



Amazon Web Services Data Engineering Immersion Day

Lab 2. ETL with AWS Glue

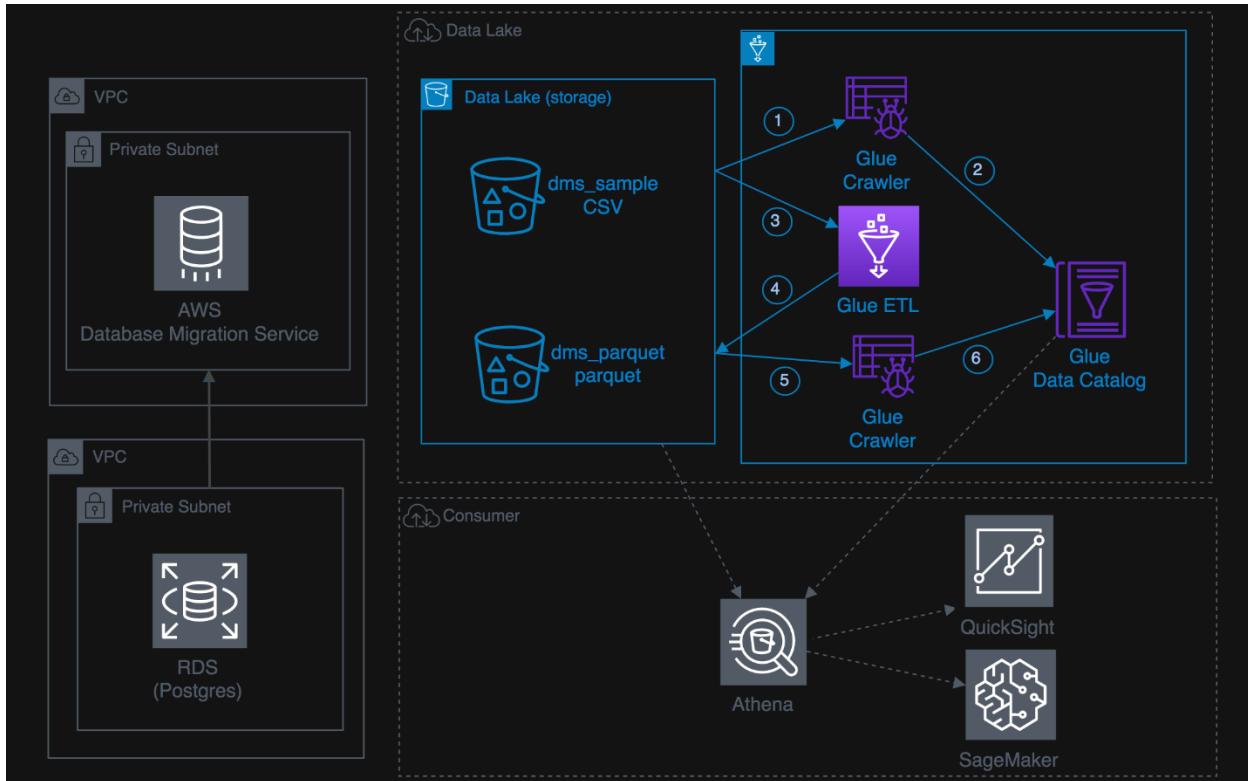
March 2021

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	11
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>	38
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/>

