



# Amazon Web Services

## Data Engineering Immersion Day

---

Lab 2. ETL with AWS Glue

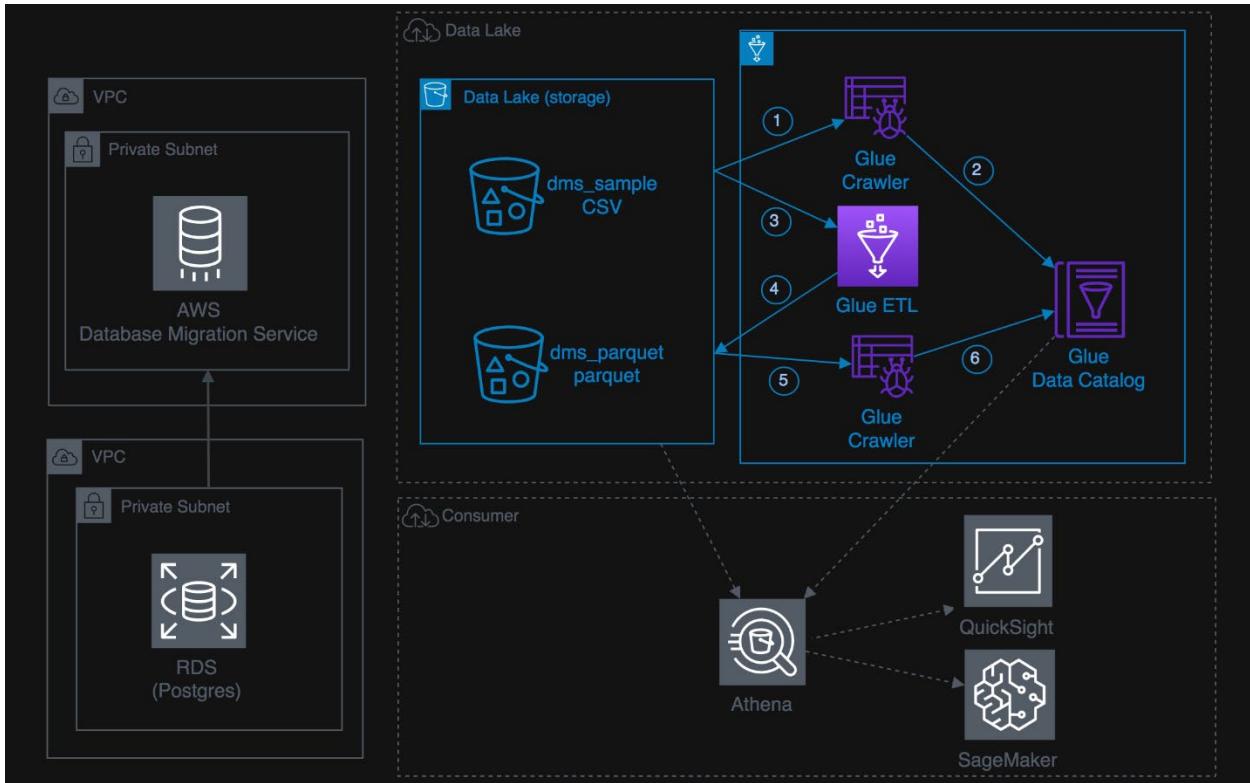
*July 2021*

## Table of Contents

<i>Introduction</i> .....	2
<i>Get Started Using the Lab Environment</i> .....	3
<i>PART A: Data Validation and ETL</i> .....	6
Create Glue Crawler for initial full load data .....	6
Data Validation Exercise .....	10
Data ETL Exercise .....	12
Create Glue Crawler for Parquet Files .....	19
<i>PART B: Glue Job Bookmark (Optional):</i> .....	23
Step 1: Create Glue Crawler for ongoing replication (CDC Data) .....	23
Step 2: Create a Glue Job with Bookmark Enabled .....	28
Step 3: Create Glue crawler for Parquet data in S3 .....	32
Step 4: Generate CDC data and to observe bookmark functionality .....	37
<i>PART C: Glue Workflows (Optional, self-paced)</i> .....	37
Overview: .....	38
Creating and Running Workflows:.....	38

## 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
2. Or complete Lab1. Copy Source Data

## 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/>

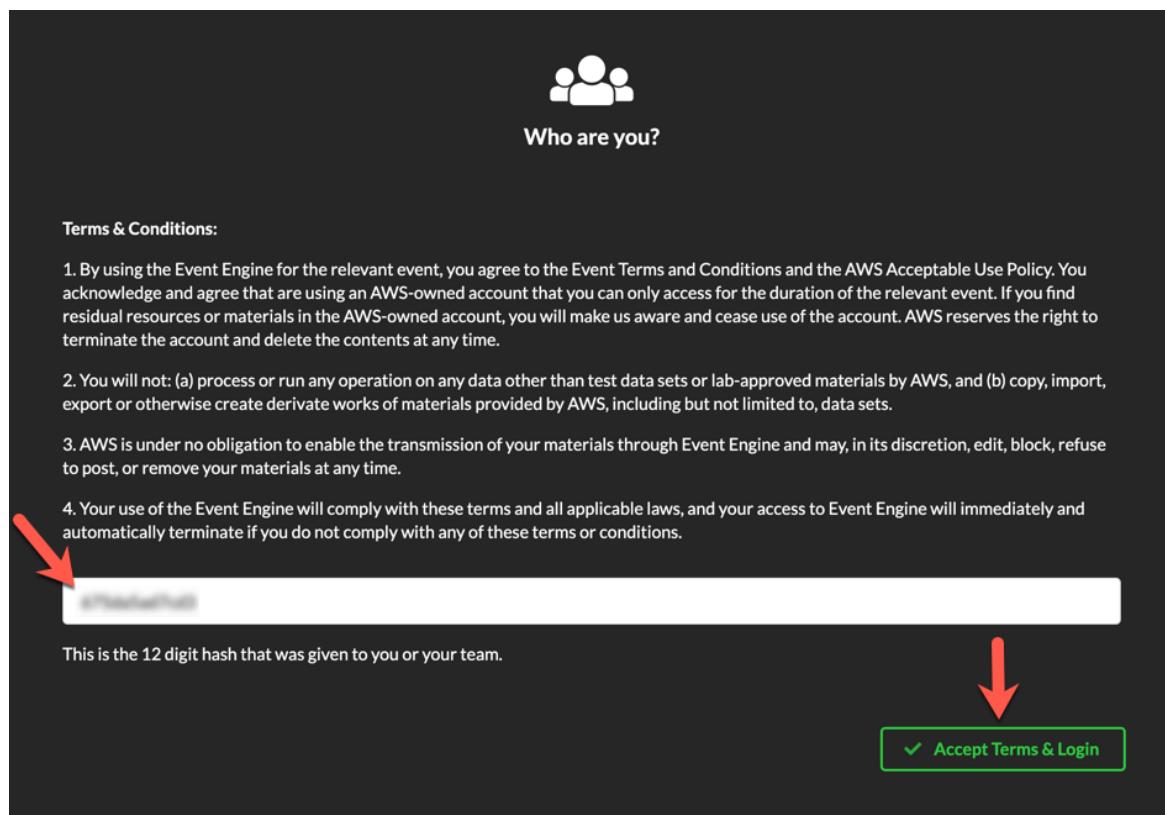
## Get Started Using the Lab Environment

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:

## Lab 2. ETL with AWS Glue

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

Outputs:  
No outputs defined

Readme

- S3 Bucket, IAM role for the Glue lab etc

Environment Setup

Outputs:  
**S3 Bucket name**  
mod-3fccddd609114925-dmslabs3bucket-1ngcgzzcnd15u  
**BusinessAnalystUser**  
mod-3fccddd609114925-BusinessAnalystUser-MB0XFZLQLOXX  
**DMSLabRoleS3 ARN**  
arn:aws:iam::377243295828:role/mod-3fccddd609114925-DMSLabRoleS3-O2VT1RSN43SG  
**Glue Lab Role**  
mod-3fccddd609114925-GlueLabRole-YLTJA13WW6WT  
**S3BucketWorkgroupA**  
mod-3fccddd609114925-s3bucketnetworkgroupa-tbon3m1mkunh  
**S3BucketWorkgroupB**  
mod-3fccddd609114925-s3bucketnetworkgroupb-18ygl8nfp8ead  
**WorkgroupManagerUser**  
mod-3fccddd609114925-WorkgroupManagerUser-5IVE0UQNIBG4

