



Amazon Web Services

Data Engineering Immersion Day

Lab 2. ETL with AWS Glue

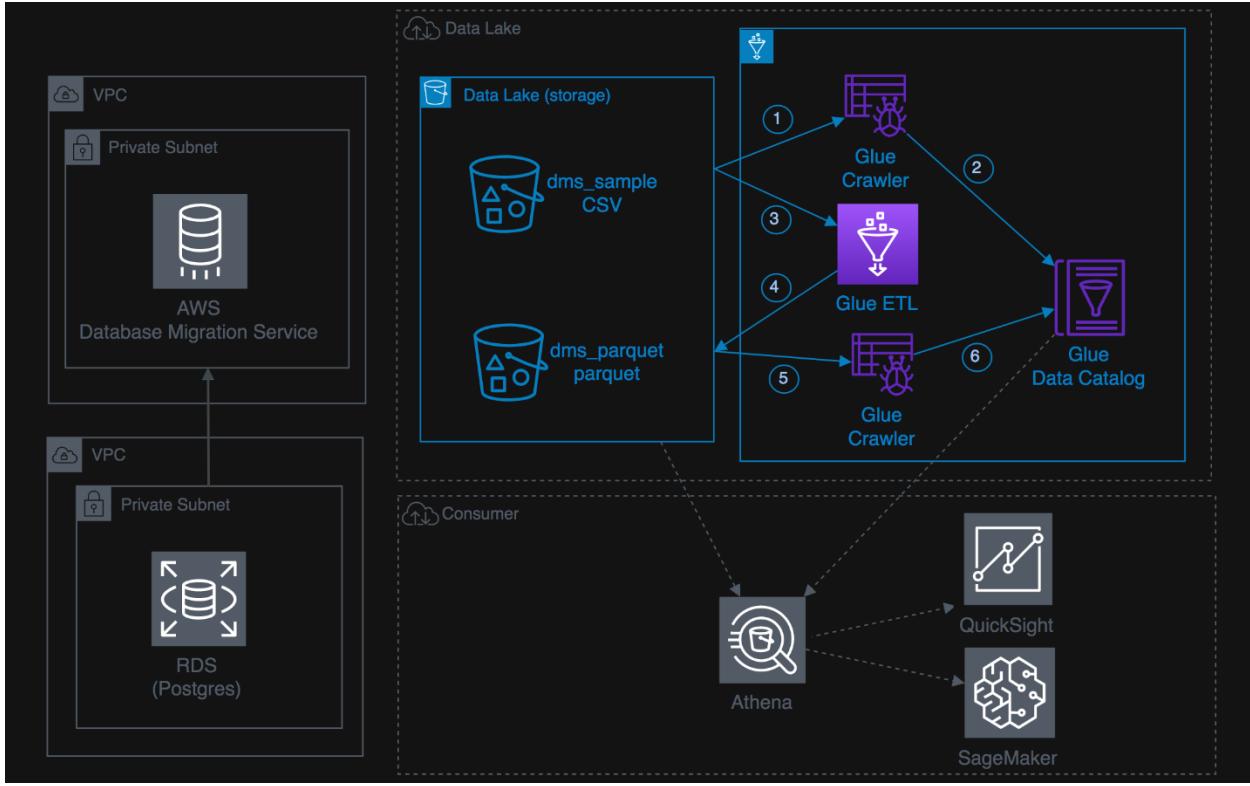
August 2020

Table of Contents

<i>Introduction</i>	2
<i>Get Started Using the Lab Environment</i>	3
<i>PART A: Data Validation and ETL</i>	7
Create Glue Crawler for initial full load data	7
Data Validation Exercise.....	12
Data ETL Exercise	13
Create Glue Crawler for Parquet Files	18
<i>PART B: Glue Job Bookmark (Optional):</i>	22
Step 1: Create Glue Crawler for ongoing replication (CDC Data).....	22
Step 2: Create a Glue Job with Bookmark Enabled	26
Step 3: Create Glue crawler for Parquet data in S3	29
Step 4: Generate CDC data and to observe bookmark functionality	32
<i>PART C: Glue Workflows (Optional, self-paced)</i>	33
Overview:.....	33
Creating and Running Workflows:	33

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 Quicksight

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/>

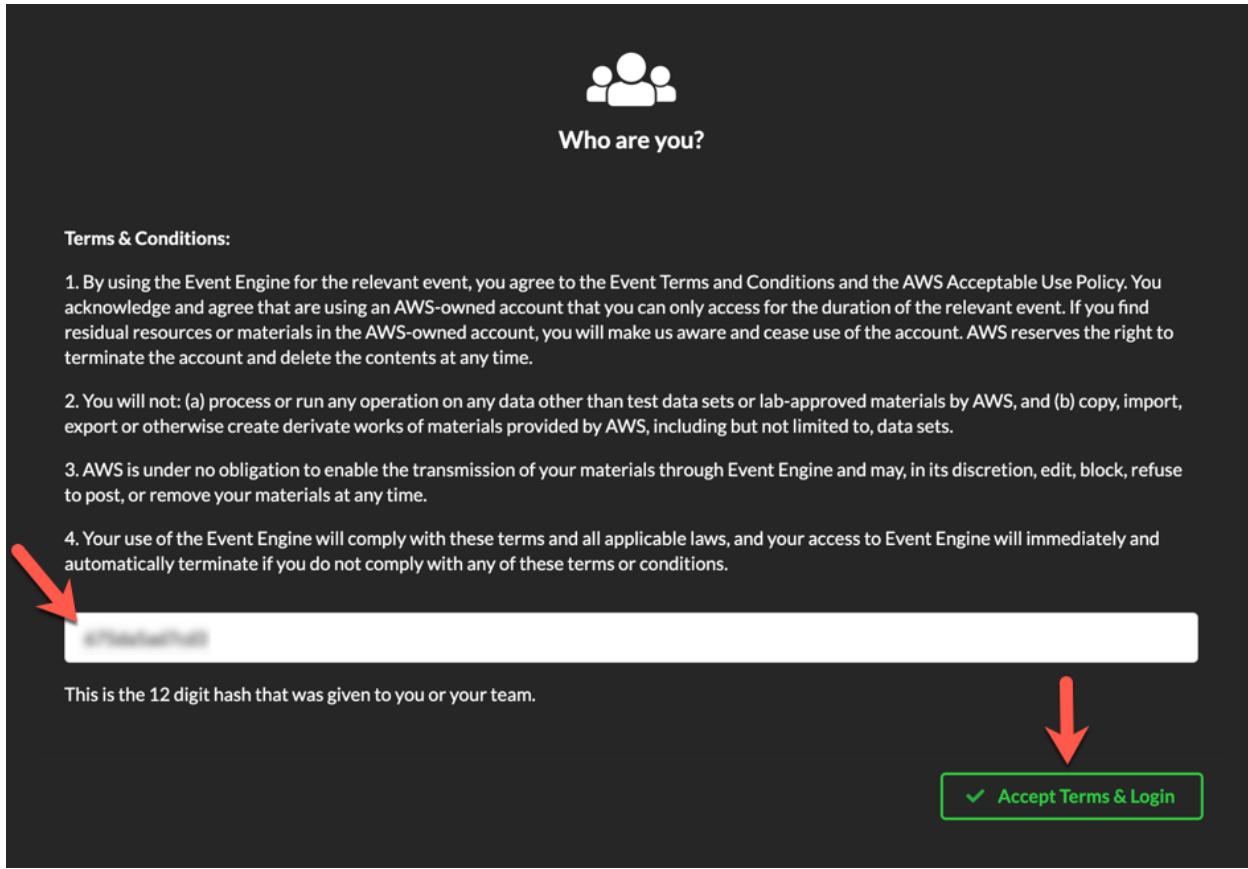
Get Started Using the Lab Environment

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

RDS DB Info	Readme
Outputs:	
No outputs defined	

- S3 Bucket, IAM role for the Glue lab etc

Lab 2. ETL with AWS Glue

Modules

Environment Setup

Readme

Outputs:

S3 Bucket name
mod-3fccddd609114925-dmslabs3bucket-1ngcgzzcnd15u [\[copy\]](#)

BusinessAnalystUser
mod-3fccddd609114925-BusinessAnalystUser-MB0XFZLQLOXX [\[copy\]](#)

DMSLabRoleS3 ARN
arn:aws:iam::377243295828:role/mod-3fccddd609114925-DMSLabRoleS3-O2VT1RSN43SG [\[copy\]](#)

Glue Lab Role
mod-3fccddd609114925-GlueLabRole-YLTJA13WW6WT [\[copy\]](#)

S3BucketWorkgroupA
mod-3fccddd609114925-s3bucketworkgroupa-tbon3m1mkunh [\[copy\]](#)

S3BucketWorkgroupB
mod-3fccddd609114925-s3bucketworkgroupb-18ygl8nfp8ead [\[copy\]](#)

WorkgroupManagerUser
mod-3fccddd609114925-WorkgroupManagerUser-5IVE0UQNIBG4 [\[copy\]](#)

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

Team Dashboard

Event

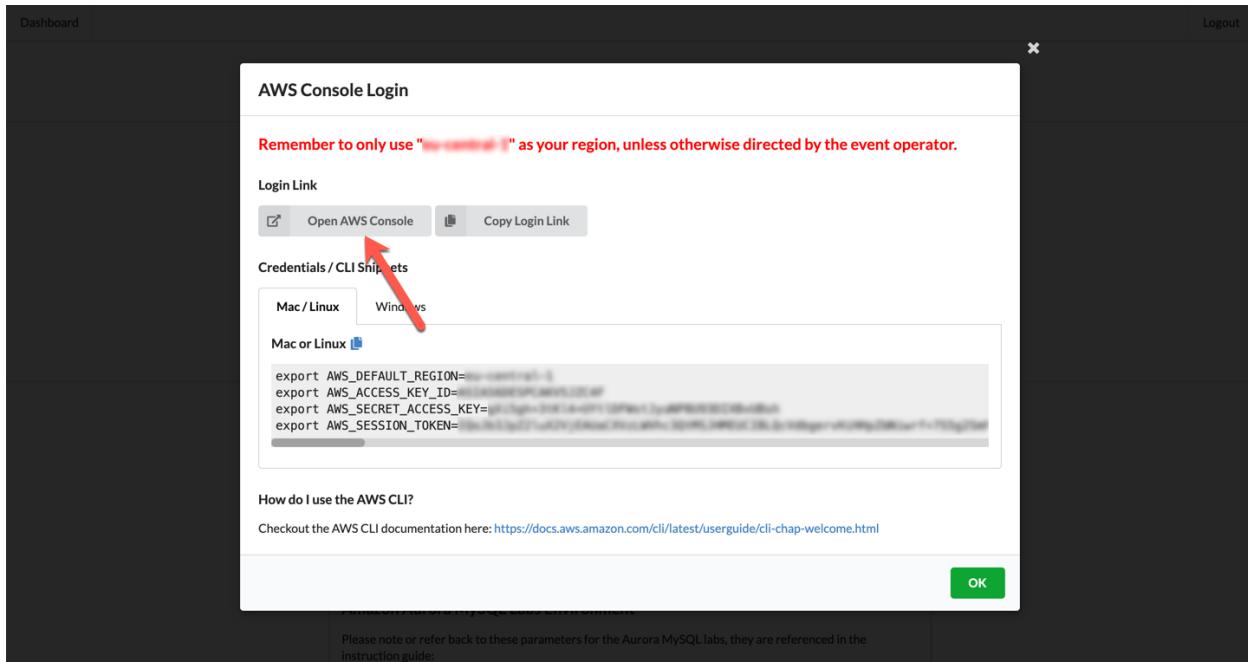
AWS Console SSH Key

Event: Data Engineering Immersion Day - Test
Team Name:

Event ID: d2302d4ae9ff4ea2857846b74f7de7e2
Team ID: 1c2f7ad7ec044b0b8276f917c5983133

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

Lab 2. ETL with AWS Glue



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: <https://console.aws.amazon.com/glue/home?region=us-east-1>

The screenshot shows the AWS Services console. In the top-left corner, there's a search bar with the text "glue" typed into it. Below the search bar, a list of services is displayed, with "AWS Glue" being the first item. Other visible items include "AWS Lake Formation" and "S3". At the bottom of the screen, there are navigation links for "All services", "S3", and "EC2".

