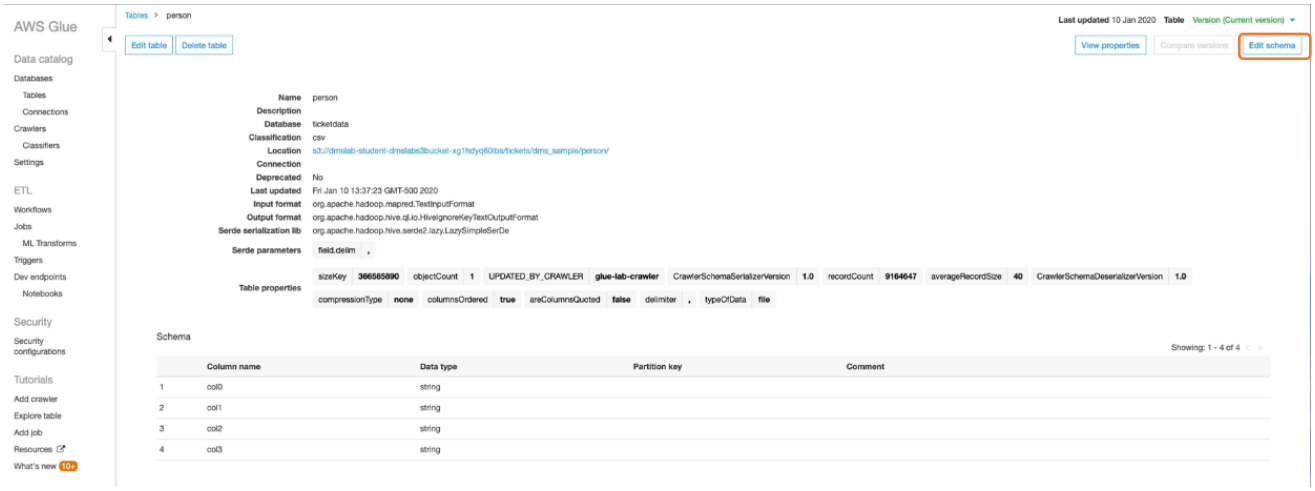
1. Within the Tables section of your **ticketdata** database, click the person table.



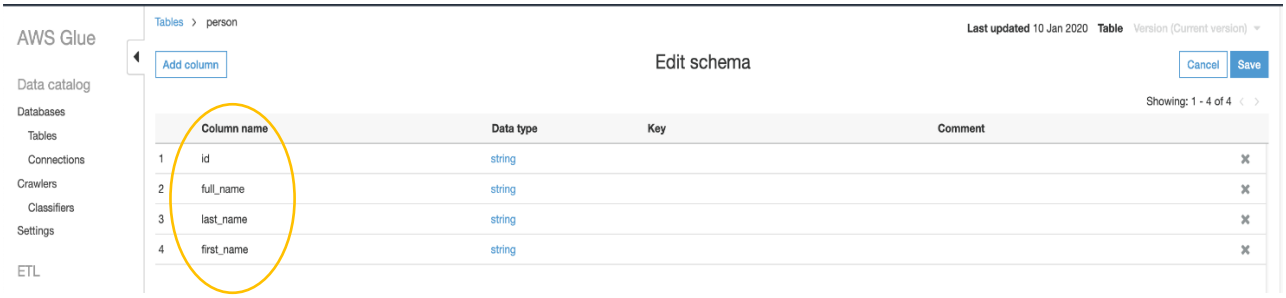
You may have noticed that some tables (such as person) have column headers such as col0,col1,col2,col3. In absence of headers or when the crawler cannot determine the header type, default column headers are specified.

This exercise uses the person table in an example of how to resolve this issue.

2. Click **Edit Schema** on the top right side.



3. In the Edit Schema section, double-click **colo** (column name) to open edit mode. Type "id" as the column name.
4. Repeat the preceding step to change the remaining column names to match those shown in the following figure.



5. Click **Save**.

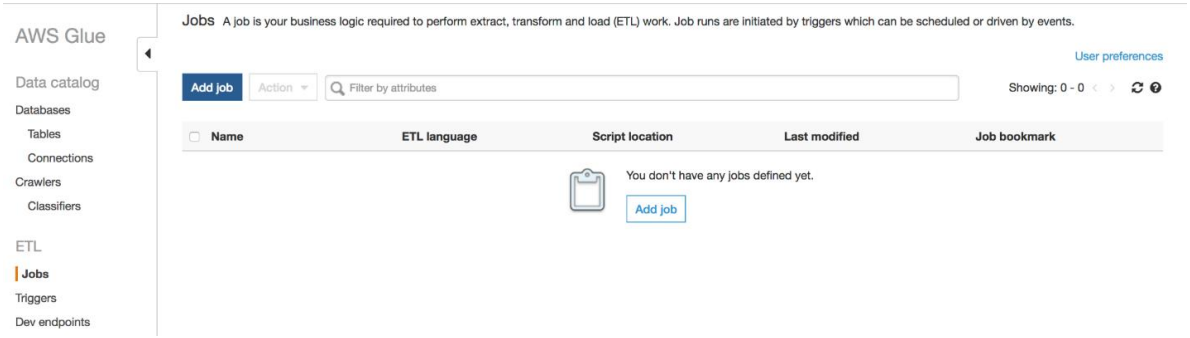
Data ETL Exercise

Pre-requisite: To store processed data in parquet format, we need a new folder location for each table, eg. the full path for sport_team table look like this –

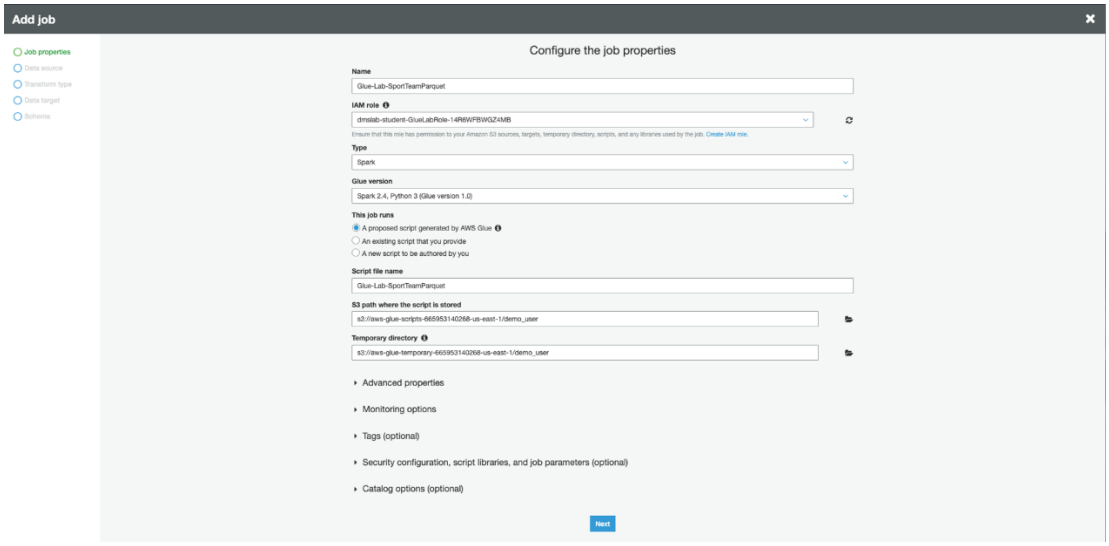
"s3://<s3_bucket_name>/tickets/**dms_parquet/sport_team**"

Glue will create the new folder automatically, based on your input of the full file path, such as the example above. Please refer to the user guide in terms of how to manually create a folder in S3 bucket.

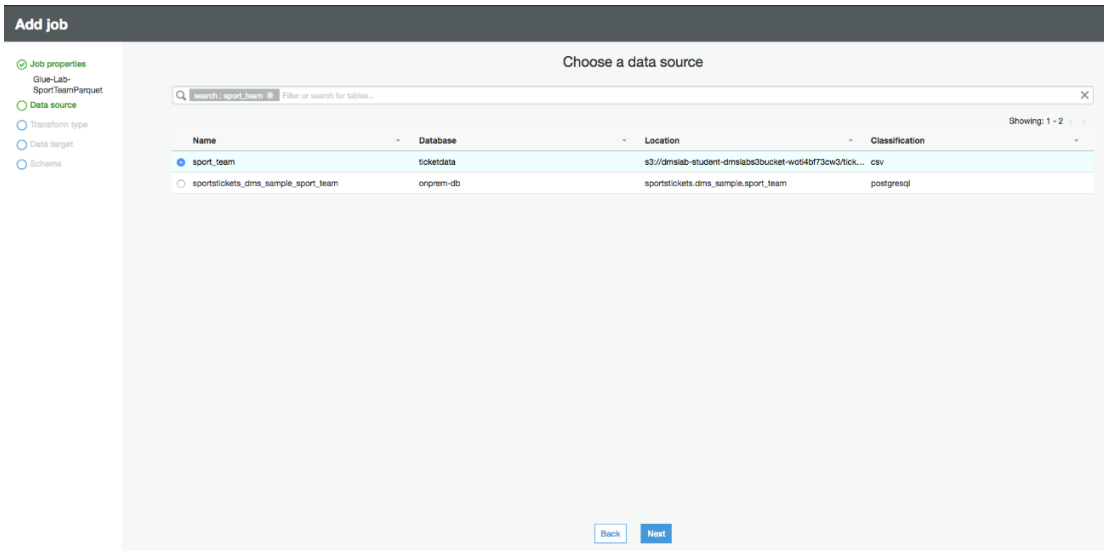
1. Go to the Glue console, in the left navigation pane, under **ETL** click **Jobs**, and then click **Add job**.



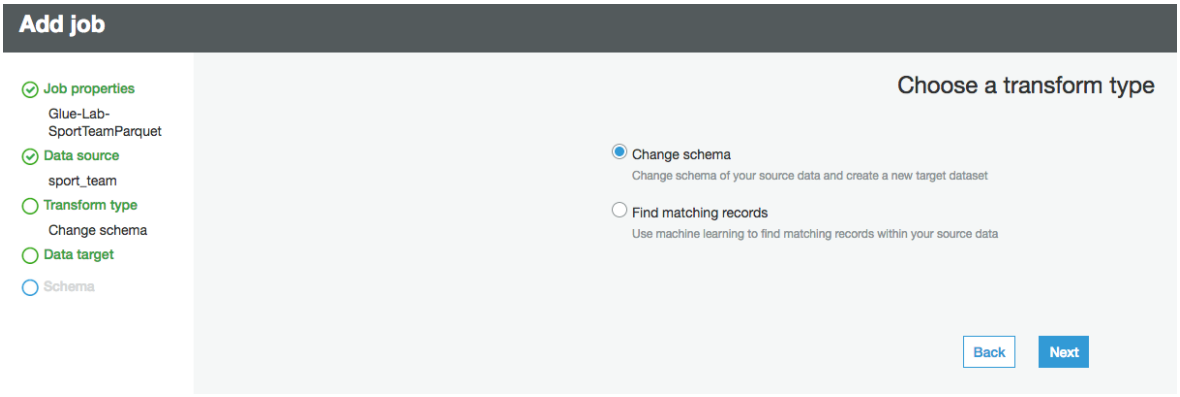
2. On the Job properties page, make the following selections:
   a. For **Name**, type "Glue-Lab-SportTeamParquet"
   b. For **IAM role**, choose existing role e.g. "xxxx-GlueLabRole-xxx"
   c. For **Type**, Select "Spark"
   d. For **Glue Version**, select "Spark 2.4, Python 3(Glue version 1.0)" or whichever is the latest version
   e. For **This job runs**, select "A proposed script generated by AWS Glue".
   f. For **Script file name**, type **Glue-Lab-SportTeamParquet**.
   g. For **S3 path where the script is stored**, provide a unique Amazon S3 path to store the scripts. (You can keep the default for this lab.)
   h. For **Temporary directory**, provide a unique Amazon S3 directory for a temporary directory. (You can keep the default for this lab.)



