



Amazon Web Services

Data Engineering Immersion Day

Lab 2. ETL with AWS Glue

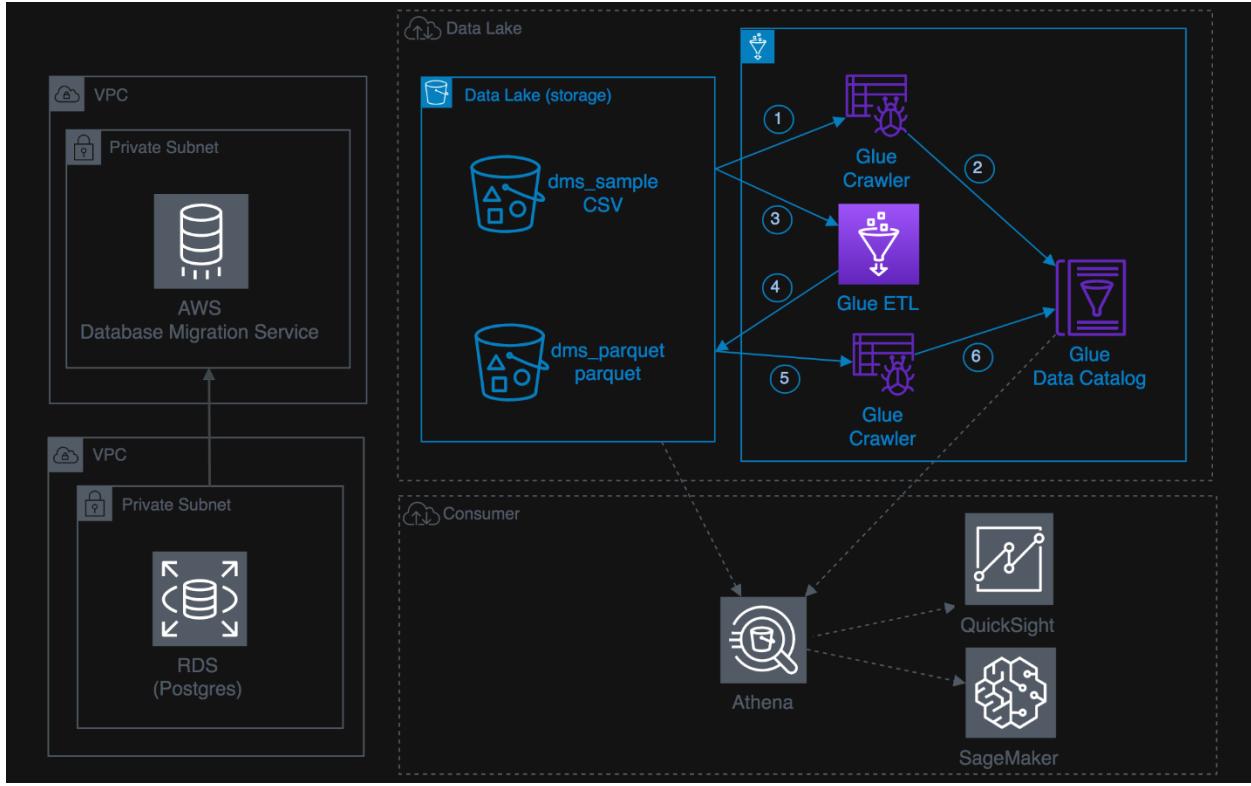
August 2020

Table of Contents

<i>Introduction</i>	2
<i>Get Started Using the Lab Environment</i>	3
<i>PART A: Data Validation and ETL</i>	6
Create Glue Crawler for initial full load data	6
Data Validation Exercise.....	10
Data ETL Exercise	11
Create Glue Crawler for Parquet Files	16
<i>PART B: Glue Job Bookmark (Optional):</i>	20
Step 1: Create Glue Crawler for ongoing replication (CDC Data).....	20
Step 2: Create a Glue Job with Bookmark Enabled	24
Step 3: Create Glue crawler for Parquet data in S3	27
Step 4: Generate CDC data and to observe bookmark functionality	30
<i>PART C: Glue Workflows (Optional, self-paced)</i>	31
Overview:.....	31
Creating and Running Workflows:	31

Introduction

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



Prerequisites

- Completed Lab 1. Hydrating the Data Lake with DMS

Tasks Completed in this Lab:

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

- [PART-\(A\): Data Validation and ETL](#)
- [PART- \(B\): Glue Job Bookmark Functionality\(Optional\)](#)
- [PART- \(C \): Glue Workflows\(Optional\)](#)