Readme

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

Team Dashboard

Event

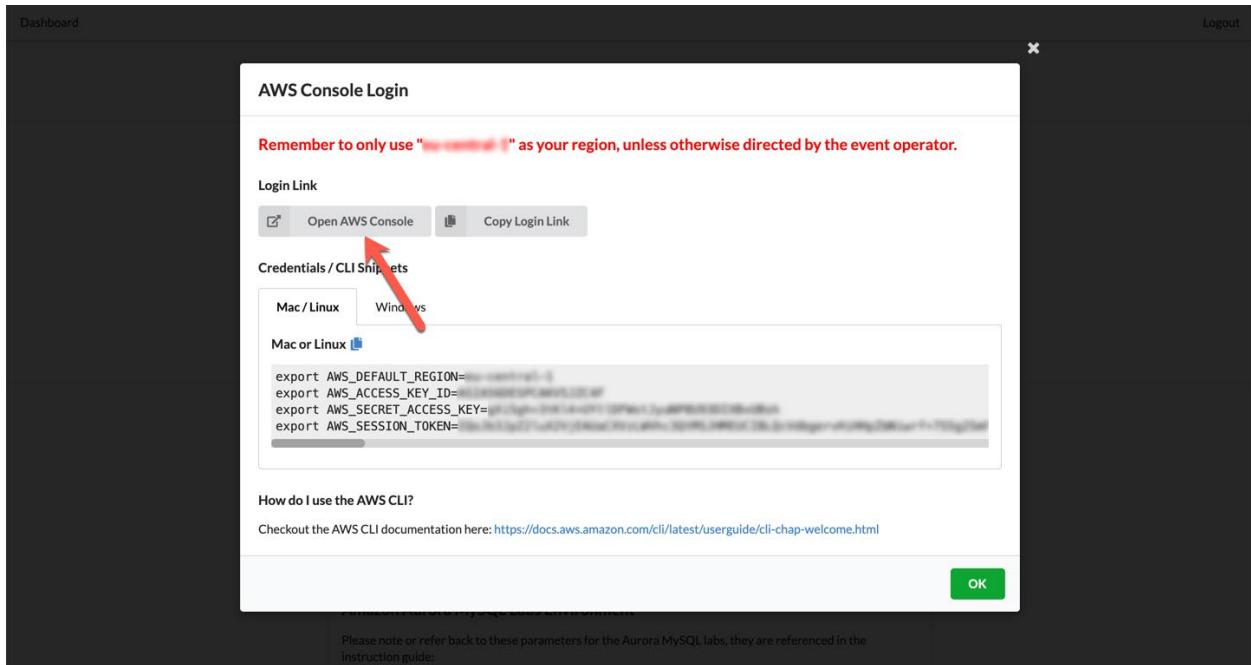
AWS Console    SSH Key

Event: Data Engineering Immersion Day - Test  
Team Name:

Event ID: d2302d4ae9ff4ea2057846b74f7de7e2  
Team ID: 1c2f7ad7ec044b0b8276f917c5983133

## Lab 2. ETL with AWS Glue

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](#)

The screenshot shows the AWS Services search interface. The search bar at the top contains the text "glue". Below the search bar, the results section displays two items: "AWS Glue" and "AWS Lake Formation". "AWS Glue" is described as a "fully managed ETL (extract, transform, and load) service". "AWS Lake Formation" is described as making it easy to set up a secure data lake. At the bottom of the results, there are links for "S3" and "EC2". On the left side, there is a sidebar with a tree view of services, and the "Crawlers" node is currently selected.

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

The screenshot shows the "Crawlers" page under the AWS Glue service. The left sidebar has "Crawlers" selected. The main area has a heading "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 is a search bar with "User preferences" and a message "Showing: 0 - 0 < >". A large button labeled "Add crawler" is visible. A message "You don't have any crawlers yet." is displayed with a hand icon. The table header includes columns for Name, Schedule, Status, Logs, Last runtime, Median runtime, Tables updated, and Tables added.

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" dialog. The left sidebar has "Crawler info" selected. The main area has a heading "Add information about your crawler". It shows a "Crawler name" field containing "glue-lab-crawler". Below the field is a note "Tags, description, security configuration, and classifiers (optional)". A "Next" button is at the bottom right.

6. Choose **Data stores, Crawl all folders** and Click **Next**

## Lab 2. ETL with AWS Glue

### 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

**Repeat crawls of S3 data stores**

Crawl all folders  
 Crawl new folders only

Only Amazon S3 folders that were added since the last crawl will be crawled. If the schemas are compatible, new partitions will be added to existing tables.

[Back](#) [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 stored CSV files, e.g., **s3://xxx-dmslabs3bucket-xxx/tickets**
8. Click **Next**.

### Add crawler

**Add a data store**

Choose a data store: S3

Crawl data in: Specified path

Include path: s3://dmsslab-student-dmsslabs3bucket-1xbv1wp8fe8iq/tickets

Chosen data stores: S3: s3://dmsslab-stud...

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

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

### Add crawler

**Add another data store**

Yes   
No

Chosen data stores: S3: s3://dmsslab-stud...

[Back](#) [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 “dmsslab-student-GlueLabRole-ZOQDII7JTBUM”

## Lab 2. ETL with AWS Glue

### 11. Click Next.

**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-ZOQDII7JTBUM

This role must provide permissions similar to the AWS managed policy, **AWSGlueServiceRole**, plus access to your data stores.  
• s3://dmslab-student-dmslabs3bucket-wotl4bf73cw3

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**

**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**

**Configure the crawler's output**

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

**Add database**

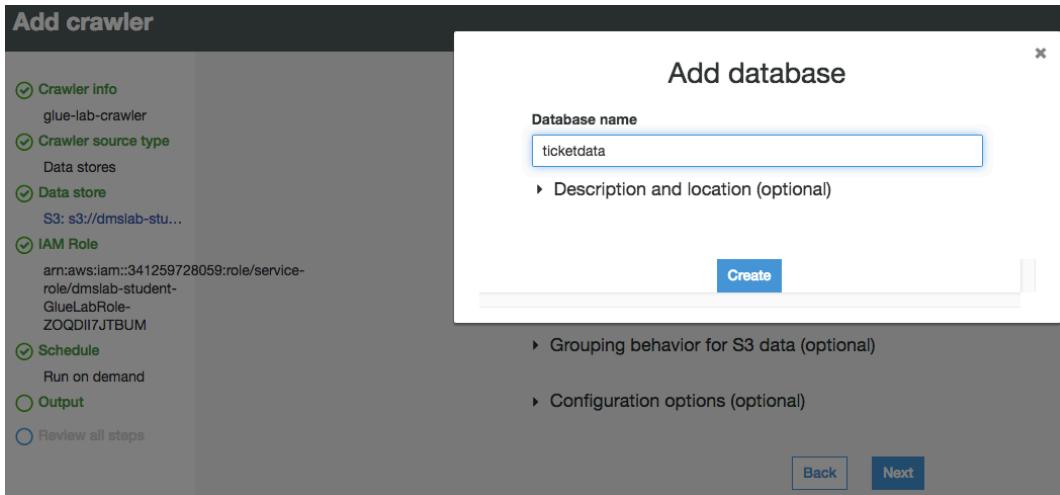
**Prefix added to tables (optional)** [?](#)  
Type a prefix added to table names

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

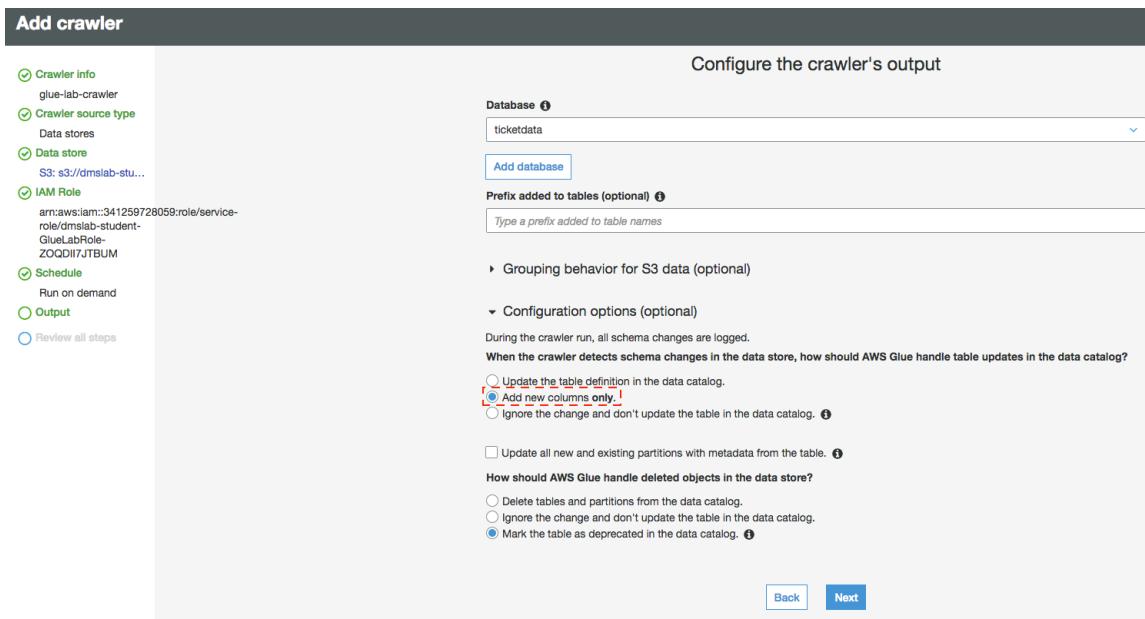
[Back](#) [Next](#)

