Amazon Web Services
Data Engineering Immersion Day

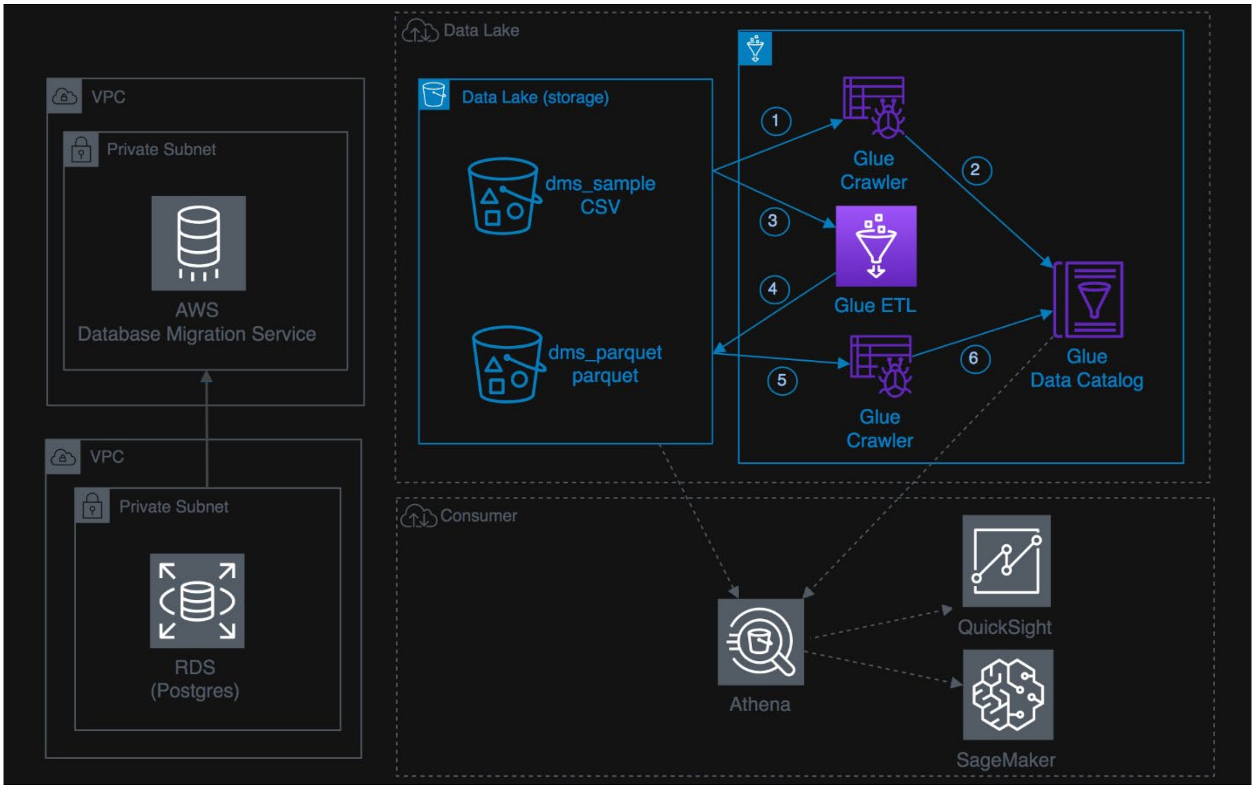
Lab 2. ETL with AWS Glue

July 2021

Table of Contents

Introduction

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



Prerequisites

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

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

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

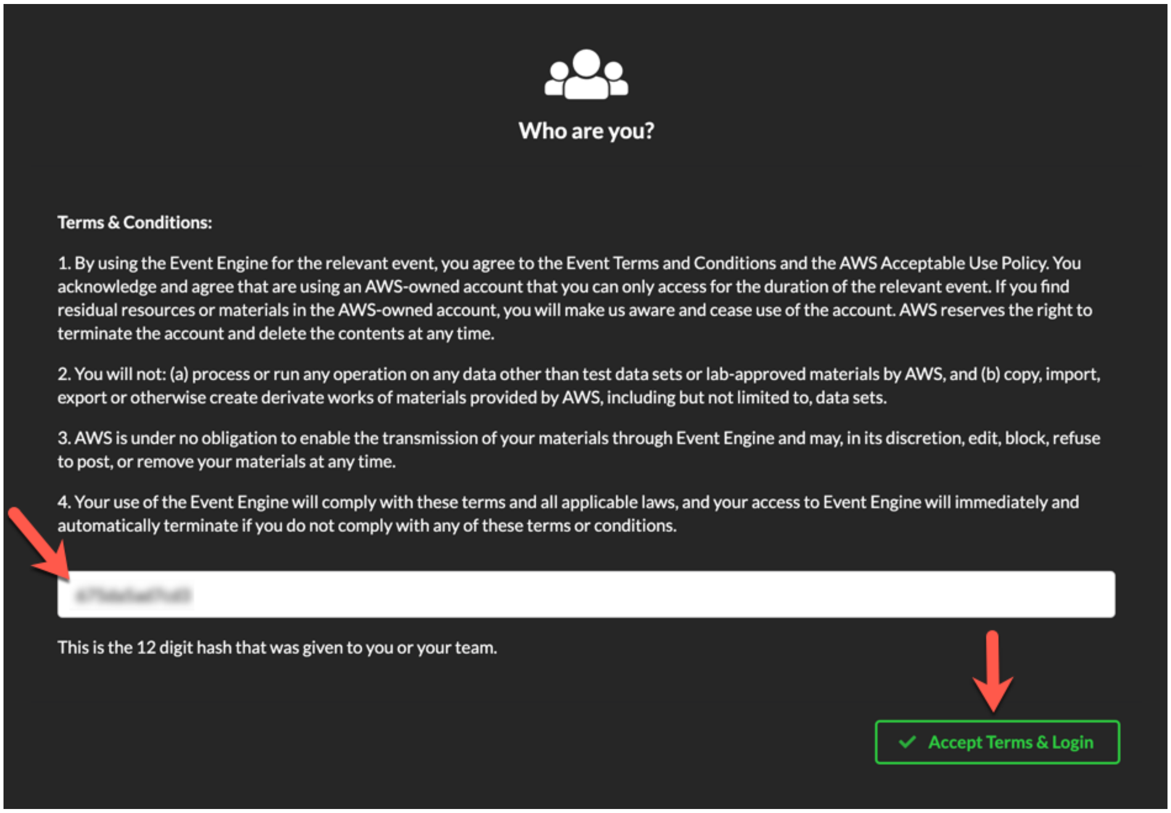
Get Started Using the Lab Environment

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

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

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

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



Who are you?