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

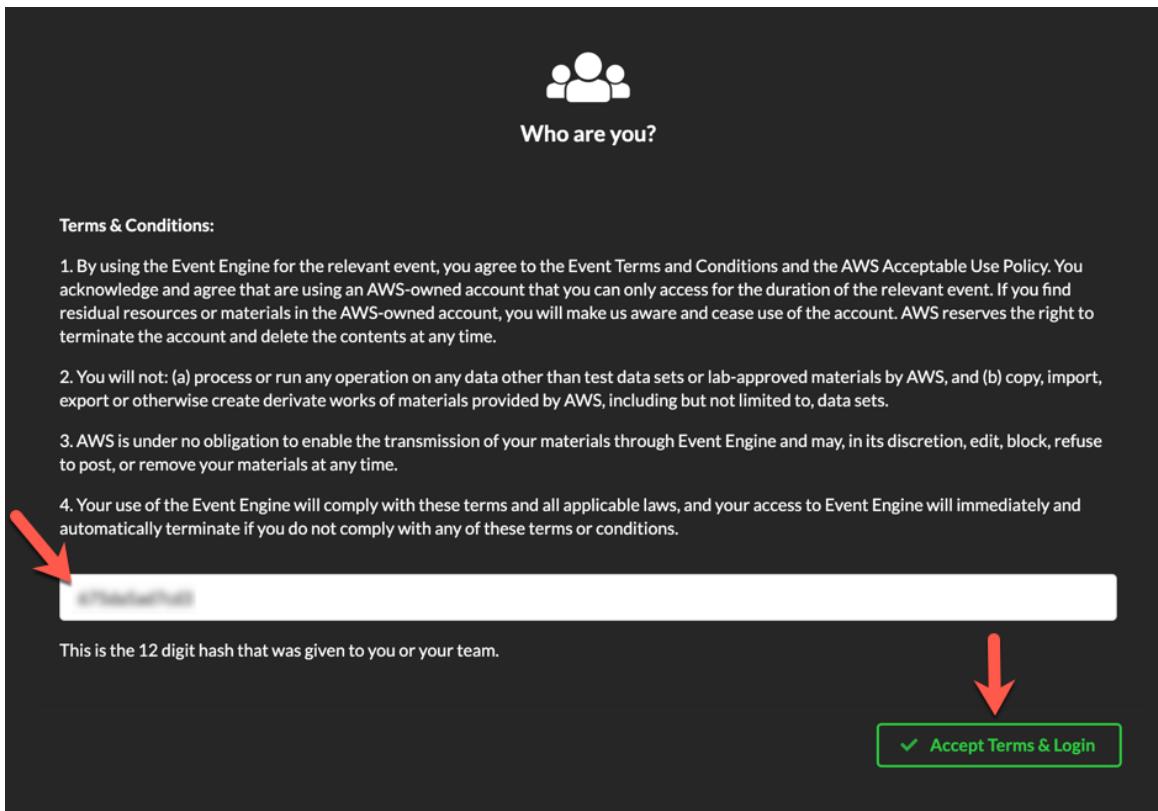
Get Started Using the Lab Environment

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-1ngcgzznd15u

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

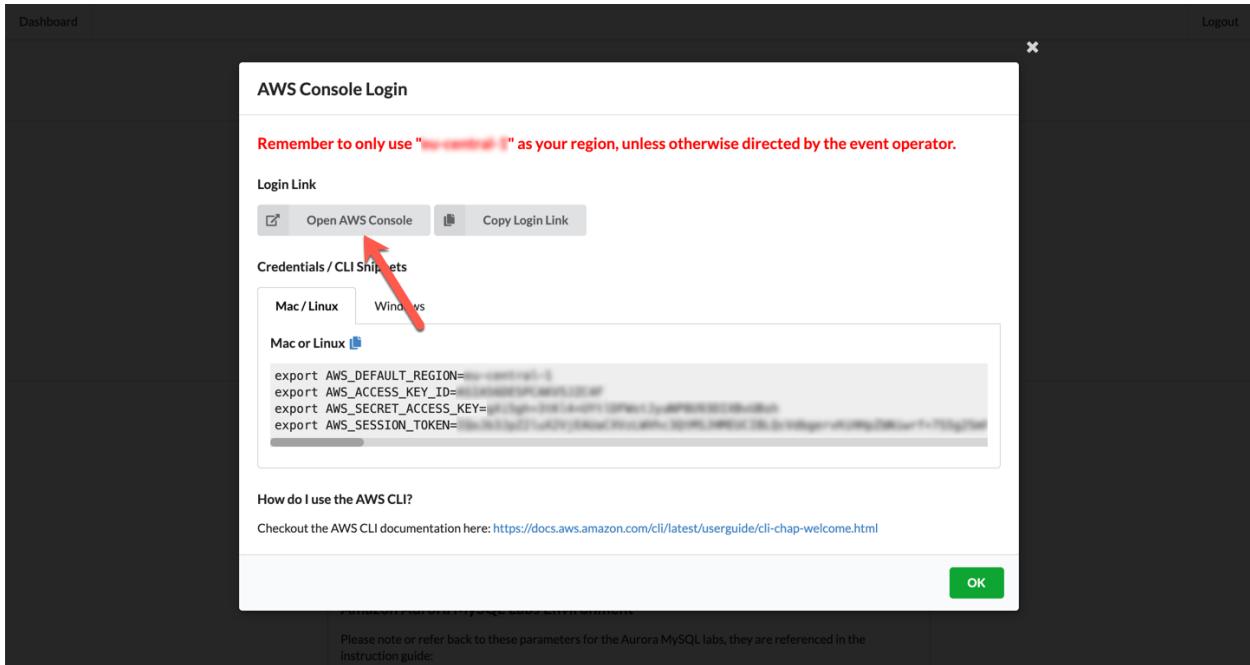
AWS Console SSH Key

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. 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", and a "All services" button.

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

The screenshot shows the "Crawlers" page under the AWS Glue service. The left sidebar has a "Crawlers" item 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 top states, "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 the table, a message says, "You don't have any crawlers yet." An "Add crawler" button is located at the bottom right of the table area.

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" step 1: Crawler info. On the left, there is a sidebar with options: Crawler info (selected), Crawler source type, Data store, IAM Role, Schedule, Output, and Review all steps. The main area has a title "Add information about your crawler". It includes a "Crawler name" field containing "glue-lab-crawler" and a note: "Tags, description, security configuration, and classifiers (optional)". At the bottom right is a "Next" button.

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

The screenshot shows the "Add crawler" step 2: Crawler source type. The sidebar on the left shows "Crawler info" (selected) and "Crawler source type" (also selected). The main area has a title "Specify crawler source type" and a note: "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." It includes a "Crawler source type" section with "Data stores" (selected) and "Existing catalog tables" options. At the bottom are "Back" and "Next" buttons.

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

Lab 2. ETL with AWS Glue

- a. For Choose a data store, click the drop-down box and select **S3**.
 - b. For Crawl data in, select **Specified path in my account**.
 - c. For Include path, browse to the target folder for your DMS initial export from Lab 1, e.g., **s3://dmslab-student-dmslabs3bucket-wotl4bf73cw3/tickets**
8. Click **Next**.

Add crawler

Choose a data store

Crawl data in

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.

Exclude patterns (optional)

Back Next

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

Add crawler

Add another data store

Chosen data stores

Yes No

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 "dmslab-student-GlueLabRole-ZOQDII7JTBUM"
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.

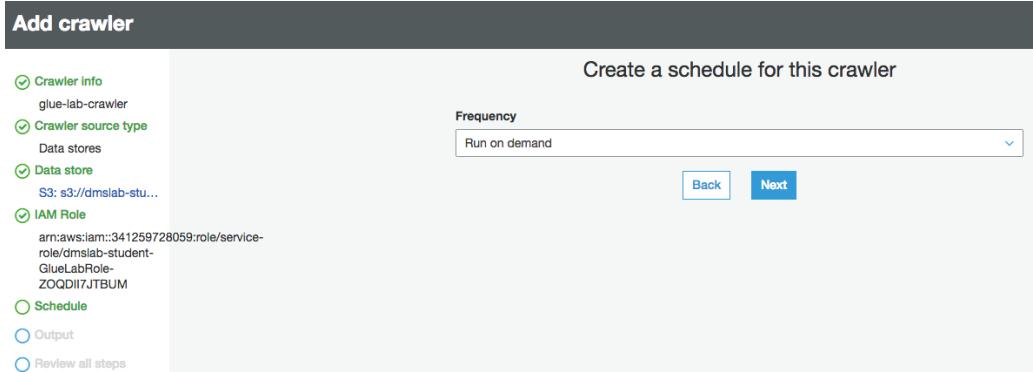
s3://dmslab-student-dmslabs3bucket-wotl4bf73cw3

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

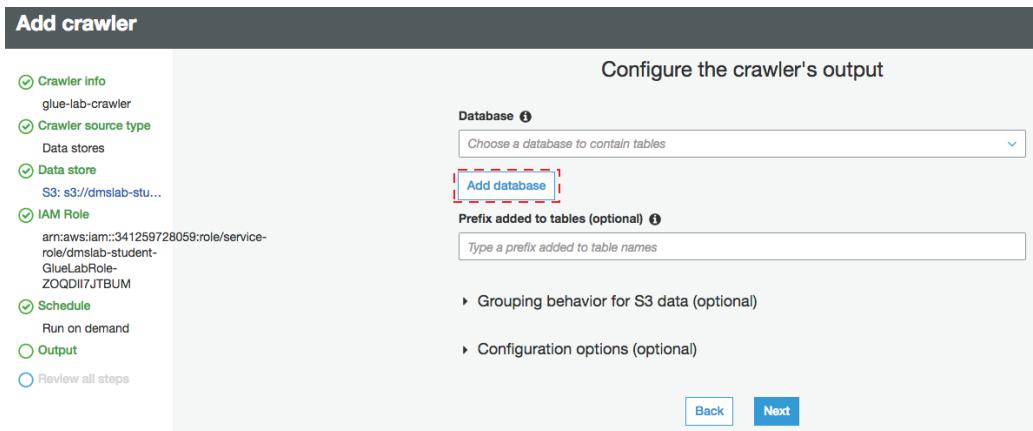
Back Next

Lab 2. ETL with AWS Glue

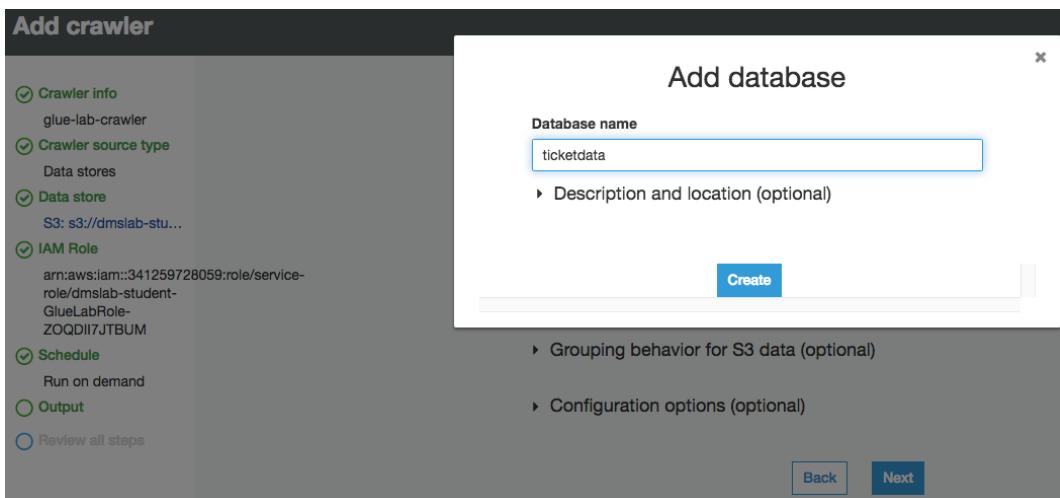
12. On the Create a schedule for this crawler page, for Frequency, select **Run on demand** and Click **Next**.