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

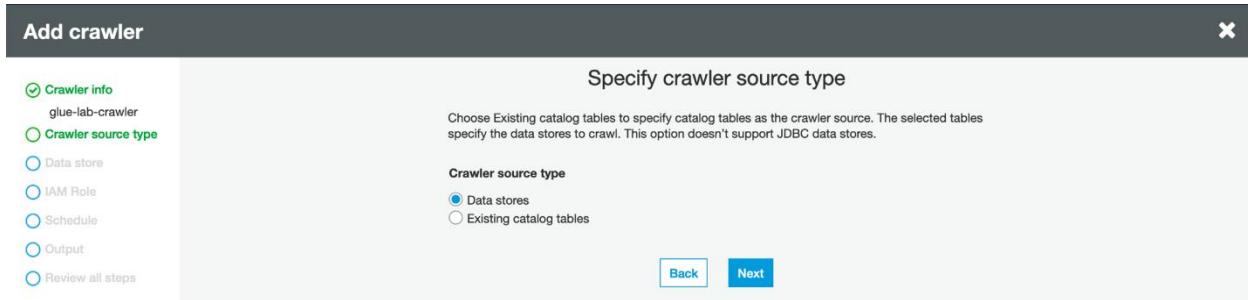
The screenshot shows the "Crawlers" page within the AWS Glue service. On the left, there's a sidebar with navigation options: "Data catalog", "Databases", "Tables", "Connections", "Crawlers" (which is selected and highlighted in orange), "Classifiers", and "Settings". The main area is titled "Crawlers" and contains a brief description: "A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog." Below this, there are buttons for "Add crawler", "Run crawler", "Action", and a search bar. A message "Showing: 0 - 0" is displayed. A table header row is shown with columns: "Name", "Schedule", "Status", "Logs", "Last runtime", "Median runtime", "Tables updated", and "Tables added". A message "You don't have any crawlers yet." is centered, and a blue "Add crawler" button is located below it.

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**.

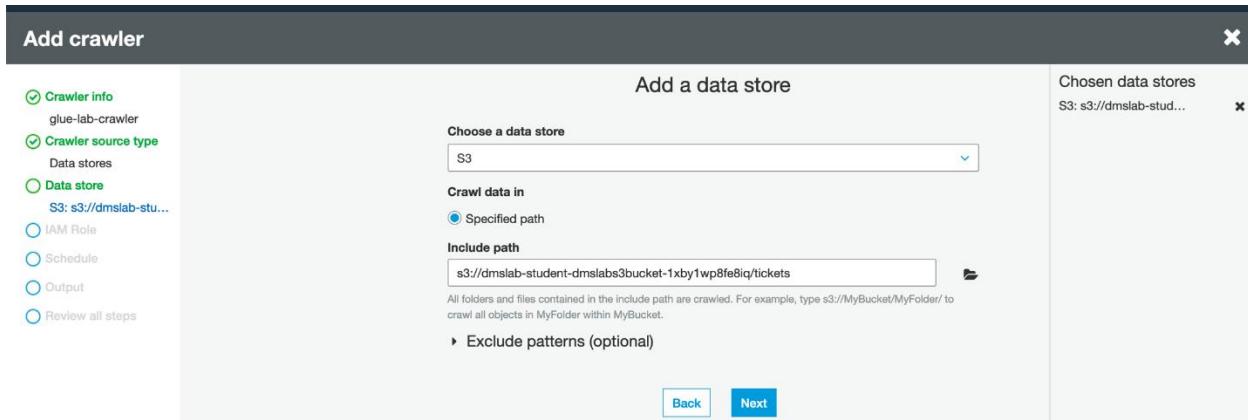
The screenshot shows the "Add crawler" wizard. The title bar says "Add crawler". The left sidebar lists steps: "Crawler info" (selected and highlighted in green), "Crawler source type", "Data store", "IAM Role", "Schedule", "Output", and "Review all steps". The main area is titled "Add information about your crawler". It has a "Crawler name" input field containing "glue-lab-crawler". Below the input field is a note: "Tags, description, security configuration, and classifiers (optional)". A blue "Next" button is at the bottom right.

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

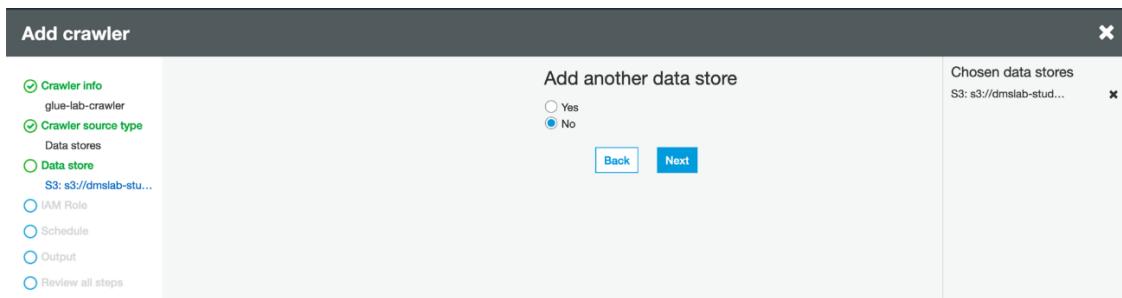
Lab 2. ETL with AWS Glue



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-wot14bf73cw3/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**.

Lab 2. ETL with AWS Glue

Add crawler

Crawler info
glue-lab-crawler

Crawler source type
Data stores

Data store
S3: s3://dmslab-stu...

IAM Role
arn:aws:iam::341259728059:role/service-role/dmslab-student-GlueLabRole-ZOQDII7JTBUM

Schedule

Output

Review all steps

Choose an IAM role

The IAM role allows the crawler to run and access your Amazon S3 data stores. [Learn more](#)

Update a policy in an IAM role
 Choose an existing IAM role
 Create an IAM role

IAM role [?](#)
dmslab-student-GlueLabRole-ZOQDII7JTBUM

This role must provide permissions similar to the AWS managed policy, **AWSGlueServiceRole**, plus access to your data stores.

• s3://dmslab-student-dmslabs3bucket-wot4bf73cw3

You can also create an IAM role on the [IAM console](#).

[Back](#) [Next](#)

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

Add crawler

Crawler info
glue-lab-crawler

Crawler source type
Data stores

Data store
S3: s3://dmslab-stu...

IAM Role
arn:aws:iam::341259728059:role/service-role/dmslab-student-GlueLabRole-ZOQDII7JTBUM

Schedule
Run on demand

Output

Review all steps

Create a schedule for this crawler

Frequency
Run on demand

[Back](#) [Next](#)

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

Add crawler

Crawler info
glue-lab-crawler

Crawler source type
Data stores

Data store
S3: s3://dmslab-stu...

IAM Role
arn:aws:iam::341259728059:role/service-role/dmslab-student-GlueLabRole-ZOQDII7JTBUM

Schedule
Run on demand

Output

Review all steps

Configure the crawler's output

Database [?](#)
Choose a database to contain tables

Add database

Prefix added to tables (optional) [?](#)
Type a prefix added to table names

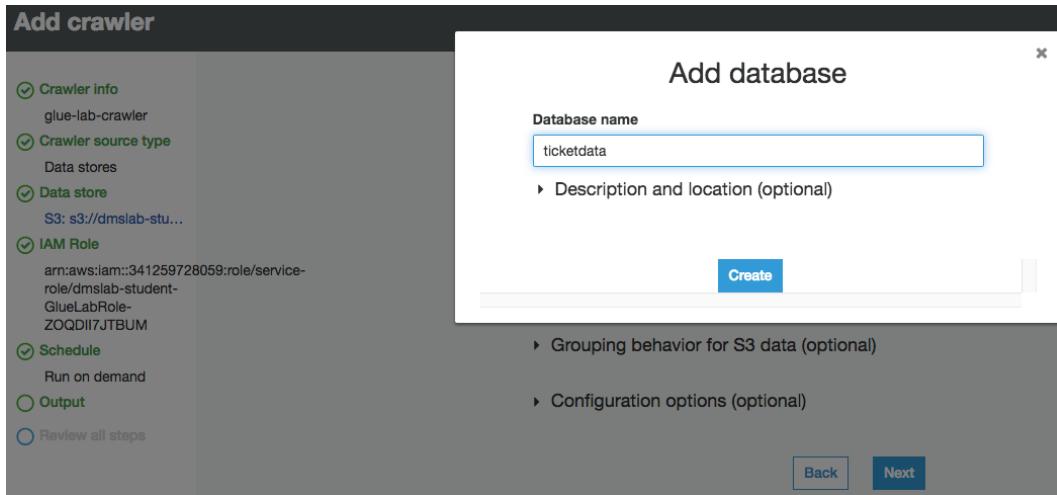
► Grouping behavior for S3 data (optional)

► Configuration options (optional)