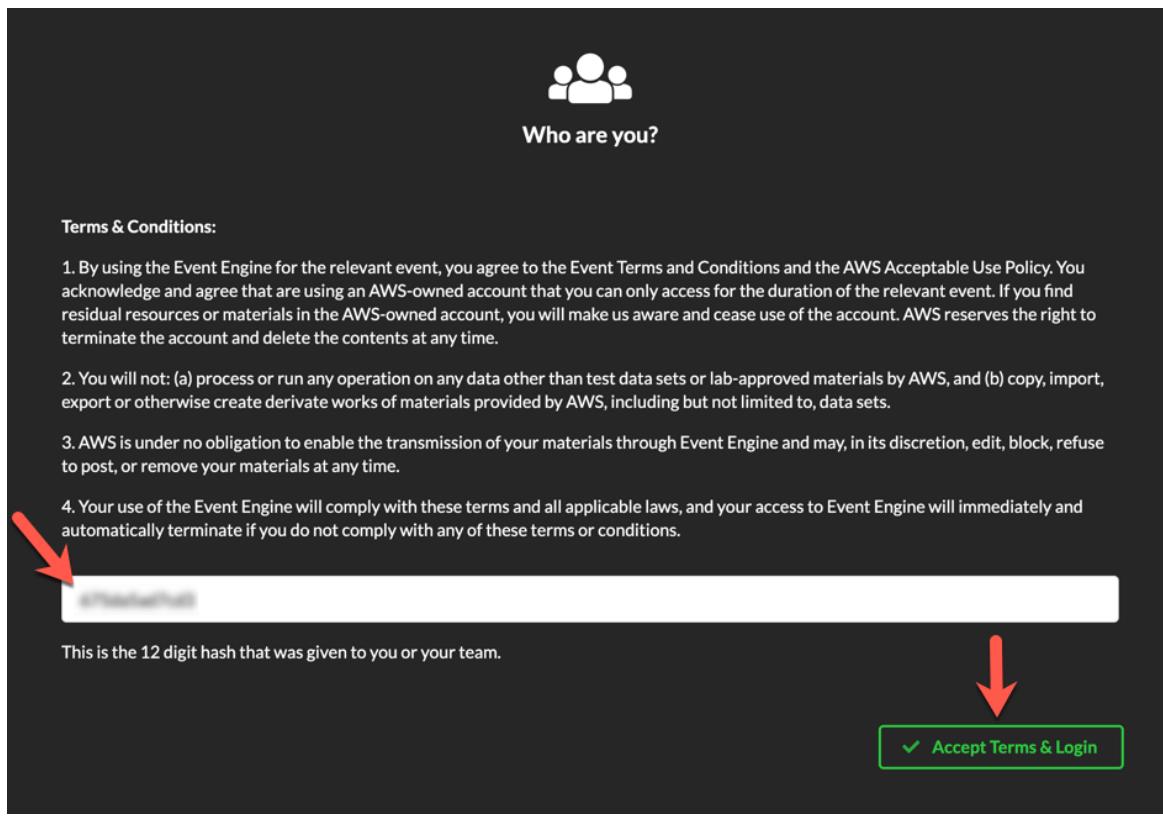
Get Started Using the Lab Environment

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

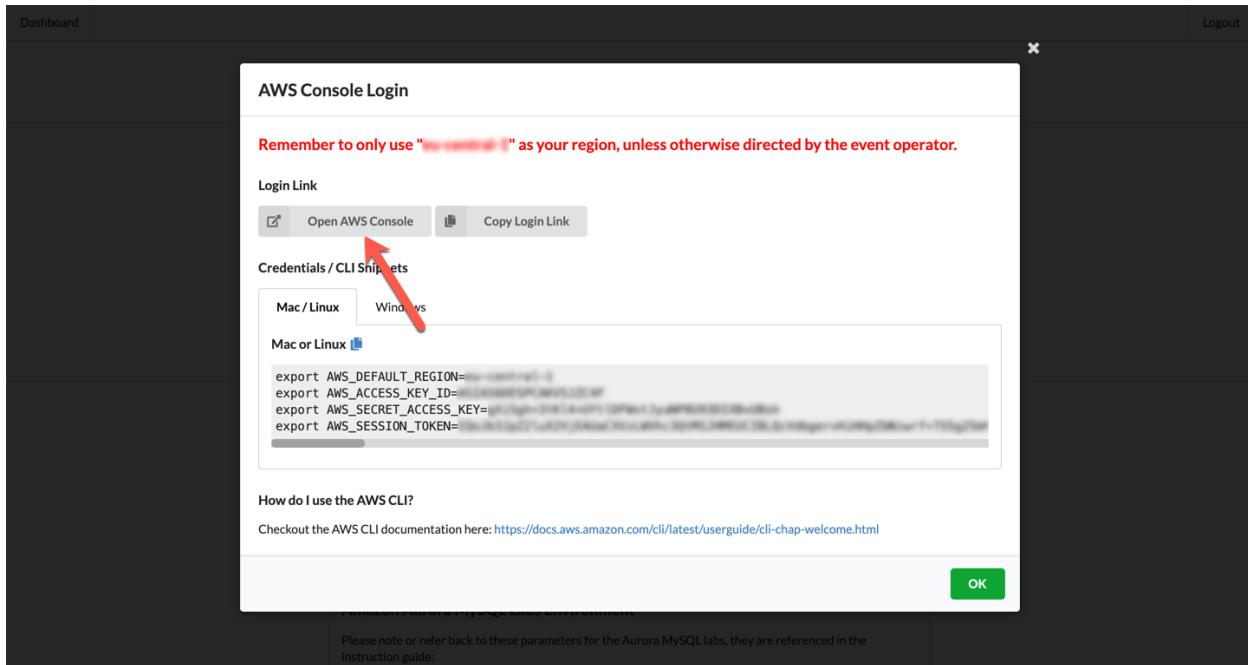
AWS Console SSH Key

Event: Data Engineering Immersion Day - Test
Team Name:

Event ID: d2302d4ae9ff4ea2857846b74f7de7e2
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. In the top-left corner, there's a search bar with the text "glue". Below it, a list of services is displayed, with "AWS Glue" being the first item. Other visible items include "AWS Lake Formation" and "S3" and "EC2" which are likely artifacts from the search or a related search.

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

The screenshot shows the "Crawlers" page under the AWS Glue service. The sidebar on the left has "Crawlers" selected. The main area displays a table with columns: Name, Schedule, Status, Logs, Last runtime, Median runtime, Tables updated, and Tables added. A message at the bottom states "You don't have any crawlers yet." and features an "Add crawler" button.

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

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

Crawler info
glue-lab-crawler

Crawler source type
Data stores

Data store
S3: s3://dmssl...
IAM Role
Schedule
Output
Review all steps

Add a data store

Choose a data store

Crawl data in
 Specified path

Include path

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

Chosen data stores
S3: s3://dmssl...
[X](#)

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

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

Add crawler

Crawler info
glue-lab-crawler

Crawler source type
Data stores

Data store
S3: s3://dmssl...
IAM Role
Schedule
Output
Review all steps

Add another data store

Yes
 No

Chosen data stores
S3: s3://dmssl...
[X](#)

[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**.

Crawler info
glue-lab-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-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**.

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

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.

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

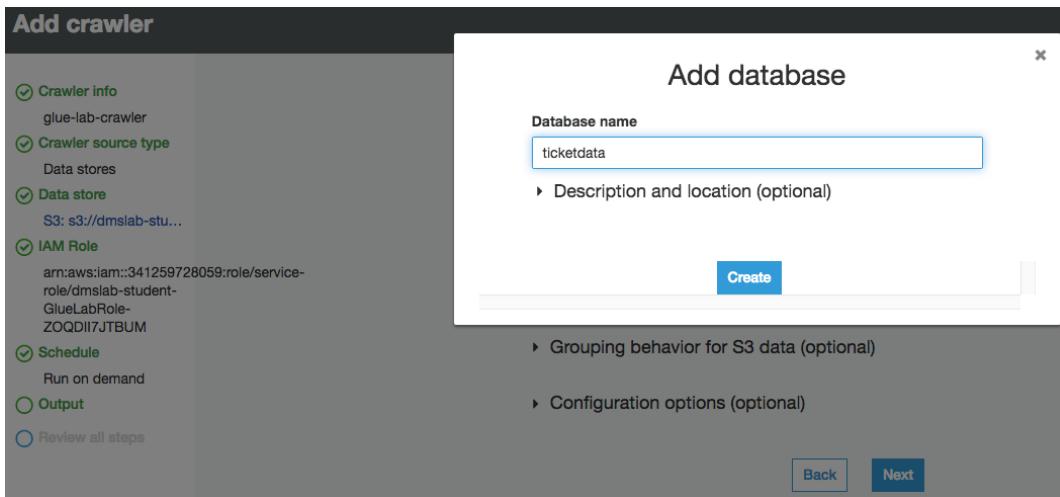
Grouping behavior for S3 data (optional)

Configuration options (optional)