Terms & Conditions:

1. By using the Event Engine for the relevant event, you agree to the Event Terms and Conditions and the AWS Acceptable Use Policy. You acknowledge and agree that are using an AWS-owned account that you can only access for the duration of the relevant event. If you find residual resources or materials in the AWS-owned account, you will make us aware and cease use of the account. AWS reserves the right to terminate the account and delete the contents at any time.

2. You will not: (a) process or run any operation on any data other than test data sets or lab-approved materials by AWS, and (b) copy, import, export or otherwise create derivate works of materials provided by AWS, including but not limited to, data sets.

3. AWS is under no obligation to enable the transmission of your materials through Event Engine and may, in its discretion, edit, block, refuse to post, or remove your materials at any time.

4. Your use of the Event Engine will comply with these terms and all applicable laws, and your access to Event Engine will immediately and automatically terminate if you do not comply with any of these terms or conditions.

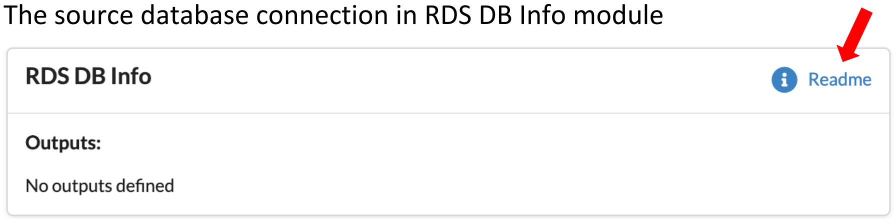
[redacted]

This is the 12 digit hash that was given to you or your team.

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

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

- The source database connection in RDS DB Info module

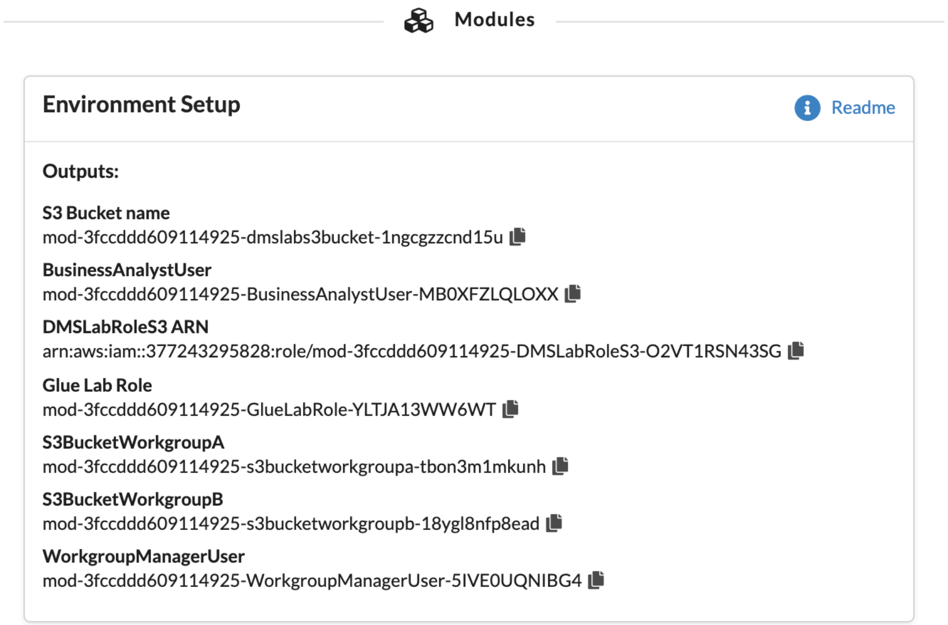


RDS DB Info

Outputs:

No outputs defined

- 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

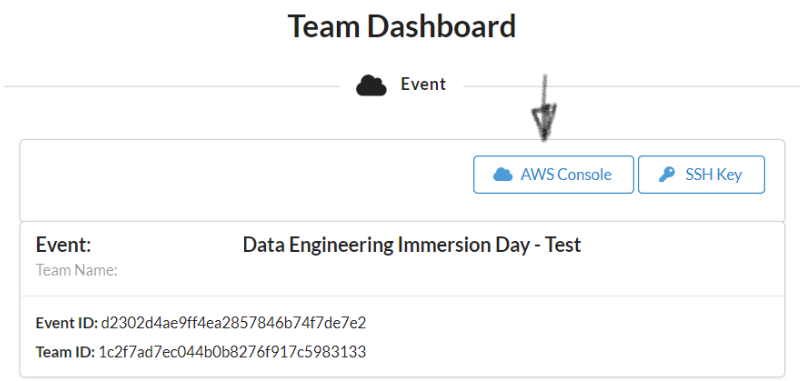
Glue Lab Role
mod-3fccddd609114925-GlueLabRole-YLTJA13WW6WT