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

## Lab 2. ETL with AWS Glue

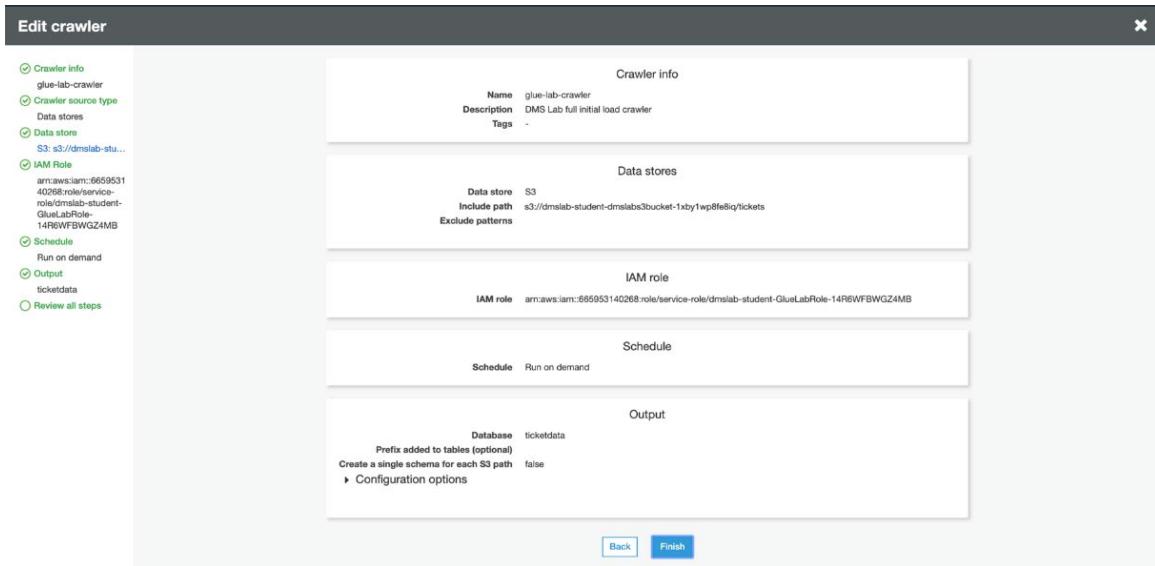


15. For **Prefix added to tables (optional)**, leave the field empty.
16. For **Configuration options (optional)**, select **Add new columns only** and keep the remaining default configuration options and Click **Next**.



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. Tick the crawler name, click **Run crawler** button.

Name	Schedule	Status	Logs	Last runtime	Median run time
glue-lab-crawler		Ready	Logs	1 min	1 min
glue-lab-parquet-crawler		Ready	Logs	1 min	1 min

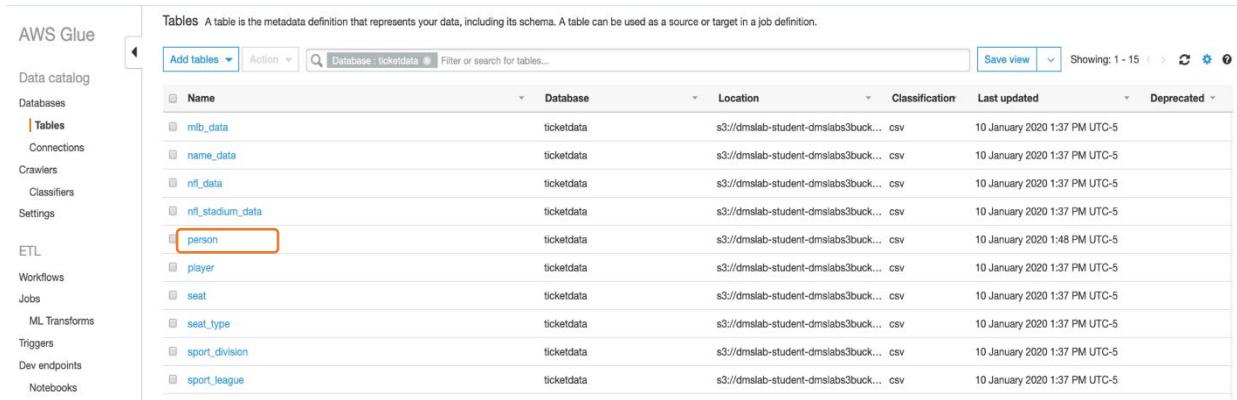
Crawler will change status from starting to stopping, wait until crawler comes back to ready state (the process will take a few minutes), you can see that it has created **15 tables**.

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.

## Lab 2. ETL with AWS Glue



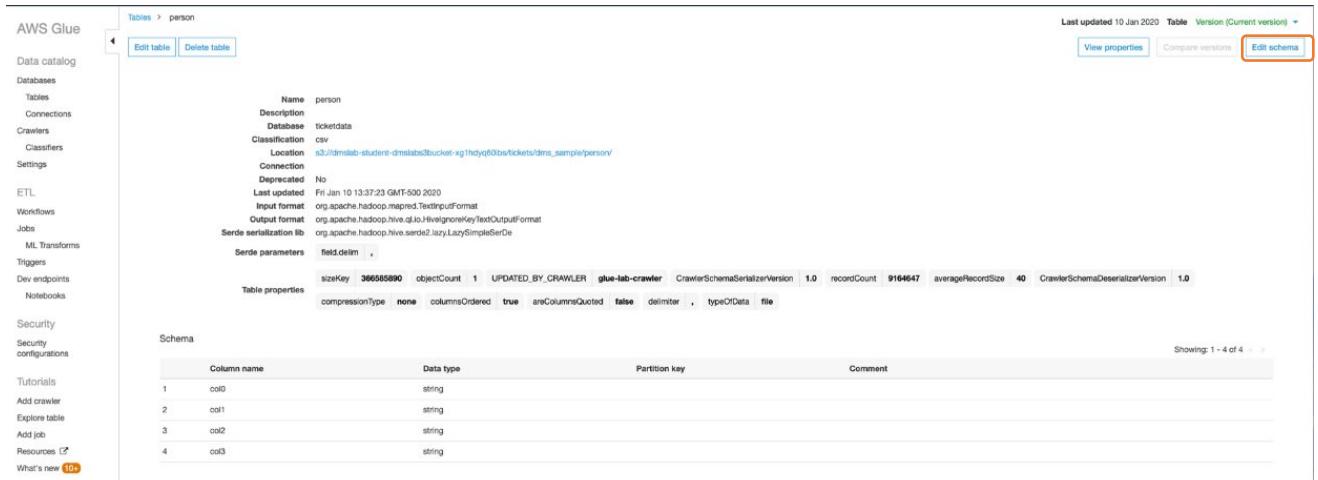
The screenshot shows the AWS Glue Data Catalog interface. On the left, there's a sidebar with navigation links for AWS Glue, Data catalog, Databases, Tables, Connections, Crawlers, Classifiers, Settings, ETL, Workflows, Jobs, ML Transforms, Triggers, Dev endpoints, Notebooks. The 'Tables' link is currently selected and highlighted in orange. The main area displays a table of tables with columns: Name, Database, Location, Classification, Last updated, and Deprecated. One row for the 'person' table is selected and highlighted with a red box.

Name	Database	Location	Classification	Last updated	Deprecated
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	
person	ticketdata	s3://dmslab-student-dmslabs3buck...	csv	10 January 2020 1:48 PM UTC-5	
player	ticketdata	s3://dmslab-student-dmslabs3buck...	csv	10 January 2020 1:37 PM UTC-5	
seat	ticketdata	s3://dmslab-student-dmslabs3buck...	csv	10 January 2020 1:37 PM UTC-5	
seat_type	ticketdata	s3://dmslab-student-dmslabs3buck...	csv	10 January 2020 1:37 PM UTC-5	
sport_division	ticketdata	s3://dmslab-student-dmslabs3buck...	csv	10 January 2020 1:37 PM UTC-5	
sport_league	ticketdata	s3://dmslab-student-dmslabs3buck...	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.

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



The screenshot shows the AWS Glue Table Editor for the 'person' table. The top navigation bar includes 'Tables > person', 'Edit table', 'Delete table', 'Last updated 10 Jan 2020', 'Table Version (Current version)', 'View properties', 'Compare versions', and 'Edit schema' (which is highlighted with a red box). The main area displays table metadata: Name (person), Description (ticketdata), Database (ticketdata), Classification (csv), Location (s3://dmslab-student-dmslabs3buck...), Connection (No), and various serde parameters like Input format, Output format, and Serde serialization lib. Below this is a 'Table properties' section with details like sizeKey, objectCount, recordCount, and averageRecordSize. At the bottom, the 'Schema' section lists four columns: col0, col1, col2, and col3, each with its data type (string).