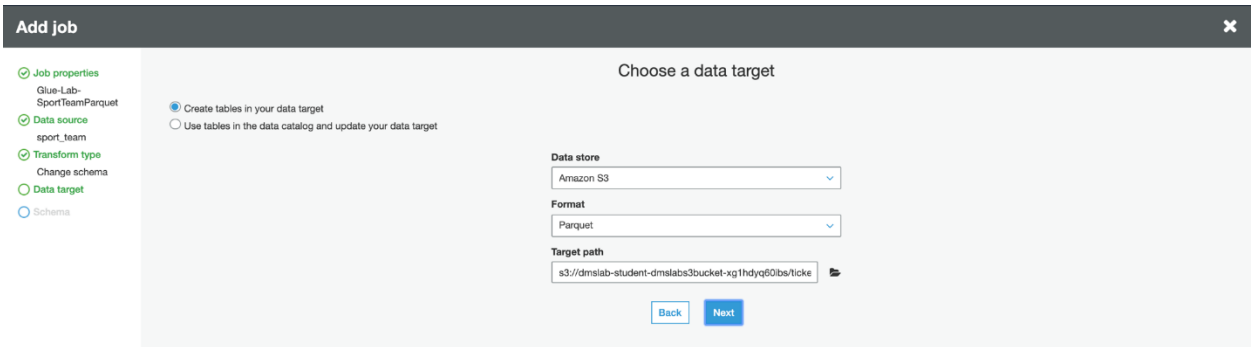
3. Click **Next**
4. On the Choose your data sources page, select **sport_team** and Click **Next**.



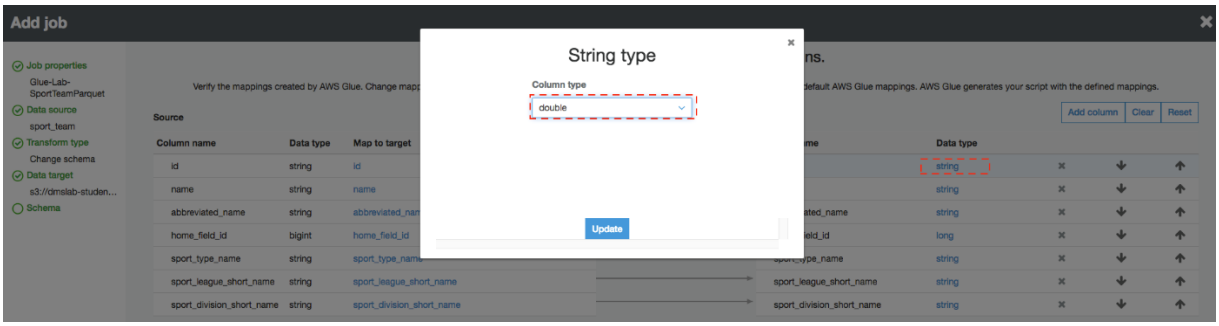
5. On the **Choose a transformation type** page, select **change schema**

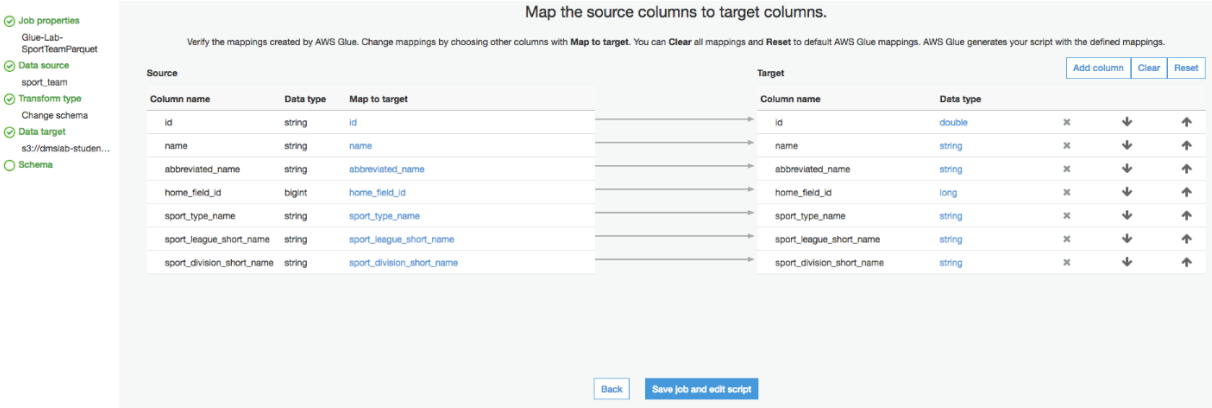


6. On the Choose your data targets page, select **Create tables in your data target**.
7. For Data store, select **Amazon S3**.
8. For Format, select **Parquet**.
9. For **Target path,** click the **folder icon** and choose the s3 bucket, then append **/tickets/dms_parquet/sport_team** to it, making the target path look like **s3://xxx-dmslabs3bucket-xxx/tickets/dms_parquet/sport_team** – Glue will create necessary folders
10. Click **Next**.

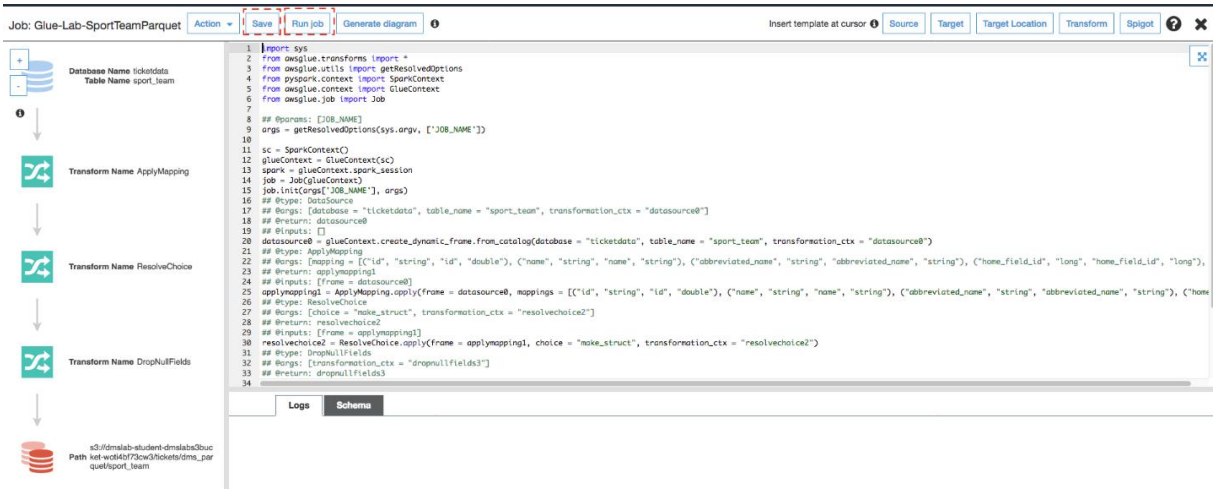


11. Click the target **Data type** to edit the schema mapping for the **id** column. In **String type** pop-up window Select **double** from **Column type** drop down and click **update**.

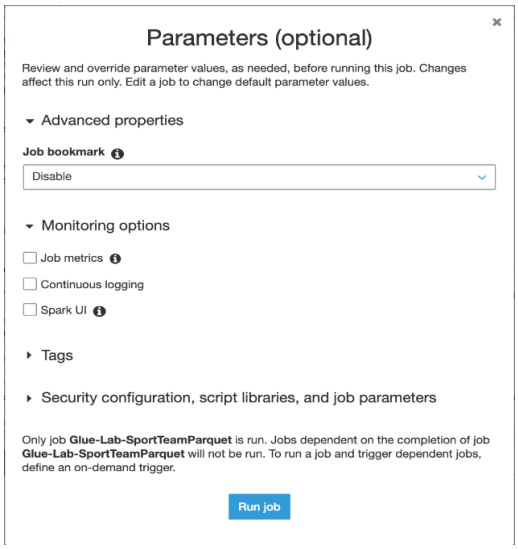




12. Click **Save job** and **edit script**.
13. View the job. (This screen provides you with the ability to customize this script as required.) Click **Save** and then **Run Job**.

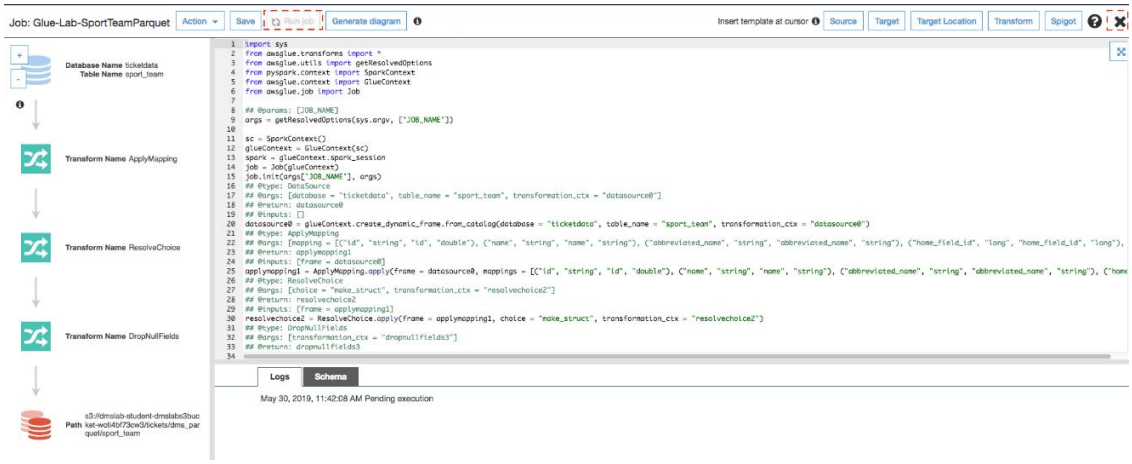


14. In **Parameters** option,
    a. you can leave **Job bookmark** as **Disable**. AWS Glue tracks data that has already been processed during a previous run of an ETL job by persisting state information from the job run.

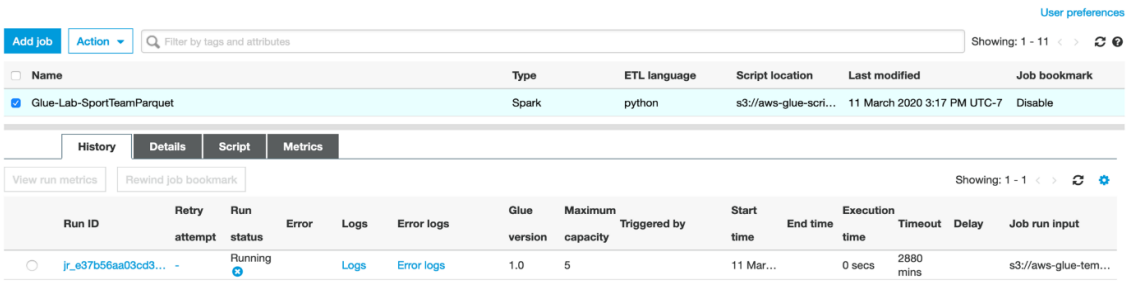


b. You can leave the **Job metrics** option **Unchecked**. You can collect metrics about AWS Glue jobs and visualize them on the AWS Glue with job metrics.

15. Click **Run Job**
16. You will see job in now running as **Run job** button became greyed out. Click the cross button located in top right corner to close the window to return to the ETL jobs.



17. Click your job to view history and verify that it ran successfully.



If you plan to continue with the Lab 3. Consuming data with Athena and Quicksight, you'll have to complete the rest of this section to create more ETL Jobs to transform additional tables to parquet, changing their schema as per instructions below. Otherwise you can proceed to the section **"Create Glue Crawler for Parquet Files"**

To enable us to join data in Athena and Quicksight lab, we also need to update the target data types in the schema. If below **Table1** is indicating Schema change as "Yes", then refer **Table 2** to find out which column needs to change with source and target data type during ETL job creation.