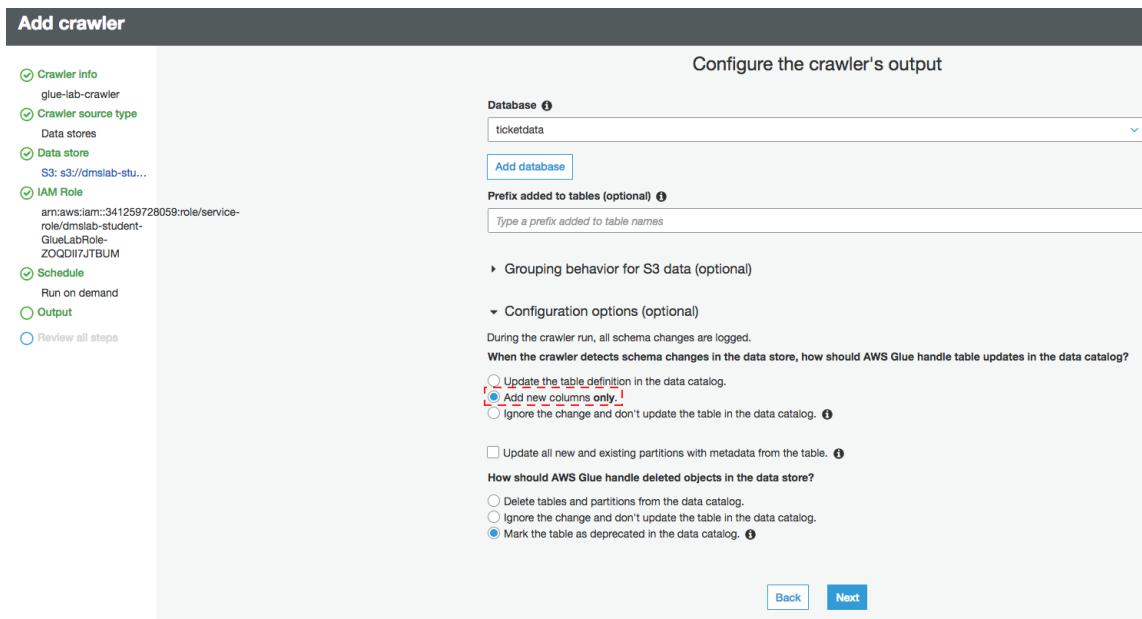
[Back](#) [Next](#)

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

Lab 2. ETL with AWS Glue

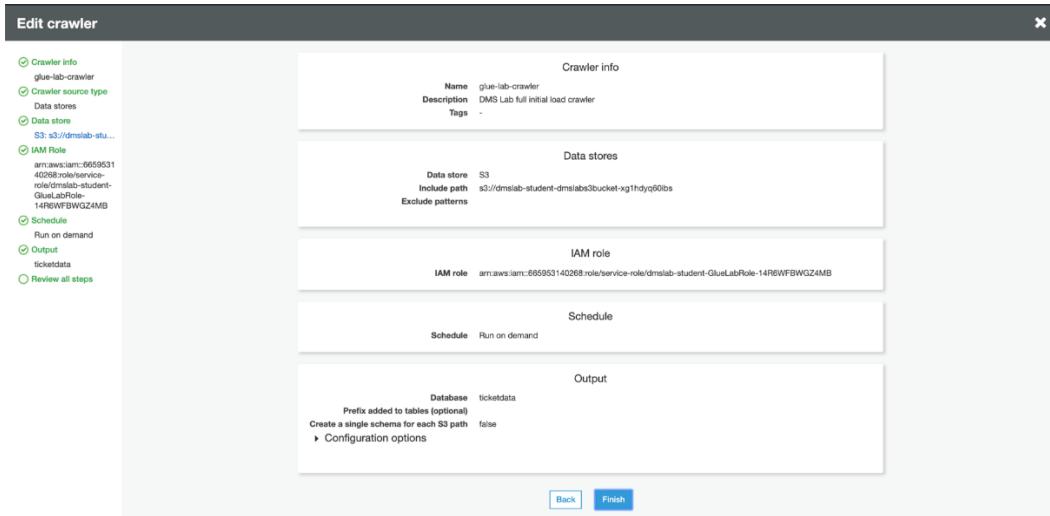


15. For **Prefix added to tables (optional)**, leave the field empty.
16. For **Configuration options (optional)**, select **Add new columns only** and keep the remaining default configuration options and Click **Next**.



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

Lab 2. ETL with AWS Glue



18. Click **Run it now**.

Name	Schedule	Catalog type	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
glue-lab-crawler		Glue	Ready		0 secs	0 secs	0	0

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.

Name	Schedule	Catalog type	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
glue-lab-crawler		Glue	Ready	Logs	1 min	1 min	0	15

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

Lab 2. ETL with AWS Glue

The screenshot shows the AWS Glue Data Catalog interface. On the left, there's a sidebar with navigation links for Services, Resource Groups, IAM, Athena, Lambda, S3, AWS Glue, and Support. Under the 'Data catalog' heading, 'Tables' is selected. The main area displays a table titled 'Tables' with columns: Name, Database, Location, Classification, Last updated, and Deprecated. There are 15 entries listed, including 'mb_data', 'name_data', 'nfl_data', 'nfl_stadium_data', 'person', etc. The 'person' table is highlighted with a blue border.

Name	Database	Location	Classification	Last updated	Deprecated
mb_data	ticketdata	s3://dmslab-student-dmst...	csv	10 January 2020 1:37 PM ...	
name_data	ticketdata	s3://dmslab-student-dmst...	csv	10 January 2020 1:37 PM ...	
nfl_data	ticketdata	s3://dmslab-student-dmst...	csv	10 January 2020 1:37 PM ...	
nfl_stadium_data	ticketdata	s3://dmslab-student-dmst...	csv	10 January 2020 1:37 PM ...	
person	ticketdata	s3://dmslab-student-dmst...	csv	10 January 2020 1:48 PM ...	
player	ticketdata	s3://dmslab-student-dmst...	csv	10 January 2020 1:37 PM ...	
seat	ticketdata	s3://dmslab-student-dmst...	csv	10 January 2020 1:37 PM ...	
seat_type	ticketdata	s3://dmslab-student-dmst...	csv	10 January 2020 1:37 PM ...	
sport_division	ticketdata	s3://dmslab-student-dmst...	csv	10 January 2020 1:37 PM ...	
sport_league	ticketdata	s3://dmslab-student-dmst...	csv	10 January 2020 1:37 PM ...	
sport_location	ticketdata	s3://dmslab-student-dmst...	csv	10 January 2020 1:37 PM ...	
sport_team	ticketdata	s3://dmslab-student-dmst...	csv	10 January 2020 1:37 PM ...	
sporting_event	ticketdata	s3://dmslab-student-dmst...	csv	10 January 2020 1:37 PM ...	
sporting_event_ticket	ticketdata	s3://dmslab-student-dmst...	csv	10 January 2020 1:37 PM ...	
ticket_purchase_hist	ticketdata	s3://dmslab-student-dmst...	csv	10 January 2020 1:37 PM ...	

Data Validation Exercise

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

This screenshot shows the same AWS Glue Data Catalog interface as the previous one, but the 'person' table is now highlighted with a red box. The rest of the table structure remains the same.

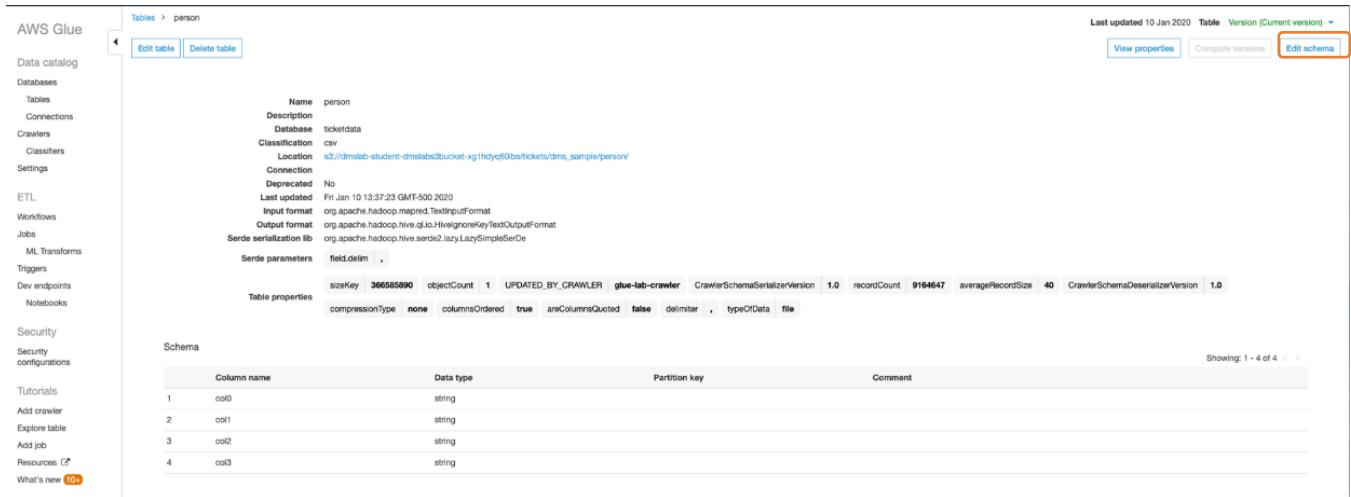
Name	Database	Location	Classification	Last updated	Deprecated
mb_data	ticketdata	s3://dmslab-student-dmst...	csv	10 January 2020 1:37 PM UTC-5	
name_data	ticketdata	s3://dmslab-student-dmst...	csv	10 January 2020 1:37 PM UTC-5	
nfl_data	ticketdata	s3://dmslab-student-dmst...	csv	10 January 2020 1:37 PM UTC-5	
nfl_stadium_data	ticketdata	s3://dmslab-student-dmst...	csv	10 January 2020 1:37 PM UTC-5	
person	ticketdata	s3://dmslab-student-dmst...	csv	10 January 2020 1:48 PM UTC-5	
player	ticketdata	s3://dmslab-student-dmst...	csv	10 January 2020 1:37 PM UTC-5	
seat	ticketdata	s3://dmslab-student-dmst...	csv	10 January 2020 1:37 PM UTC-5	
seat_type	ticketdata	s3://dmslab-student-dmst...	csv	10 January 2020 1:37 PM UTC-5	
sport_division	ticketdata	s3://dmslab-student-dmst...	csv	10 January 2020 1:37 PM UTC-5	
sport_league	ticketdata	s3://dmslab-student-dmst...	csv	10 January 2020 1:37 PM UTC-5	

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.

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

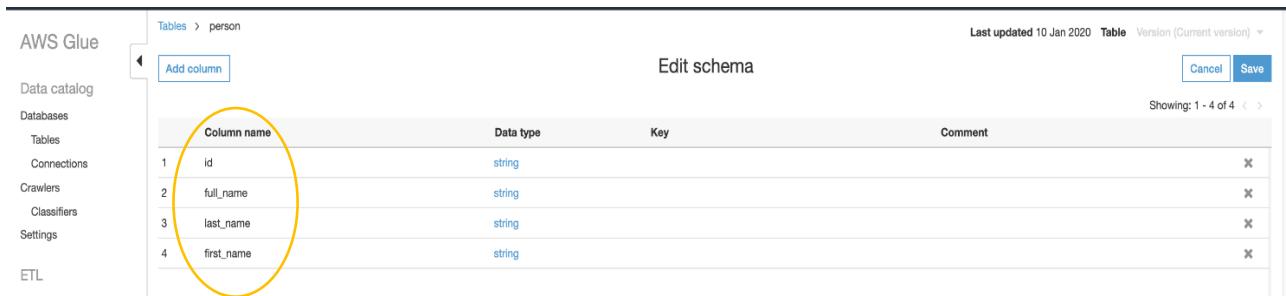
Lab 2. ETL with AWS Glue



The screenshot shows the AWS Glue Table Editor interface. On the left, there's a navigation sidebar with various options like Data catalog, Databases, Crawlers, Classifiers, Settings, ETL, Workflows, Jobs, ML Transforms, Triggers, Dev endpoints, Notebooks, Security, and Tutorials. The main area shows a table named 'person'. The 'Edit schema' button is highlighted with a red box at the top right of the table properties section.