3. In the Edit Schema section, double-click **col0** (column name) to open edit mode. Type “id” as the column name.

Repeat the preceding step to change the remaining column names to match those shown in the following figure: `full_name`, `last_name` and `first_name`

## Lab 2. ETL with AWS Glue

Last updated 10 Jan 2020 Table Version (Current version) ▾  
Showing: 1 - 4 of 4 Cancel Save

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

- 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.

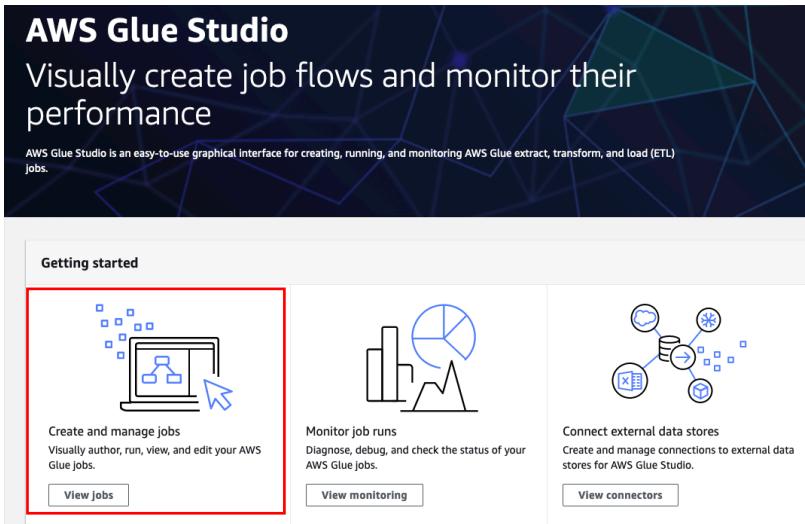
- In the left navigation pane, under ETL, click **AWS Glue Studio**.

Tables A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.  
Add tables Action Filter by attributes or search by keyword Save view Showing: 1 - 15

Name	Database	Location	Classification	Last updated	Deprec
sporting_event	ticketdata	s3://mod-3fcddd60911...	csv	17 March 2021 1:10 PM...	
sport_location	ticketdata	s3://mod-3fcddd60911...	csv	17 March 2021 1:10 PM...	
sport_division	ticketdata	s3://mod-3fcddd60911...	csv	17 March 2021 1:10 PM...	
seat_type	ticketdata	s3://mod-3fcddd60911...	csv	17 March 2021 1:10 PM...	
nfl_data	ticketdata	s3://mod-3fcddd60911...	csv	17 March 2021 1:10 PM...	
ticket_purchase_hist	ticketdata	s3://mod-3fcddd60911...	csv	17 March 2021 1:10 PM...	
person	ticketdata	s3://mod-3fcddd60911...	csv	17 March 2021 1:14 PM...	

- Choose “View Jobs”

## Lab 2. ETL with AWS Glue



- Leave the “Visual with a source and target” option selected, and press “Create”
- Jobs** Info

**Create job** Info

**Create**

Visual with a source and target  
Start with a source, ApplyMapping transform, and target.

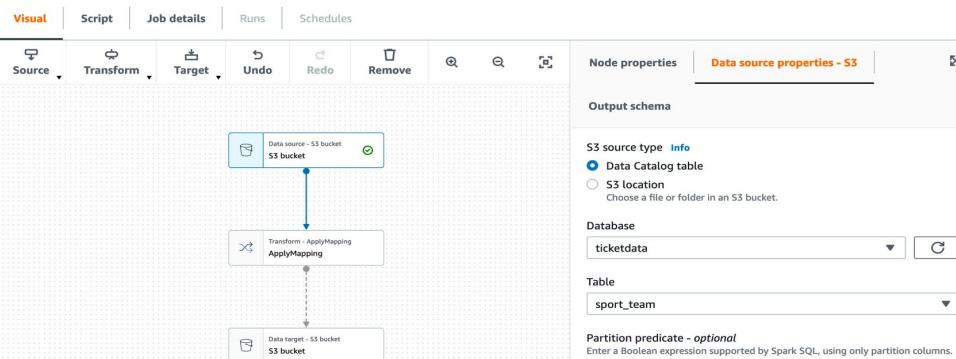
Visual with a blank canvas  
Author using an interactive visual interface.

Script editor  
Write or upload your own code.

**Source**  
Amazon S3  
JSON, CSV, or Parquet files stored in S3.

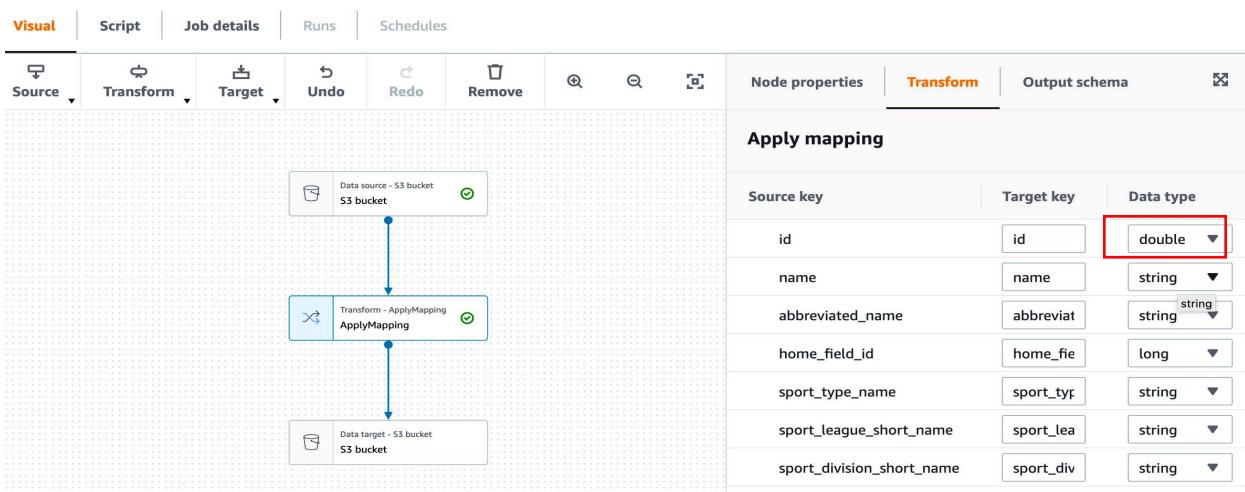
**Target**  
Amazon S3  
S3 bucket by specifying a bucket path as the data target.

- Select the “Data source - S3 bucket” at the top of the graph.
- In the panel on the right under “Data source properties - S3”, choose the “ticketdata” database from the drop down.
- For Table, select the **sport\_team** table.

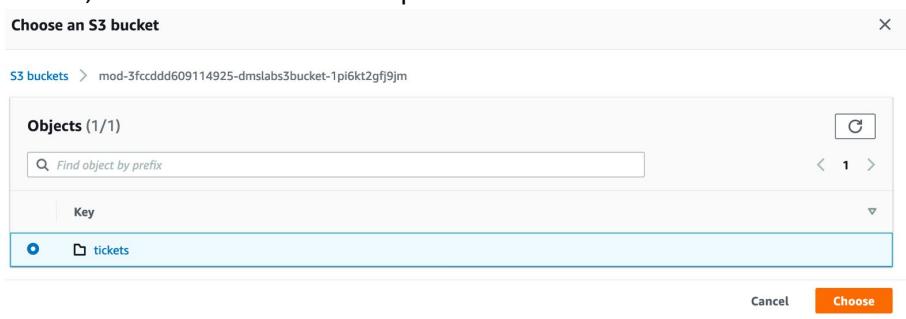


- Select the “ApplyMapping” node. In the Transform panel on the right and change the data type of “id” column to double in the dropdown.

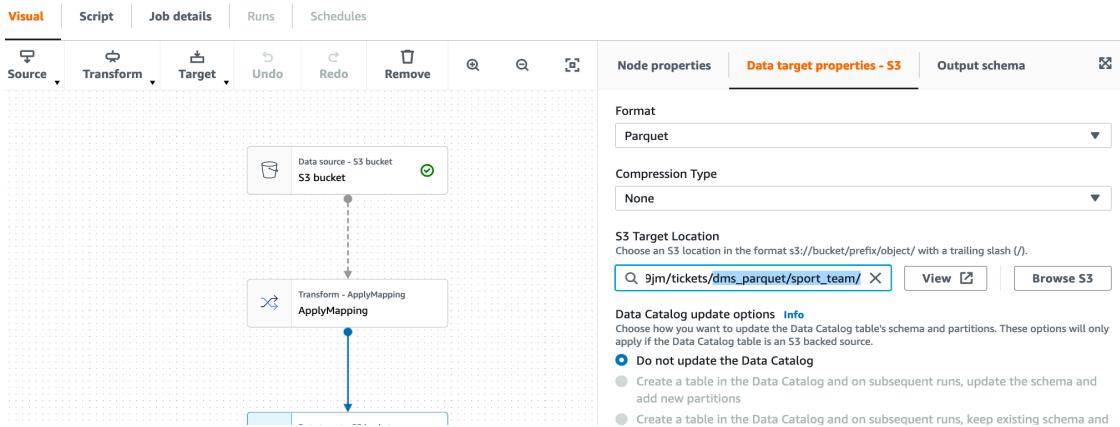
## Lab 2. ETL with AWS Glue



8. Select the “Data target - S3 bucket” node at the bottom of the graph, and change the Format to **Parquet** in the dropdown.
9. Under “S3 Target Location”, select “**Browse S3**” browse to the “mod-xxx-dmslabs3bucket-xxx” bucket, select “**tickets**” item and press “**Choose**”.