Table 1:

Job Name & Script Filename	Source Table	S3 Target Path	Need Schema Change?
Glue-Lab-SportLocationParquet	sport_location	tickets/dms_parquet/sport_location	No
Glue-Lab-SportingEventParquet	sporting_event	tickets/dms_parquet/sporting_event	Yes
Glue-Lab-SportingEventTicketParquet	sporting_event_ticket	tickets/dms_parquet/sporting_event_ticket	Yes
Glue-Lab-PersonParquet	person	tickets/dms_parquet/person	Yes

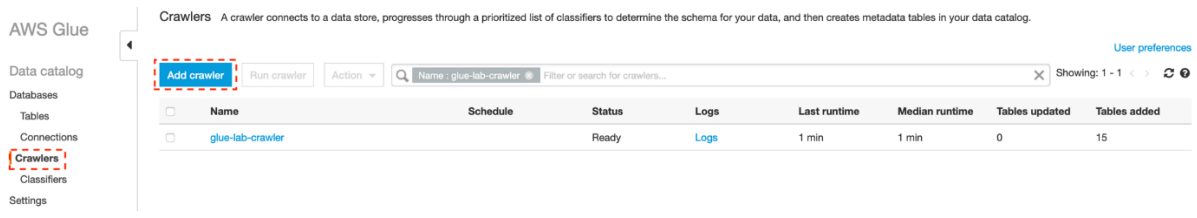
Table 2:

Job Name	Table	Column	Source Data Type	Target Data Type
Glue-Lab-SportingEventParquet	sporting_event	start_date_time	STRING	TIMESTAMP
Glue-Lab-SportingEventParquet	sporting_event	start_date	STRING	DATE
Glue-Lab-SportingEventTicketParquet	sporting_event_ticket	id	STRING	DOUBLE
Glue-Lab-SportingEventTicketParquet	sporting_event_ticket	sporting_event_id	STRING	DOUBLE
Glue-Lab-SportingEventTicketParquet	sporting_event_ticket	ticketholder_id	STRING	DOUBLE
Glue-Lab-PersonParquet	person	id	STRING	DOUBLE

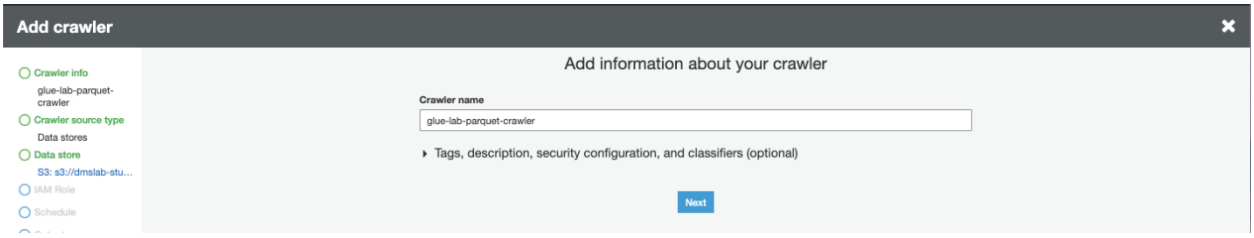
Once these jobs have completed, we can create a crawler to index these parquet files.

Create Glue Crawler for Parquet Files

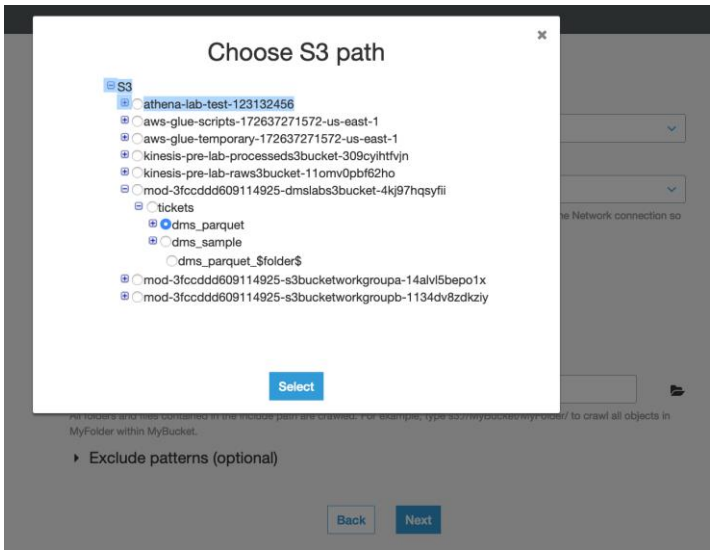
1. In the AWS Glue navigation menu, click **Crawlers**, and then click **Add crawler**.

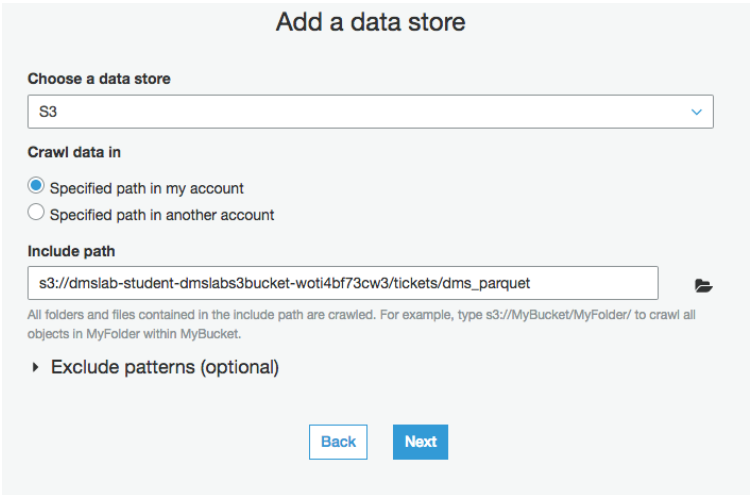


2. For **Crawler name**, type "glue-lab-parquet-crawler" and Click **Next**.

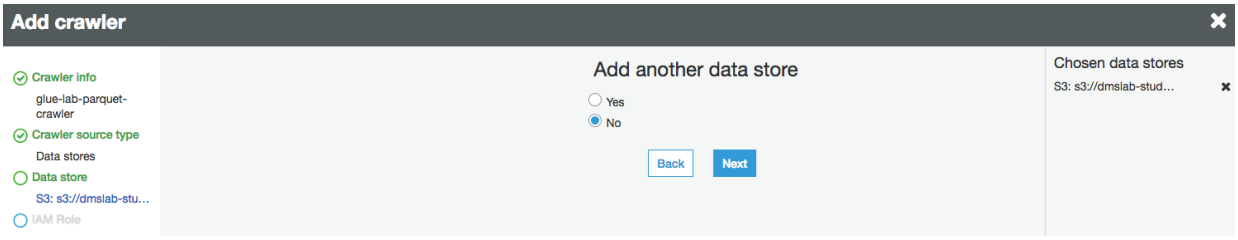


3. In next screen **Specify crawler source type,** select **Data Stores** as choice **for Crawler source type** and click **Next.**
4. In Add a data store screen
    a. For **Choose a data store,** select "S3".
    b. For **Crawl data** in, select "Specified path in account".
    c. For Include path, specify the S3 Path (Parent Parquet folder) that contains the nested parquet files e.g., s3://xxx-dmslabs3bucket-xxx/tickets/dms_parquet
    d. Click **Next**.

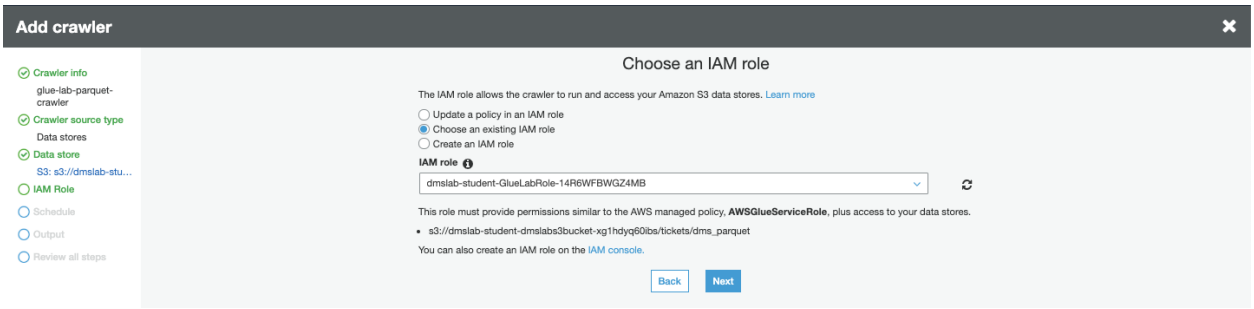




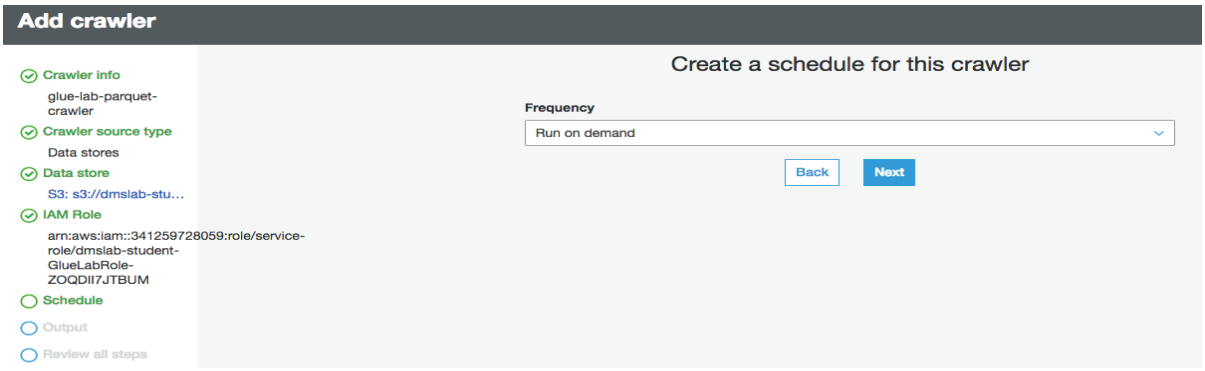
5. For Add another data store, select **No** and Click **Next**.



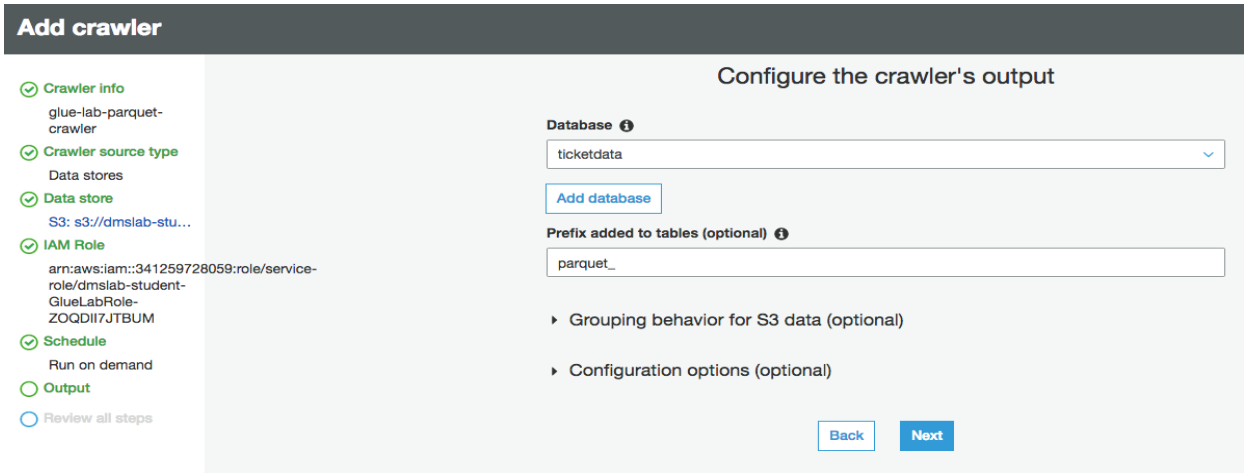
6. On the Choose an IAM role page, select **Choose an existing IAM role**.
   For IAM role, select the existing role "xxx-GlueLabRole-xxx" and Click **Next**.



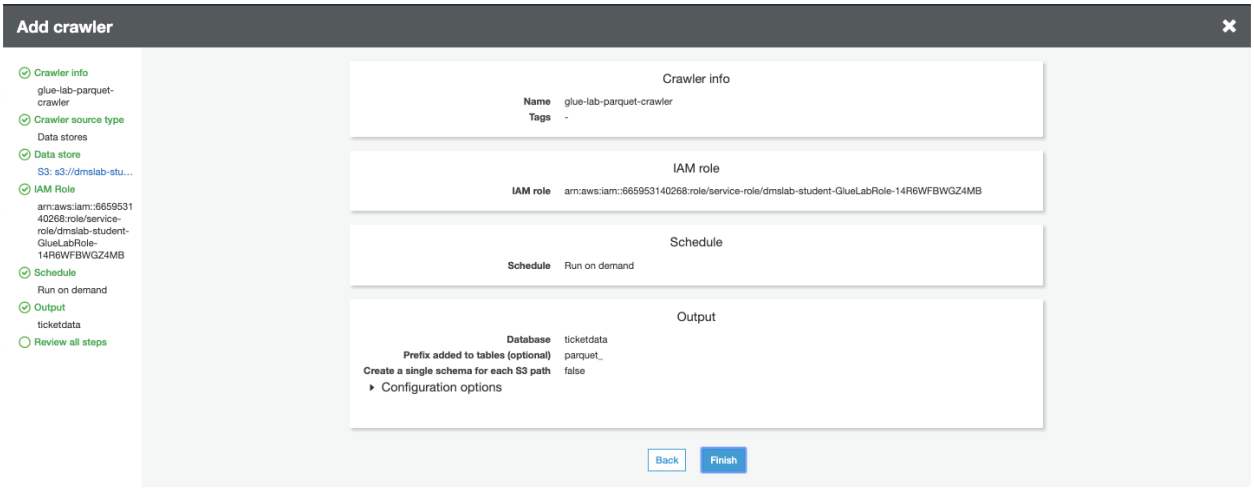
7. For **Frequency**, select "Run On Demand" and Click **Next**.