S3BucketWorkgroupA
mod-3fccddd609114925-s3bucketworkgroupa-tbon3m1mkunh

S3BucketWorkgroupB
mod-3fccddd609114925-s3bucketworkgroupb-18ygl8nfp8ead

WorkgroupManagerUser
mod-3fccddd609114925-WorkgroupManagerUser-5IVE0UQNIBG4

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



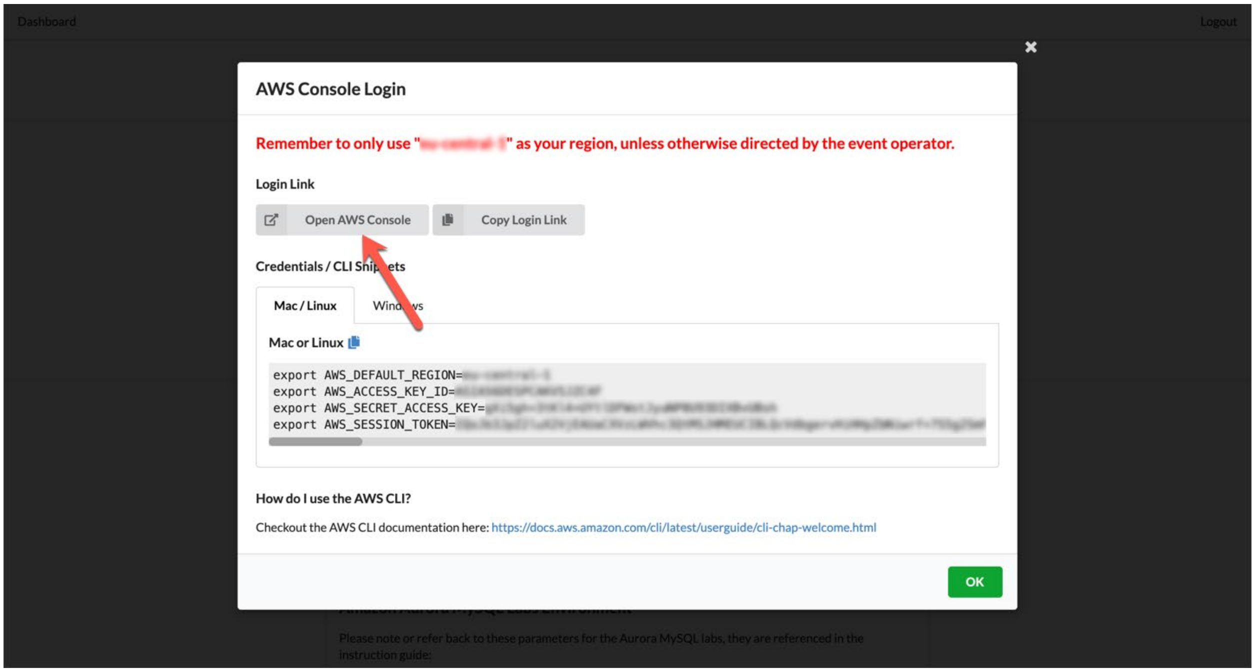
Event: Data Engineering Immersion Day - Test

Team Name:

Event ID: d2302d4ae9ff4ea2857846b74f7de7e2

Team ID: 1c2f7ad7ec044b0b8276f917c5983133

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



Remember to only use "[redacted]" as your region, unless otherwise directed by the event operator.

```
export AWS_DEFAULT_REGION=[redacted]
export AWS_ACCESS_KEY_ID=[redacted]
export AWS_SECRET_ACCESS_KEY=[redacted]
export AWS_SESSION_TOKEN=[redacted]
```

How do I use the AWS CLI?

Checkout the AWS CLI documentation here: https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-welcome.html

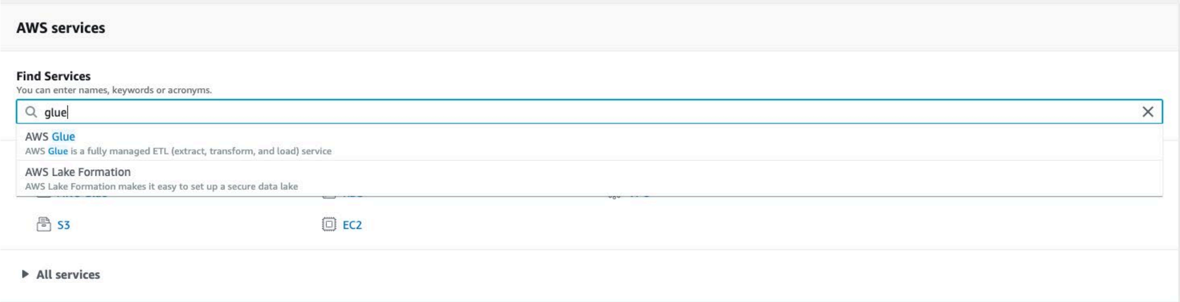
Please note or refer back to these parameters for the Aurora MySQL labs, they are referenced in the instruction guide:

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



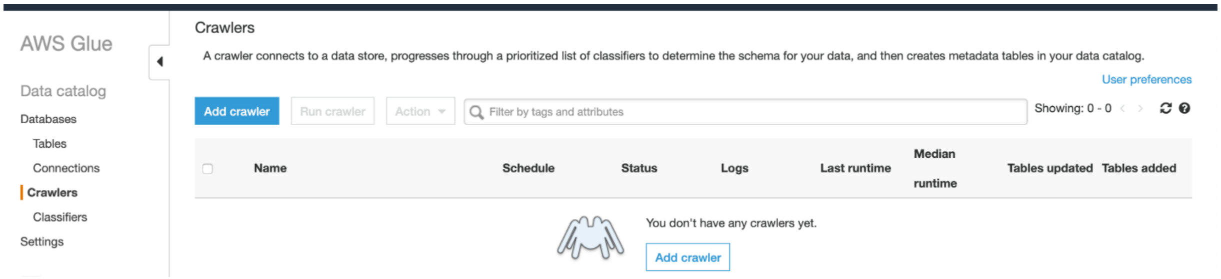
You can enter names, keywords or acronyms.

glue

AWS Glue
AWS Glue is a fully managed ETL (extract, transform, and load) service

AWS Lake Formation
AWS Lake Formation makes it easy to set up a secure data lake

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



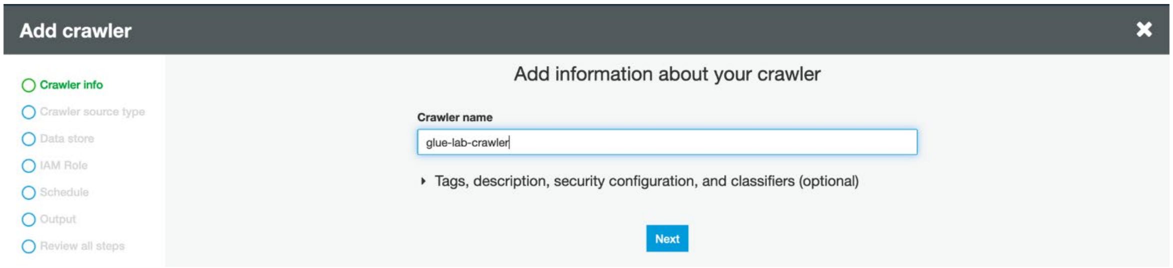
Crawlers

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.

	Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added

You don't have any crawlers yet.

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**.

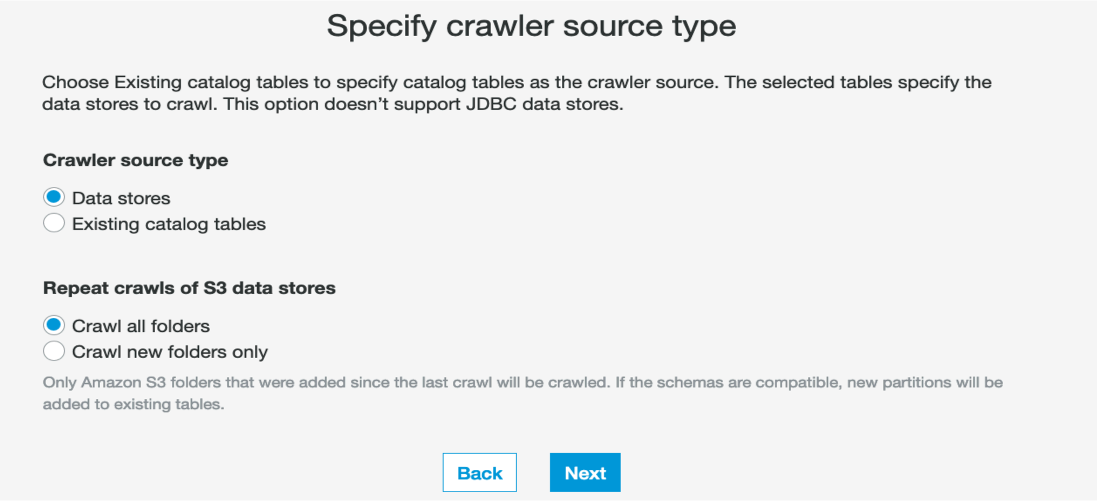


Add information about your crawler

Crawler name: glue-lab-crawler

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

6. Choose **Data stores**, **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

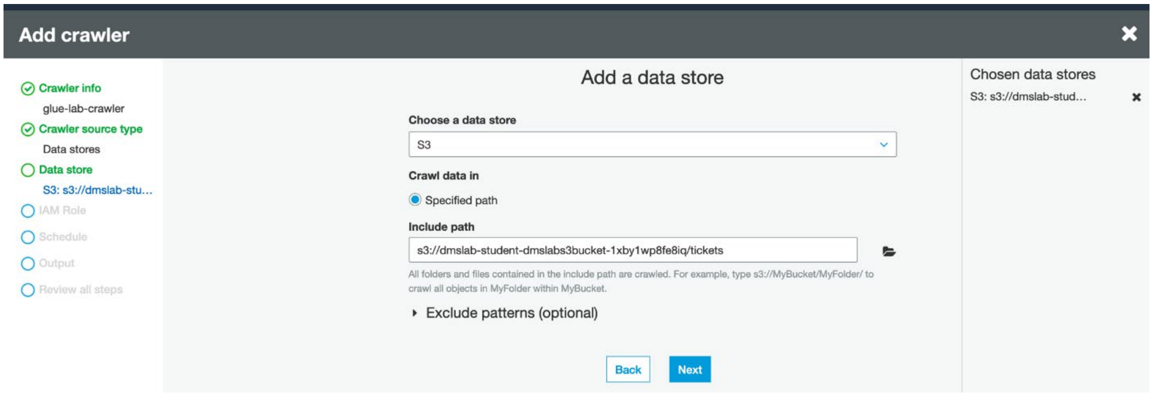
● Data stores
○ Existing catalog tables

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.

7. On the **Add a data store** page, make the following selections:
   a. For Choose a data store, click the drop-down box and select **S3**.
   b. For Crawl data in, select **Specified path in my account**.
   c. For Include path, browse to the target folder stored CSV files, e.g., **s3://xxx-dmslabs3bucket-xxx/tickets**
8. Click **Next**.



Add crawler

Add a data store

Choose a data store: S3

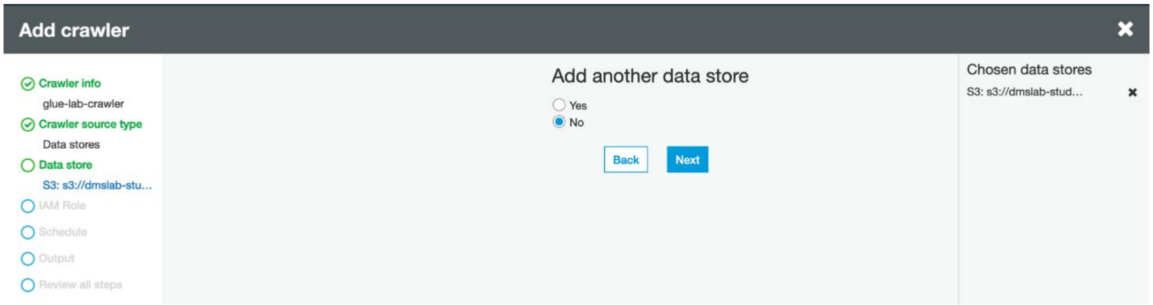
Crawl data in: ● Specified path

Include path: s3://dmslab-student-dmslabs3bucket-1xby1wp8fe8iq/tickets

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.