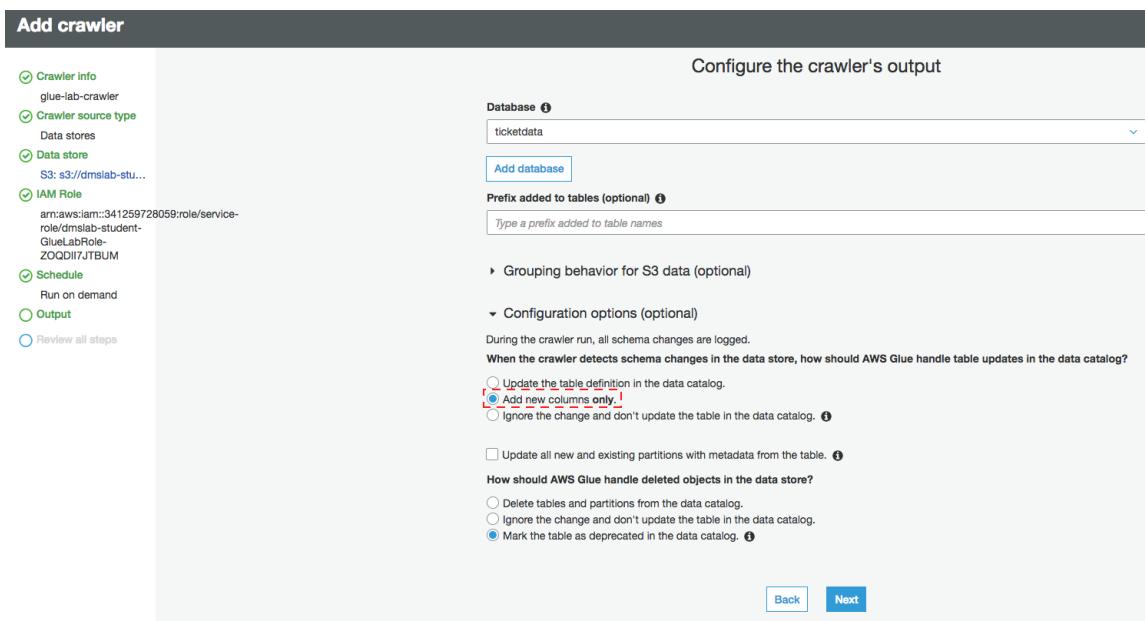
Back Next

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

Lab 2. ETL with AWS Glue

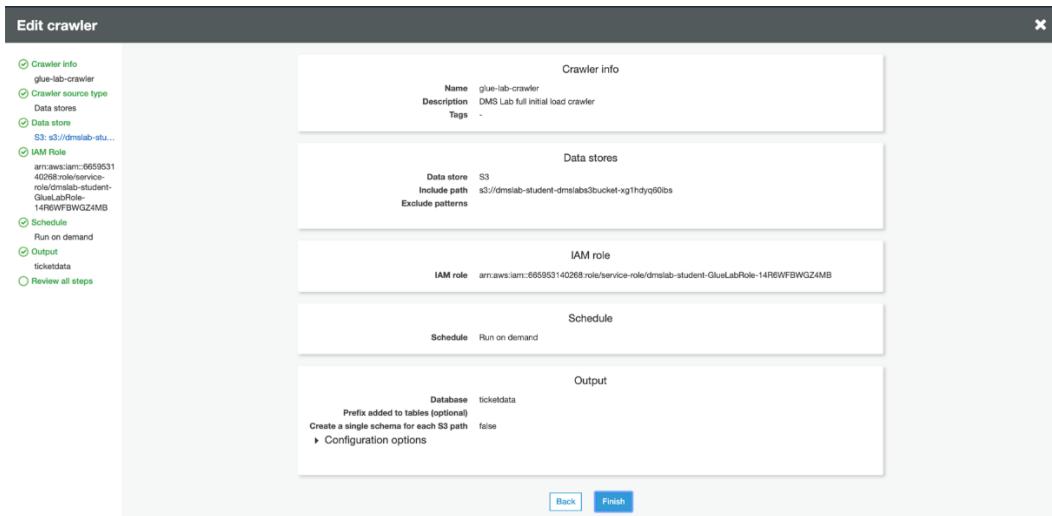


15. For **Prefix added to tables (optional)**, leave the field empty.
16. For **Configuration options (optional)**, select **Add new columns only** and keep the remaining default configuration options and Click **Next**.



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

Name	Schedule	Status	Logs	Last runtime	Median run
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.

Name	Database	Location	Classification	Last updated	Deprecated
mlb_data	ticketdata	s3://dmslab-student-dmslslabs3bucket-xg1hdqj601os	csv	10 January 2020 1:37 PM UTC-5	
name_data	ticketdata	s3://dmslab-student-dmslslabs3bucket-xg1hdqj601os	csv	10 January 2020 1:37 PM UTC-5	
nfl_data	ticketdata	s3://dmslab-student-dmslslabs3bucket-xg1hdqj601os	csv	10 January 2020 1:37 PM UTC-5	
nfl_stadium_data	ticketdata	s3://dmslab-student-dmslslabs3bucket-xg1hdqj601os	csv	10 January 2020 1:37 PM UTC-5	
person	ticketdata	s3://dmslab-student-dmslslabs3bucket-xg1hdqj601os	csv	10 January 2020 1:48 PM UTC-5	
player	ticketdata	s3://dmslab-student-dmslslabs3bucket-xg1hdqj601os	csv	10 January 2020 1:37 PM UTC-5	
seat	ticketdata	s3://dmslab-student-dmslslabs3bucket-xg1hdqj601os	csv	10 January 2020 1:37 PM UTC-5	
seat_type	ticketdata	s3://dmslab-student-dmslslabs3bucket-xg1hdqj601os	csv	10 January 2020 1:37 PM UTC-5	
sport_division	ticketdata	s3://dmslab-student-dmslslabs3bucket-xg1hdqj601os	csv	10 January 2020 1:37 PM UTC-5	
sport_league	ticketdata	s3://dmslab-student-dmslslabs3bucket-xg1hdqj601os	csv	10 January 2020 1:37 PM UTC-5	

Lab 2. ETL with AWS Glue

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 interface for a table named 'person'. The left sidebar contains navigation links for AWS Glue, Data catalog, Databases, Crawlers, Classifiers, Settings, ETL, Workflows, Jobs, ML Transforms, Triggers, Dev endpoints, Notebooks, Security, and Tutorials. The main area displays table metadata: Name (person), Description (ticketdata), Database (ticketdata), Classification (csv), Location (s3://dmelab-student-dmslab3bucket-xg1hdyqf0bs/tickets/dms_sample/person/), Connection (No), Last updated (Fri Jan 10 13:37:23 GMT+500 2020), Input format (org.apache.hadoop.mapred.TextInputFormat), Output format (org.apache.hadoop.mapreduce.lib.output.FileOutputFormat), and Serde serialization lib (org.apache.hadoop.mapreduce.lib.serde2.lazy.LazySimpleSerDe). Table properties include sizeKey (3666565990), objectCount (1), UPDATED_BY_CRAWLER (glue-lab-crawler), CrawlerSchemaSerializerVersion (1.0), recordCount (9164647), averageRecordSize (40), and CrawlerSchemaDeserializerVersion (1.0). Below this, the 'Schema' section lists four columns: col0, col1, col2, and col3, all of type string. A red box highlights the 'Edit schema' button in the top right corner of the main content area.

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**

The screenshot shows the 'Edit schema' dialog for the 'person' table. The left sidebar is identical to the previous screenshot. The main area shows the 'Edit schema' dialog with four columns: id, full_name, last_name, and first_name. A yellow circle highlights the 'Column name' column header. Another yellow circle highlights the 'id' entry in the first row. A red box highlights the 'Save' button in the top right corner of the dialog. The status bar at the bottom right indicates 'Showing: 1 - 4 of 4'.

4. 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"

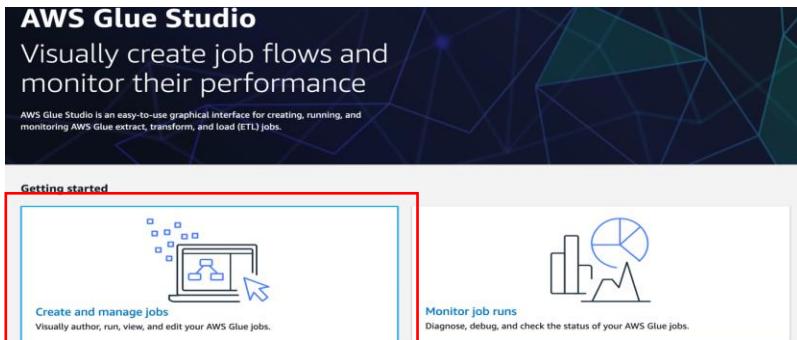
Lab 2. ETL with AWS Glue

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

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

Name	Database	Location	Classification	Last updated	Deprecation
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...	