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



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



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

Lab 2. ETL with AWS Glue

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

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

18. Click **Run it now**.

Lab 2. ETL with AWS Glue

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.

The screenshot shows the AWS Glue interface with the 'Crawlers' section selected. A message at the top indicates that the crawler 'glue-lab-crawler' completed and created 15 tables. The table below lists the crawler details:

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

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

The screenshot shows the AWS Glue interface with the 'Tables' section selected under 'Database: ticketdata'. The table lists 15 tables with their names, databases, locations, and last update times:

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

Data Validation Exercise

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

The screenshot shows the AWS Glue interface with the 'Tables' section selected under 'Database: ticketdata'. The table lists 15 tables, with the 'person' table highlighted by a red box. The columns are Name, Database, Location, Classification, Last updated, and Deprecated.

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

Lab 2. ETL with AWS Glue

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

The screenshot shows the AWS Glue Table Editor for a table named 'person'. In the top right corner, there is a red box around the 'Edit schema' button. The table properties section shows the following details:

Name	person
Description	
Database	tickerdata
Classification	csv
Location	abf://dmelab-student-dmelabs3buciel-xg1hdyof0bs/tickers/dmz_sample/person/
Connection	No
Deprecated	No
Last updated	Fri Jan 10 13:37:23 GMT-500 2020
Input format	org.apache.hadoop.mapred.TextInputFormat
Output format	org.apache.hadoop.hive.io.HiveIgnoreKeyTextOutputFormat
Serde serialization lib	org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
Serde parameters	field.delim = ,
Table properties	sizeKey 366656890 objectCount 1 UPDATED_BY_CRAWLER glue-lab-crawler CrawlerSchemaSerializerVersion 1.0 recordCount 9164647 averageRecordSize 40 CrawlerSchemaDeserializerVersion 1.0 compressionType none columnsOrdered true areColumnsQuoted false delimiter , typeOfData file

The 'Schema' section lists four columns:

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

- 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.

The screenshot shows the 'Edit schema' interface for the 'person' table. A yellow circle highlights the 'Column name' column header. The table data is as follows:

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

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

The screenshot shows the AWS Glue Jobs page. The left navigation pane has 'Jobs' selected. The main area displays the following information:

Jobs A job is your business logic required to perform extract, transform and load (ETL) work. Job runs are initiated by triggers which can be scheduled or driven by events.

User preferences

Action ▾ Filter by attributes

Showing: 0 - 0

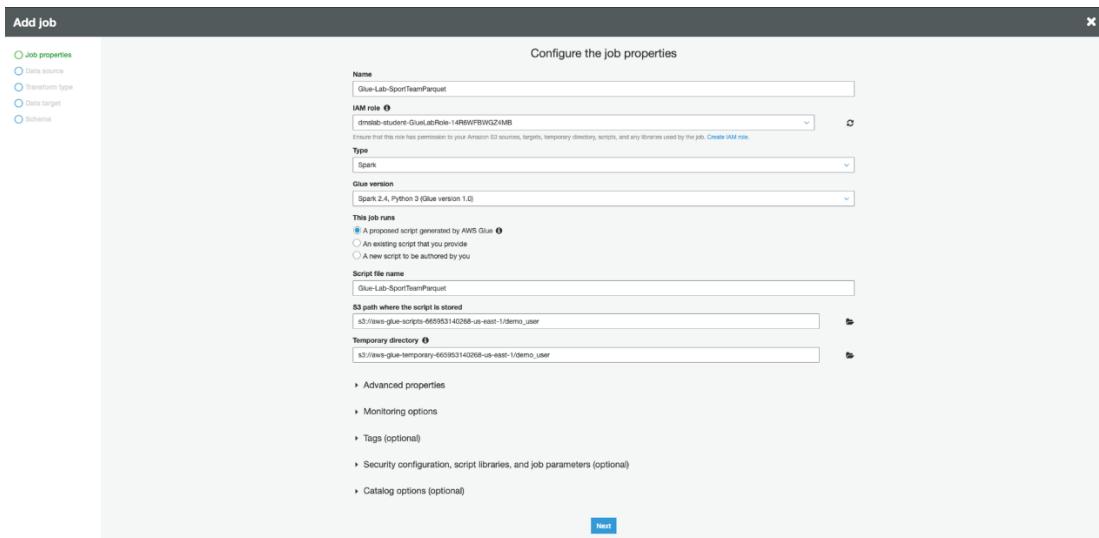
A table with columns: Name, ETL language, Script location, Last modified, and Job bookmark. There are no rows in the table.

You don't have any jobs defined yet.

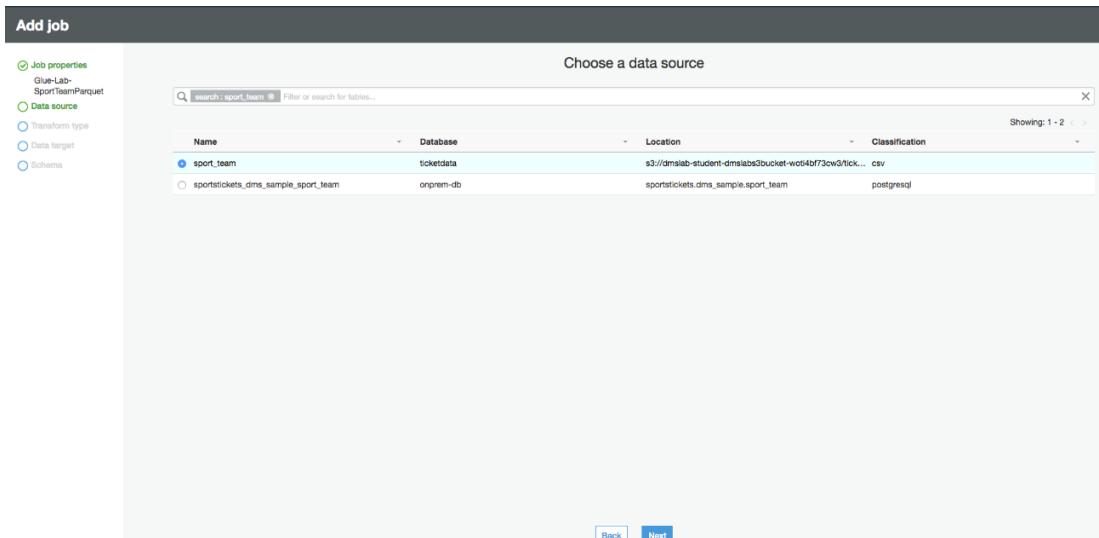
Add job

Lab 2. ETL with AWS Glue

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



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



Lab 2. ETL with AWS Glue

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

The screenshot shows the 'Add job' wizard with the title 'Choose a transform type'. On the left, there is a sidebar with the following options:

- Job properties: Glue-Lab-SportTeamParquet
- Data source: sport_team
- Transform type: Change schema (this is the selected option)
- Data target
- Schema

On the right, the 'Change schema' option is selected, with the sub-instruction: 'Change schema of your source data and create a new target dataset'. Below it is the 'Find matching records' option. At the bottom are 'Back' and 'Next' buttons.

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

The screenshot shows the 'Add job' wizard with the title 'Choose a data target'. On the left, the sidebar includes:

- Job properties: Glue-Lab-SportTeamParquet
- Data source: sport_team
- Transform type: Change schema
- Data target (this is the selected option)
- Schema

On the right, the 'Create tables in your data target' option is selected. It also shows the 'Data store' set to 'Amazon S3', 'Format' set to 'Parquet', and the 'Target path' field containing 's3://dmslab-student-dmslabs3bucket-xg1hdyq60ls/ticke'. At the bottom are 'Back' and 'Next' buttons.

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

The screenshot shows the 'Add job' wizard with the title 'Data target'. The sidebar shows the 'Data target' section with the target path 's3://dmslab-student-dmslabs3bucket-xg1hdyq60ls/ticke'. The main area displays a table of column mappings:

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

A modal window titled 'String type' is open over the table, focusing on the 'id' row. The 'Column type' dropdown is highlighted and set to 'double'. At the bottom of the modal are 'Update' and 'Cancel' buttons.