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

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



Add crawler

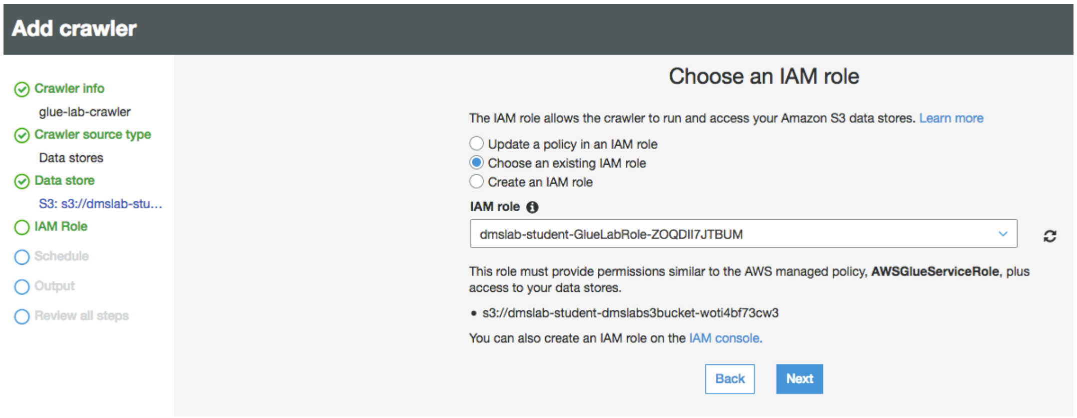
Add another data store

○ Yes
● No

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

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

- 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

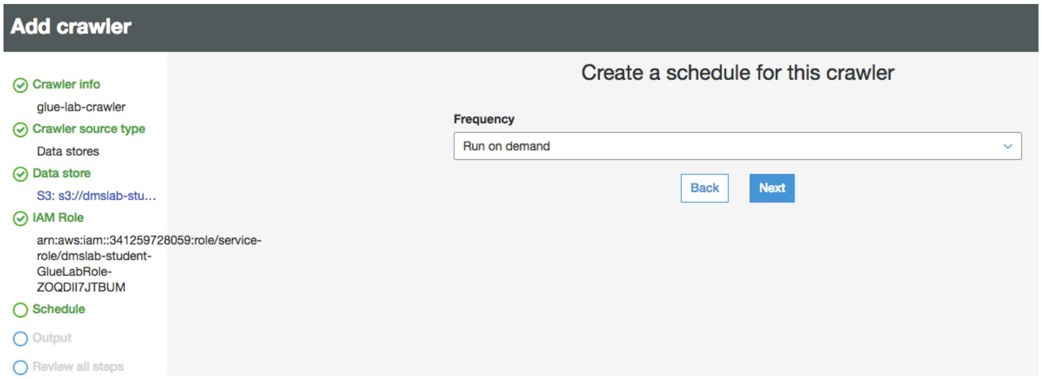
This role must provide permissions similar to the AWS managed policy, **AWSGlueServiceRole**, plus access to your data stores.

- s3://dmslab-student-dmslabs3bucket-woti4bf73cw3

You can also create an IAM role on the IAM console.

Back | Next

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



Add crawler

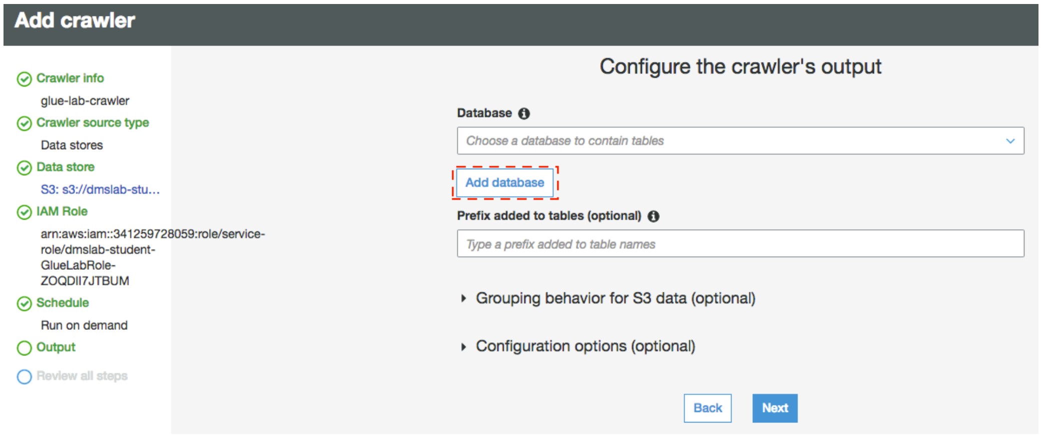
- Crawler info: glue-lab-crawler
- Crawler source type: Data stores
- Data store: S3: s3://dmslab-stu...
- IAM Role: arn:aws:iam::341259728059:role/service-role/dmslab-student-GlueLabRole-ZOQDII7JTBUM
- Schedule
- Output
- Review all steps

Create a schedule for this crawler

Frequency: Run on demand

Back | Next

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



Add crawler

- Crawler info: glue-lab-crawler
- Crawler source type: Data stores
- Data store: S3: s3://dmslab-stu...
- IAM Role: arn:aws:iam::341259728059:role/service-role/dmslab-student-GlueLabRole-ZOQDII7JTBUM
- Schedule: Run on demand
- Output
- Review all steps

Configure the crawler's output

Database: *Choose a database to contain tables*

Add database

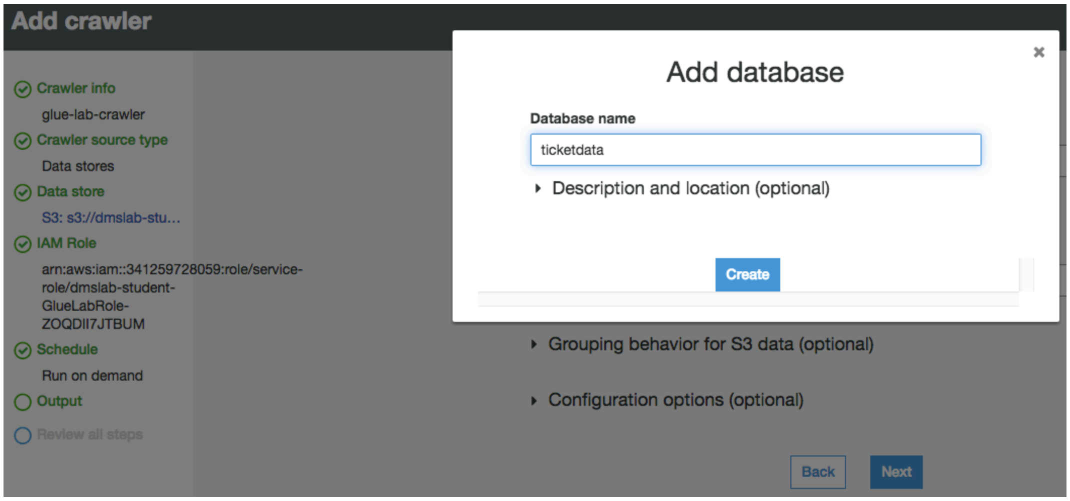
Prefix added to tables (optional): *Type a prefix added to table names*