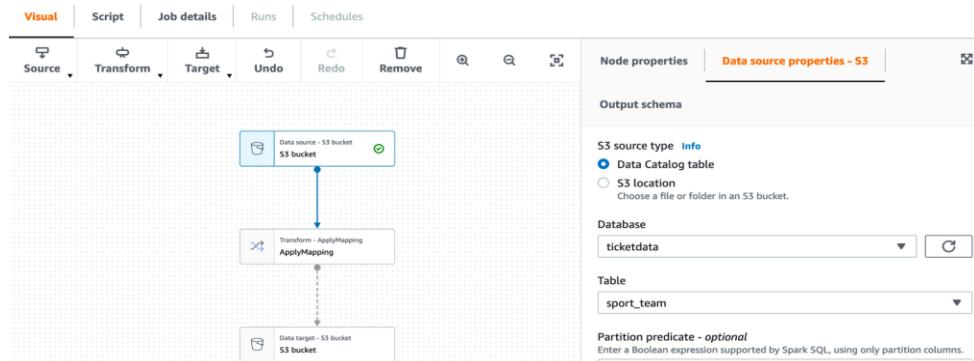
2. Choose "Create and Manage Jobs"



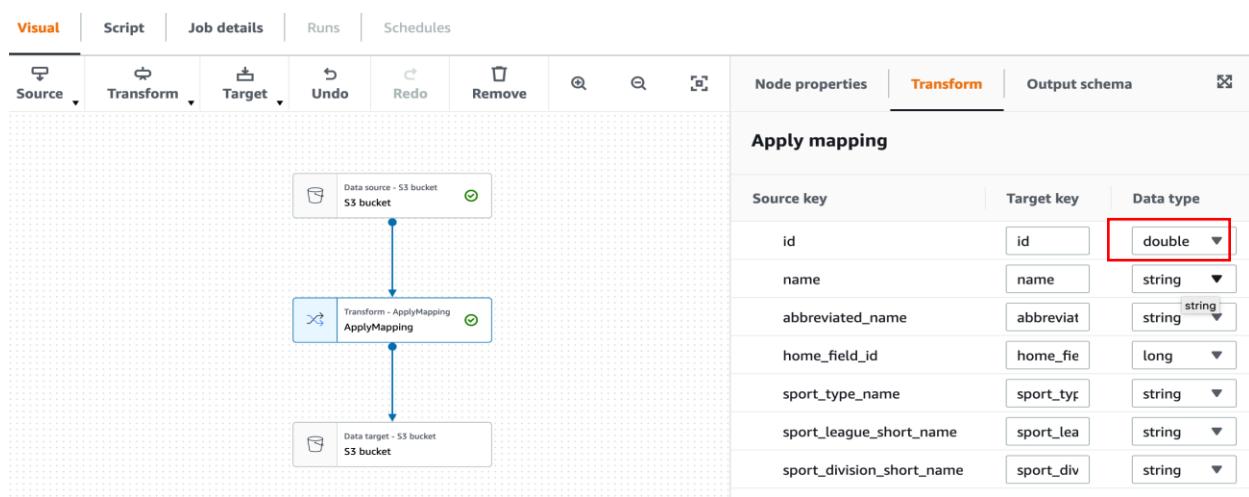
3. Leave the "Source and target added to the graph" option selected, and press "Create"

4. Select the "Data source - S3 bucket" at the top of the graph.
5. In the panel on the right under "Data source properties - S3", choose the "ticketdata" database from the drop down.
6. For Table, select the **sport_team** table.

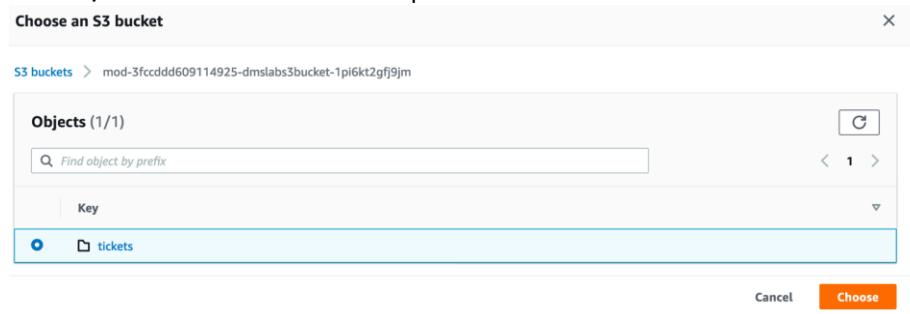
Lab 2. ETL with AWS Glue



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

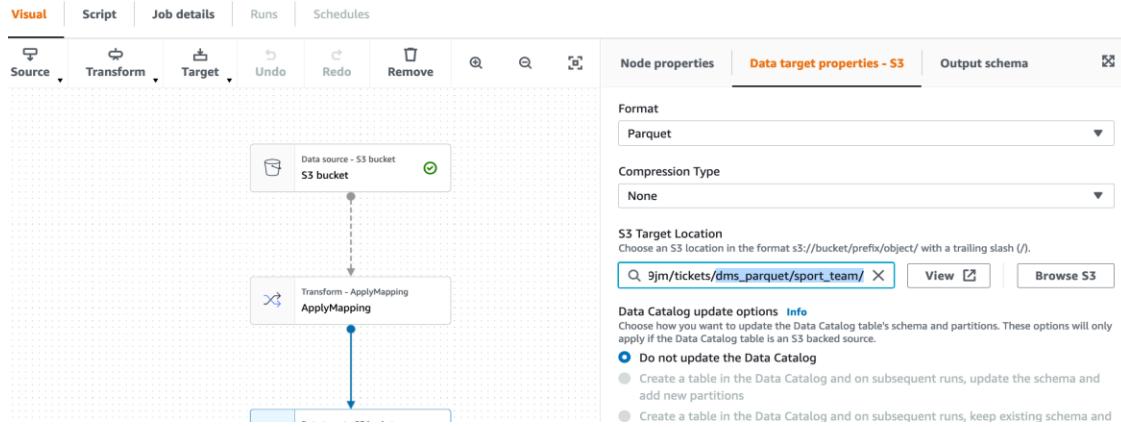


- Select the "Data target - S3 bucket" node at the bottom of the graph, and change the Format to **Parquet** in the dropdown.
- Under "S3 Target Location", select "**Browse S3**" browse to the "mod-xxx-dmslabs3bucket-xxx" bucket, select "**tickets**" item and press "**Choose**".



- 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.

Lab 2. ETL with AWS Glue



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.
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.