Lab 2. ETL with AWS Glue

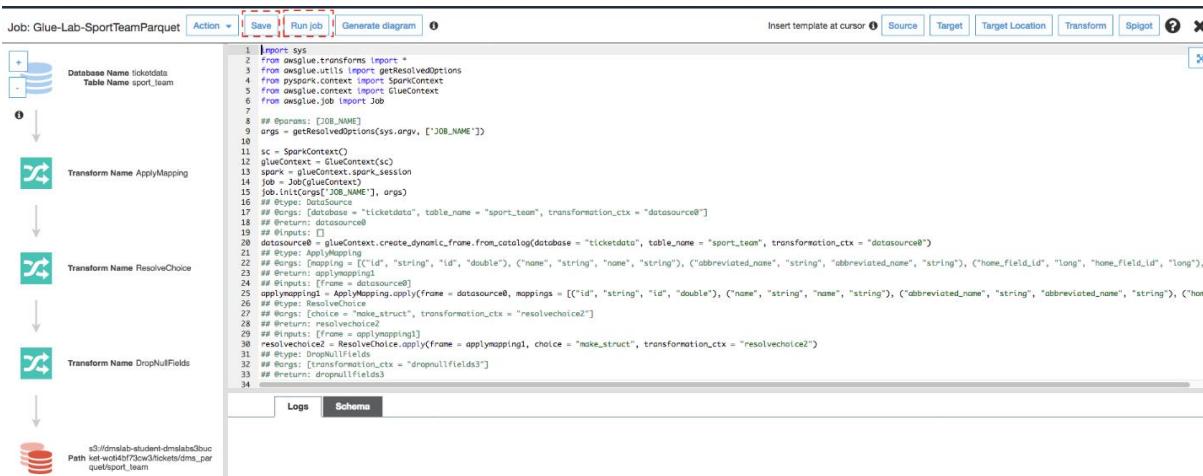
Map the source columns to target columns.

Verify the mappings created by AWS Glue. Change mappings by choosing other columns with **Map to target**. You can **Clear** all mappings and **Reset** to default AWS Glue mappings. AWS Glue generates your script with the defined mappings.

Source			Target				
Column name	Data type	Map to target	Column name	Data type			
id	string	id	id	double	x	▼	▲
name	string	name	name	string	x	▼	▲
abbreviated_name	string	abbreviated_name	abbreviated_name	string	x	▼	▲
home_field_id	bigint	home_field_id	home_field_id	long	x	▼	▲
sport_type_name	string	sport_type_name	sport_type_name	string	x	▼	▲
sport_league_short_name	string	sport_league_short_name	sport_league_short_name	string	x	▼	▲
sport_division_short_name	string	sport_division_short_name	sport_division_short_name	string	x	▼	▲

Add column **Clear** **Reset**

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



14. In **Parameters** option,

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

Parameters (optional)

Review and override parameter values, as needed, before running this job. Changes affect this run only. Edit a job to change default parameter values.

▼ Advanced properties

Job bookmark ⓘ

▼

▼ Monitoring options

Job metrics ⓘ

Continuous logging

Spark UI ⓘ

► Tags

► Security configuration, script libraries, and job parameters

Only job **Glue-Lab-SportTeamParquet** is run. Jobs dependent on the completion of job **Glue-Lab-SportTeamParquet** will not be run. To run a job and trigger dependent jobs, define an on-demand trigger.

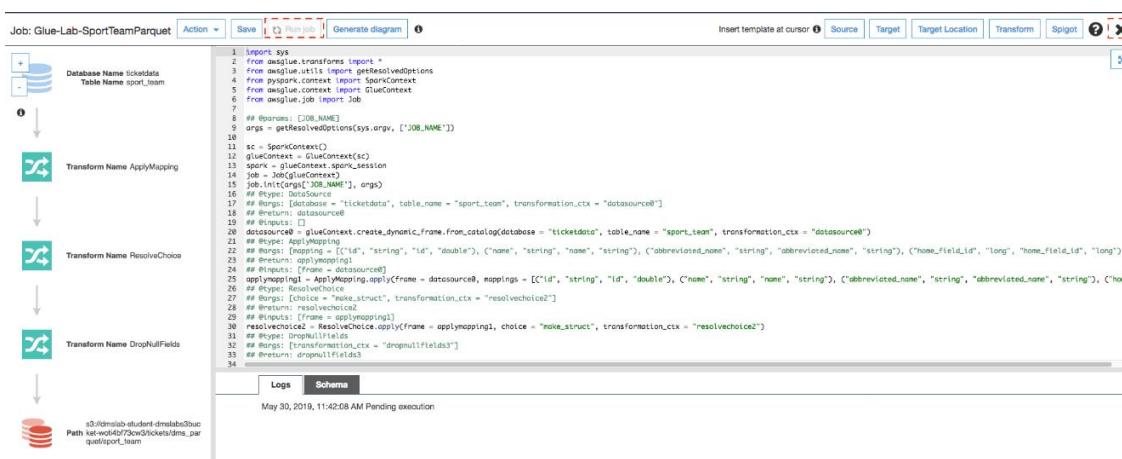
Run job

Lab 2. ETL with AWS Glue

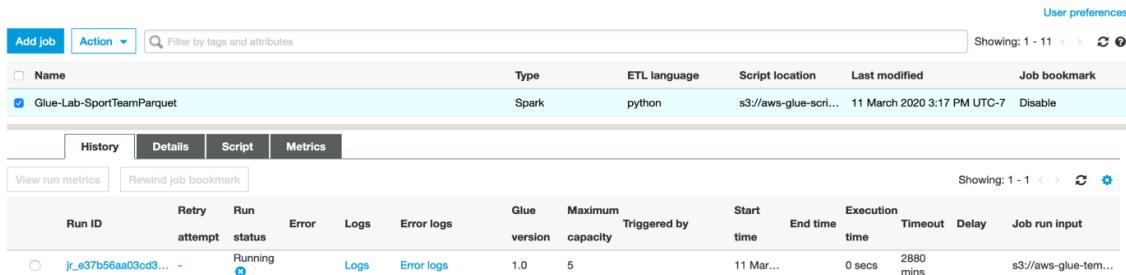
- b. You can leave the **Job metrics** option **Unchecked**. You can collect metrics about AWS Glue jobs and visualize them on the AWS Glue with job metrics.

15. Click Run Job

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



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



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

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

Table 1:

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

Lab 2. ETL with AWS Glue

Table 2:

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

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

Create Glue Crawler for Parquet Files

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

The screenshot shows the AWS Glue interface under the 'Crawlers' section. A red box highlights the 'Add crawler' button. The crawler 'glue-lab-crawler' is listed with a status of 'Ready'. Other columns include Schedule, Status, Logs, Last runtime, Median runtime, Tables updated, and Tables added.

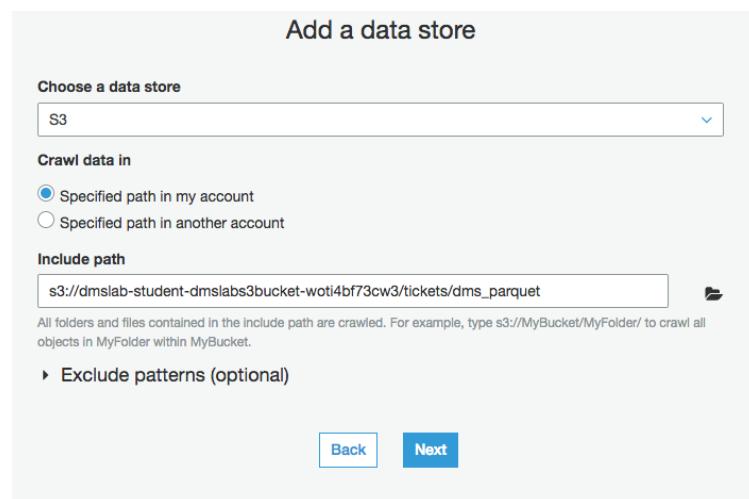
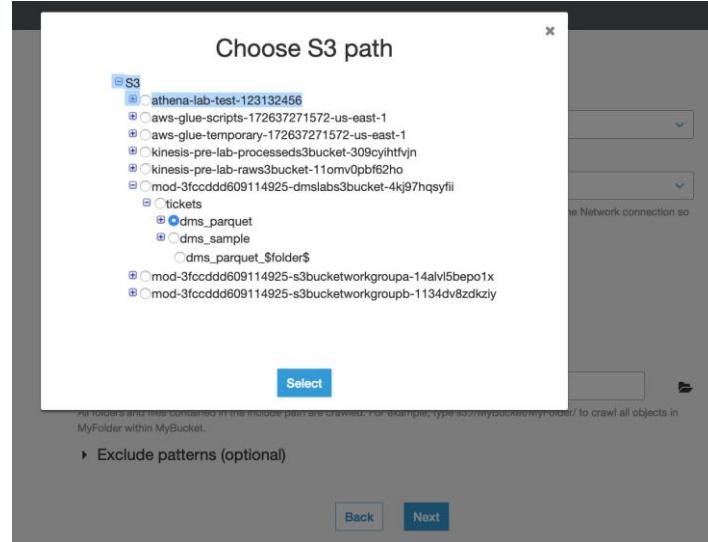
Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
glue-lab-crawler		Ready	Logs	1 min	1 min	0	15