▸ Grouping behavior for S3 data (optional)

▸ Configuration options (optional)

Back | Next

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



Add crawler

- Crawler info
  glue-lab-crawler
- Crawler source type
  Data stores
- Data store
  S3: s3://dmslab-stu…
- IAM Role
  arn:aws:iam::341259728059:role/service-role/dmslab-student-GlueLabRole-ZOQDII7JTBUM
- Schedule
  Run on demand
- Output
- Review all steps

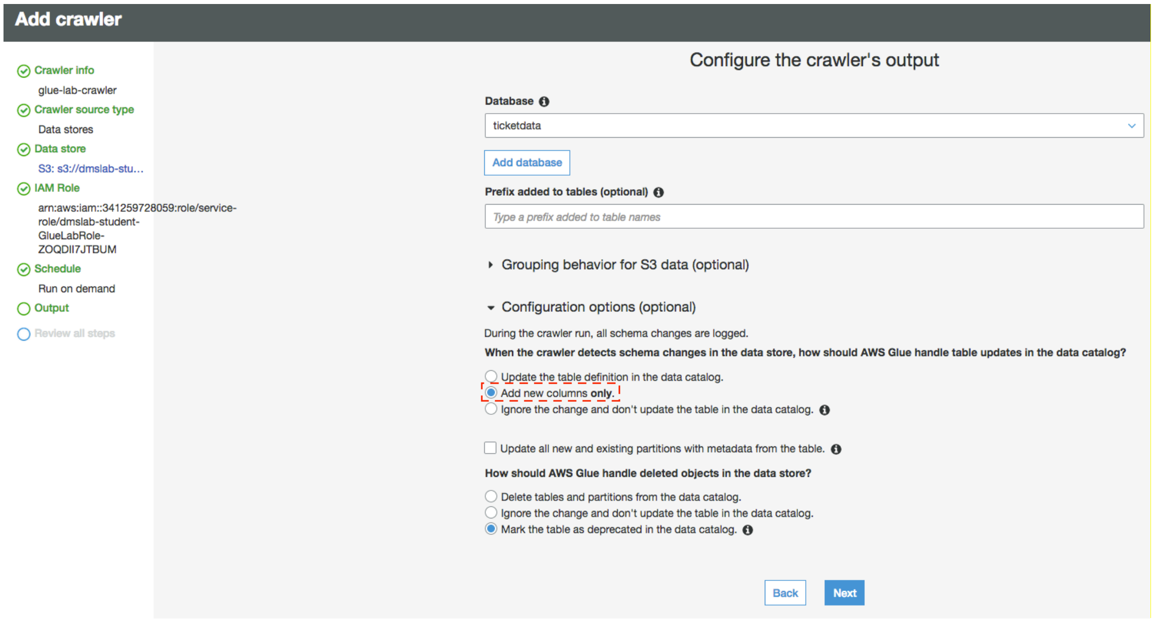
Add database

Database name: ticketdata

▸ Description and location (optional)

Create
▸ Grouping behavior for S3 data (optional)
▸ Configuration options (optional)
Back
Next

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**.



Add crawler

- Crawler info
  glue-lab-crawler
- Crawler source type
  Data stores
- Data store
  S3: s3://dmslab-stu…
- IAM Role
  arn:aws:iam::341259728059:role/service-role/dmslab-student-GlueLabRole-ZOQDII7JTBUM
- Schedule
  Run on demand
- Output
- Review all steps

Configure the crawler's output

Database: ticketdata

Prefix added to tables (optional): Type a prefix added to table names

▸ 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?

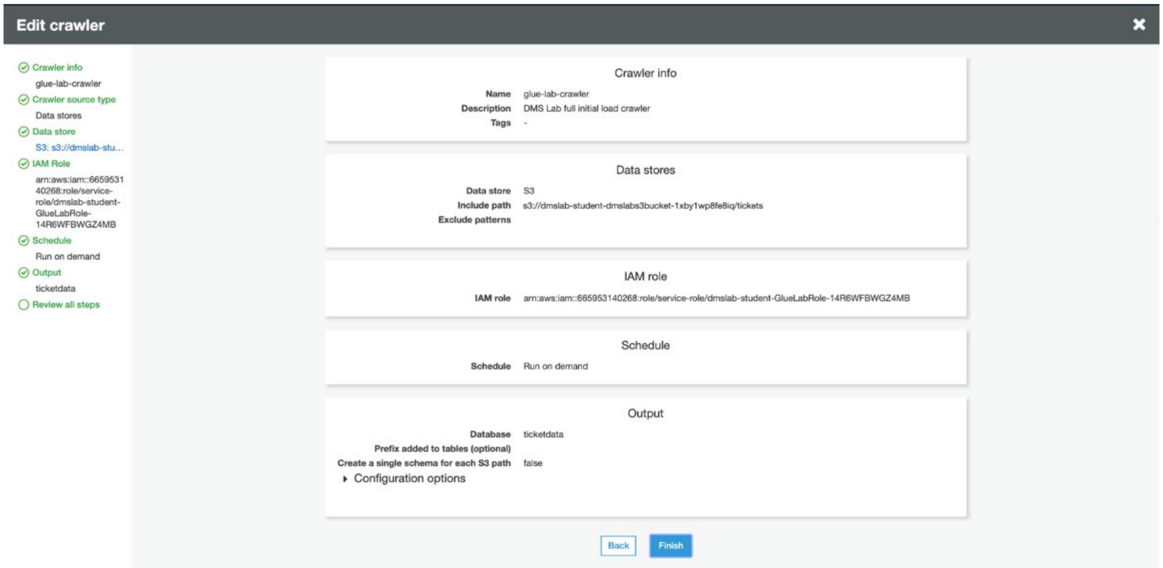
○ Update the table definition in the data catalog.
● Add new columns only.
○ Ignore the change and don't update the table in the data catalog.

□ 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.