Column name	Data type	Partition key	Comment
1 col0	string		
2 col1	string		
3 col2	string		
4 col3	string		

3. In the Edit Schema section, double-click **col0** (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.



The screenshot shows the 'Edit schema' dialog box for the 'person' table. The 'Column name' column has its first row ('1 id') circled in yellow. The dialog includes a 'Cancel' and 'Save' button at the top right. The table structure is identical to the one in the previous screenshot:

Column name	Data type	Key	Comment
1 id	string		
2 full_name	string		
3 last_name	string		
4 first_name	string		

5. Click **Save**.

Data ETL Exercise

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

Lab 2. ETL with AWS Glue

The screenshot shows the AWS Glue console. On the left, there's a navigation sidebar with sections for Data catalog, Databases, Tables, Connections, Crawlers, Classifiers, ETL, Jobs (which is selected and highlighted in orange), Triggers, and Dev endpoints. The main area is titled "Jobs" with a sub-instruction: "A job is your business logic required to perform extract, transform and load (ETL) work. Job runs are initiated by triggers which can be scheduled or driven by events." Below this is a search bar with "Add job" and "Action" buttons, and a filter bar with a search icon and placeholder "Filter by attributes". A message says "Showing: 0 - 0" with icons for refresh, sort, and details. The main table has columns: Name, ETL language, Script location, Last modified, and Job bookmark. It displays a single row with a small icon, the message "You don't have any jobs defined yet.", and a blue "Add job" button.

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.)

The screenshot shows the "Add job" dialog box. At the top, it says "Configure the job properties". The left sidebar lists "Job properties", "Data source", "Transform type", "Data target", and "Schemas". The main form fields include:

- Name:** Glue-Lab-SportTeamParquet
- IAM role:** Ensure this role has permission to your Amazon S3 resources, targets, temporary directory, scripts, and any libraries used by the job.
- Type:** Spark
- Glue version:** Spark 2.4, Python 3 (Glue version 1.0)
- This job runs:** A proposed script generated by AWS Glue
- Script file name:** Glue-Lab-SportTeamParquet
- S3 path where the script is stored:** s3://aws-glue-scripts-660953140268-us-east-1/demo_user
- Temporary directory:** s3://aws-glue-temporary-660953140268-us-east-1/demo_user

 At the bottom right is a "Next" button.

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

Lab 2. ETL with AWS Glue

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**.

Lab 2. ETL with AWS Glue

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**.

The screenshot shows the AWS Glue 'Add job' interface. On the left, under 'Job properties', there are several checked options: 'Glue-Lab-SportTeamParquet', 'Data source sport_team', 'Transform type Change schema', 'Data target s3://dmslab-studen...', and 'Schema'. The main area shows a table of columns and their mappings:

Source Column name	Data type	Map to target
id	string	id
name	string	name
abbreviated_name	string	abbreviated_name
home_field_id	bigint	home_field_id
sport_type_name	string	sport_type_name
sport_league_short_name	string	sport_league_short_name
sport_division_short_name	string	sport_division_short_name

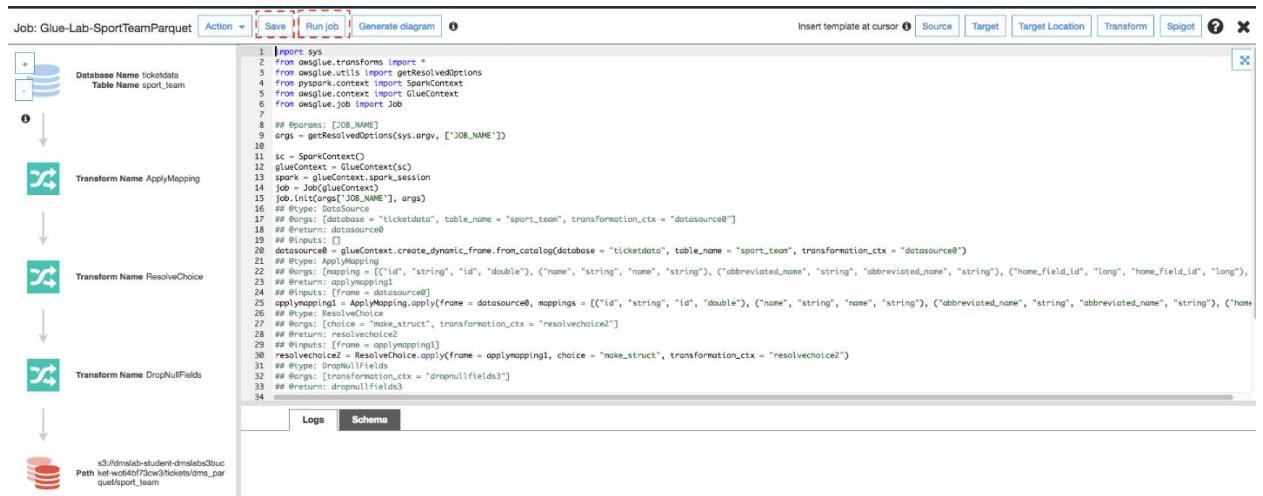
A modal window titled 'String type' is open over the table, focusing on the 'id' row. The 'Column type' dropdown is set to 'double' and has a red dashed box around it. A blue 'Update' button is at the bottom of the modal.

Below the table, another section titled 'Map the source columns to target columns.' shows the same mappings with arrows indicating the flow from source to target. At the bottom right of this section are 'Back' and 'Save job and edit script' buttons.

12. Click **Save job and edit script**.

13. View the job. (This screen provides you with the ability to customize this script as required.)

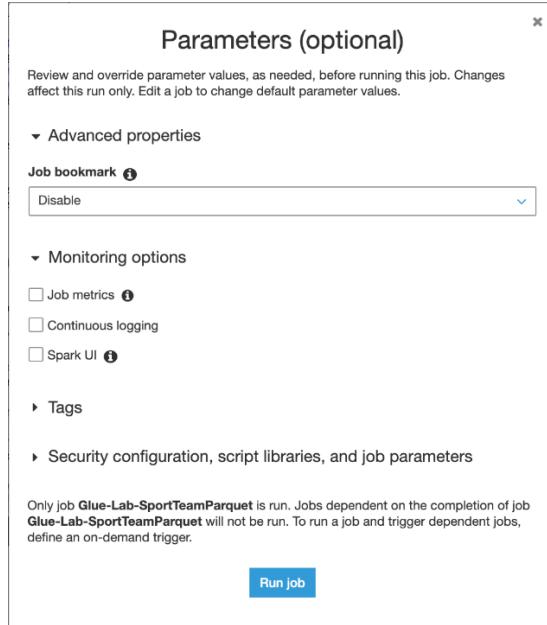
Click **Save** and then **Run Job**.



14. In **Parameters** option,

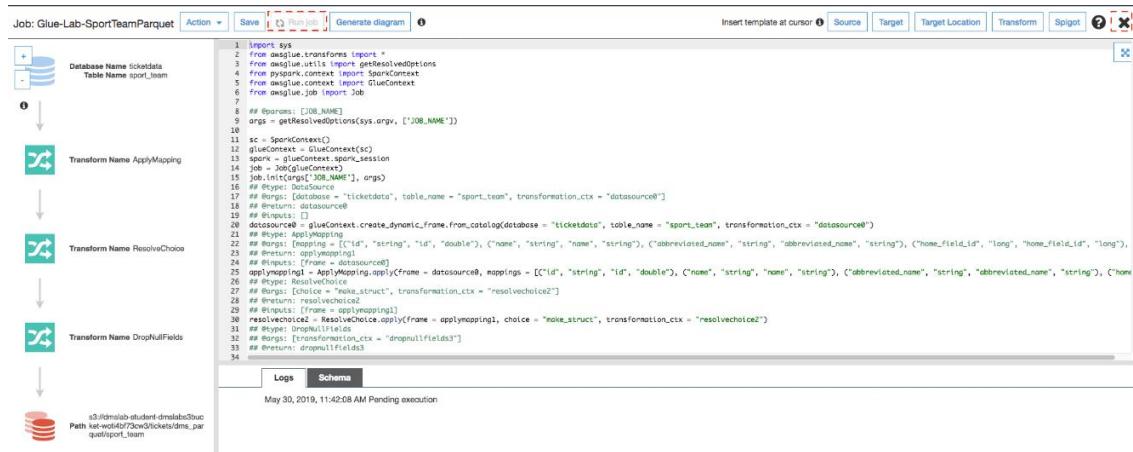
- you can leave **Job bookmark** as **Disable**. AWS Glue tracks data that has already been processed during a previous run of an ETL job by persisting state information from the job run.
- You can leave the **Job metrics** option **Unchecked**. You can collect metrics about AWS Glue jobs and visualize them on the AWS Glue with job metrics.

Lab 2. ETL with AWS Glue



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.

User preferences														
Add job	Action	Filter by tags and attributes	Showing: 1 - 11 < > ⌂ ⌂											
Name	Type	ETL language	Script location	Last modified	Job bookmark									
Glue-Lab-SportTeamParquet	Spark	python	s3://aws-glue-scri...	11 March 2020 3:17 PM UTC-7	Disable									
History		Metrics			Logs									
View run metrics		Rewind job bookmark			Showing: 1 - 1 < > ⌂ ⌂									
Run ID	Retry attempt	Run status	Error	Logs	Error logs	Glue version	Maximum capacity	Triggered by	Start time	End time	Execution time	Timeout	Delay	Job run input
jr_e37b56aa03cd3...	-	Running	Logs	Error logs		1.0	5		11 Mar...	0 secs	2880 mins			s3://aws-glue-tem...

Lab 2. ETL with AWS Glue

(Optional) If you plan to continue with other labs outside of this event (for example, the [Athena lab](#)), 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 section "Create Glue Crawler for Parquet Files"

To enable us to join data, we will also update the target data types in the schema. If below **Table 1** is indicating need Schema changes as "Yes". Refer **Table 2** find out column which need to changes 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

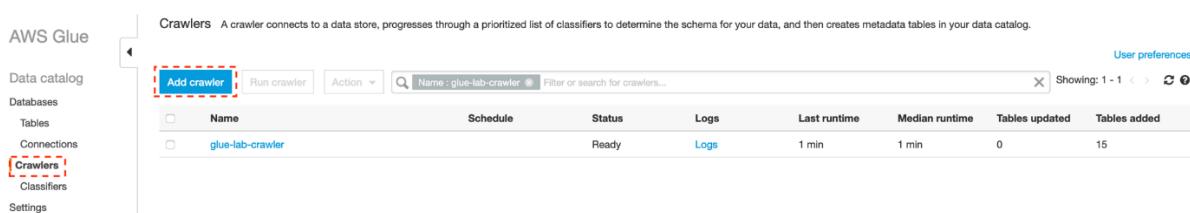
Table 2:

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**.