10. In the textbox, append **dms\_parquet/sport\_team/** to the S3 url. The path should look similar to s3://mod-xxx-dmslabs3bucket-xxx/tickets/**dms\_parquet/sport\_team/** - don't forget the “/” at the end. The job will automatically create the folder.



11. Finally, select the **Job details** tab at the top. Enter **Glue-Lab-SportTeamParquet** under Name.
12. For “**IAM Role**”, select the role named similar to mod-xxx-GlueLabRole-xxx.

## Lab 2. ETL with AWS Glue

13. Scroll down the page and under “**Job bookmark**”, select “**Disable**” in the drop down. You can try out the bookmark functionality later in this lab.

**Glue-Lab-SportTeamParquet** 

Visual | Script | **Job details**  Runs | Schedules

**Basic properties** [Info](#)

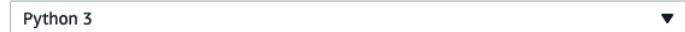
Name   
Glue-Lab-SportTeamParquet

Description - *optional*  
  
Descriptions can be up to 2048 characters long.

IAM Role  
Role assumed by the job with permission to access your data stores. Ensure that this role has permission to your Amazon S3 sources, targets, temporary directory, scripts, and any libraries used by the job.  
  
mod-3fccddd609114925-GlueLabRole-7OEMGU9C9TZ7  
No description available.

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

Glue version [Info](#)  
  
Glue 2.0 - Supports spark 2.4, Scala 2, Python 3

Language  
  
Python 3

Worker type  
Set the type of predefined worker that is allowed when a job runs.  
  
G.1X

Number of workers  
The number of workers of a defined workerType that are allocated when a job runs. The maximum number of workers you can define are 299 for G.1X, and 149 for G.2X.  
  
10

Job bookmark [Info](#)  
Specifies how AWS Glue processes job bookmark when the job runs. It can remember previously processed data (Enable), update state information (Pause), or ignore state information (Disable).  
  
Disable

Number of retries  
  
3

Job timeout (minutes)  
Set the execution time. The default is 2,880 minutes (48 hours).  
  
2880

14. Press the “**Save**” button in the top right-hand corner to create the job.

## Lab 2. ETL with AWS Glue

15. Once you see the “Successfully created job” message in the banner, click the “Run” button to start the job.
16. Select “Jobs” from the navigation panel on the left-hand side to see a list of your jobs.
17. Select “Monitoring” from the navigation panel on the left-hand side to view your running jobs, success/failure rates and various other statistics.

The screenshot shows the AWS Glue Studio interface with the 'Monitoring' tab selected. On the left, the navigation menu includes 'Jobs', 'Monitoring' (which is highlighted in orange), 'Connectors', 'Glue console' (with sub-options 'Glue catalog', 'Crawlers', and 'Security configurations'), 'Marketplace', and 'Documentation'. The main area displays a 'Job runs summary' card with the following data:

Total runs	Running	Canceled	Success	Failed
1	0	0	1	0

A date range selector at the top right shows '7 Day'. Below the summary, there is a chart with a single green bar reaching up to 0.6 on the y-axis, labeled '2021-03-16' at the bottom.

18. Scroll down to the “Job runs” list to verify that the ETL job has completed successfully. This should take about 1 minute to complete.

The screenshot shows the 'Job runs' list in AWS Glue Studio. The left sidebar remains the same as the previous screenshot. The main area shows a table titled 'Job runs (1) Info' with one entry:

Job name	Type	Start time	End time	Run status	Run time
Glue-Lab-SportTeamParquet	Glue ETL	03/17/2021 02:49:04	03/17/2021 02:49:58	Succeeded	1 minute

There are buttons for 'Actions', 'View CloudWatch logs', and 'View run details' above the table. A search bar with 'Filter job runs' placeholder text is also present.

19. We need to repeat this process for an additional 4 jobs, to transform the **sport\_location**, **sporting\_event**, **sporting\_event\_ticket** and **person** tables.

During this process, we will need to modify different column data types. We can either repeat the process above for each table, or we can clone the first job and update the details. The steps below describe how to clone the job - if creating manually each time, follow the above steps but make sure you use the updated values from the tables below.

20. Return to the “Jobs” menu, and select the “Glue-Lab-SportsTeamParquet” job by clicking the small circle next to the name.

## Lab 2. ETL with AWS Glue

The screenshot shows the AWS Glue Studio interface. On the left, there's a sidebar with 'Jobs', 'Monitoring', 'Connectors', 'Glue console' (with sub-options like 'Glue catalog', 'Crawlers', 'Security configurations'), 'Marketplace', and 'Documentation'. The main area is titled 'Create job' with a 'Create' button. It has two options: 'Blank graph' (beginning with an empty canvas) and 'Source and target added to the graph' (beginning with a source, ApplyMapping transform, and target). The 'Source' is set to 'S3' (JSON, CSV, or Parquet files stored...), and the 'Target' is also set to 'S3' (S3 bucket by specifying a bucket ...). Below this, there's a table titled 'Your jobs (1)' with one entry: 'Glue-Lab-SportTeamParquet' (Type: Glue ETL, Last modified: 3/17/2021, 2:48:53 AM).

21. Under the “Actions” dropdown, select “Clone job”. Update the job as per the following tables, then “Save” and “Run”.

### ***1. Sport\_Location:***

Create a **Glue-Lab-SportLocationParquet** job with the following attributes:

Task / Action	Attribute	Values
“Data source - S3 bucket” node	Database	ticketdata
	Table	sport_location
“Transform - ApplyMapping” node	Schema transformations	None
“Data target - S3 bucket” node	Format	Parquet
	S3 target path	tickets/dms_parquet/sport_location/
“Job details tab”	Job Name	Glue-Lab-SportLocationParquet
	IAM Role	xxx-GlueLabRole-xxx
	Job bookmark	Disable

### ***2. Sporting\_Event:***

Create a **Glue-Lab-SportingEventParquet** job with the following attributes:

## Lab 2. ETL with AWS Glue

Task / Action	Attribute	Values
“Data source - S3 bucket” node	Database	ticketdata
	Table	sporting_event
“Transform - ApplyMapping” node	Schema transformations	column “start_date_time” => TIMESTAMP
		column “start_date” => DATE
“Data target - S3 bucket” node	Format	Parquet
	S3 target path	tickets/dms_parquet/sporting_event/
“Job details tab”	Job Name	Glue-Lab-SportingEventParquet
	IAM Role	xxx-GlueLabRole-xxx
	Job bookmark	Disable

### 3. Sporting\_Event\_Ticket:

Create a **Glue-Lab-SportingEventTicketParquet** job with the following attributes:

Task / Action	Attribute	Values
“Data source - S3 bucket” node	Database	ticketdata
	Table	sporting_event_ticket
“Transform - ApplyMapping” node	Schema transformations	column “id” => DOUBLE
		column “sporting_event_id” => DOUBLE
		column “ticketholder_id” => DOUBLE
“Data target - S3 bucket” node	Format	Parquet
	S3 target path	tickets/dms_parquet/sporting_event_ticket/
“Job details tab”	Job Name	Glue-Lab-SportingEventTicketParquet

## Lab 2. ETL with AWS Glue

Task / Action	Attribute	Values
	IAM Role	xxx-GlueLabRole-xxx
	Job bookmark	Disable

### 4. Person:

Create a **Glue-Lab-PersonParquet** job with the following attributes:

Task / Action	Attribute	Values
“Data source - S3 bucket” node	Database	ticketdata
	Table	person
“Transform - ApplyMapping” node	Schema tranformations	column “id” => DOUBLE
“Data target - S3 bucket” node	Format	Parquet
	S3 target path	tickets/dms_parquet/person/
“Job details tab”	Job Name	Glue-Lab-PersonParquet
	IAM Role	xxx-GlueLabRole-xxx
	Job bookmark	Disable