Lab 2. ETL with AWS Glue

Glue-Lab-SportTeamParquet 

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

Basic properties 

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 

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 

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

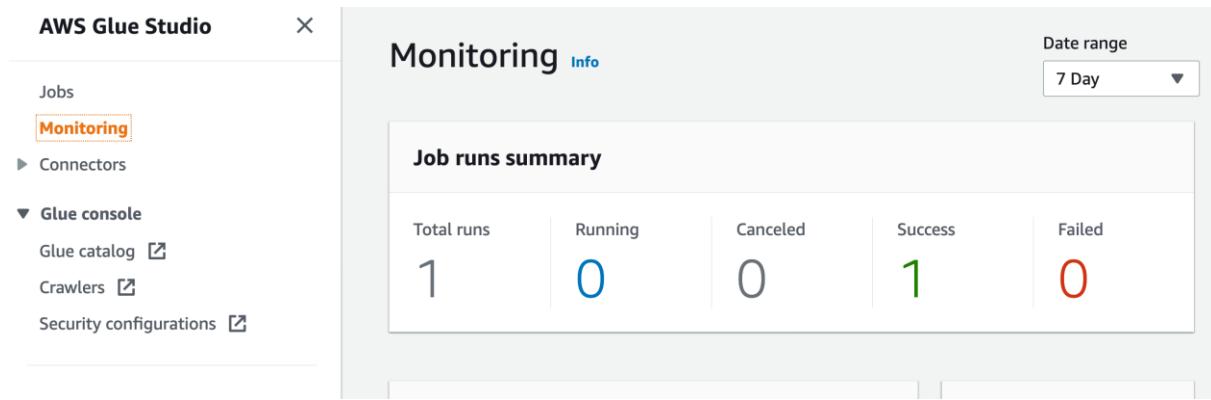
Number of retries
3

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

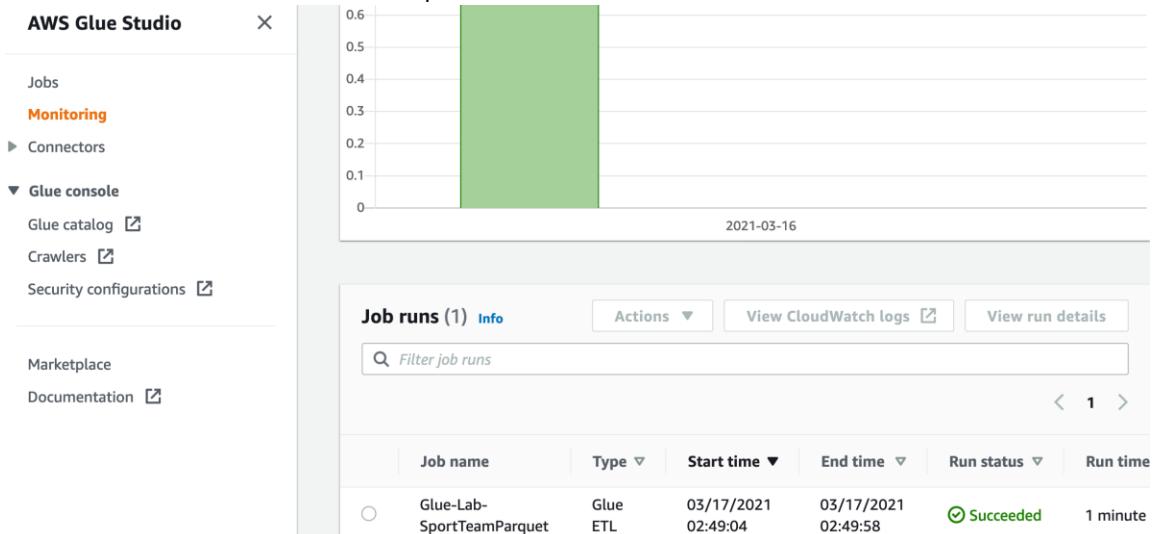
14. Press the “Save” button in the top right-hand corner to create the job.
15. Once you see the “Successfully created job” message in the banner, click the “Run” button to start the job.
16. Select “Jobs” from the navigation panel on the left-hand side to see a list of your jobs.

Lab 2. ETL with AWS Glue

17. Select “**Monitoring**” from the navigation panel on the left-hand side to view your running jobs, success/failure rates and various other statistics.



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.



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 shows two boxes: 'Source' (S3) and 'Target' (S3), connected by an arrow. Below this, a table lists 'Your jobs (1)' with one entry: 'Glue-Lab-SportTeamParquet' (Type: Glue ETL, Last modified: 3/17/2021, 2:48:53 AM). There are 'Actions' and 'Run job' buttons.

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 tranformations	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 tranformations	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

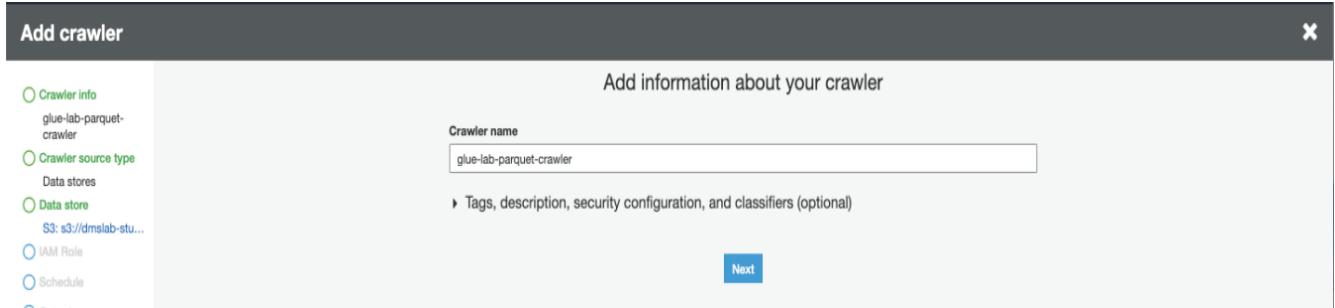
1. In the Glue Studio navigation menu, select **Crawlers** to open the Glue Crawlers page in a new tab. Click **Add crawler**.

The screenshot shows the AWS Glue Studio interface. On the left, there's a navigation sidebar with options like Data catalog, Databases, Tables, Connections, Crawlers (which is highlighted with a red box), Classifiers, and Settings. The main area is titled 'Crawlers' and contains a sub-header: '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 'Name : glue-lab-crawler' and a 'Filter or search for crawlers...' placeholder. There are also 'Run crawler' and 'Action' buttons. To the right, there's a 'User preferences' section and a status message 'Showing: 1 - 1'. The table below has columns: Name, Schedule, Status, Logs, Last runtime, Median runtime, Tables updated, and Tables added. One row is shown for 'glue-lab-crawler'.

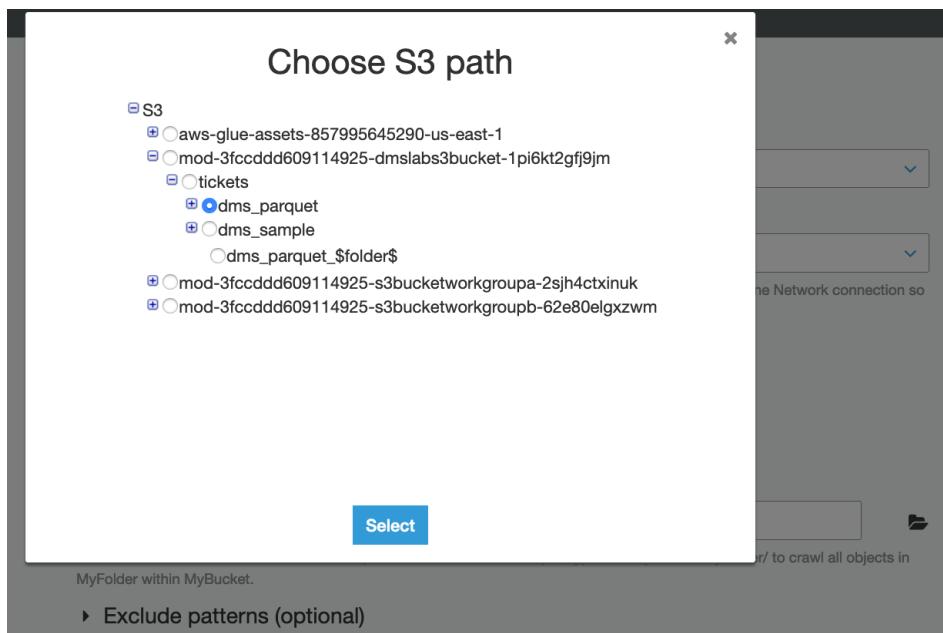
Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
glue-lab-crawler		Ready	Logs	1 min	1 min	0	15