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

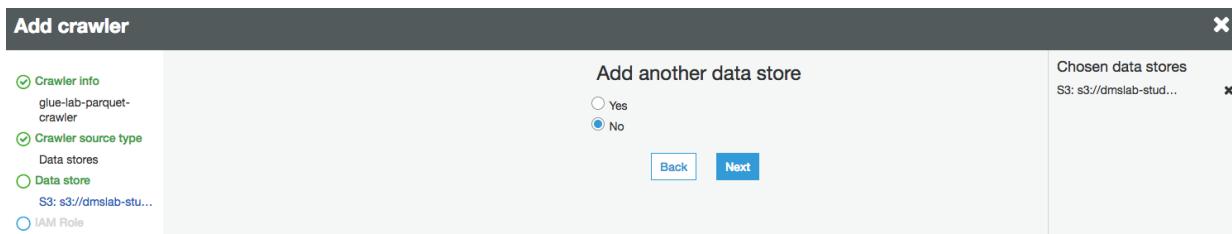
The screenshot shows the 'Add crawler' configuration screen. The 'Crawler source type' section is expanded, showing 'Data stores' selected. The 'Crawler name' field contains 'glue-lab-parquet-crawler'. The 'Next' button is visible at the bottom right.

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

Lab 2. ETL with AWS Glue



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



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

Lab 2. ETL with AWS Glue

Add crawler

Crawler info
glue-lab-parquet-crawler

Crawler source type
Data stores

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

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

Schedule

Output

Review all steps

Choose an IAM role

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

Update a policy in an IAM role

Choose an existing IAM role

Create an IAM role

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

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

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

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

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

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

Add crawler

Crawler info
glue-lab-parquet-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](#)

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

Crawler info
glue-lab-parquet-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 [?](#)
ticketdata

Add database

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

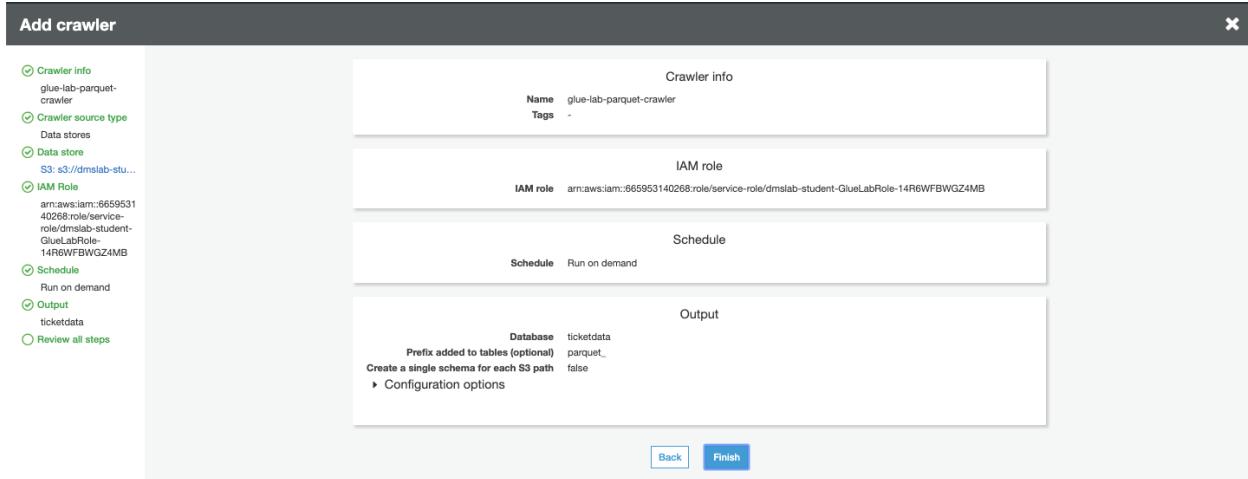
▶ Grouping behavior for S3 data (optional)

▶ Configuration options (optional)

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

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

Lab 2. ETL with AWS Glue



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

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

Confirm you can see the tables:

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

Name	Database	Location	Classification	Last updated	Deprecated
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	ticketdata	s3://dmslab-student-dmslabs3buck... parquet		23 January 2020 1:49 PM UTC-5	
parquet_sporting_event	ticketdata	s3://dmslab-student-dmslabs3buck... parquet		23 January 2020 1:49 PM UTC-5	
parquet_sporting_event_ticket	ticketdata	s3://dmslab-student-dmslabs3buck... parquet		23 January 2020 1:49 PM UTC-5	
person	ticketdata	s3://dmslab-student-dmslabs3buck... csv		10 January 2020 1:48 PM UTC-5	

PART B: Glue Job Bookmark (Optional):

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

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

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

1. On the AWS Glue menu, select Crawlers.

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

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

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

Lab 2. ETL with AWS Glue

Add crawler

Crawler info
glue-lab-cdc-crawler

Crawler source type
Data stores

Data store

IAM Role

Schedule

Output

Review all steps

Add a data store

Choose a data store
S3

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

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

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

Exclude patterns (optional)

Back Next

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

Add crawler

Crawler info
glue-lab-cdc-crawler

Crawler source type
Data stores

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

IAM Role

Schedule

Output

Review all steps

Add another data store

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

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**.

Add crawler

Crawler info
glue-lab-cdc-crawler

Crawler source type
Data stores

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

IAM Role
dmslab-student-GlueLabRole-14R6WFBWGZ4MB

Choose an IAM role

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

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

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

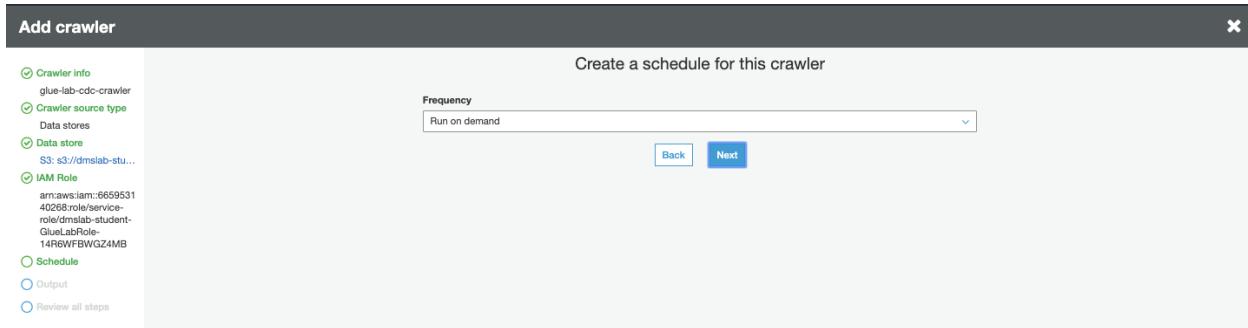
This role must provide permissions similar to the AWS managed policy, **AWSGlueServiceRole**, plus access to your data stores.
s3://dmslab-student-dmslabs3bucket-xg1hydq60lbs/cdc/dms_sample