## Create Glue Crawler for Parquet Files

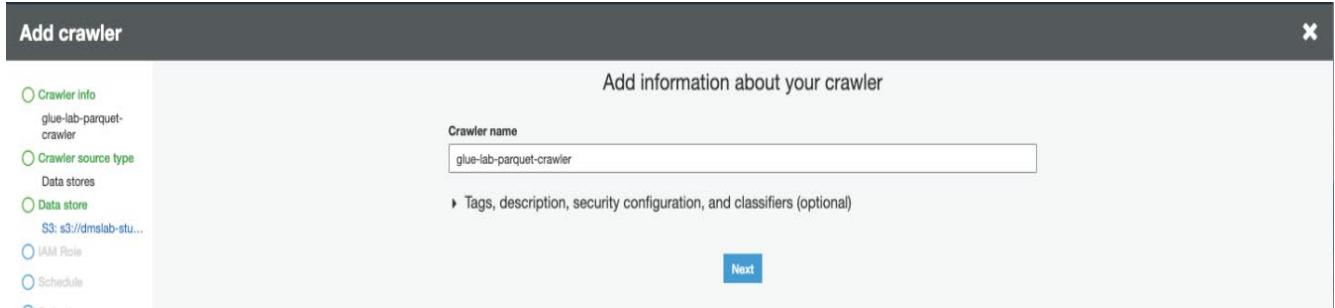
1. In the Glue Studio navigation menu, select **Crawlers** to open the Glue Crawlers page in a new tab.  
Click **Add crawler**.

The screenshot shows the AWS Glue Studio interface. On the left, there's a sidebar with options: Data catalog, Databases, Tables, Connections, Crawlers (which is highlighted with a red box), Classifiers, and Settings. The main area is titled 'Crawlers' and contains the following information:

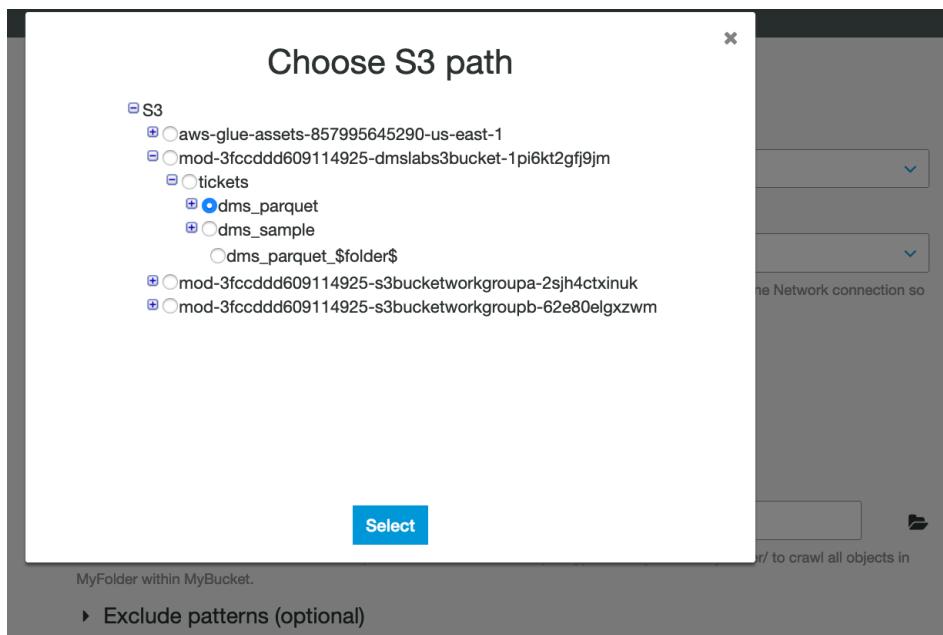
- A header bar with 'Add crawler', 'Run crawler', 'Action', a search bar ('Name: glue-lab-crawler'), and filter options.
- A table with columns: Name, Schedule, Status, Logs, Last runtime, Median runtime, Tables updated, and Tables added.
- In the table, there is one row for 'glue-lab-crawler' with the status 'Ready'.
- At the bottom right of the table, it says 'Showing: 1 - 1'.
- At the top right of the main area, there are 'User preferences' and other navigation icons.

## Lab 2. ETL with AWS Glue

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 my 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**.



## Lab 2. ETL with AWS Glue

### Add a data store

**Choose a data store**  
S3

**Connection**  
Select a connection

Optionaly include a Network connection to use with this S3 target. Note that each crawler is limited to one Network connection so any future S3 targets will also use the same connection (or none, if left blank).

**Add connection**

**Crawl data in**  
 Specified path

**Include path**  
s3://mod-3fccddd609114925-dmslabs3bucket-1pi6kt2gfj9jm/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**.

### 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** [?](#)  
mod-3fccddd609114925-GlueLabRole-7OEMGU9C9TZ7

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

s3://mod-3fccddd609114925-dmslabs3bucket-1pi6kt2gfj9jm/tickets/dms\_parquet

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

**Back** **Next**

## Lab 2. ETL with AWS Glue

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

Add crawler

Create a schedule for this crawler

Frequency

Run on demand

Back Next

Crawler info  
glue-lab-parquet-crawler

Crawler source type  
Data stores

Data store  
S3: s3://mod-3fccc...

IAM Role  
arn:aws:iam::857995645290:role/service-role/mod-3fcdd609114925-GlueLabRole-7OEMGU9C9TZ7

Schedule  
Run on demand

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

Crawler info  
glue-lab-parquet-crawler

Crawler source type  
Data stores

Data store  
S3: s3://mod-3fccc...

IAM Role  
arn:aws:iam::857995645290:role/service-role/mod-3fcdd609114925-GlueLabRole-7OEMGU9C9TZ7

Schedule  
Run on demand

Output  
parquet\_

Review all steps

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

11. Click **Run Crawler**. 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.

Data catalog

Databases

Tables

Connections

Crawlers

Classifiers

Schema registries

Schemas

...

Crawlers 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.

User | Showing: 1 - 2

Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables
glue-lab-crawler		Ready	Logs	1 min	1 min	0	15
glue-lab-parquet-crawler		Ready		0 secs	0 secs	0	0

Confirm you can see the tables:

## Lab 2. ETL with AWS Glue

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

The screenshot shows the AWS Glue Tables list. On the left, the navigation pane is visible with 'Tables' selected. The main area displays a table of tables, with the 'parquet\_person' table highlighted by a red box. The table columns include Name, Database, Location, Classification, Last updated, and Deprecated. The 'parquet\_person' table is listed under the 'ticketdata' database and has a 'parquet' classification.

Name	Database	Location	Classification	Last updated	Deprecated
mib_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.

The screenshot shows the AWS Glue Crawlers list. On the left, the navigation pane is visible with 'Crawlers' selected. The main area displays a table of crawlers, which is currently empty. A message indicates 'You don't have any crawlers yet.' and a blue 'Add crawler' button is visible.

Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
You don't have any crawlers yet.							

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").

## Lab 2. ETL with AWS Glue

4. Optionally, enter the description. This should also be descriptive and easily recognized and Click

Add information about your crawler

Crawler name

glue-lab-cdc-crawler

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

Next

**Next.**

5. Choose **Data Stores** as Crawler Source Type, **Crawl all folders** and 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

**Repeat crawls of S3 data stores**

- Crawl all folders  
 Crawl new folders only

Only Amazon S3 folders that were added since the last crawl will be crawled. If the schemas are compatible, new partitions will be added to existing tables.

Back

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**.

## Lab 2. ETL with AWS Glue

Add a data store

Choose a data store  
S3

Connection  
Select a connection

Optional note: Optionally include a Network connection to use with this S3 target. Note that each crawler is limited to one Network connection so any future S3 targets will also use the same connection (or none, if left blank).

Add connection

Crawl data in  
 Specified path

Include path  
s3://mod-3fccddd609114925-dmslabs3bucket-1pi6kt2gfj9jm/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://mod-3fc...

Add another data store  
 Yes  
 No

Back 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**.

## 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** [?](#)

mod-3fccddd609114925-GlueLabRole-7OEMGU9C9TZ7

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

- s3://mod-3fccddd609114925-dmslabs3bucket-1pi6kt2gfi9jm/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**.

## Create a schedule for this crawler

**Frequency**

Run on demand

[Back](#) [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**, specify "**cdc\_**"
14. For Configuration options (optional), keep the **default** selections and click **Next**.

## Lab 2. ETL with AWS Glue

**Add crawler**

- Crawler info  
glue-lab-cdc-crawler
- Crawler source type  
Data stores
- Data store  
S3: s3://mod-3fccd...
- IAM Role  
arn:aws:iam::857995645290:role/service-role/mod-3fccdd609114925-GlueLabRole-7OEMGU9C9TZ7
- Schedule  
Run on demand
- Output  
ticketdata
- Review all steps

**Database** i

**Add database**

**Prefix added to tables (optional)** i

**Grouping behavior for S3 data (optional)**

**Configuration options (optional)**

During the crawler run, all schema changes are logged.