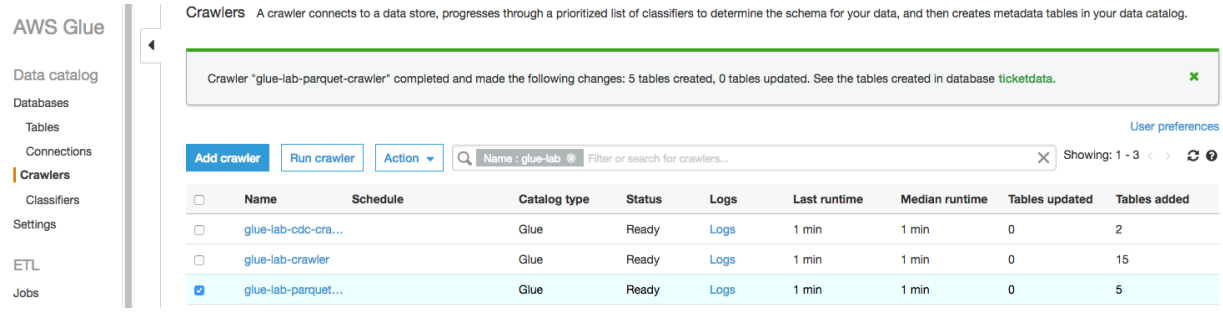
8. For the crawler's output database, choose your existing database which you created earlier e.g. "**ticketdata**"
9. For the **Prefix added to tables** (optional), type "**parquet_**"



10. Review the summary page and click **Finish**.

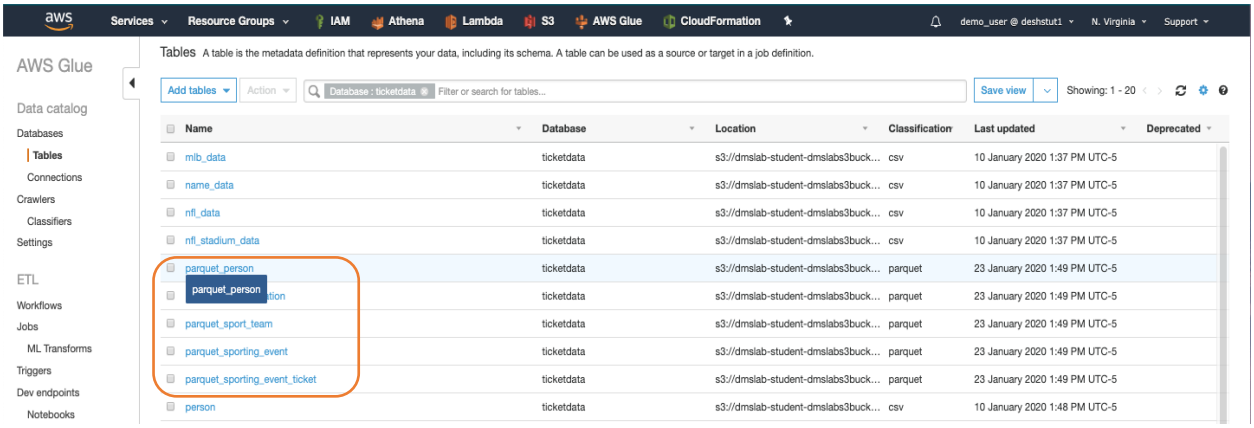


11. On the notification bar, click **Run it now**. Once your crawler has finished running, you should report that tables were added from 1 to 5, depending on how many parquet ETL conversions you set up in the previous section



Confirm you can see the tables:

1. In the left navigation pane, click **Tables**.
2. Add the filter "parquet" to return the newly created tables.



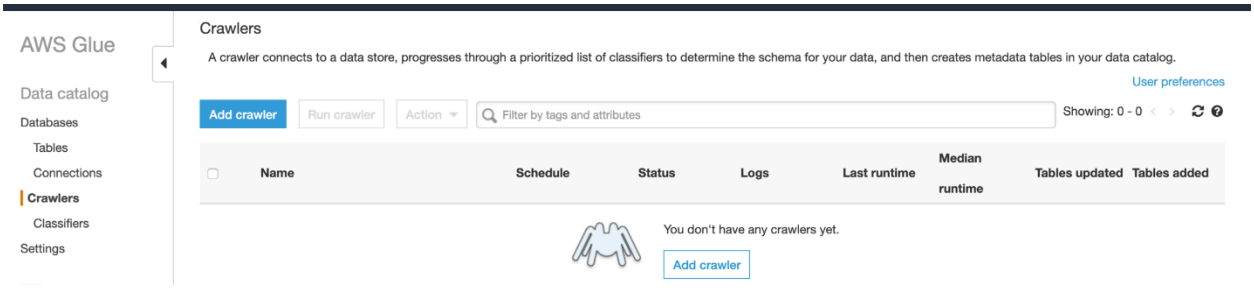
PART B: Glue Job Bookmark (Optional):

****Pre-requisite: Completion of CDC part of DMS Lab ****

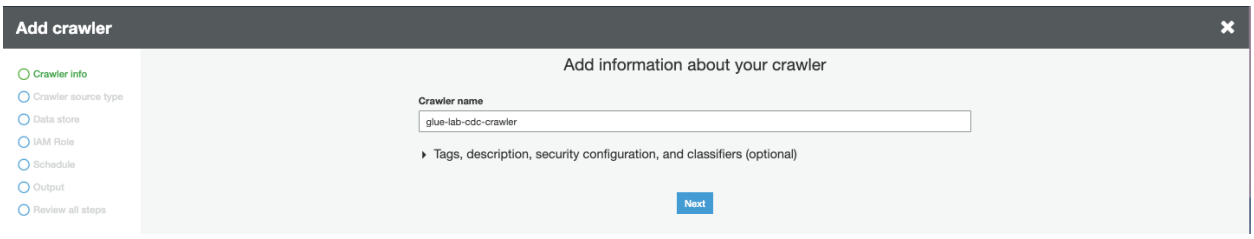
Step 1: Create Glue Crawler for ongoing replication (CDC Data)

Now, let's repeat this process to load the data from change data capture.

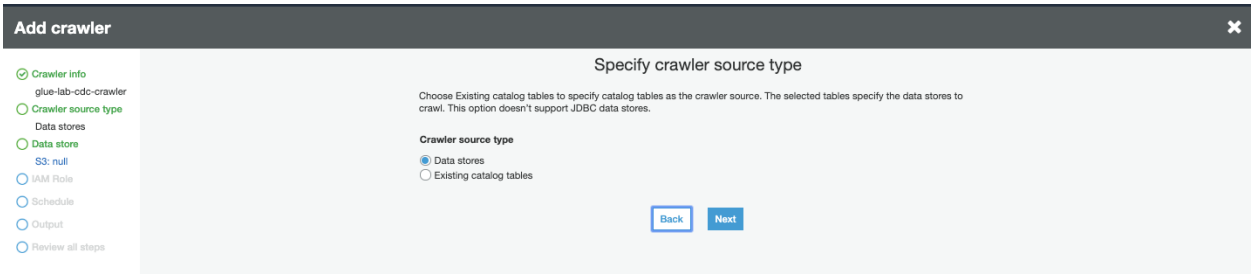
1. On the AWS Glue menu, select Crawlers.



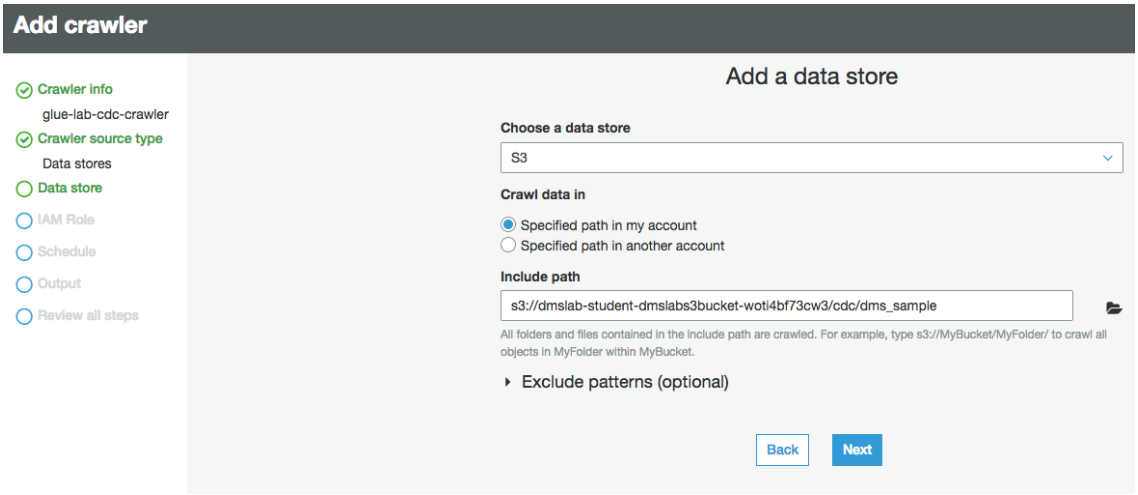
2. Click **Add crawler**.
3. Enter the crawler name for ongoing replication. This name should be descriptive and easily recognized (e.g., "**glue-lab-cdc-crawler**").
4. Optionally, enter the description. This should also be descriptive and easily recognized and Click **Next**.



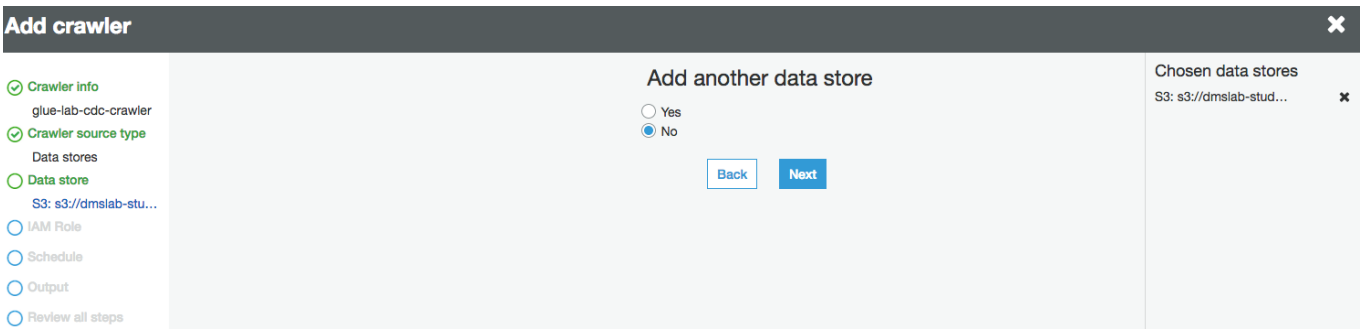
5. Choose **Data Stores** as **Crawler Source Type** and Click **Next**



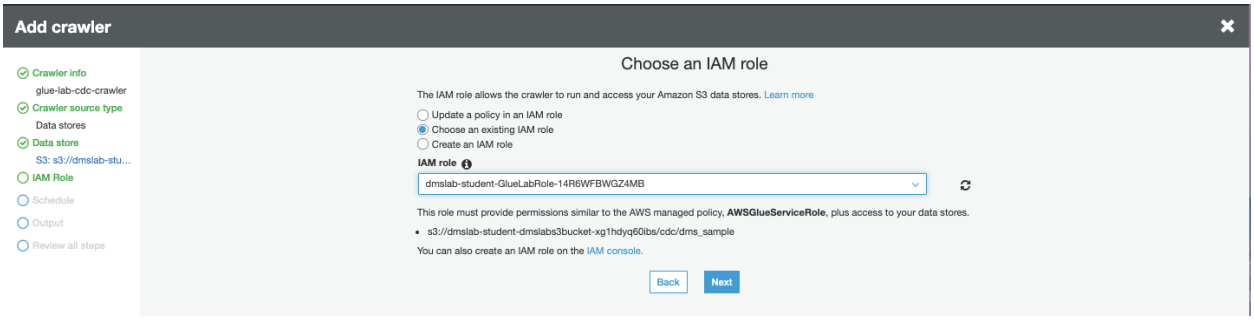
6. On the Add a data store page, make the following selections:
    a. For **Choose a data store**, click the drop-down box and select **S3**.
    b. For **Crawl data in**, select **Specified path in my account**.
    c. For **Include path**, enter the **target folder** for your DMS ongoing replication, e.g., "**s3://xxx-dmslabs3bucket-xxx/cdc/dms_sample**"
7. Click **Next**.