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

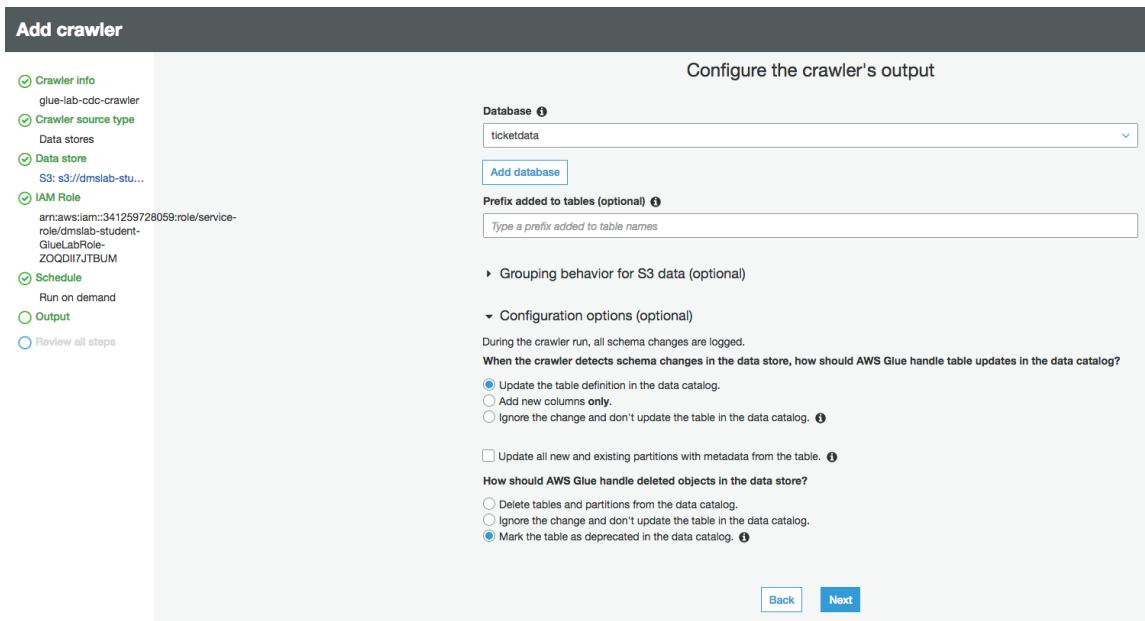
Back Next

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

Lab 2. ETL with AWS Glue



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



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

Lab 2. ETL with AWS Glue

Crawler info

Name: glue-lab-cdc-crawler
Tags: -

IAM role

IAM role: arn:aws:iam:665953140268:role/service-role/dmslab-student-GlueLabRole-14R6WFBWGZ4MB

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. Click Run it now.

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.

Crawler glue-lab-cdc-crawler was created to run on demand. Run it now?

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

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

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.

Crawler "glue-lab-cdc-crawler" completed and made the following changes: 2 tables created, 0 tables updated. See the tables created in database ticketdata.

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

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

Lab 2. ETL with AWS Glue

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

Step 2: Create a Glue Job with Bookmark Enabled

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

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

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

Lab 2. ETL with AWS Glue

Add job

Configure the job properties

Name	Glue-Lab-TicketHistory-Parquet-with-bookmark
IAM role	dmslab-student-GlueJobRole-14R6WFBWGZ4MB
Type	Spark
Glue version	Spark 2.4, Python 3 (Glue Version 1.0)
This job runs	
<input checked="" type="radio"/> A proposed script generated by AWS Glue <small>(1)</small> <input type="radio"/> An existing script that you provide <input type="radio"/> 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-665993140268-us-east-1/deshtut1
Temporary directory	s3://aws-glue-temporary-666993140268-us-east-1/deshtut1
Advanced properties	
Job bookmark	Enable
Monitoring options	
<input checked="" type="checkbox"/> Job metrics <small>(1)</small> <input type="checkbox"/> Continuous logging <input type="checkbox"/> Spark UI <small>(1)</small>	
Tags (optional)	

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

Add job

Choose a data source

Name	Database	Location	Classification
bookmark_parquet_ticket_purchase_history	ticketdata	s3://dmslab-student-dmslabs3bucket-xg1hdyg0ls/cdc_bookmark/ticket.../parquet	
cdc_sporting_event_ticket	ticketdata	s3://dmslab-student-dmslabs3bucket-xg1hdyg0ls/obcdms_sample/.../csv	
cdc_ticket_purchase_hist	ticketdata	s3://dmslab-student-dmslabs3bucket-xg1hdyg0ls/obcdms_sample/.../csv	
clickstream_data	processed-data	s3://rawdataset-deshtut1/Clickstream_data/	json
csv_clickstream_data	processed-data	s3://processed-deshtut1/clickstream-data/	csv

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

Add job

Choose a transform type

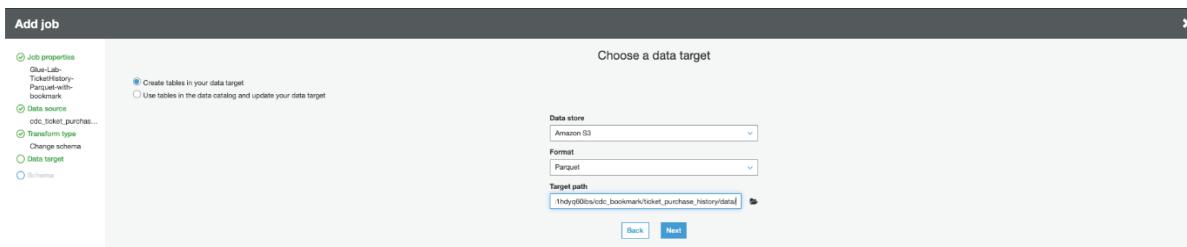
<input checked="" type="radio"/> Change schema	Change schema of your source data and create a new target dataset
<input type="radio"/> Find matching records	Use machine learning to find matching records within your source data

Back **Next**

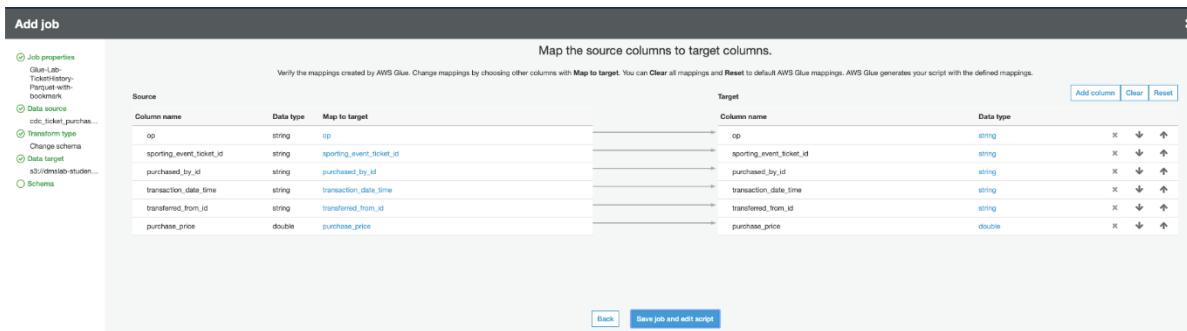
8. In Choose a data target:

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

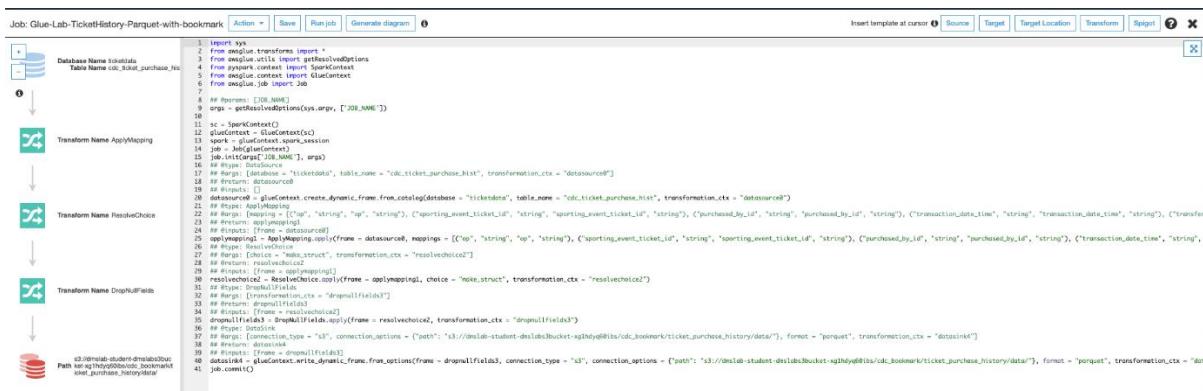
Lab 2. ETL with AWS Glue



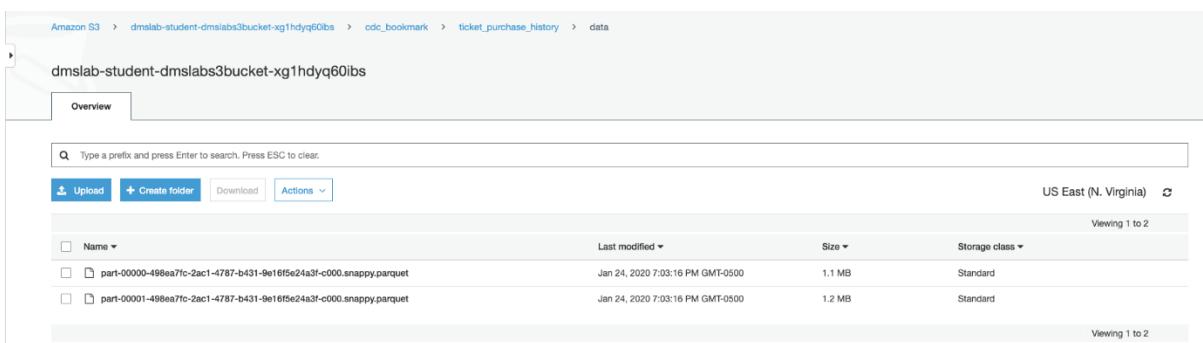
9. In map the source columns to target columns window, leave everything default and Click on **Save job and edit script**.



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



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

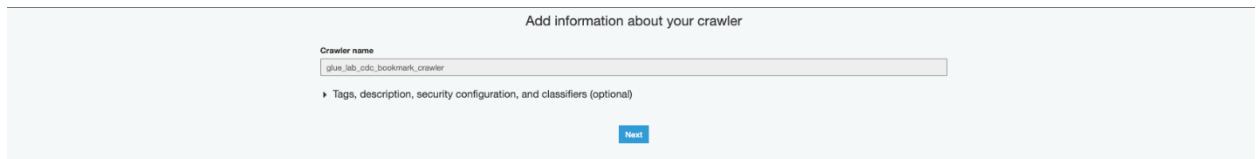


Lab 2. ETL with AWS Glue

Step 3: Create Glue crawler for Parquet data in S3

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

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



Add information about your crawler

Crawler name
glue_lab_cdc_bookmark_crawler

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

Next

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



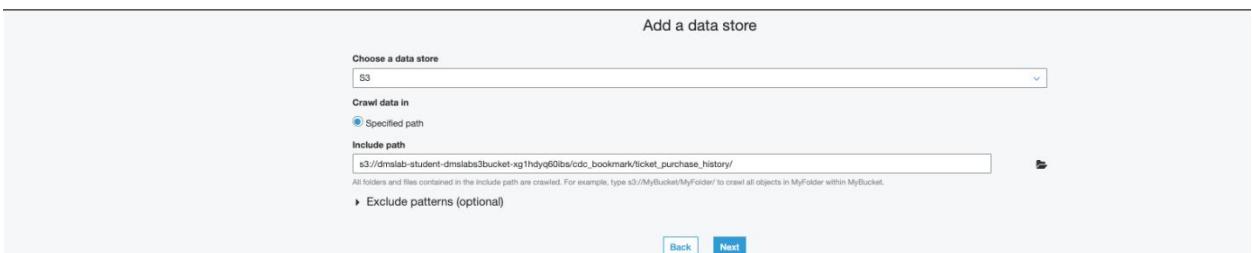
Specify crawler source type

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

Crawler source type
 Data stores
 Existing catalog tables

Back Next

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



Add a data store

Choose a data store
S3

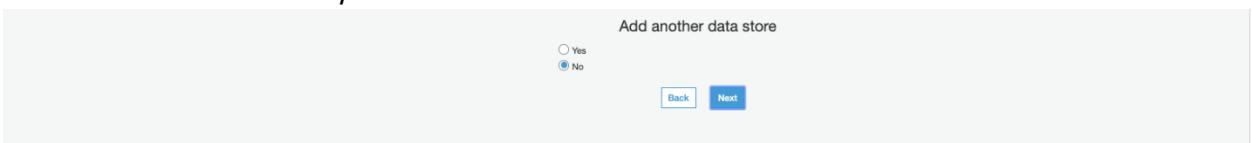
Crawl data in
 Specified path
Include path
`s3://dmslab-student-dmslabs3bucket-xg1hdyq60bs/cdc_bookmark/ticket_purchase_history/`

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

Exclude patterns (optional)

Back Next

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



Add another data store

Yes
 No

Back Next

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

Lab 2. ETL with AWS Glue

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

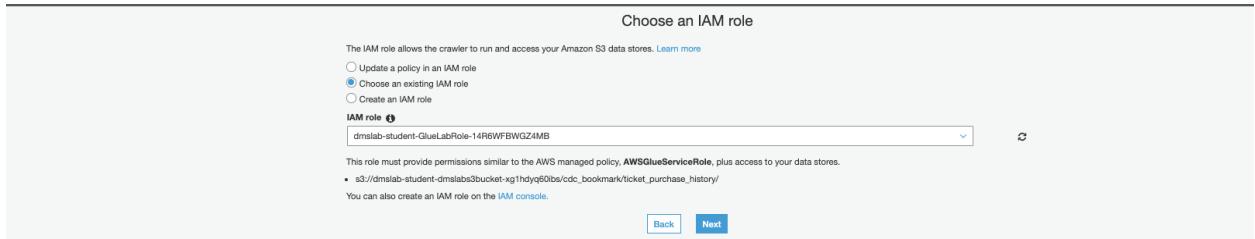
IAM role [?](#)
dmlab-student-GlueLabRole-14R8WF8WQZ4MB