**When the crawler detects schema changes in the data store, how should AWS Glue handle table updates in the data catalog?**

- Update the table definition in the data catalog.
- Add new columns only.
- Ignore the change and don't update the table in the data catalog. i

Update all new and existing partitions with metadata from the table. i

**How should AWS Glue handle deleted objects in the data store?**

- Delete tables and partitions from the data catalog.
- Ignore the change and don't update the table in the data catalog.
- Mark the table as deprecated in the data catalog. i

**Back** **Next**

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

**Add crawler**

- Crawler info  
glue-lab-cdc-crawler
- Crawler source type  
Data stores
- Data store  
S3: s3://dmslab-stu...
- IAM Role  
arn:aws:iam::665953140268:role/service-role/dmslab-student-GlueLabRole-14R6WFBBWGZ4MB
- Schedule  
Run on demand
- Output  
ticketdata
- Review all steps

**Crawler info**

Name: glue-lab-cdc-crawler

Tags: -

**IAM role**

IAM role: arn:aws:iam::665953140268:role/service-role/dmslab-student-GlueLabRole-14R6WFBBWGZ4MB

**Schedule**

Schedule: Run on demand

**Output**

Database: ticketdata

Prefix added to tables (optional): cdc\_

Create a single schema for each S3 path: false

**Configuration options**

Schema updates in the data store	Update the table definition in the data catalog.
Object deletion in the data store	Mark the table as deprecated in the data catalog.

**Back** **Finish**

16. Tick the crawler name “**glue-lab-cdc-crawler**”, click **Run crawler** button.

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**.

## Lab 2. ETL with AWS Glue

Crawlers 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.

**Add crawler** Run crawler Action ▾ Filter or search for crawlers... Name : glue-lab User preferences Showing: 1 - 2

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.

Tables A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

**Add tables** Action Database : ticketdata Filter or search for tables... Save view Showing: 1 - 22

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

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

Jobs A job is your business logic required to perform extract, transform and load (ETL) work. Job runs are initiated by triggers.

**New in AWS Glue**  
Author jobs visually in [AWS Glue Studio](#).  
With AWS Glue version 2.0, jobs start 10x faster and get 1-minute minimum billing. Test your existing jobs on the new version.

**Add job** Action Filter by tags and attributes

Name	Type	ETL
Glue-Lab-PersonParquet	Spark	pytho
Glue-Lab-SportLocationParquet	Spark	pytho
Glue-Lab-SportTeamParquet	Spark	pytho
Glue-Lab-SportingEventParquet	Spark	pytho
Glue-Lab-SportingEventTicketParquet	Spark	pytho

## Lab 2. ETL with AWS Glue

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 2.0)** or whichever is the latest version
  - e. For **This job runs**, select **A proposed script generated by AWS Glue**.
  - f. For **Script file name**, use the **default**.
  - 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**

## Lab 2. ETL with AWS Glue

**Configure the job properties**

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

**IAM role**   
mod-3fccddd609114925-GlueLabRole-7OEMGU9C9TZ7 

Ensure that this role has permission to your Amazon S3 sources, targets, temporary directory, scripts, and any libraries used by the job. [Create IAM role](#).

**Type**  
Spark

**Glue version**  
Spark 2.4, Python 3 with improved job startup times (Glue Version 2.0)

**This job runs**  
 A proposed script generated by AWS Glue   
 An existing script that you provide  
 A new script to be authored by you

**Script file name**  
Glue-Lab-TicketHistory-Parquet-with-bookmark

**S3 path where the script is stored**  
s3://aws-glue-scripts-857995645290-us-east-1/admin 

**Temporary directory**   
s3://aws-glue-temporary-857995645290-us-east-1/admin 

▼ Advanced properties

**Job bookmark**   
Enable

▼ Monitoring options

Job metrics   
 Continuous logging  
 Spark UI 

► Tags (optional)

► Security configuration, script libraries, and job parameters (optional)

► Catalog options (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** 

**Job properties**  
Glue-Lab-TicketHistory-Parquet-with-bookmark

**Data source**  
cdc.ticket.purchase...   
Transform type   
Data target   
Systems

**Choose a data source**

Showing: 1 - 29 

Name	Database	Location	Classification
bookmark_parquet_ticket_purchase_history	ticketdata	s3://dmstlab-student-dmstlab3bucket-xg1hdyg0lsr/cdc_bookmark/ticke...	parquet
cdc_sporting_event_ticket	ticketdata	s3://dmstlab-student-dmstlab3bucket-xg1hdyg0lsr/cdc/dms_sample/s...	csv
<b>cdc_ticket_purchase_hist</b>	ticketdata	s3://dmstlab-student-dmstlab3bucket-xg1hdyg0lsr/cdc/dms_sample/tic...	csv
clickstream_data	processed-data	s3://rawdataset-dehnalt/Clickstream_data/	json
csv_clickstream_data	processed-data	s3://processed-dehnalt/Clickstream_data/	csv

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

## Lab 2. ETL with AWS Glue

**Choose a transform type**

Machine learning transforms are currently not supported for Glue 2.0.

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:

- a. Create tables in your data target
- b. For **Data store**: select **Amazon S3**
- c. Format: **parquet**
- d. **Target path**: s3://xxx-dmslabs3bucket-xxx/cdc\_bookmark/ticket\_purchase\_history/data/
- e. Click **Next**

**Choose a data target**

Create tables in your data target  
 Use tables in the data catalog and update your data target

**Data store**  
Amazon S3

**Format**  
Parquet

**Connection**  
- Select one -

[Add connection](#)

**Target path**  
s3://xxx-dmslabs3bucket-xxx/cdc\_bookmark/ticket\_purchase\_history/data/

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

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

## Lab 2. ETL with AWS Glue

10. In the next window, review the job script and click on **Run job**, then 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

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 ## Environment: DFO_NHIC
9 args = getResolvedOptions(sys.argv, ['JOB_NAME'])
10
11 sc = SparkContext()
12 glueContext = GlueContext(sc)
13 spark = glueContext.spark_session
14 job = Job(glueContext)
15 job.init(args['JOB_NAME'], args)
16
17 ### Input: Datasource
18 ### Output: Datasink
19 ### Datasource
20 datasource = glueContext.create_dynamic_frame.from_catalog(database = "ticketdata", table_name = "cdc_ticket_purchase_hist", transformation_ctx = "datasource")
21 ### Datasink
22 ### Transformation
23 ### Transformation
24 ### Transformation
25 applymapping1 = ApplyMapping.apply(frame = datasource, mappings = [{"op": "string", "arg": "string"}, {"sporting_event_ticket_id": "string", "sporing_event_ticket_id": "string"}, {"purchased_by_id": "string", "purchased_by_id": "string"}, {"transaction_date_time": "string", "transaction_date_time": "string"}, {"transferred_from_id": "string", "transferred_from_id": "string"}, {"purchase_price": "double", "purchase_price": "double"}])
26 ### Transformation
27 ### Transformation
28 ### Transformation
29 ### Transformation
30 resolvechoice1 = ResolveChoice.apply(frame = applymapping1, choice = "make_struct", transformation_ctx = "resolvechoice1")
31 ### Transformation
32 ### Transformation
33 dropnullfields1 = DropNullFields.apply(frame = resolvechoice1, transformation_ctx = "dropnullfields1")
34 ### Transformation
35 ### Transformation
36 ### Transformation
37 ### Transformation
38 ### Transformation
39 ### Transformation
40 datasource1 = glueContext.write_dynamic_frame(frame_options = dropnullfields1, connection_type = "s3", connection_options = {"path": "s3://dmslab-student-dmslslabs3bucket-xg1hydq60ls/cdc_bookmark/ticket_purchase_history/data"}, format = "parquet", transformation_ctx = "datasink1")
41 job.commit()

```

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

Name	Last modified	Size	Storage class
part_00000-498ea7fc-2ac1-4787-b431-9e16f5e24a3f-c000.snappy.parquet	Jan 24, 2020 7:03:16 PM GMT-0500	1.1 MB	Standard
part_00001-498ea7fc-2ac1-4787-b431-9e16f5e24a3f-c000.snappy.parquet	Jan 24, 2020 7:03:16 PM GMT-0500	1.2 MB	Standard

### 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**

## Lab 2. ETL with AWS Glue

Name	Schedule	Status	Logs
glue-lab-crawler		Ready	Logs
glue-lab-parquet-crawler		Ready	Logs

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

4. In **Specify crawler source type**, select **Data stores** and **Crawl all folders**. 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

Repeat crawl of S3 data stores  
 Crawl all folders  
 Crawl new folders only

Only Amazon S3 folders that were added since the last crawl will be crawled. If the schemas are compatible, new partitions will be added to existing tables.

Back 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**

## Lab 2. ETL with AWS Glue

Add a data store

Choose a data store  
S3

Connection  
Select a connection

Optional include a Network connection to use with this S3 target. Note that each crawler is limited to one Network connection so any future S3 targets will also use the same connection (or none, if left blank).

[Add connection](#)

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

Include path  
3://mod-3fccddd609114925-dmslabs3bucket-1pi6kt2gfj9jm/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](#)

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

Add crawler

Crawler info  
glue\_lab\_cdc\_bookmark\_crawler

Crawler source type  
Data stores

Data store

Add another data store  
 Yes  
 No

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

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

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 [?](#)  
mod-3fccddd609114925-GlueLabRole-7OEMGU9C9TZ7

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

- s3://mod-3fccddd609114925-dmslabs3bucket-1pi6kt2gfj9jm/cdc\_bookmark/ticket\_purchase\_history

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

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

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.

## Lab 2. ETL with AWS Glue

- 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**. Then **Run 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**”

Name	Database	Location	Classification	Last updated	Deprecated
bookmark_parquet_ticket_purchase_history	ticketdata	s3://dmslab-student-dmlabs3bucket-xg1hdyg90...	parquet	24 January 2020 7:14 PM UTC-5	
cdc_sparking_event_ticket	ticketdata	s3://dmslab-student-dmlabs3bucket-xg1hdyg90...	csv	24 January 2020 5:13 PM UTC-5	
cdc_ticket_purchase_hist	ticketdata	s3://dmslab-student-dmlabs3bucket-xg1hdyg90...	csv	24 January 2020 5:13 PM UTC-5	
mrb_data	ticketdata	s3://dmslab-student-dmlabs3bucket-xg1hdyg90...	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

Sources Workgroup : primary

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](#)

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:

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-00000-0e5d80b4-2bc-47c2-d249-d9100b75ddad-c000.snappy.parquet	Jan 24, 2020 9:21:13 PM GMT+0000	9.3 KB	Standard
part-00000-496ea7fc-2bc1-47f7-b431-9e1f5e24a31-c000.snappy.parquet	Jan 24, 2020 7:03:16 PM GMT+0000	1.1 MB	Standard
part-00000-d166723-3158-45fa-b8be-a65238402348-c000.snappy.parquet	Jan 25, 2020 11:24:20 PM GMT+0000	1.7 MB	Standard
part-00000-efcc0200f9d4-4f5c-b036-3cb2a21519ba-c000.unsnappy.parquet	Jan 25, 2020 10:24:27 PM GMT+0000	7.2 KB	Standard
part-00000-0e5d80b4-2bc-47c2-d249-d9100b75ddad-c000.snappy.parquet	Jan 24, 2020 9:21:13 PM GMT+0000	66.5 KB	Standard
part-00001-496ea7fc-2bc1-47f7-b431-9e1f5e24a31-c000.snappy.parquet	Jan 24, 2020 7:03:16 PM GMT+0000	1.2 MB	Standard
part-00001-d166723-3158-45fa-b8be-a65238402348-c000.unsnappy.parquet	Jan 25, 2020 11:24:20 PM GMT+0000	1.7 MB	Standard
part-00002-0e5d80b4-2bc-47c2-d249-d9100b75ddad-c000.snappy.parquet	Jan 24, 2020 9:21:15 PM GMT+0000	1.7 MB	Standard
part-00002-d166723-3158-45fa-b8be-a65238402348-c000.unsnappy.parquet	Jan 25, 2020 11:24:19 PM GMT+0000	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";
```