Lab 2. ETL with AWS Glue

Add crawler

Add information about your crawler

Crawler name: glue-lab-parquet-crawler

Tags, description, security configuration, and classifiers (optional)

Next

Crawler info
glue-lab-parquet-crawler

Crawler source type
Data stores

Data store
S3: s3://dmslab-stu...

IAM Role

Schedule

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**.

Add a data store

Choose a data store: S3

Crawl data in:
 Specified path in my account
 Specified path in another account

Include path: s3://dmslab-student-dmslabs3bucket-wotl4bf73cw3/tickets/dms_parquet

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

Exclude patterns (optional)

Back Next

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

Add crawler

Crawler info
glue-lab-parquet-crawler

Crawler source type
 Data stores
 Data store
S3: s3://dmslab-stu...
 IAM Role

Add another data store
 Yes
 No

Chosen data stores
S3: s3://dmslab-stu...

Back 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**.

Lab 2. ETL with AWS Glue

Add crawler

Choose an IAM role

The IAM role allows the crawler to run and access your Amazon S3 data stores. [Learn more](#)

Update a policy in an IAM role
 Choose an existing IAM role
 Create an IAM role

IAM role [?](#)
 dmslab-student-GlueLabRole-14R6WFBWGZ4MB

This role must provide permissions similar to the AWS managed policy, [AWSGlueServiceRole](#), plus access to your data stores.

- s3://dmslab-student-dmslabs3bucket-xg1hdq60bs/tickets/dms_parquet

You can also create an IAM role on the [IAM console](#).

[Back](#) [Next](#)

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

Add crawler

Create a schedule for this crawler

Frequency
 Run on demand

[Back](#) [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_**"

Add crawler

Configure the crawler's output

Database [?](#)
 ticketdata

[Add database](#)

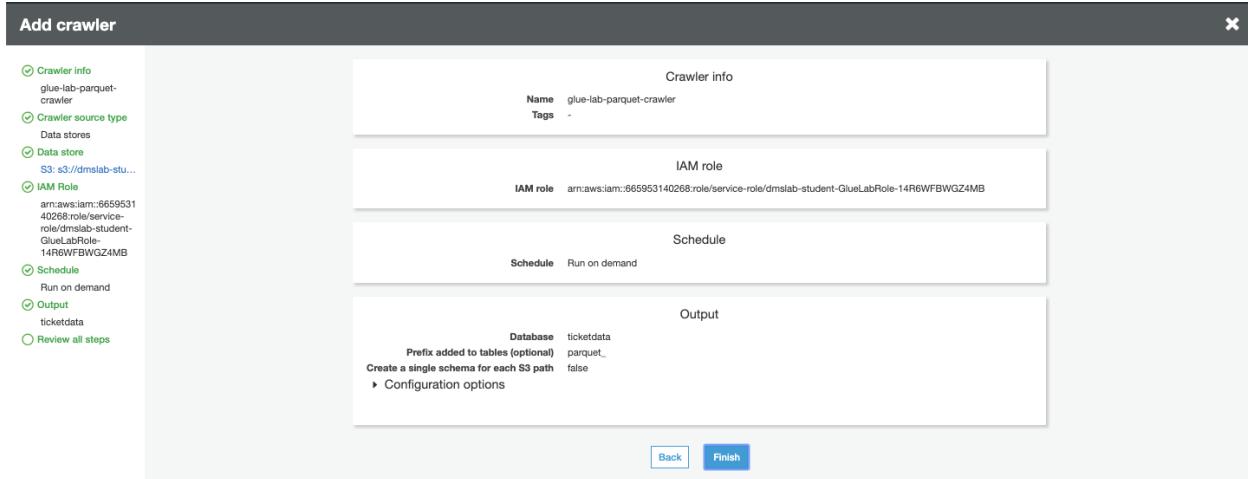
Prefix added to tables (optional) [?](#)
 parquet_

▶ Grouping behavior for S3 data (optional)
 ▶ Configuration options (optional)

[Back](#) [Next](#)

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

Lab 2. ETL with AWS Glue



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

Name	Schedule	Catalog type	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
glue-lab-cdc-cra...		Glue	Ready	Logs	1 min	1 min	0	2
glue-lab-crawler		Glue	Ready	Logs	1 min	1 min	0	15
glue-lab-parquet...		Glue	Ready	Logs	1 min	1 min	0	5

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.

Name	Database	Location	Classification	Last updated	Deprecated
mb_data	ticketdata	s3://dmslab-student-dmslabs3buck... csv		10 January 2020 1:37 PM UTC-5	
name_data	ticketdata	s3://dmslab-student-dmslabs3buck... csv		10 January 2020 1:37 PM UTC-5	
nfl_data	ticketdata	s3://dmslab-student-dmslabs3buck... csv		10 January 2020 1:37 PM UTC-5	
nfl_stadium_data	ticketdata	s3://dmslab-student-dmslabs3buck... csv		10 January 2020 1:37 PM UTC-5	
parquet_person	ticketdata	s3://dmslab-student-dmslabs3buck... parquet		23 January 2020 1:49 PM UTC-5	
parquet_person_annotation	ticketdata	s3://dmslab-student-dmslabs3buck... parquet		23 January 2020 1:49 PM UTC-5	
parquet_sport_team	ticketdata	s3://dmslab-student-dmslabs3buck... parquet		23 January 2020 1:49 PM UTC-5	
parquet_sporting_event	ticketdata	s3://dmslab-student-dmslabs3buck... parquet		23 January 2020 1:49 PM UTC-5	
parquet_sporting_event_ticket	ticketdata	s3://dmslab-student-dmslabs3buck... parquet		23 January 2020 1:49 PM UTC-5	
person	ticketdata	s3://dmslab-student-dmslabs3buck... csv		10 January 2020 1:48 PM UTC-5	

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**"

Lab 2. ETL with AWS Glue

7. Click **Next**.

Add crawler

Crawler info
glue-lab-cdc-crawler

Crawler source type
Data stores

Data store

IAM Role

Schedule

Output

Review all steps

Add a data store

Choose a data store
S3

Crawl data in
 Specified path in my account
 Specified path in another account

Include path
s3://dmslab-student-dmslabs3bucket-wotl4bf73cw3/cdc/dms_sample

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

Exclude patterns (optional)

Back **Next**

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

Add crawler

Crawler info
glue-lab-cdc-crawler

Crawler source type
Data stores

Data store
S3: s3://dmslab-stu...

IAM Role

Schedule

Output

Review all steps

Add another data store

Yes
 No

Back **Next**

Chosen data stores
S3: s3://dmslab-stu...

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**.

Add crawler

Crawler info
glue-lab-cdc-crawler

Crawler source type
Data stores

Data store
S3: s3://dmslab-stu...

IAM Role

Schedule

Output

Review all steps

Choose an IAM role

The IAM role allows the crawler to run and access your Amazon S3 data stores. [Learn more](#)

Update a policy in an IAM role
 Choose an existing IAM role
 Create an IAM role

IAM role [?](#)
dmslab-student-GlueLabRole-14R6WFBWGZ4MB

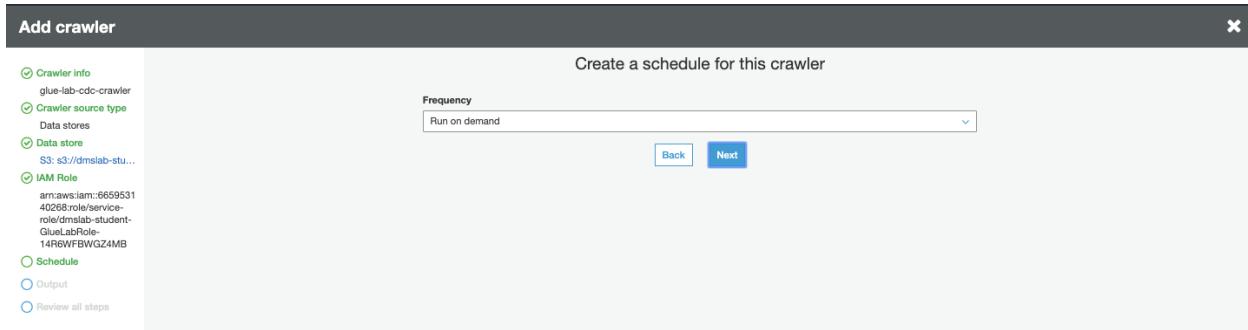
This role must provide permissions similar to the AWS managed policy, **AWSGlueServiceRole**, plus access to your data stores.
s3://dmslab-student-dmslabs3bucket-xg1hydny60lbs/cdc/dms_sample

You can also create an IAM role on the [IAM console](#).