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



Crawler info

Name	glue-lab-crawler
Description	DMS Lab full initial load crawler
Tags	-

Data stores

Data store	S3
Include path	s3://dmslab-student-dmslabs3bucket-1xby1wp8fe8iq/tickets
Exclude patterns	

IAM role

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

Schedule

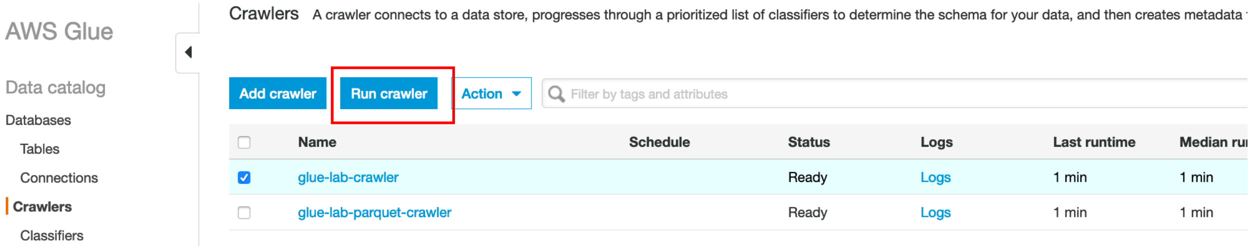
Schedule	Run on demand

Output

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

▸ Configuration options

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



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

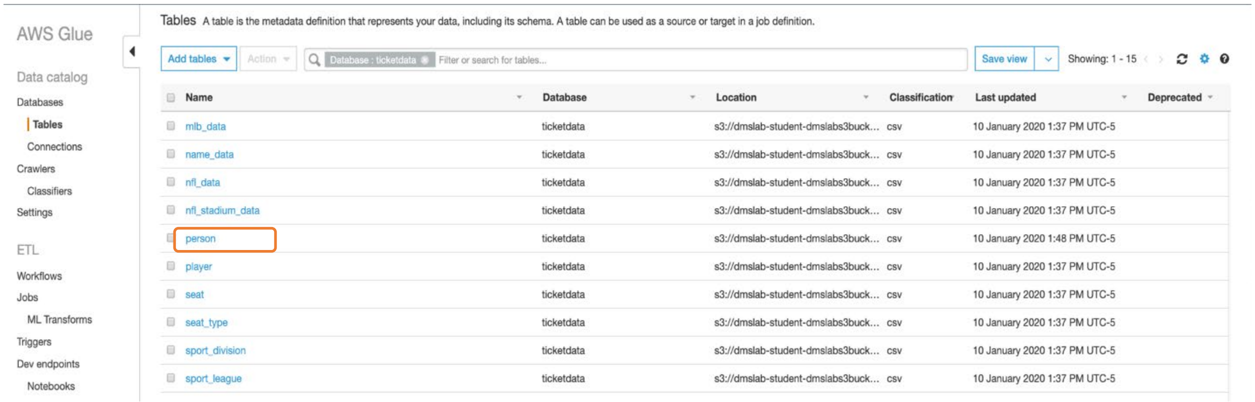
□	Name	Schedule	Status	Logs	Last runtime	Median ru
☑	glue-lab-crawler		Ready	Logs	1 min	1 min
□	glue-lab-parquet-crawler		Ready	Logs	1 min	1 min

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.



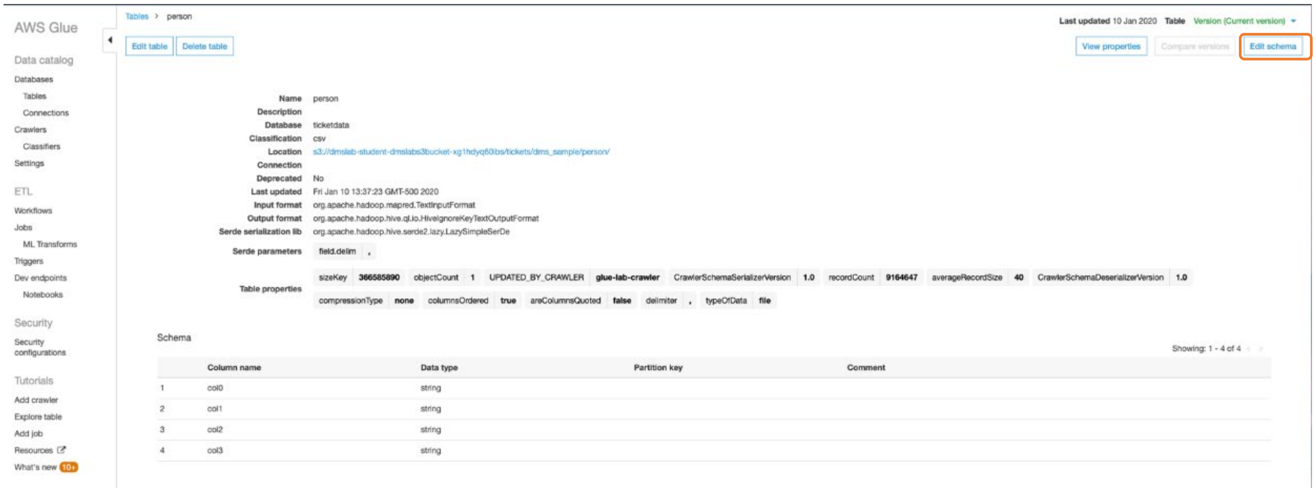
Tables A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

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

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

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

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



Last updated 10 Jan 2020 Table Version (Current version)

Name	person
Description	
Database	ticketdata
Classification	csv
Location	s3://dmslab-student-dmslabs3bucket-xg1hdyq60ibs/tickets/dms_sample/person/
Connection	
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.ql.io.HiveIgnoreKeyTextOutputFormat
Serde serialization lib	org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
Serde parameters	field.delim ,
Table properties	sizeKey 366585890 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