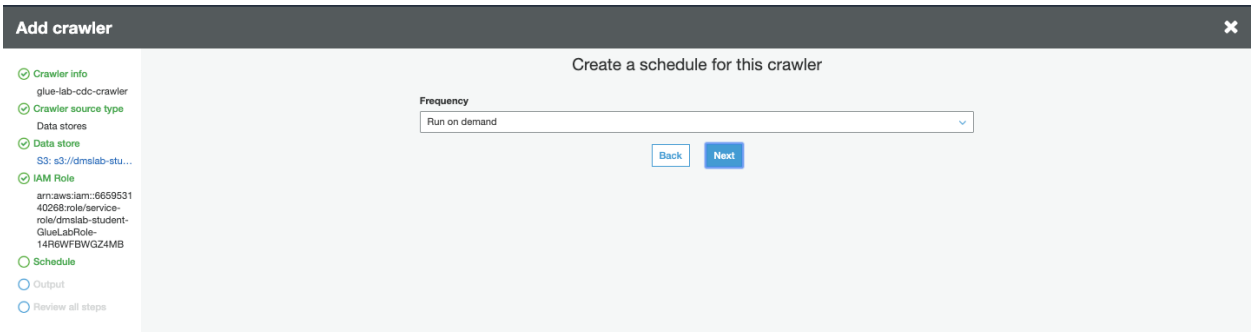
8. On the **Add another data store page**, select **No** and Click **Next**.



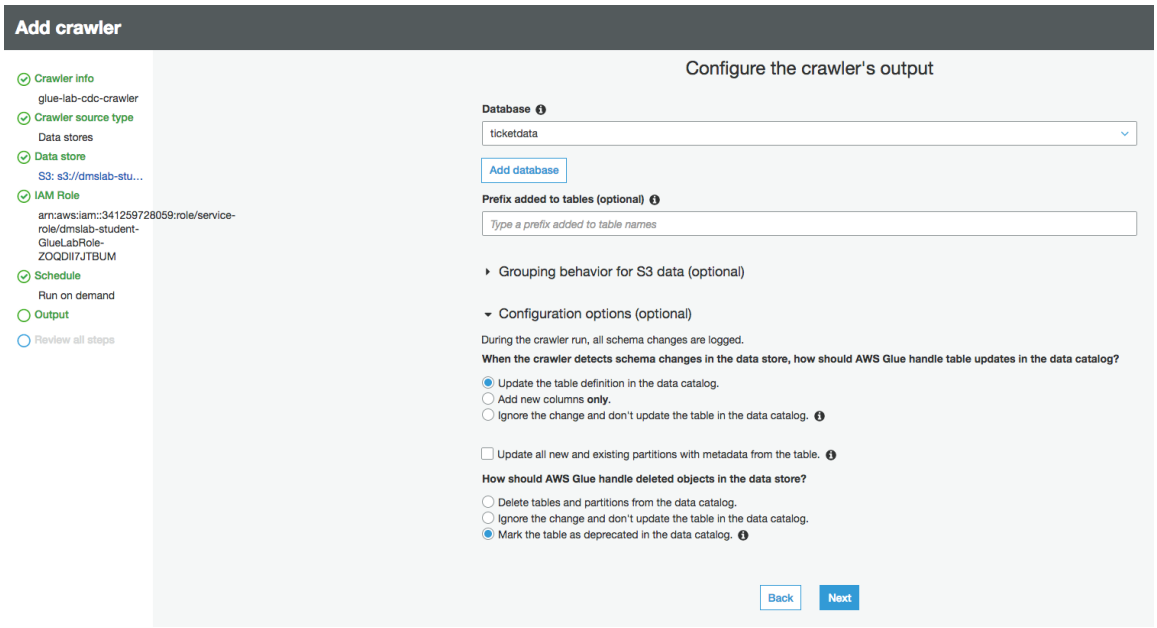
9. On the **Choose an IAM role** page, make the following selections:
    a. Select **Choose an existing IAM role**.
    b. **For IAM role**, select **xxx-GlueLabRole-xxx**. E.g. "dmslab-student-GlueLabRole-ZOQDII7JTBUM"
10. Click **Next**.



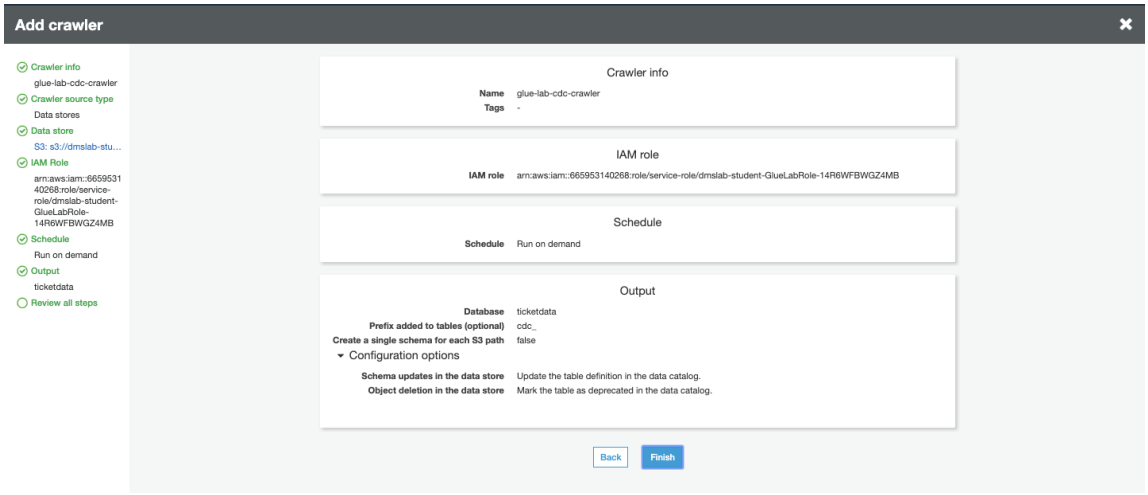
11. On the Create a schedule for this crawler page, for Frequency, select **Run on demand** and Click **Next**.



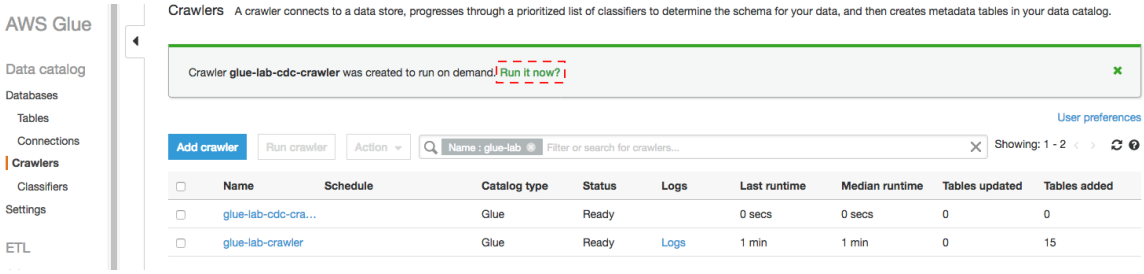
12. On the Configure the crawler's output page, select the existing **Database** for crawler output (e.g., "ticketdata").
13. For **Prefix added to tables (optional)**, specify "cdc_"
14. For Configuration options (optional), keep the default selections and click **Next**.



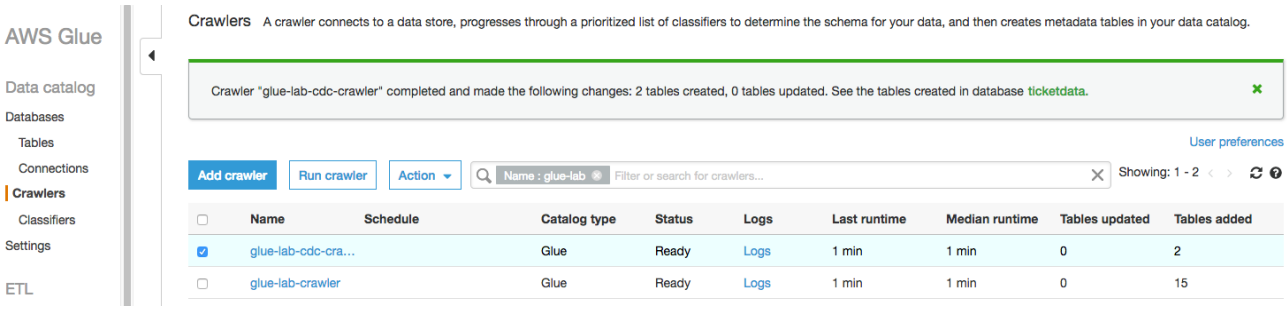
15. Review the summary page noting the Include path and Database target and Click **Finish**. The crawler is now ready to run.



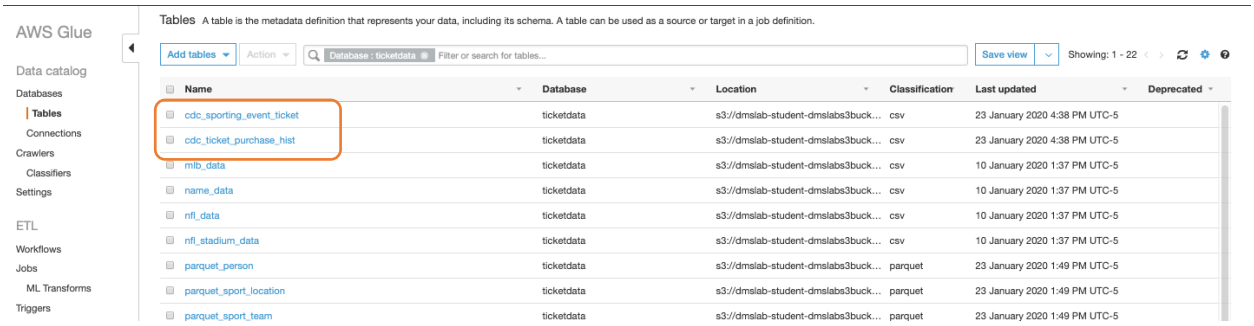
16. Click **Run it now**.



17. When the crawler is completed, you can see it has "Status" as **Ready,** Crawler will change status from starting to stopping, wait until crawler comes back to ready state, you can see that it has created 2 tables.



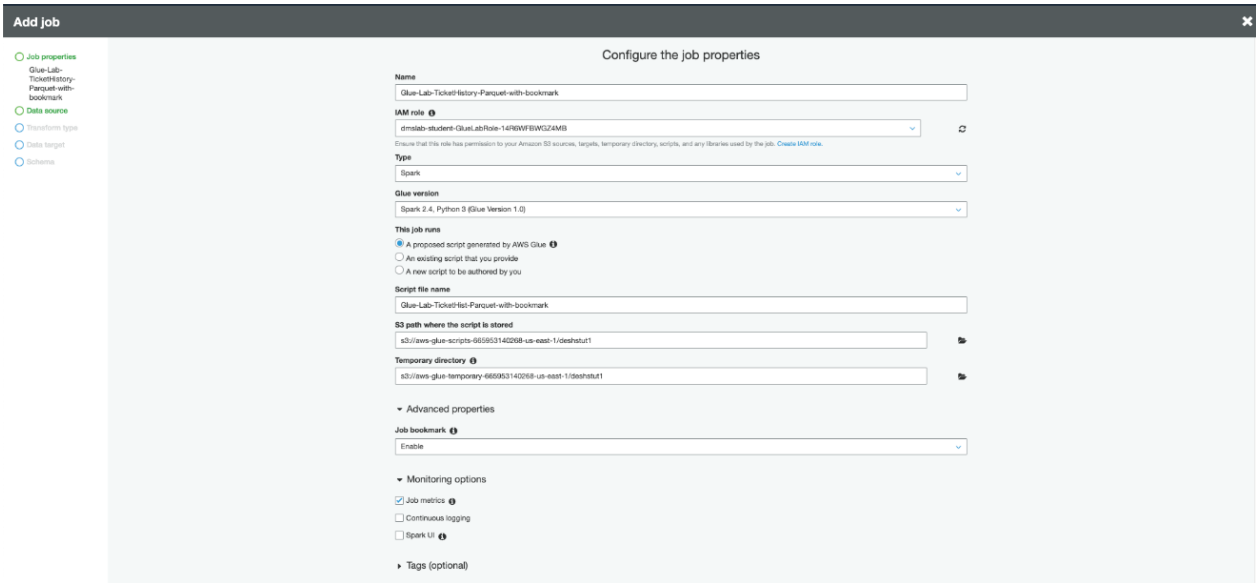
18. Click the database name (e.g., "ticketdata") to browse the tables. Specify "cdc" as the filter to list only newly imported tables.