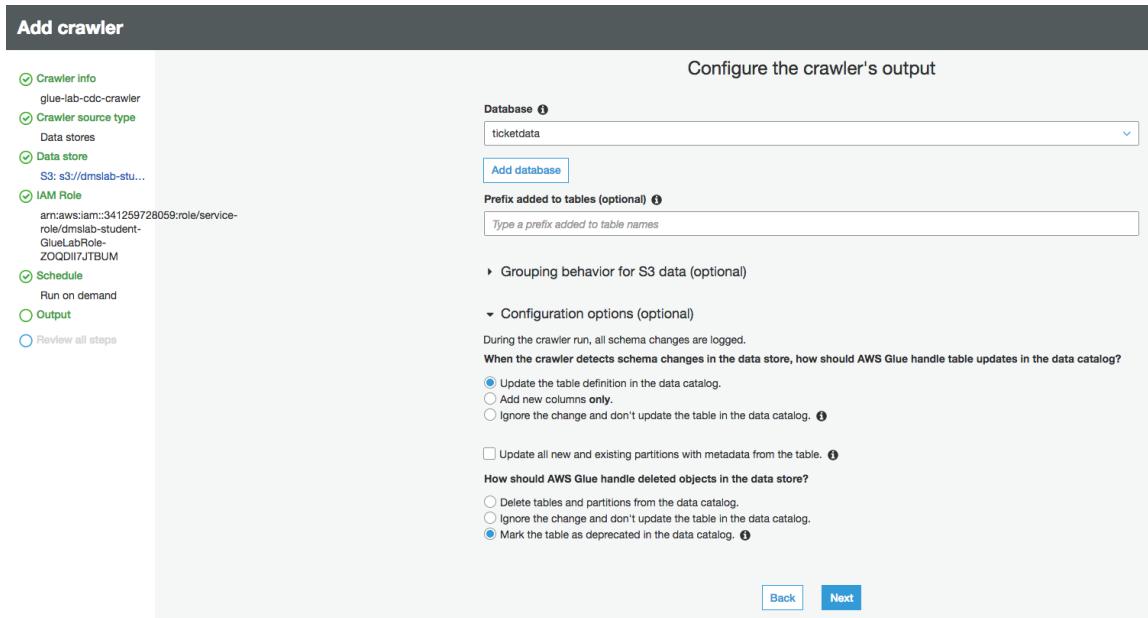
Back **Next**

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

Lab 2. ETL with AWS Glue

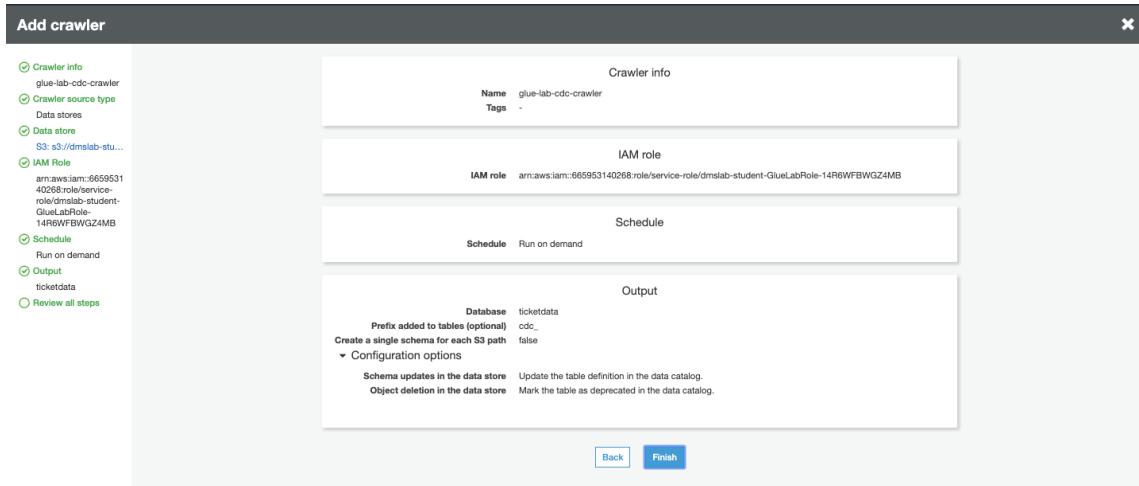


12. On the Configure the crawler's output page, select the existing **Database** for crawler output (e.g., "ticketdata").
13. For **Prefix added to tables (optional)**, specify "cdc_"
14. For Configuration options (optional), keep the default selections and click **Next**.



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

Lab 2. ETL with AWS Glue



16. Click Run it now.

The screenshot shows the AWS Glue console under the 'Crawlers' section. A message box displays 'Crawler glue-lab-cdc-crawler was created to run on demand.' with a red border around the 'Run it now?' button. Below is a table listing crawlers:

Name	Schedule	Catalog type	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
glue-lab-cdc-cra...		Glue	Ready		0 secs	0 secs	0	0
glue-lab-crawler		Glue	Ready	Logs	1 min	1 min	0	15

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.

The screenshot shows the AWS Glue console under the 'Crawlers' section. A message box displays 'Crawler "glue-lab-cdc-crawler" completed and made the following changes: 2 tables created, 0 tables updated. See the tables created in database ticketdata.' Below is a table listing crawlers:

Name	Schedule	Catalog type	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
glue-lab-cdc-cra...		Glue	Ready	Logs	1 min	1 min	0	2
glue-lab-crawler		Glue	Ready	Logs	1 min	1 min	0	15

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

Lab 2. ETL with AWS Glue

Tables						
Name	Database	Location	Classification	Last updated	Deprecated	
cdc_sporting_event_ticket	ticketdata	s3://dmslab-student-dmslabs3buck...	csv	23 January 2020 4:38 PM UTC-5		
cdc_ticket_purchase_hist	ticketdata	s3://dmslab-student-dmslabs3buck...	csv	23 January 2020 4:38 PM UTC-5		
mlb_data	ticketdata	s3://dmslab-student-dmslabs3buck...	csv	10 January 2020 1:37 PM UTC-5		
name_data	ticketdata	s3://dmslab-student-dmslabs3buck...	csv	10 January 2020 1:37 PM UTC-5		
nfl_data	ticketdata	s3://dmslab-student-dmslabs3buck...	csv	10 January 2020 1:37 PM UTC-5		
nfl_stadium_data	ticketdata	s3://dmslab-student-dmslabs3buck...	csv	10 January 2020 1:37 PM UTC-5		
parquet_person	ticketdata	s3://dmslab-student-dmslabs3buck...	parquet	23 January 2020 1:49 PM UTC-5		
parquet_sport_location	ticketdata	s3://dmslab-student-dmslabs3buck...	parquet	23 January 2020 1:49 PM UTC-5		
parquet_sport_team	ticketdata	s3://dmslab-student-dmslabs3buck...	parquet	23 January 2020 1:49 PM UTC-5		

Step 2: Create a Glue Job with Bookmark Enabled

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

Jobs					
Name	Type	ETL language	Script location	Last modified	Job bookmark
Glue-Lab-TicketHistory-Parquet-with-bookmark					

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

Lab 2. ETL with AWS Glue

Add job

Configure the job properties

Name: Glue-Lab-TicketHistory-Parquet-with-bookmark

IAM role: drinstudent-GlueLabRole-1418WF9WSZAMB

Type: Spark

Glue version: Spark 2.4, Python 3 (Glue Version 1.0)

This job runs: A proposed script generated by AWS Glue

Script file name: Glue-Lab-TicketHist-Parquet-with-bookmark

S3 path where the script is stored: s3://aws-glue-scripts-66595314298-us-east-1/deashut1

Temporary directory: s3://aws-glue-temporary-66595314298-us-east-1/deashut1

Advanced properties

Job bookmark: Enable

Monitoring options

Log metrics

Continuous logging

Spark UI

Tags (optional)

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**

Add job

Choose a data source

Name	Database	Location	Classification
bookmark_parquet_ticket	ticketdata	s3://dmslab-student-dmslabs3bucket-xg1hydg0bs/cdc_bookmark/ticket...	parquet
cdc_sporting_event_ticket	ticketdata	s3://dmslab-student-dmslabs3bucket-xg1hydg0bs/cdc/dms_sample/sp...	csv
cdc_ticket_purchase_hist	ticketdata	s3://dmslab-student-dmslabs3bucket-xg1hydg0bs/cdc/dms_sample/tic...	csv
clickstream_data	processed-data	s3://rawdatastaged-deashut/Clickstream_data/	json
csv_clickstream_data	processed-data	s3://processed-deashut/clickstream-data/	csv

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

Add job

Choose a transform type

Change schema
Change schema of your source data and create a new target dataset

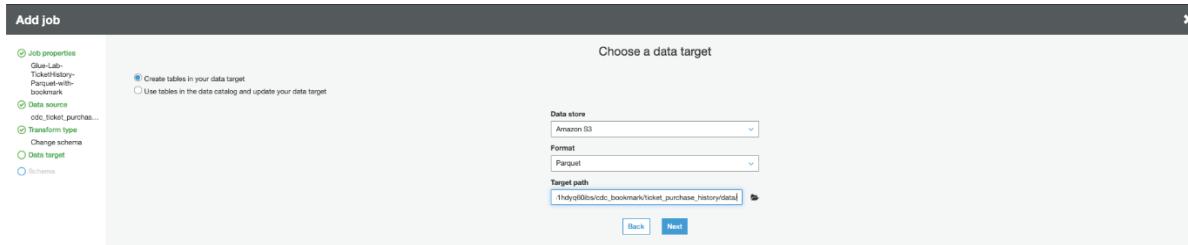
Find matching records
Use machine learning to find matching records within your source data

Back **Next**

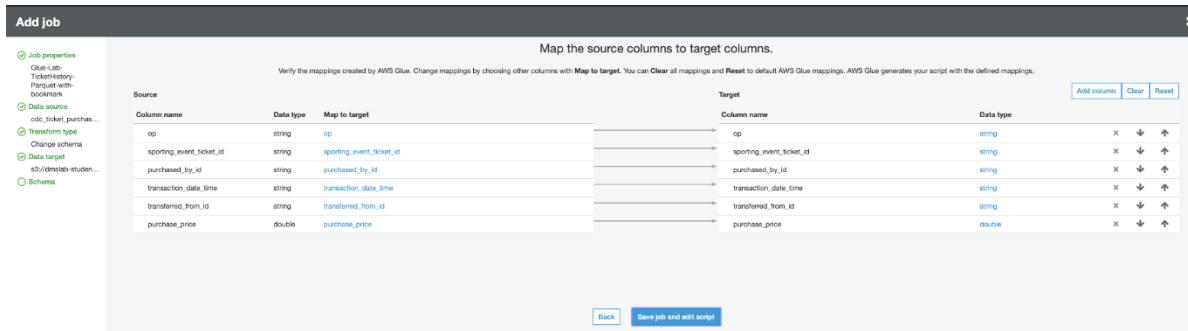
8. In Choose a data target:

- For **Data store:** select **amazon S3**
- Format: **parquet**
- Target path:** **s3://xxx-dmslabs3bucket-xxx/cdc_bookmark/ticket_purchase_history_data/**
- Click **Next**