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**

## Lab 2. ETL with AWS Glue

### Overview:

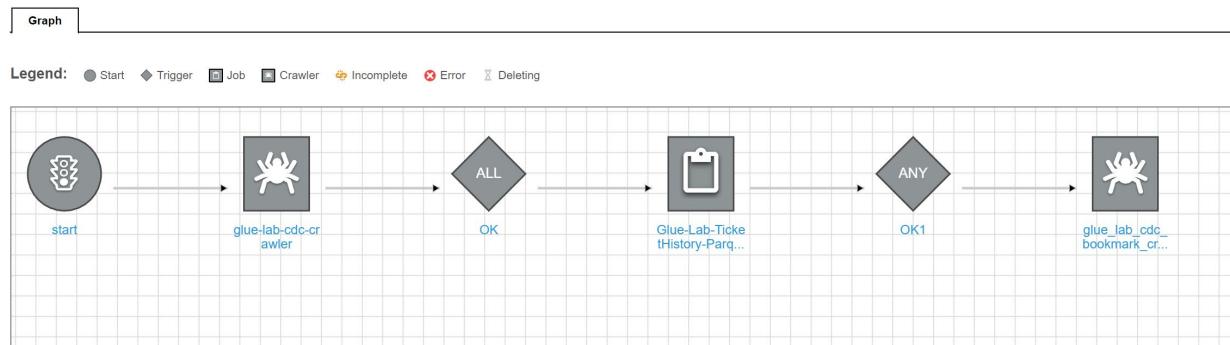
In AWS Glue, you can use workflows to create and visualize complex extract, transform, and load (ETL) activities involving multiple crawlers, jobs, and triggers. Each workflow manages the execution and monitoring of all its components. As a workflow runs each component, it records execution progress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

## Lab 2. ETL with AWS Glue

The screenshot shows the AWS Glue Workflows console. The left sidebar has a tree view with 'Data catalog', 'Databases', 'Tables', 'Connections', 'Crawlers', 'Classifiers', 'Settings', 'ETL' (which is expanded), 'Workflows' (selected), 'Jobs', and 'ML Transforms'. The main content area is titled 'Workflows (0)' with a sub-instruction: 'A workflow is an orchestration used to visualize and manage the relationship and execution of multiple triggers, jobs and crawlers.' Below this is a search bar 'Filter workflows' and a table header with columns: Name, Last run, Last run status, and Last modified. A message 'No workflows' is displayed, followed by a blue button 'Add a new ETL 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.

The screenshot shows the AWS Glue Workflows console with a single workflow listed: 'Workflow\_MLB\_Data'. The table columns are: Name, Last run, Last run status, and Last modified. The 'Last run' and 'Last run status' columns show '-' and '-' respectively. The 'Last modified' column shows 'Sun, 26 Jan 2020 05:12:03 GMT'. At the bottom of the page, there is a legend with icons for Start, Trigger, Job, Crawler, Incomplete, Error, and Deleting. Below the legend, a button labeled 'Add trigger' is highlighted with an orange box.

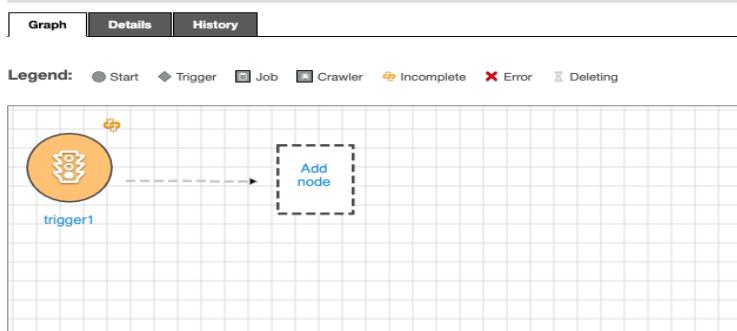
4. In **Add Trigger** window, From Clone Existing and Add New options, click on **Add New**.
  - a. Provide **Name** as "**trigger1**"
  - b. Provide a **description**: Trigger to start workflow
  - c. **Trigger type**: On-demand.
  - d. Click on **Add**

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

The screenshot shows the 'Add trigger' dialog box. It has two tabs: 'Clone existing' (which is selected) and 'Add new'. The 'Add new' tab contains fields for 'Name' (with 'trigger1' entered), 'Description (optional)' (with 'Trigger to start the workflow' entered), and 'Trigger type' (with 'On demand' selected). At the bottom of the dialog are 'Cancel' and 'Add' buttons.

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.

## Lab 2. ETL with AWS Glue



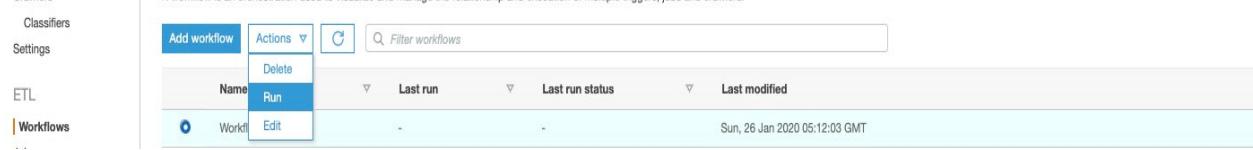
6. Click on **Add node**, a new window to add jobs or crawlers will open. Select the Crawler **glue-lab-cdc-crawler**, then **Add**.
7. Click on the crawler and **Add Trigger** provide the following:
  - a. **Name:** trigger2
  - b. **Description:** Trigger to execute job
  - c. **Trigger type:** Event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

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

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**

## Lab 2. ETL with AWS Glue

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”



Name	Last run	Last run status	Last modified
Run	-	-	Sun, 26 Jan 2020 05:12:03 GMT
Workflow	Edit		

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