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

Add another data store
 Yes
 No

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

Crawler info
glue-lab-parquet-crawler

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

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-7OEMGU9C9TZT

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**

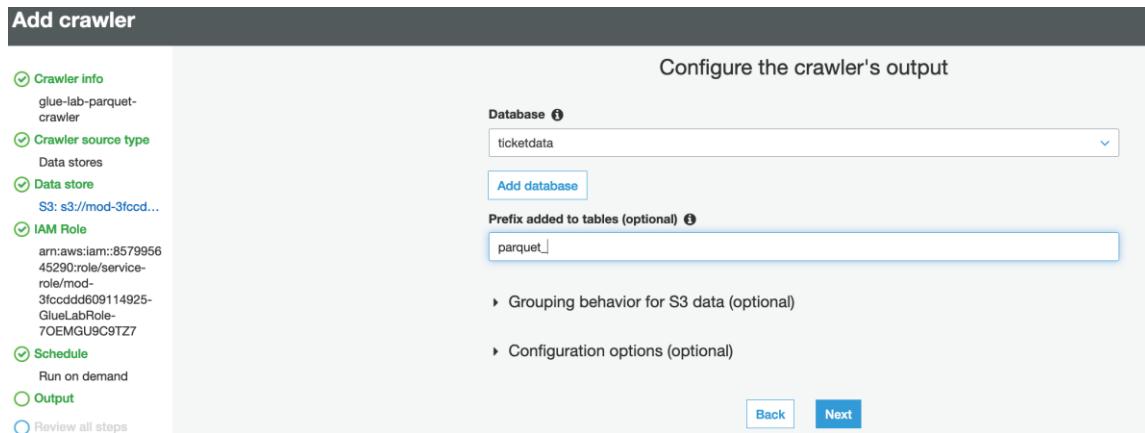
Lab 2. ETL with AWS Glue

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



8. For the crawler's output database, choose your existing database which you created earlier e.g. "ticketdata"

9. For the **Prefix added to tables** (optional), type "parquet_"



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

11. 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.

The screenshot shows the AWS Glue Data Catalog console under the 'Crawlers' section. On the left is a navigation sidebar with links for Data catalog, Databases, Tables, Connections, Crawlers (which is selected and highlighted in orange), Classifiers, Schema registries, and Schemas. The main area has a heading 'Crawlers' with a sub-instruction: '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' and a filter button 'Action'. A table lists two crawlers: 'glue-lab-crawler' and 'glue-lab-parquet-crawler'. The 'glue-lab-parquet-crawler' row has a checked checkbox in the first column. The table columns are Name, Schedule, Status, Logs, Last runtime, Median runtime, Tables updated, and Tables. The 'Run crawler' button for the 'glue-lab-parquet-crawler' row is highlighted with a red box. The status for both crawlers is 'Ready'. The 'Logs' column for 'glue-lab-parquet-crawler' is blue, indicating it's currently running or has run recently. The 'Last runtime' and 'Median runtime' columns show '1 min' and '0 secs' respectively. The 'Tables updated' and 'Tables' columns both show '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. A search filter 'Database : ticketdata' is applied. The results show several tables, with one group expanded to show five sub-tables under 'parquet_person'. These sub-tables are highlighted with a red rounded rectangle. The columns in the table are Name, Database, Location, Classification, Last updated, and Deprecated.

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. A message indicates 'You don't have any crawlers yet.' An 'Add crawler' button is visible. The columns in the table are Name, Schedule, Status, Logs, Last runtime, Median runtime, Tables updated, and Tables added.

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

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
Add information about your crawler

Crawler name
glue-lab-cdc-crawler

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

Next

Click **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:
- For **Choose a data store**, click the drop-down box and select **S3**.
 - For **Crawl data in**, select **Specified path in my account**.
 - 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-3fccddd609114925-dmslabs3bucket-1pi6kt2gfj9jm/cdc/dms_sample

Add another data store
 Yes
 No

Back Next

9. On the **Choose an IAM role** page, make the following selections:
 - Select **Choose an existing IAM role**.
 - 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 i

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/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-3fc...
 IAM Role
arn:aws:iam::857995645280:role/service-role/mod-3fcdd609114925-GlueLabRole-7OEMGU9C9TZ7

Schedule
Run on demand

Output
 Review all steps

Database ticketdata

Prefix added to tables (optional) cdc_

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. ⓘ

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

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. ⓘ

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.

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

Name	Schedule	Catalog type	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
<input checked="" type="checkbox"/> glue-lab-cdc-cra...		Glue	Ready	Logs	1 min	1 min	0	2
<input type="checkbox"/> 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.

Name	Database	Location	Classification	Last updated	Deprecated
<input type="checkbox"/> cdc_sporting_event_ticket	ticketdata	s3://dmslab-student-dmslabs3buck...	csv	23 January 2020 4:38 PM UTC-5	
<input type="checkbox"/> cdc_ticket_purchase_hist	ticketdata	s3://dmslab-student-dmslabs3buck...	csv	23 January 2020 4:38 PM UTC-5	
<input type="checkbox"/> mlb_data	ticketdata	s3://dmslab-student-dmslabs3buck...	csv	10 January 2020 1:37 PM UTC-5	
<input type="checkbox"/> name_data	ticketdata	s3://dmslab-student-dmslabs3buck...	csv	10 January 2020 1:37 PM UTC-5	
<input type="checkbox"/> nfl_data	ticketdata	s3://dmslab-student-dmslabs3buck...	csv	10 January 2020 1:37 PM UTC-5	
<input type="checkbox"/> nfl_stadium_data	ticketdata	s3://dmslab-student-dmslabs3buck...	csv	10 January 2020 1:37 PM UTC-5	
<input type="checkbox"/> parquet_person	ticketdata	s3://dmslab-student-dmslabs3buck...	parquet	23 January 2020 1:49 PM UTC-5	
<input type="checkbox"/> parquet_sport_location	ticketdata	s3://dmslab-student-dmslabs3buck...	parquet	23 January 2020 1:49 PM UTC-5	
<input type="checkbox"/> 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**.