Lab 2. ETL with AWS Glue



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



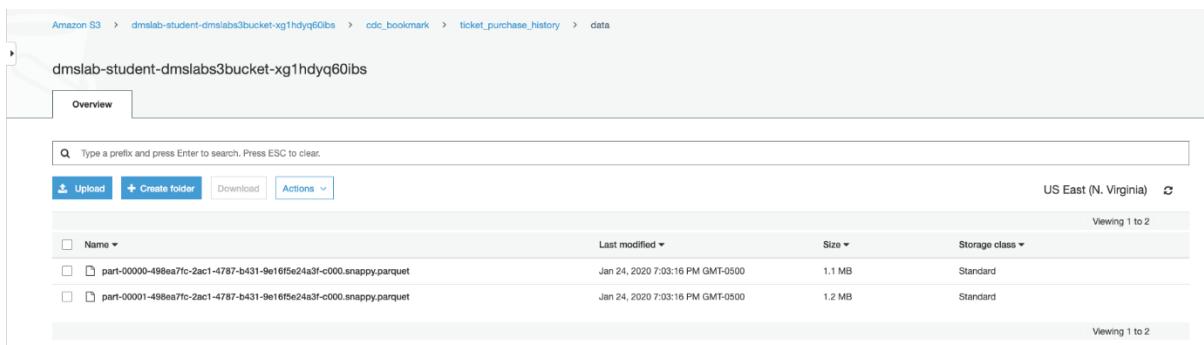
10. In the next window, review the job script and click on **Run job**. Click on close mark on the top right of the window to close the screen.

```

Job: Glue-Lab-TicketHistory-Parquet-with-bookmark Action: Save Run job Generate diagram
Insert template at cursor Source Target Target Location Transform Split X
Database Name ticketdata Table Name cdc_ticket_purchase_hist
Transform Name ApplyMapping
Transform Name ResolveChoice
Transform Name DropNullFields
Path s3://dmslab-student-dmslabs3bucket-xg1hydq60ibs/cdc_bookmark/ticket_purchase_history/data
  1. import sys
  2. from awsglue.transforms import *
  3. from awsglue.utils import getResolvedOptions
  4. from pyspark.context import SparkContext
  5. from awsglue.context import GlueContext
  6. from awsglue.job import Job
  7.
  8. ## Parameters: [JOB_NAME]
  9. args = getResolvedOptions(sys.argv, ['JOB_NAME'])
  10.
  11. sc = SparkContext()
  12. glueContext = GlueContext(sc)
  13. spark = glueContext.sparkSession
  14. job = Job(glueContext)
  15. job.init(args['JOB_NAME'], args)
  16. ## #use: DatabaseName
  17. ## #use: TableName
  18. ## #return: datasource
  19. ## #return: dataFrame
  20. datasource = glueContext.create_dynamic_frame.from_catalog(database = "ticketdata", table_name = "cdc_ticket_purchase_hist", transformation_ctx = "datasource0")
  21. ## #use: ApplyMapping
  22. ## #return: dataFrame
  23. ## #return: applymapping0
  24. ## #return: applymapping0
  25. applymapping0 = ApplyMapping.apply(frame = datasource0, mapping = [{"op": "string", "name": "op"}, {"sporting_event_ticket_id": "string", "name": "sporting_event_ticket_id"}, {"purchased_by_id": "string", "name": "purchased_by_id"}, {"transaction_date_time": "string", "name": "transaction_date_time"}, {"transferred_from_id": "string", "name": "transferred_from_id"}, {"purchase_price": "double", "name": "purchase_price"}], transformation_ctx = "applymapping0")
  26. ## #use: ResolveChoice
  27. ## #return: resolvechoice0
  28. ## #return: resolvechoice0
  29. ## #use: ResolveChoice
  30. resolvechoice0 = ResolveChoice.apply(frame = applymapping0, choice = "make_struct", transformation_ctx = "resolvechoice0")
  31. ## #return: resolvechoice0
  32. ## #use: DropNullFields
  33. ## #return: dropnullfields0
  34. ## #return: dropnullfields0
  35. dropnullfields0 = DropNullFields.apply(frame = resolvechoice0, transformation_ctx = "dropnullfields0")
  36. ## #return: dropnullfields0
  37. ## #use: connection_type
  38. ## #return: connection_type
  39. ## #use: connection_type
  40. datasource1 = glueContext.create_dynamic_frame.from_options(frame_options = "dropnullfields0", connection_type = "s3", connection_options = {"path": "s3://dmslab-student-dmslabs3bucket-xg1hydq60ibs/cdc_bookmark/ticket_purchase_history/data"}, format = "parquet", transformation_ctx = "datasink1")
  41. datasource1.write_parquet(path = "s3://dmslab-student-dmslabs3bucket-xg1hydq60ibs/cdc_bookmark/ticket_purchase_history/data")
  42. job.commit()

```

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

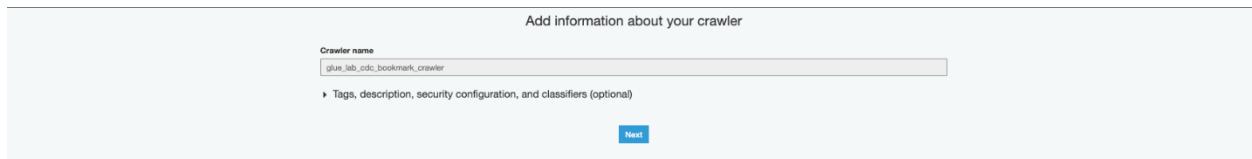


Lab 2. ETL with AWS Glue

Step 3: Create Glue crawler for Parquet data in S3

- 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.
- Click on **Add crawler**

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



Add information about your crawler

Crawler name
glue_lab_cdc_bookmark_crawler

Tags, description, security configuration, and classifiers (optional)

Next

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



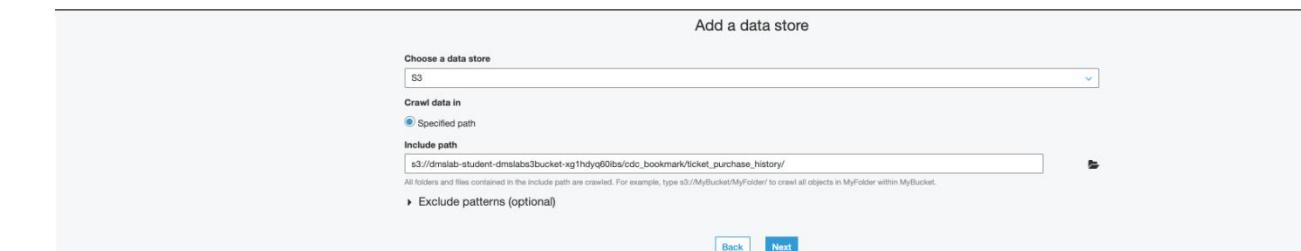
Specify crawler source type

Choose Existing catalog tables to specify catalog tables as the crawler source. The selected tables specify the data stores to crawl. This option doesn't support JDBC data stores.

Crawler source type
 Data stores
 Existing catalog tables

Back Next

- In **Add a data store**:
 - For **Choose a data store**, select **S3**
 - For the path, provide this: `s3://xxx-dmslabs3bucket-xxx` and append `/cdc_bookmark/ticket_purchase_history/`.
- Click on **Next**



Add a data store

Choose a data store
S3

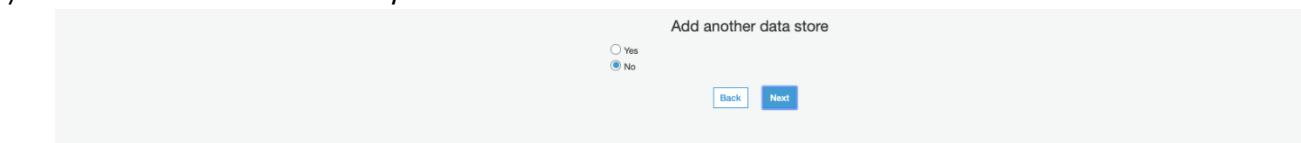
Crawl data in
 Specified path
Include path
`s3://dmslab-student-dmslabs3bucket-xg1hdyq60bs/cdc_bookmark/ticket_purchase_history/`

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

Exclude patterns (optional)

Back Next

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



Add another data store

Yes
 No

Back Next

- In **Choose an IAM role**, select Choose an existing IAM role and select the role that you created as part of the DMS_Student Lab. (for eg, this role name looks something like this: `dmslab-student-GlueLabRole-<random-alphanumeric-characters>`)

Lab 2. ETL with AWS Glue

The IAM role allows the crawler to run and access your Amazon S3 data stores. [Learn more](#)

Update a policy in an IAM role
 Choose an existing IAM role
 Create an IAM role

IAM role [?](#)
dmslab-student-GlueLabRole-14R8WF8WQZ4MB

This role must provide permissions similar to the AWS managed policy, [AWSGlueServiceRole](#), plus access to your data stores.

- s3://dmslab-student-dmlabs3bucket-xg1hdqg60bs/cdc_bookmark/ticket_purchase_history/

You can also create an IAM role on the [IAM console](#).

[Back](#) [Next](#)

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

Create a schedule for this crawler

Frequency
Run on demand

[Back](#) [Next](#)

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

Configure the crawler's output

Database [?](#)
ticketdata

Add database

Prefix added to tables (optional) [?](#)
bookmark_parquet_

Grouping behavior for S3 data (optional)
Configuration options (optional)

[Back](#) [Next](#)

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

Crawler info

Name: glue_lab_cdc_bookmark_crawler
Tags: -

Data stores

Data store: S3
Include path: s3://dmslab-student-dmlabs3bucket-xg1hdqg60bs/cdc_bookmark/ticket_purchase_history/
Exclude patterns: -

IAM role

IAM role: arn:aws:iam::665953140208:role/service-role/dmslab-student-GlueLabRole-14R8WF8WQZ4MB

Schedule

Schedule: Run on demand

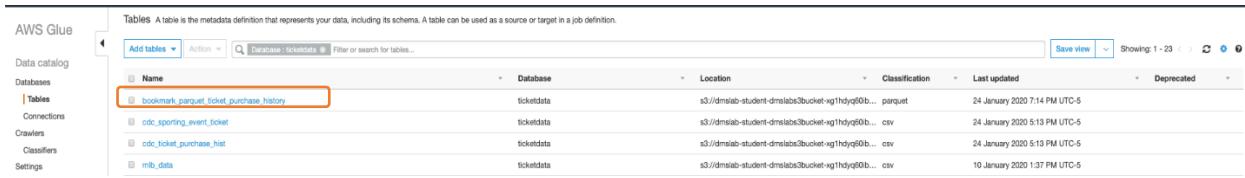
Output

Database: ticketdata
Prefix added to tables (optional): bookmark_parquet_
Create a single schema for each S3 path: false
Configuration options: -

[Back](#) [Finish](#)

- 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**"