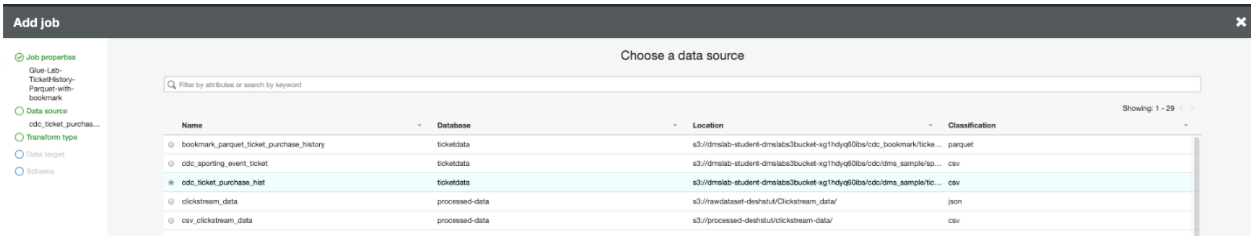
Step 2: Create a Glue Job with Bookmark Enabled

1. On the left-hand side of Glue Console, click on Jobs and then Click on Add Job

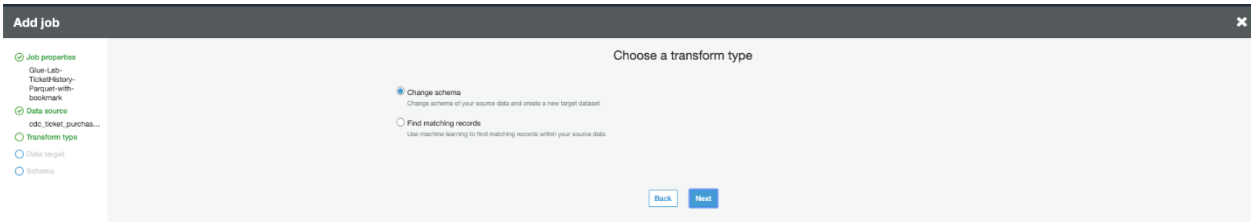
2. On the Job properties page, make the following selections:
    a. For **Name**, type **Glue-Lab-TicketHistory-Parquet-with-bookmark**.
    b. For **IAM role**, choose existing role "xxx-GlueLabRole-xxx"
    c. For **Type**, Select **Spark**
    d. For **Glue Version**, select **Spark 2.4, Python 3 (Glue version 1.0)** or whichever is the latest version
    e. For **This job runs**, select **A proposed script generated by AWS Glue**.
    f. For **Script file name**, type **Glue-Lab-TicketHistory-Parquet-with-bookmark**.
    g. For **S3 path where the script is stored**, provide a unique Amazon S3 path to store the scripts. (You can keep the default for this lab.)
    h. For **Temporary directory**, provide a unique Amazon S3 directory for a temporary directory. (You can keep the default for this lab.)
3. Expand the **Advanced properties** section. For Job bookmark, select **Enable** from the drop-down option.
4. Expand on the **Monitoring** options, enable **Job metrics**.
5. Click **Next**



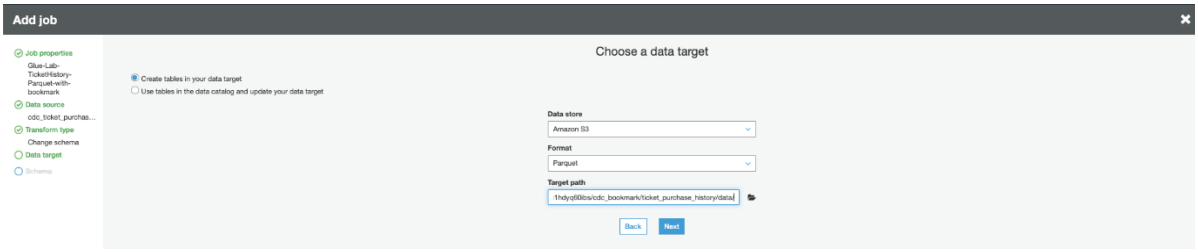
6. In **Choose a data source,** select **cdc_ticket_purchase_hist** as we are generating new data entries for **ticket_purchase_hist table**. Click **Next**



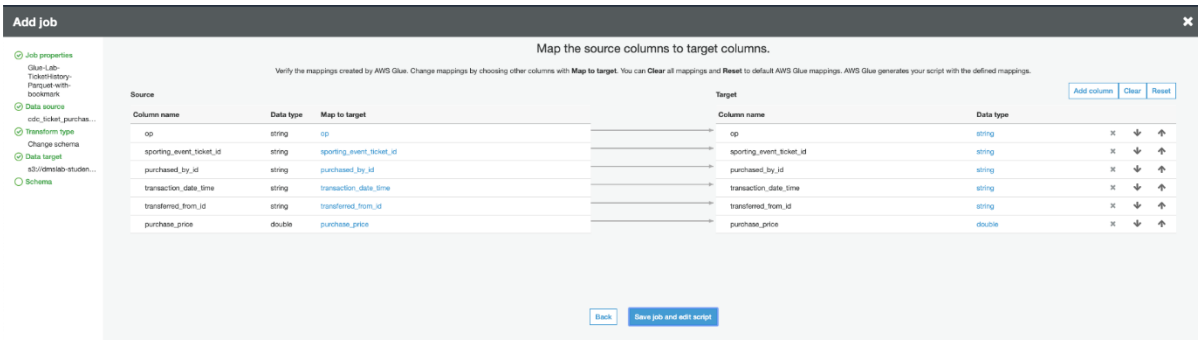
7. In **Choose a transform type,** select **Change** Schema and Click **Next**



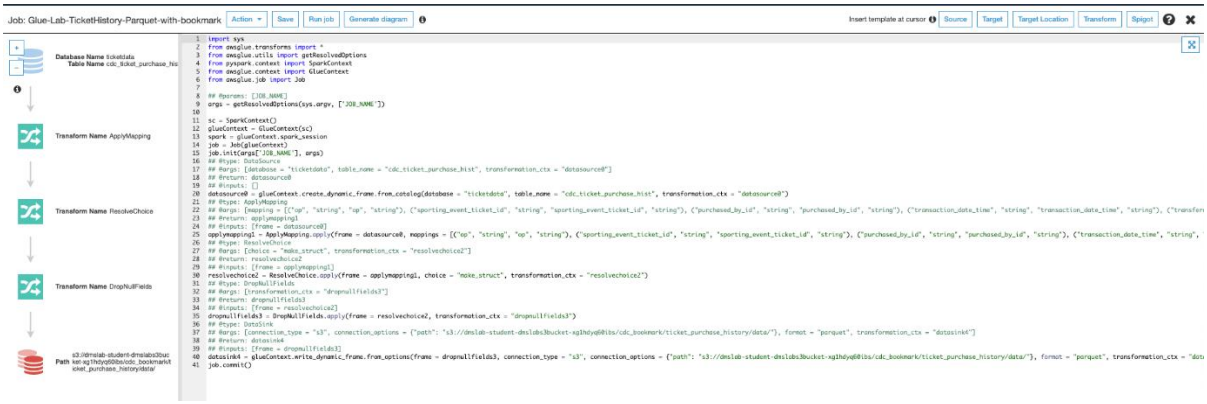
8. In Choose a data target:
    a. For **Data store**: select amazon **S3**
    b. Format: **parquet**
    c. **Target path**: s3://xxx-dmslabs3bucket-xxx/cdc_bookmark/ticket_purchase_history/data/
    d. Click **Next**



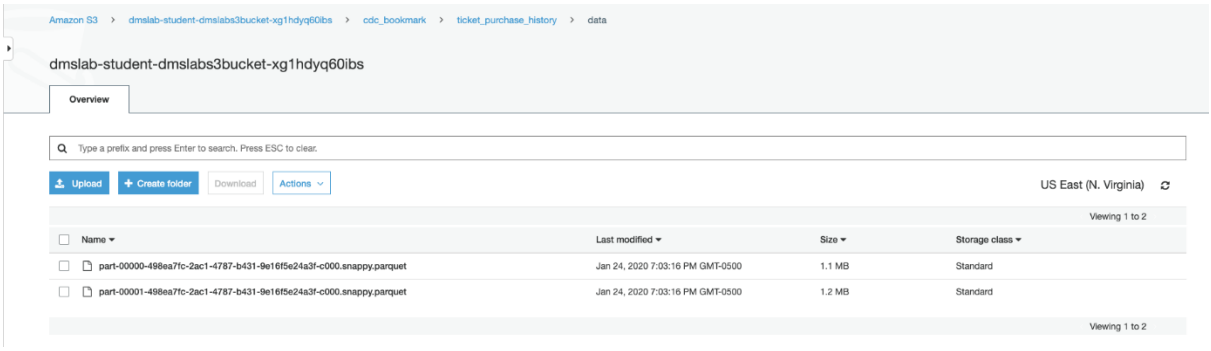
9. In map the source columns to target columns window, leave everything default and Click on **Save job and edit script**.



10. In the next window, review the job script and click on **Run job**. Click on close mark on the top right of the window to close the screen.

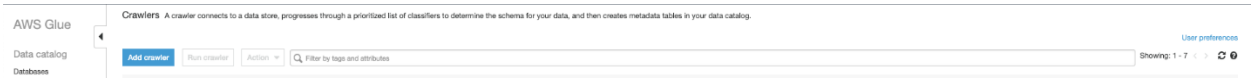


11. Once the job finishes its run, check the **S3 bucket** for the parquet partitioned data.

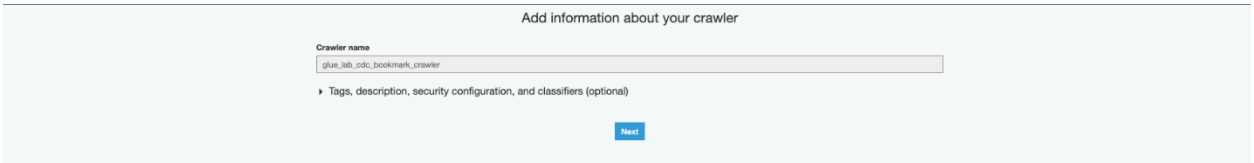


Step 3: Create Glue crawler for Parquet data in S3

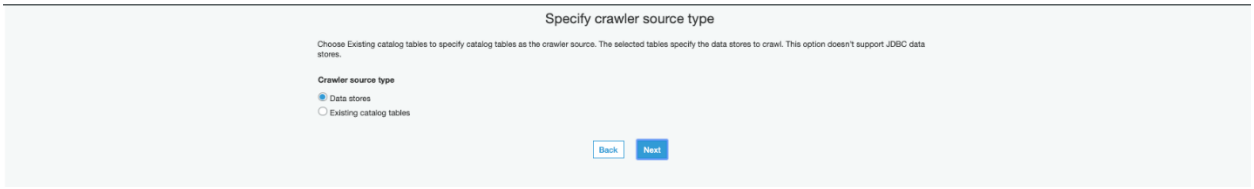
1. Once you have the data in S3 bucket, navigate to **Glue Console** and now we will crawl the parquet data in S3 to create data catalog.
2. Click on **Add crawler**



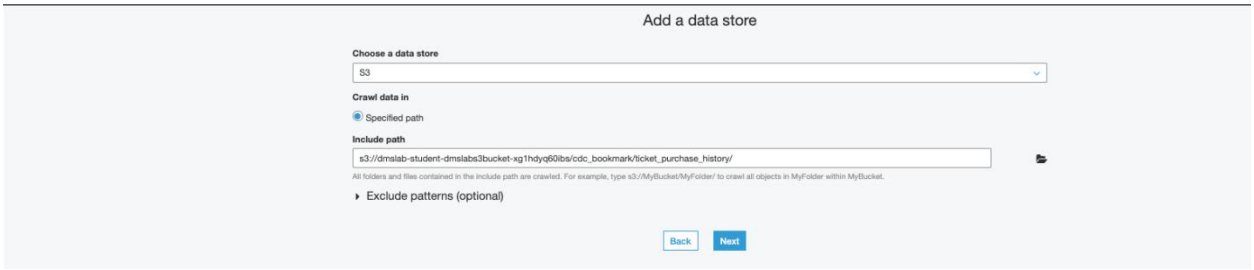
3. In crawler configuration window, provide crawler name as **glue_lab_cdc_bookmark_crawler** and Click **Next**.