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

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-3fcddd609114925-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)

- 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_hist... 
Transform type 
Data target 
Schema

Choose a data source

Showing: 1 - 29 

Name	Database	Location	Classification
bookmark_parquet_ticket_purchase_history	ticketdata	s3://dmslab-student-dmslab3bucket-vghdyqg0bs/cdc_bookmark/ticke...	parquet
cdc_sporting_event_ticket	ticketdata	s3://dmslab-student-dmslab3bucket-vghdyqg0bs/cdc/dms_sample/so...	csv
cdc_ticket_purchase_hist	ticketdata	s3://dmslab-student-dmslab3bucket-vghdyqg0bs/cdc/dms_sample/nc...	csv
clickstream_data	processed-data	s3://rawdataset-deshift/Clickstream_data/	json
csv_clickstream_data	processed-data	s3://processed-deshift/Clickstream-data/	csv

- 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
`>i6kt2gfj9jm/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

The screenshot shows the 'Add job' configuration screen for AWS Glue. On the left, there's a sidebar with options like 'Add properties', 'Glue Lab - TicketHistory-Parquet-with-bookmarks', 'Data source', 'Transform type', 'Change schema', 'Data target', and 'S3/dmslab-student... Schema'. The main area is titled 'Map the source columns to target columns.' It shows a table with two rows: 'Source' and 'Target'. The Source row lists columns from a database named 'ticketdata' with their types (e.g., string, double) and the mapping to target columns ('op', 'purchase_price'). The Target row lists the target columns with their types (string, double) and the mapping back to source columns. Buttons at the bottom include 'Back', 'Save and edit script', 'Add column', 'Clear', and 'Reset'.

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.

The screenshot shows the AWS Glue job editor. At the top, it says 'Job: Glue-Lab-TicketHistory-Parquet-with-bookmark'. Below that is a 'Job Flow' section with four stages: 'Data Name ticketdata' (with 'Table Name cdc_ticket_purchase_hist'), 'Transform Name ApplyMapping' (with 'Transform type Map', 'Name applymapping1', 'Script type Python'), 'Transform Name ResolveChoice' (with 'Transform type Map', 'Name resolvechoice1', 'Script type Python'), and 'Transform Name DropNullFields' (with 'Transform type Map', 'Name dropnullfields1', 'Script type Python'). To the right of the flow is the generated Python code:

```

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 ## Libraries: [DYNAMIC]
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 #datasource = glueContext.create_dynamic_frame.from_catalog(database = "ticketdata", table_name = "cdc_ticket_purchase_hist", transformation_ctx = "datasource0")
19 #datasource = datasource0.resolvechoice("applymapping1", transformation_ctx = "datasource1")
20 #datasource = glueContext.create_dynamic_frame.from_catalog(database = "ticketdata", table_name = "cdc_ticket_purchase_hist", transformation_ctx = "datasource0")
21 ## # Stage 1: Apply Mapping
22 #applymapping1 = ApplyMapping.apply(frame = datasource0, mappings = [{"op": "string", "args": ["string", "string", "string"]}, {"op": "string", "args": ["string", "string", "string"]}], transformation_ctx = "applymapping1")
23 #applymapping1 = applymapping1.resolvechoice("resolvechoice1", transformation_ctx = "applymapping1")
24 #applymapping1 = applymapping1.resolvechoice("dropnullfields1", transformation_ctx = "applymapping1")
25 #applymapping1 = applymapping1.resolvechoice("dropnullfields1", transformation_ctx = "applymapping1")
26 ## # Stage 2: Resolve Choice
27 #resolvechoice1 = ResolveChoice.apply(frame = applymapping1, choice = "make_struct", transformation_ctx = "resolvechoice1")
28 #resolvechoice1 = resolvechoice1.resolvechoice("dropnullfields1", transformation_ctx = "resolvechoice1")
29 #resolvechoice1 = resolvechoice1.resolvechoice("dropnullfields1", transformation_ctx = "resolvechoice1")
30 #resolvechoice1 = resolvechoice1.resolvechoice("dropnullfields1", transformation_ctx = "resolvechoice1")
31 #resolvechoice1 = resolvechoice1.resolvechoice("dropnullfields1", transformation_ctx = "resolvechoice1")
32 ## # Stage 3: Drop Null Fields
33 #dropnullfields1 = DropNullFields.apply(frame = resolvechoice1, transformation_ctx = "dropnullfields1")
34 #dropnullfields1 = dropnullfields1.resolvechoice("dropnullfields1", transformation_ctx = "dropnullfields1")
35 #dropnullfields1 = dropnullfields1.resolvechoice("dropnullfields1", transformation_ctx = "dropnullfields1")
36 #dropnullfields1 = dropnullfields1.resolvechoice("dropnullfields1", transformation_ctx = "dropnullfields1")
37 #dropnullfields1 = dropnullfields1.resolvechoice("dropnullfields1", transformation_ctx = "dropnullfields1")
38 #dropnullfields1 = dropnullfields1.resolvechoice("dropnullfields1", transformation_ctx = "dropnullfields1")
39 ## # Stage 4: Write Parquet
40 datasource1 = glueContext.create_dynamic_frame.from_options(frame_options = "dropnullfields1", connection_type = "s3", connection_options = {"path": "s3://dmslab-student-dmslslabs3bucket-xg1hydq60bs/cdc_bookmark/ticket_purchase_history/data"}, format = "parquet", transformation_ctx = "datasource1")
41 datasource1.write.parquet("s3://dmslab-student-dmslslabs3bucket-xg1hydq60bs/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.

The screenshot shows the AWS S3 console. The path is 'Amazon S3 > dmslab-student-dmslslabs3bucket-xg1hydq60bs > cdc_bookmark > ticket_purchase_history > data'. The 'Overview' tab is selected. A search bar at the top says 'Type a prefix and press Enter to search. Press ESC to clear.' Below it are buttons for 'Upload', '+ Create folder', 'Download', and 'Actions'. The status bar indicates 'US East (N. Virginia)' and 'Viewing 1 to 2'. A table lists two objects:

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

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

Lab 2. ETL with AWS Glue

The screenshot shows the AWS Glue console's Crawler list page. On the left, a sidebar menu includes options like Tables, Connections, Crawlers (which is selected and highlighted in orange), Classifiers, Schema registries, Schemas, and Settings. The main area displays a table with columns: Name, Schedule, Status, and Logs. Two crawlers are listed:

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

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 in my account
 Specified path in another account

Include path
3://mod-3fcddd609114925-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

Add another data store

Crawler info
glue_lab_cdc_bookmark_crawler

Crawler source type
Data stores

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-3fcddd609114925-GlueLabRole-7OEMGU9C9TZ7

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

- s3://mod-3fcddd609114925-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:
- 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**.

Name	Schedule
glue-lab-crawler	
glue-lab-parquet-crawler	
glue_lab_cdc_bookmark_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**”

Tables						
Name		Database	Location	Classification	Last updated	Deprecated
bookmark_parquet_ticket_purchase_history		ticketdata	s3://dmstlab-student-dmlslabs3bucket-xg1hdyg90b...	parquet	24 January 2020 7:14 PM UTC-5	
cdc_sporting_event_ticket		ticketdata	s3://dmstlab-student-dmlslabs3bucket-xg1hdyg90b...	csv	24 January 2020 5:13 PM UTC-5	
cdc_ticket_purchase_hist		ticketdata	s3://dmstlab-student-dmlslabs3bucket-xg1hdyg90b...	csv	24 January 2020 5:13 PM UTC-5	
mrb_data		ticketdata	s3://dmstlab-student-dmlslabs3bucket-xg1hdyg90b...	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

A screenshot of the Amazon Athena Settings page. It shows a "Sources" section and a "Workgroup : primary" section. A callout box highlights the "Query result location" field, which contains the placeholder text "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](#)

A screenshot of a "Query result location" settings dialog box. It has a "Query result location" input field containing "s3://query-results-bucket/folder/" with a help icon. Below it is an "Example: s3://query-results-bucket/folder/" placeholder. There are two checkboxes: "Encrypt query results" and "Autocomplete", both with help icons. At the bottom are "Cancel" and "Save" buttons.

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:

Lab 2. ETL with AWS Glue

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

Before moving on to next step, note the rowcount.

Step 4: Generate CDC data and to observe bookmark functionality

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

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

Name	Last modified	Size	Storage class
part-00000-052927b4-2fbc-4fc2-0249-d9100b75ddad-c000.snappy.parquet	Jan 24, 2020 9:20:13 PM GMT-0500	9.3 KB	Standard
part-00000-49fe37c7-2e11-4f7f-9431-de1f5e2fa3d3-c000.snappy.parquet	Jan 24, 2020 7:03:16 PM GMT-0500	1.1 MB	Standard
part-00000-d166f723-315b-05fb-868e-a6f239402348-c000.snappy.parquet	Jan 25, 2020 11:24:20 PM GMT-0500	1.7 MB	Standard
part-00000-4ec020f9-dc40-4fc6-8230-c3b2d8151ba-c000.snappy.parquet	Jan 25, 2020 10:24:27 PM GMT-0500	7.2 KB	Standard
part-00001-052927b4-2fbc-4fc2-0249-d9100b75ddad-c000.snappy.parquet	Jan 24, 2020 9:20:27 PM GMT-0500	66.5 KB	Standard
part-00001-49fe37c7-2e11-4f7f-9431-de1f5e2fa3d3-c000.snappy.parquet	Jan 24, 2020 7:03:16 PM GMT-0500	1.2 MB	Standard
part-00002-d166f723-315b-05fb-868e-a6f239402348-c000.snappy.parquet	Jan 25, 2020 11:24:20 PM GMT-0500	1.7 MB	Standard
part-00002-052927b4-2fbc-4fc2-0249-d9100b75ddad-c000.snappy.parquet	Jan 24, 2020 9:20:18 PM GMT-0500	1.7 MB	Standard
part-00002-d166f723-315b-05fb-868e-a6f239402348-c000.snappy.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";
```