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

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

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

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

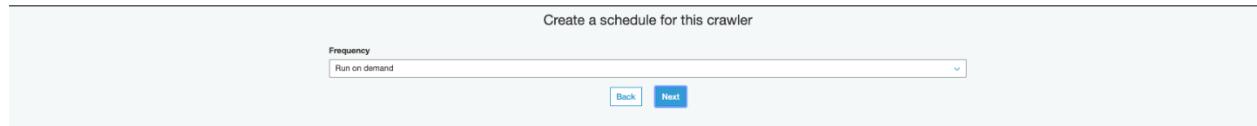


9. For setting the **frequency** in create a schedule for this crawler, select "Run on demand". Click **Next**

Create a schedule for this crawler

Frequency
Run on demand

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



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

Configure the crawler's output

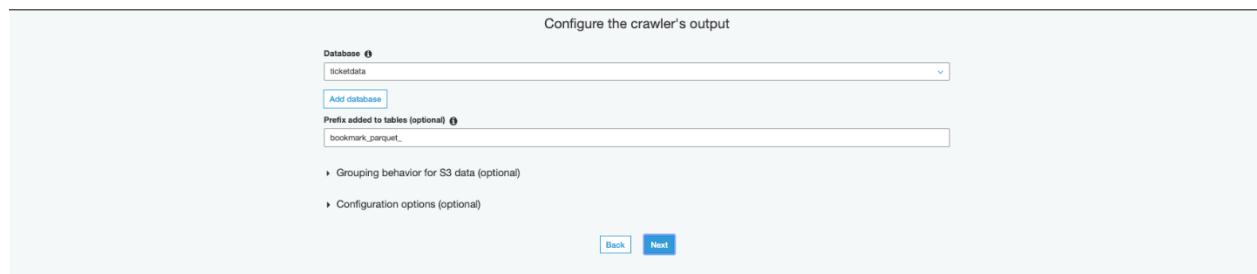
Database [?](#)
ticketdata

Add database

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

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

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



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

Crawler info
glue_lab_cdc_bookmark_crawler

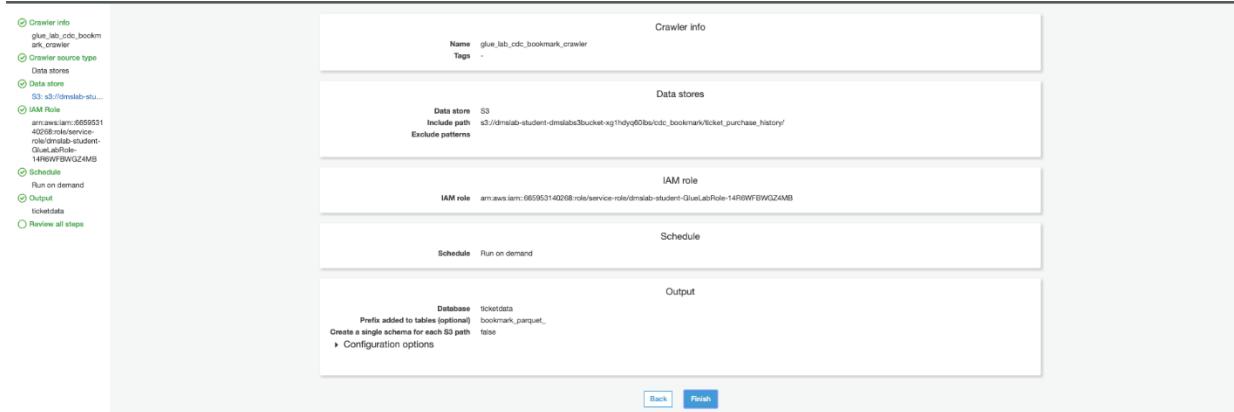
Data stores
S3: s3://dmlab-student-dmlabs3bucket-xg1hdqg60bs/cdc_bookmark/ticket_purchase_history/

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

Schedule
Run on demand

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

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



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**"

Lab 2. ETL with AWS Glue

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

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

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



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

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

Query result location: Example: s3://query-results-bucket/folder/

Encrypt query results:

Autocomplete:

Cancel **Save**

Lab 2. ETL with AWS Glue

To select some rows from the table, try running:

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

To get a row count, run:

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

Before moving on to next step, note the rowcount.

Step 4: Generate CDC data and to observe bookmark functionality

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

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

Name	Last modified	Size	Storage class
part-00000-05e20504-2fbc-47c2-8240-d9100b75edad-c000.snappy.parquet	Jan 24, 2020 9:20:13 PM GMT-0500	9.3 KB	Standard
part-00000-49feba7c-29c1-47f7-9431-9e159e4f4a39-c000.snappy.parquet	Jan 24, 2020 7:52:16 PM GMT-0500	1.1 MB	Standard
part-00000-d1666723-3158-43f8-b68e-a65238402348-c000.snappy.parquet	Jan 25, 2020 11:24:29 PM GMT-0500	1.7 MB	Standard
part-00000-05e20504-2fbc-47c2-8240-d9100b75edad-c000.snappy.parquet	Jan 25, 2020 10:24:27 PM GMT-0500	7.2 KB	Standard
part-00000-05e20504-2fbc-47c2-8240-d9100b75edad-c000.snappy.parquet	Jan 24, 2020 9:20:13 PM GMT-0500	66.5 KB	Standard
part-00001-49feba7c-29c1-47f7-9431-9e159e4f4a39-c000.snappy.parquet	Jan 24, 2020 7:53:16 PM GMT-0500	1.2 MB	Standard
part-00001-d1666723-3158-43f8-b68e-a65238402348-c000.snappy.parquet	Jan 25, 2020 11:24:29 PM GMT-0500	1.7 MB	Standard
part-00002-05e20504-2fbc-47c2-8240-d9100b75edad-c000.snappy.parquet	Jan 24, 2020 9:20:15 PM GMT-0500	1.7 MB	Standard
part-00002-d1666723-3158-43f8-b68e-a65238402348-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_data";
```

To review the latest transactions, run:

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

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

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

Overview:

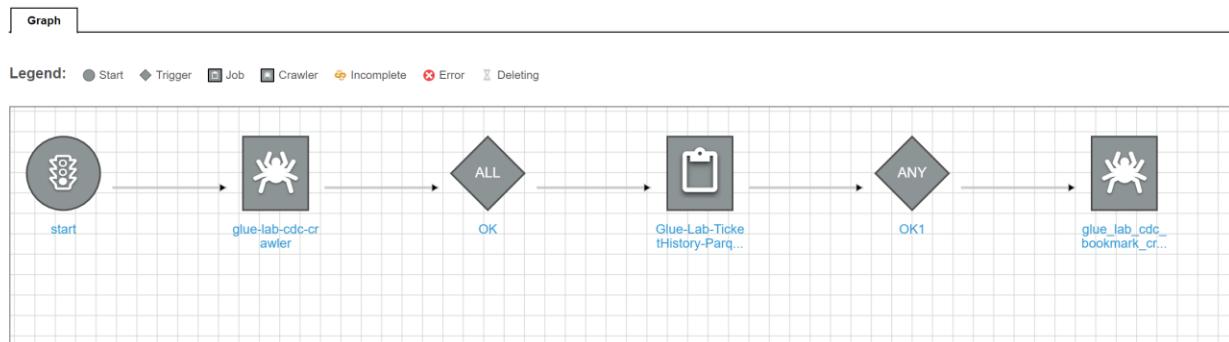
In AWS Glue, you can use workflows to create and visualize complex extract, transform, and load (ETL) activities involving multiple crawlers, jobs, and triggers. Each workflow manages the execution and monitoring of all its components. As a workflow runs each component, it records execution progress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

Creating and Running Workflows:

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

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

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



To create a workflow:

Lab 2. ETL with AWS Glue

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

Workflows (0)
A workflow is an orchestration used to visualize and manage the relationship and execution of multiple triggers, jobs and crawlers.

Name	Last run	Last run status	Last modified
No workflows			

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.

Workflow is empty
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

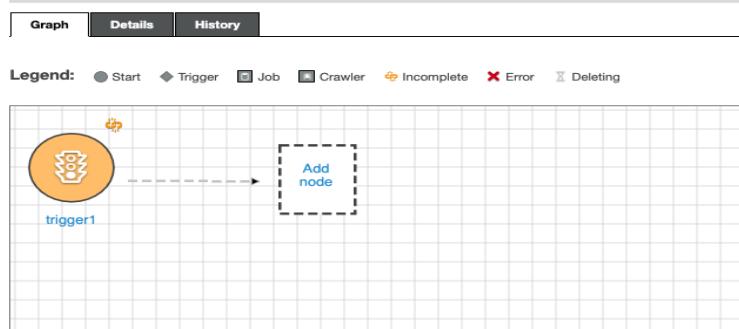
Description (optional)
Trigger to start the workflow

Trigger type
 Schedule Event On demand

Cancel Add

Lab 2. ETL with AWS Glue

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



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

The screenshot shows the 'Add trigger' dialog box. At the top, it says 'Add trigger' and has a close button. Below that are two buttons: '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:
 - Name:** trigger3
 - Description:** Trigger to execute crawler
 - Trigger type:** Event

Lab 2. ETL with AWS Glue

- d. **Trigger logic: Start after ANY watched event.** This will make sure that crawler starts once Glue job finishes processing of ALL data.
 - e. Click **Add**
10. Click on **Add node**, Select the Crawler **glue_lab_cdc_bookmark_crawler**, then **Add**.
11. Select your workflow, click on **Actions->Run** and this will start the first trigger "trigger1"

The screenshot shows the AWS Glue Workflows console interface. On the left, there's a sidebar with 'Classifiers' and 'Settings' under 'ETL'. The main area has a 'Workflows' tab selected, indicated by an orange border. At the top, there are buttons for 'Add workflow' and 'Actions' (with a dropdown menu), a search bar ('Filter workflows'), and a refresh icon. The 'Actions' dropdown has 'Run' selected. Below this is a table with columns: Name, Last run, Last run status, and Last modified. There is one row visible with the name 'Workflow'.

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