Lab 2. ETL with AWS Glue



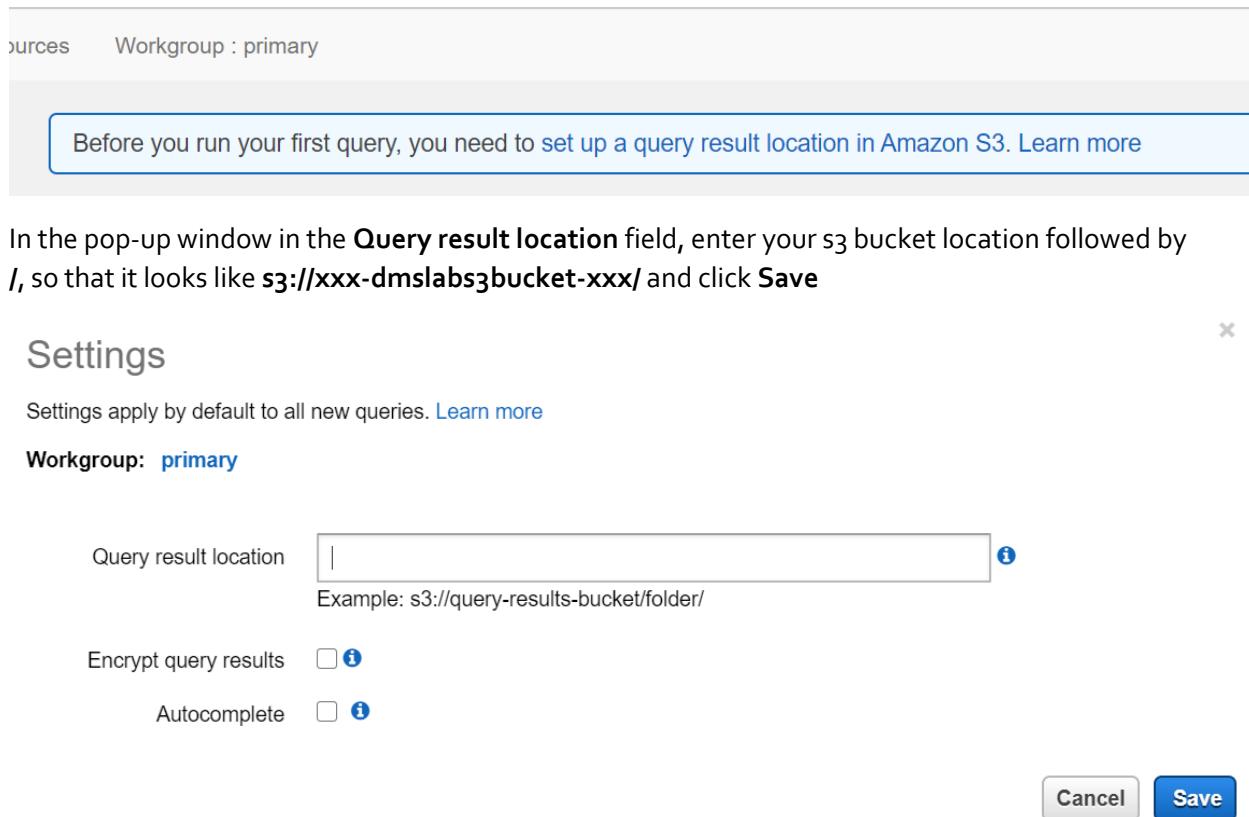
Name	Database	Location	Classification	Last updated	Deprecated
bookmark_parquet_ticket_purchase_history	ticketdata	s3://dmslab-student-dmrlabs3bucket-xg1hdygq0b...	parquet	24 January 2020 7:14 PM UTC-5	
cdc_sporting_event_ticket	ticketdata	s3://dmslab-student-dmrlabs3bucket-xg1hdygq0b...	csv	24 January 2020 5:13 PM UTC-5	
cdc_ticket_purchase_hist	ticketdata	s3://dmslab-student-dmrlabs3bucket-xg1hdygq0b...	csv	24 January 2020 5:13 PM UTC-5	
mb_data	ticketdata	s3://dmslab-student-dmrlabs3bucket-xg1hdygq0b...	csv	10 January 2020 1:37 PM UTC-5	

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



Before you run your first query, you need to set up a query result location in Amazon S3. Learn more

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**

Settings

Settings apply by default to all new queries. [Learn more](#)

Workgroup: [primary](#)

Query result location [?](#)
Example: `s3://query-results-bucket/folder/`

Encrypt query results [?](#)

Autocomplete [?](#)

Cancel **Save**

Lab 2. ETL with AWS Glue

To select some rows from the table, try running:

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

To get a row count, run:

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

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.

Name	Last modified	Size	Storage class
part-00000-0e030204-2fbc-4f12-0249-d9100b77edaa-c000.anappy.parquet	Jan 24, 2020 9:20:13 PM GMT-0500	9.3 KB	Standard
part-00000-0e030204-2fbc-4f12-0249-d9100b77edaa-c000.anappy.parquet	Jan 24, 2020 7:02:16 PM GMT-0500	1.1 MB	Standard
part-00000-d1666723-3158-459a-b8be-a65239452348-c000.anappy.parquet	Jan 25, 2020 11:24:20 PM GMT-0500	1.7 MB	Standard
part-00000-0e030204-2fbc-4f12-0249-d9100b77edaa-c000.anappy.parquet	Jan 25, 2020 10:24:27 PM GMT-0500	7.2 KB	Standard
part-00001-0e030204-2fbc-4f12-0249-d9100b77edaa-c000.anappy.parquet	Jan 24, 2020 9:20:13 PM GMT-0500	66.5 KB	Standard
part-00001-49fe37c2-2fbc-4f12-0249-d9100b77edaa-c000.anappy.parquet	Jan 24, 2020 7:02:16 PM GMT-0500	1.2 MB	Standard
part-00001-d1666723-3158-459a-b8be-a65239452348-c000.anappy.parquet	Jan 25, 2020 11:24:20 PM GMT-0500	1.7 MB	Standard
part-00002-0e030204-2fbc-4f12-0249-d9100b77edaa-c000.anappy.parquet	Jan 24, 2020 9:20:18 PM GMT-0500	1.7 MB	Standard
part-00002-d1666723-3158-459a-b8be-a65239452348-c000.anappy.parquet	Jan 25, 2020 11:24:19 PM GMT-0500	1.5 MB	Standard

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_data";
```

Lab 2. ETL with AWS Glue

To review the latest transactions, run:

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

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

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

Overview:

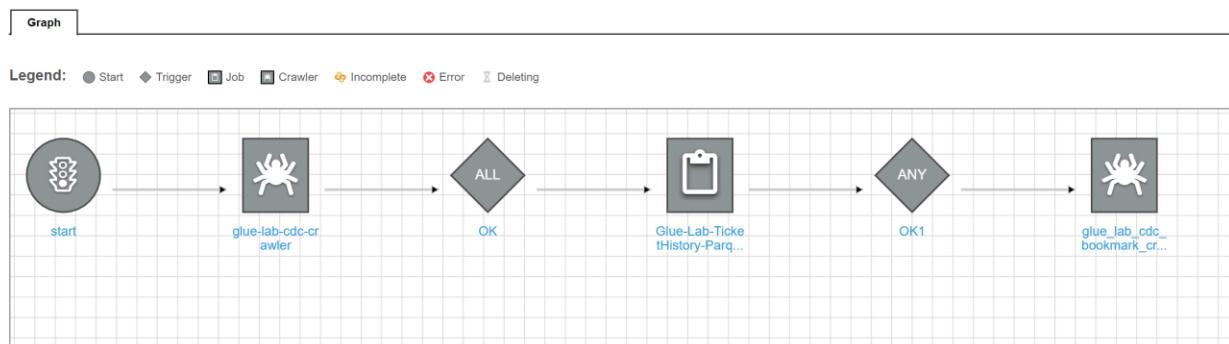
In AWS Glue, you can use workflows to create and visualize complex extract, transform, and load (ETL) activities involving multiple crawlers, jobs, and triggers. Each workflow manages the execution and monitoring of all its components. As a workflow runs each component, it records execution progress 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:

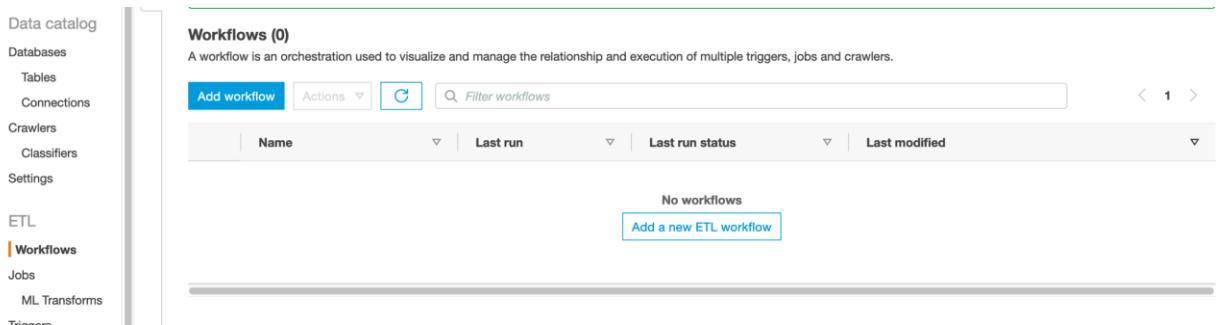
On-demand trigger -> glue-lab-cdc-crawler -> Glue-Lab-TicketHistory-Parquet-with-bookmark -> glue_lab_cdc_bookmark_crawler



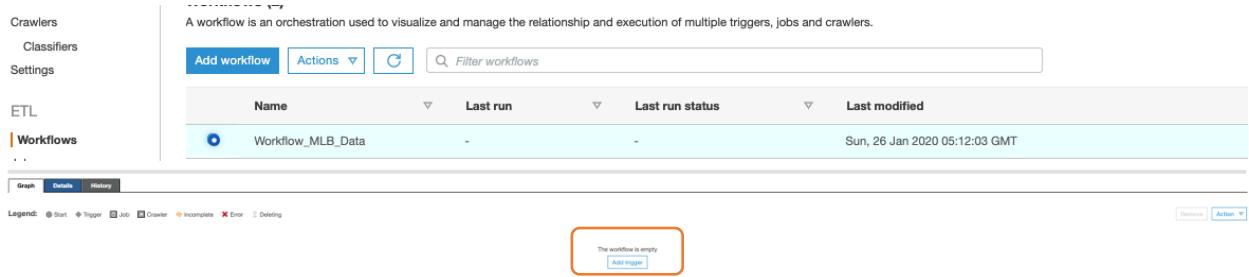
Lab 2. ETL with AWS Glue

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

Lab 2. ETL with AWS Glue

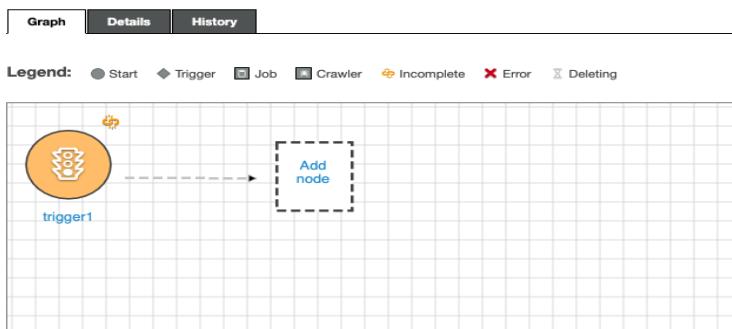
Add trigger

Name
trigger1

Description (optional)
Trigger to start the workflow

Trigger type
 Schedule Event On demand

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:
 - Name:** trigger2
 - Description:** Trigger to execute job
 - Trigger type:** Event
 - Trigger logic:** Start after ALL watched event. This will make sure that job starts once Glue Crawler finishes.
 - Click Add**

Lab 2. ETL with AWS Glue

The screenshot shows the 'Add trigger' dialog box. At the top, there are two buttons: 'Clone existing' and 'Add new'. The 'Add new' button is highlighted. Below it, the 'Name' field contains 'trigger2'. The 'Description (optional)' field has the text 'Trigger to execute crawler'. Under 'Trigger type', the 'Event' radio button is selected. In the 'Trigger logic' section, the 'Start after ANY watched event' radio button is selected. At the bottom right are 'Cancel' and 'Add' buttons, with 'Add' being the active button.

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"

The screenshot shows the AWS Glue Workflows console. On the left, there are tabs for 'Classifiers', 'Settings', 'ETL', and 'Workflows'. The 'Workflows' tab is selected. In the center, a table lists a single workflow named 'Workflow'. The 'Actions' column for this workflow has a dropdown menu open, showing options: 'Add workflow', 'Actions' (with 'Run' highlighted), 'Delete', and a 'C' icon. There is also a 'Edit' button. At the top right, there is a search bar with the placeholder 'Filter workflows'.

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