To review the latest transactions, run:

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

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

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

Overview:

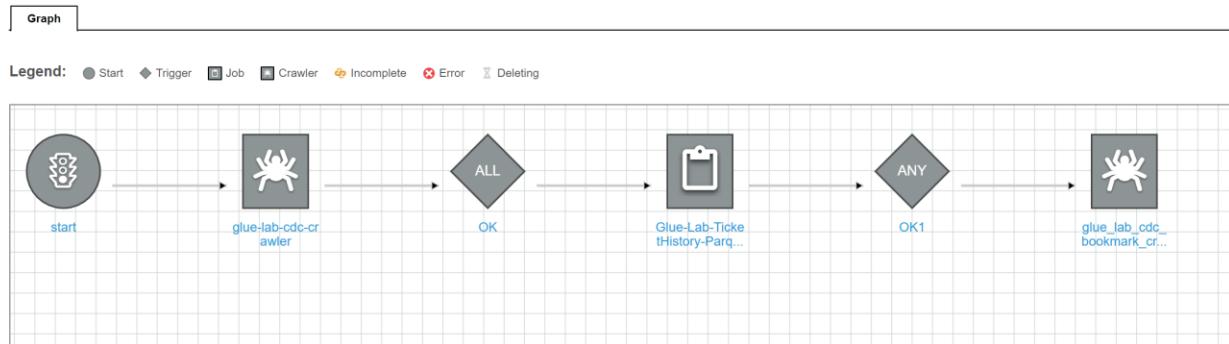
In AWS Glue, you can use workflows to create and visualize complex extract, transform, and load (ETL) activities involving multiple crawlers, jobs, and triggers. Each workflow manages the execution and monitoring of all its components. As a workflow runs each component, it records execution progress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

Creating and Running Workflows:

Above mentioned Part A (ETL with Glue) and Part B (Glue Job Bookmarks) can be created and executed using workflows. Complex ETL jobs involving multiple crawlers and jobs can also be created and executed using workflows in an automated fashion. Below is a simple example to demonstrate how to create and run workflows.

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

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



To create a workflow:

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

Lab 2. ETL with AWS Glue

Workflows (0)

A workflow is an orchestration used to visualize and manage the relationship and execution of multiple triggers, jobs and crawlers.

Add workflow Actions Filter workflows < 1 >

Name	Last run	Last run status	Last modified
No workflows			
Add a new ETL workflow			

ETL
Workflows
Jobs
ML Transforms

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.

Workflows (1)

A workflow is an orchestration used to visualize and manage the relationship and execution of multiple triggers, jobs and crawlers.

Add workflow Actions Filter workflows

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

Graph Details History

Legend: Start Trigger Job Crawler Incomplete Error Deleting

[Add trigger](#)

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

Add trigger

Clone existing Add new

Name
trigger1

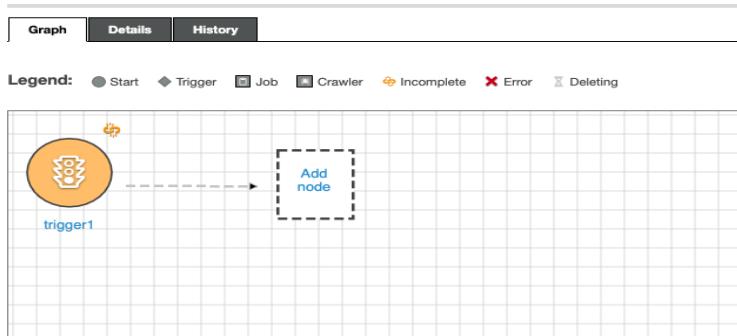
Description (optional)
Trigger to start the workflow

Trigger type
 Schedule Event On demand

Cancel Add

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**

Add trigger

Name: trigger2

Description (optional): Trigger to execute crawler

Trigger type: Event Schedule On demand

Trigger logic: Start after ANY watched event Start after ALL watched event

Add

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

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"

The screenshot shows the AWS Glue Workflows console. On the left, there's a sidebar with 'Classifiers', 'Settings', 'ETL', and 'Workflows'. The 'Workflows' section is currently selected. In the main area, there's a table with columns: 'Name', 'Last run', 'Last run status', and 'Last modified'. One row is visible, labeled 'Run'. Above the table, there's a toolbar with 'Add workflow', 'Actions' (with a dropdown menu), and a search bar. The 'Actions' dropdown has 'Run' highlighted.

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