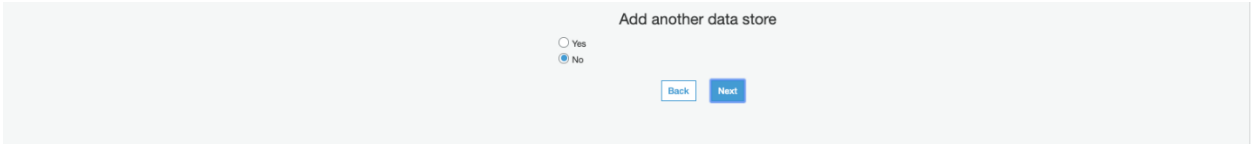
4. In specify **crawler source type**, for crawler source type, select **Data stores**. Click **Next**



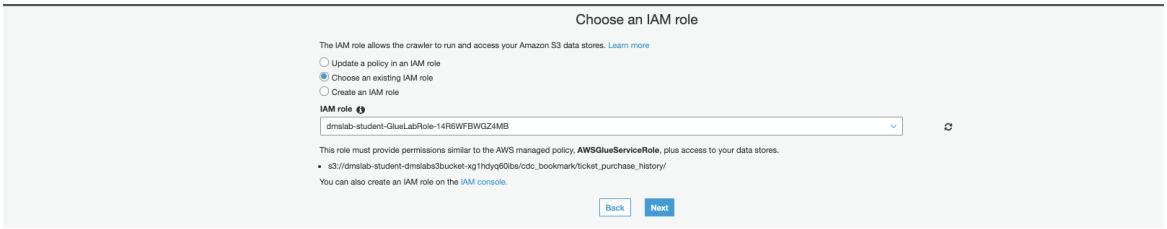
5. In **Add a data store**:
   a. For **Choose a data store**, select **S3**
   a. For the **Include path**, click the folder icon and choose your target S3 bucket, then append **/cdc_bookmark/ticket_purchase_history** , e.g., "s3://xxx-dmslabs3bucket-xxx/cdc_bookmark/ticket_purchase_history"
6. Click on **Next**



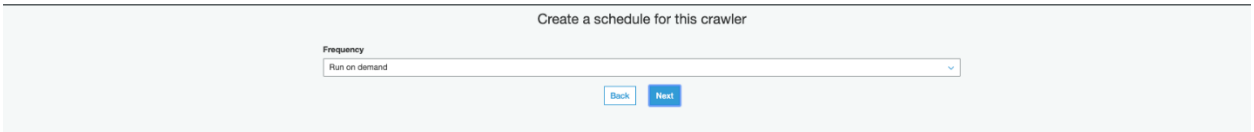
7. For **Add another data** store, select **No** and click **Next**.



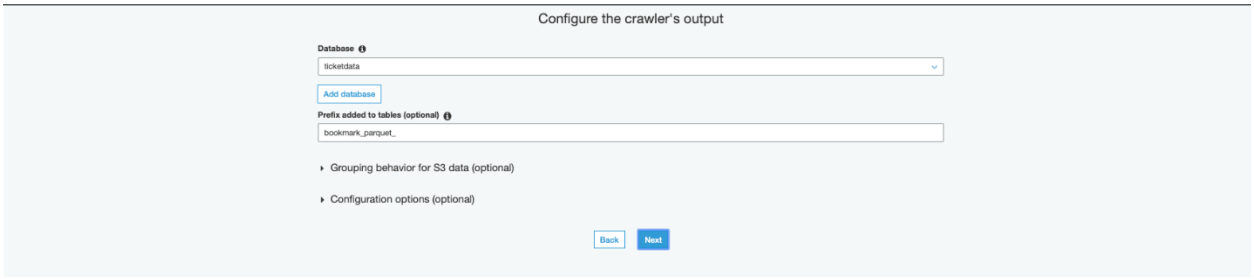
8. In **Choose an IAM role**, select an existing IAM role contains **GlueLabRole** text. Something looks like this: xxx-GlueLabRole-xxx



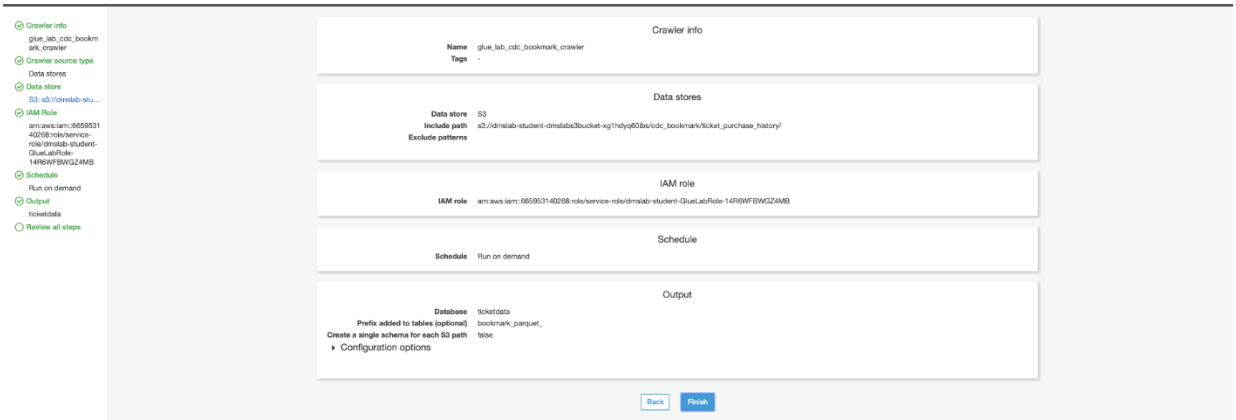
9. For setting the **frequency** in create a schedule for this crawler, select "Run on demand". Click **Next**



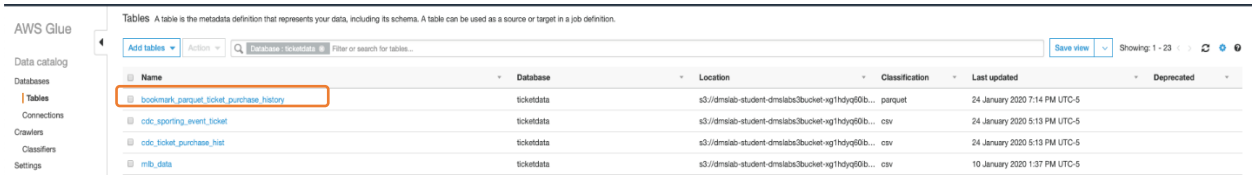
10. For the crawler's output:
    a. For Database, select "**ticketdata**" database.
    b. Optionally, add prefix to the newly created tables for easy identification. Provide the prefix as "**bookmark_parquet_"**
    c. Click **Next**



11. Review all the details and click on **Finish**. Next, **run the crawler**.



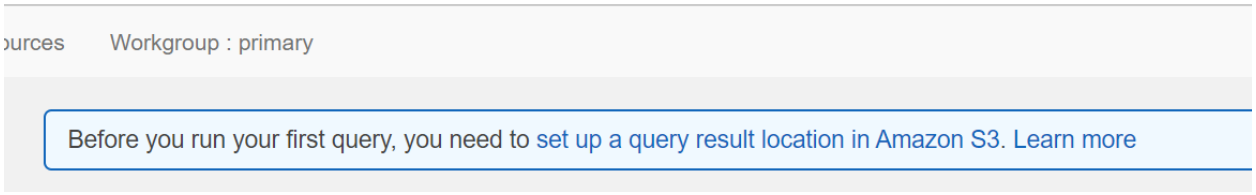
12. After the crawler finishes running, click on Databases, select "**ticketdata**" and view tables in this database. You will find the newly created table as "**bookmark_parquet_ticket_purchase_history**"



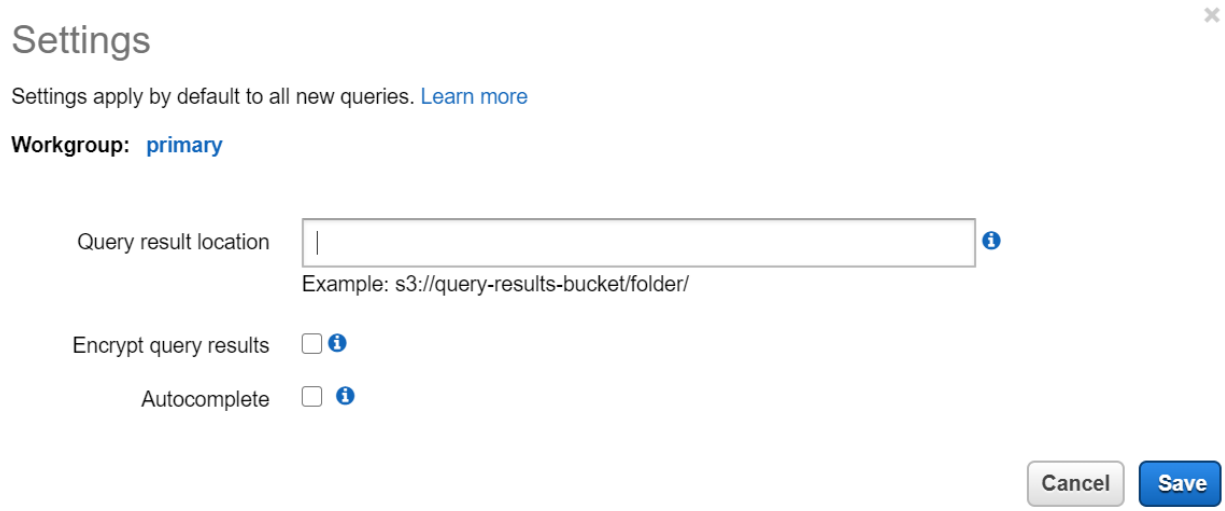
13. Once the table is created, click on **Action** and from **dropdown** select **View Data.**

If it's the first time you are using Athena in your AWS Account, click **Get Started**

Then click **set up a query result location in Amazon S3** at the top



In the pop-up window in the **Query result location** field, enter your s3 bucket location followed by **/**, so that it looks like **s3://xxx-dmslabs3bucket-xxx/** and click **Save**



To select some rows from the table, try running:

```
SELECT * FROM "ticketdata"."bookmark_parquet_ticket_purchase_history"
limit 10;
```

To get a row count, run:

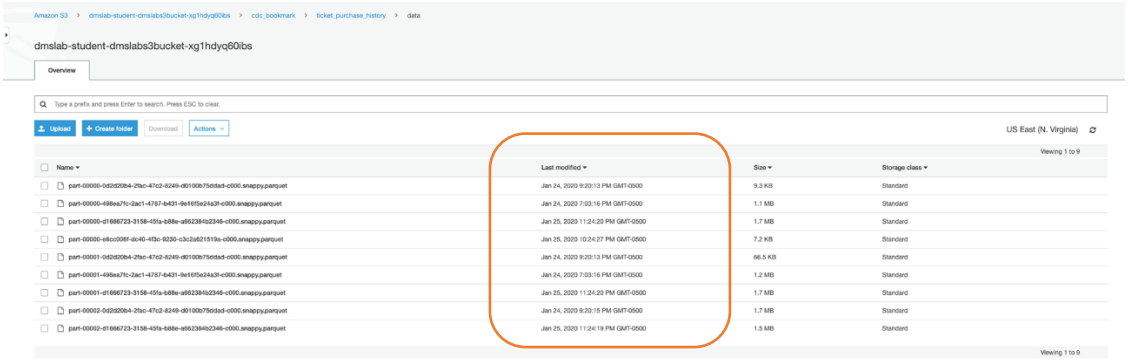
```
SELECT count(*) as recordcount FROM
"ticketdata"."bookmark_parquet_ticket_purchase_history";
```

Before moving on to next step, note the rowcount.

Step 4: Generate CDC data and to observe bookmark functionality

Ask your instructor generate more CDC data at source database, if you ran the instructor setup on your own, then make sure to follow "**Generate the CDC Data"** section from instructor prelab.

1. To make sure the new data has been successfully generated, check the S3 bucket for cdc data, you will see new files generated. Note the time when the files were generated.



2. Rerun the Glue job **Glue-Lab-TicketHistory-Parquet-with-bookmark** you created in Step 2

3. Go to the Athena Console, and rerun the following query to notice the increase in row count:

```
SELECT count(*) as recordcount FROM
"ticketdata"."bookmark_parquet_ticket_purchase_history";
```

To review the latest transactions, run:

```
SELECT * FROM "ticketdata"."bookmark_parquet_ticket_purchase_history"
order by transaction_date_time desc limit 100;
```

PART C: Glue Workflows (Optional, self-paced)

****Pre-requisite before creating workflow** - completed Part B**

Overview:

In AWS Glue, you can use workflows to create and visualize complex extract, transform, and load (ETL) activities involving multiple crawlers, jobs, and triggers. Each workflow manages the execution and monitoring of all its components. As a workflow runs each component, it records execution progress and status, providing you with an overview of the larger task and the details of each step. The AWS Glue console provides a visual representation of a workflow as a graph.

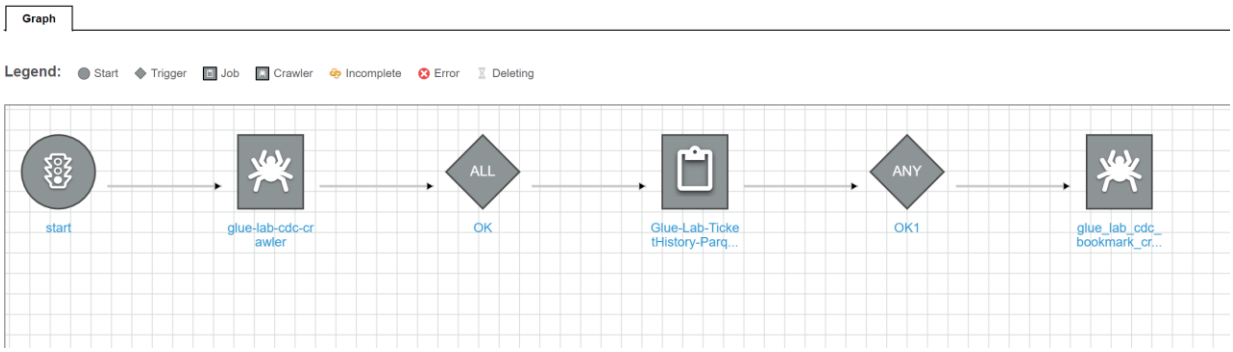
Creating and Running Workflows:

Above mentioned Part A (ETL with Glue) and Part B (Glue Job Bookmarks) can be created and executed using workflows. Complex ETL jobs involving multiple crawlers and jobs can also be created and

executed using workflows in an automated fashion. Below is a simple example to demonstrate how to create and run workflows.

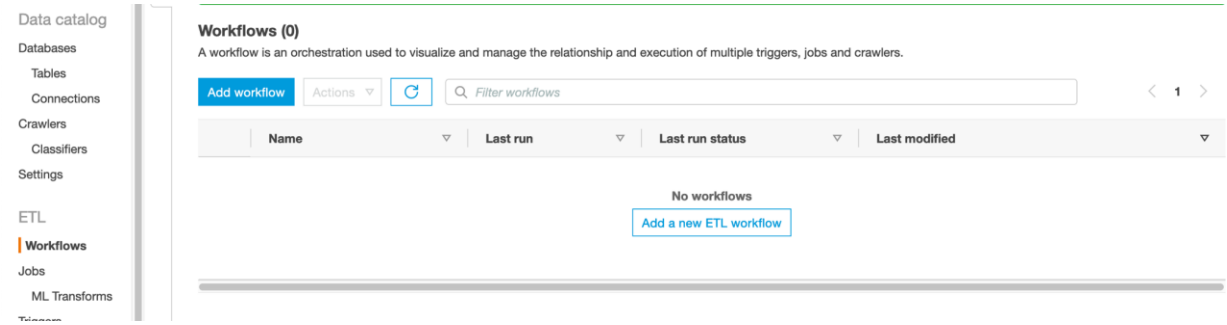
Try creating a new Glue Workflow to string together the two Crawlers and one Job from part B as follows:

On-demand trigger -> glue-lab-cdc-crawler -> Glue-Lab-TicketHistory-Parquet-with-bookmark -> glue_lab_cdc_bookmark_crawler

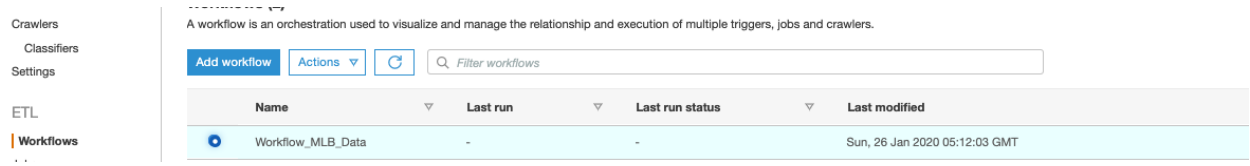


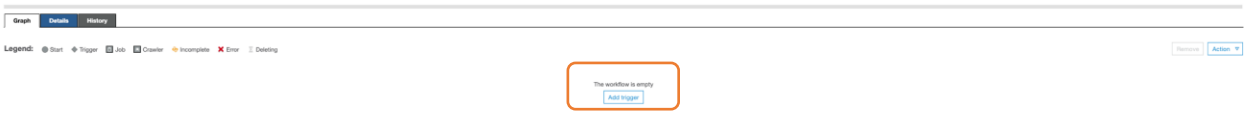
To create a workflow:

1. Navigate to **AWS Glue Console** and under **ETL,** click on **Workflows**. Then Click on **Add Workflow**.



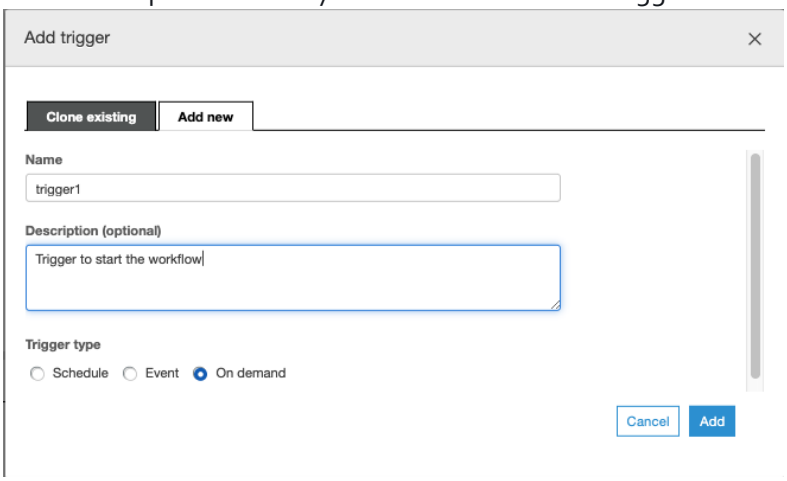
2. Give the workflow name as "**Workflow_tickethistory**". Provide a description (optional) and click on **Add Workflow** to create it.
3. Click on the **workflow** and scroll to the bottom of the page. You will see an option **Add Trigger**. Click on that button.



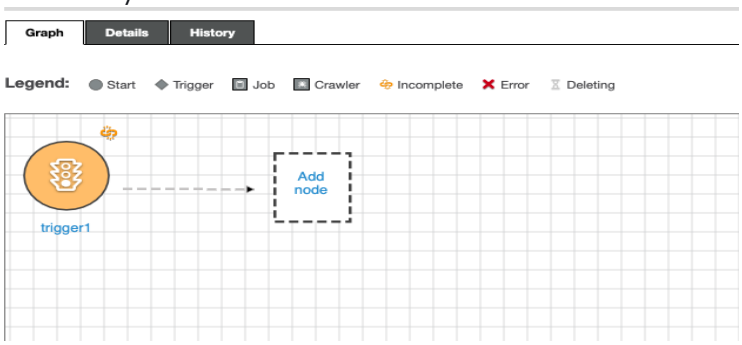


4. In **Add Trigger** window, From Clone Existing and Add New options, click on **Add New**.
    a. Provide **Name** as "**trigger1**"
    b. Provide a **description**: Trigger to start workflow
    c. **Trigger type**: **On-demand**.
    d. Click on **Add**

    Triggers are used to initiate the workflow and there are multiple ways to invoke the trigger. Any scheduled operation or any event can activate the trigger which in turn starts the workflow

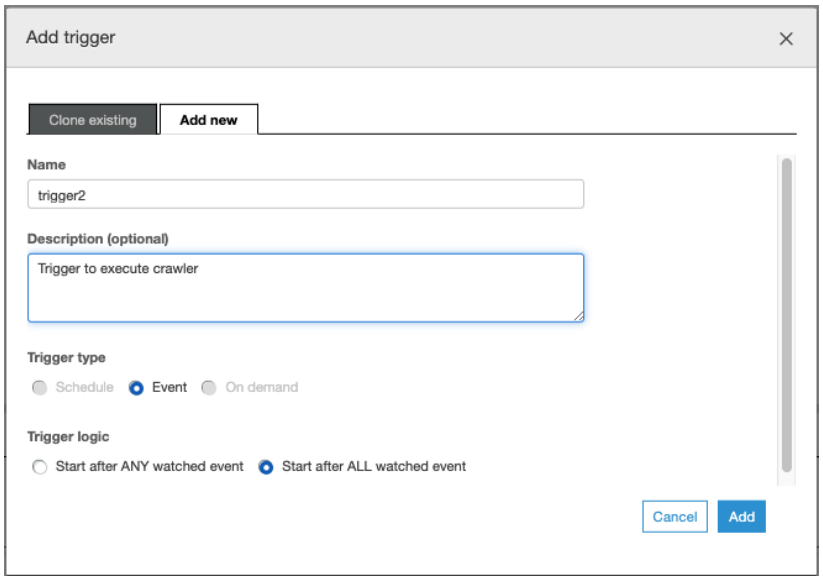


5. Click on **trigger1** to add a **new node**. New Node can be a crawler or job, depending upon the workflow you want to build.



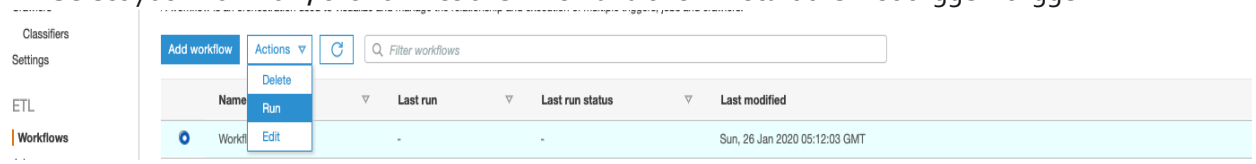
6. Click on **Add node,** a new window to add jobs or crawlers will open. Select the Crawler **glue-lab-cdc-crawler**, then **Add.**
7. Click on the crawler and **Add Trigger** provide the following:
    a. **Name**: **trigger2**
    b. **Description**: Trigger to execute job
    c. **Trigger type**: **Event**
    d. **Trigger logic**: **Start after ALL watched event.** This will make sure that job starts once Glue Crawler finishes.

e. Click **Add**



8. After **trigger2** is added to workflow, Click on **Add node,** select job **Glue-Lab-TicketHistory-Parquet-with-bookmark,** click **Add.**
9. Click on the job and **Add Trigger** provide the following:
    a. **Name**: **trigger3**
    b. **Description**: Trigger to execute crawler
    c. **Trigger type**: **Event**
    d. **Trigger logic**: **Start after ANY watched event.** This will make sure that crawler starts once Glue job finishes processing of ALL data.
    e. Click **Add**

10. Click on **Add node,** Select the Crawler **glue_lab_cdc_bookmark_crawler**, then **Add.**
11. Select your workflow, click on **Actions->Run** and this will start the first trigger "trigger1"



12. Once the workflow is completed, you will observe that glue job and crawlers have been successfully executed.

Congratulations!! You have successfully completed this lab