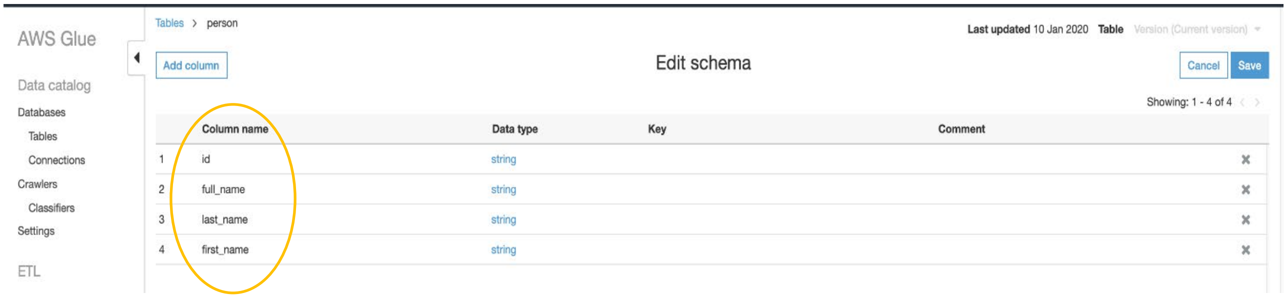
Schema

Showing: 1 - 4 of 4

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

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

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`



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

4. Click **Save**.

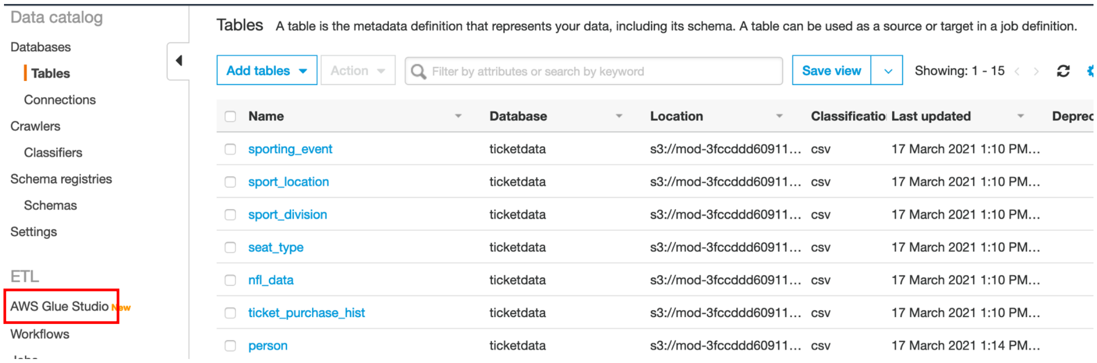
Data ETL Exercise

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**”

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.

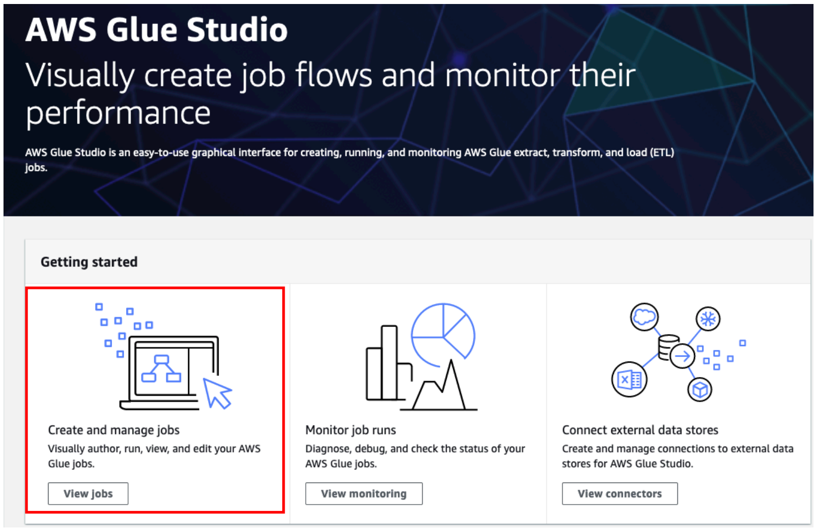
1. In the left navigation pane, under ETL, click **AWS Glue Studio**.



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

Name	Database	Location	Classificatio	Last updated	Deprec
sporting_event	ticketdata	s3://mod-3fccddd60911…	csv	17 March 2021 1:10 PM…	
sport_location	ticketdata	s3://mod-3fccddd60911…	csv	17 March 2021 1:10 PM…	
sport_division	ticketdata	s3://mod-3fccddd60911…	csv	17 March 2021 1:10 PM…	
seat_type	ticketdata	s3://mod-3fccddd60911…	csv	17 March 2021 1:10 PM…	
nfl_data	ticketdata	s3://mod-3fccddd60911…	csv	17 March 2021 1:10 PM…	
ticket_purchase_hist	ticketdata	s3://mod-3fccddd60911…	csv	17 March 2021 1:10 PM…	
person	ticketdata	s3://mod-3fccddd60911…	csv	17 March 2021 1:14 PM…	

2. Choose “**View Jobs**”



AWS Glue Studio

Visually create job flows and monitor their performance

AWS Glue Studio is an easy-to-use graphical interface for creating, running, and monitoring AWS Glue extract, transform, and load (ETL) jobs.

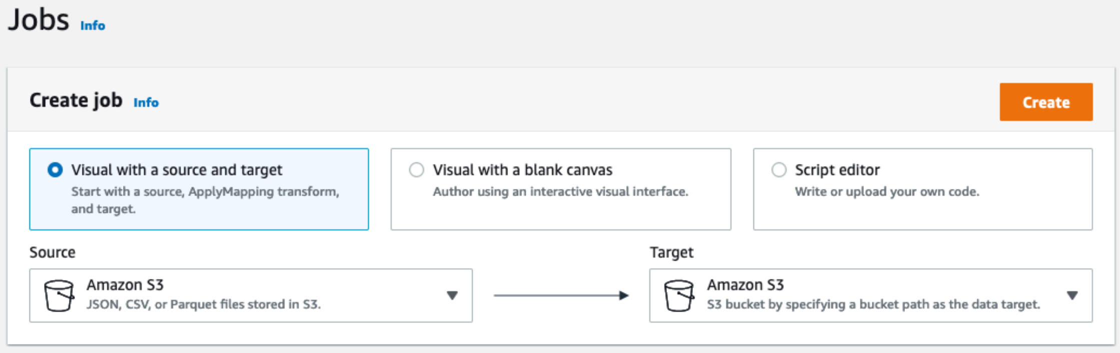
Getting started

Create and manage jobs
Visually author, run, view, and edit your AWS Glue jobs.
View jobs

Monitor job runs
Diagnose, debug, and check the status of your AWS Glue jobs.
View monitoring

Connect external data stores
Create and manage connections to external data stores for AWS Glue Studio.
View connectors

3. Leave the “Visual with a source and target” option selected, and press “**Create**”



Jobs Info

Create job Info

Create

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

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

○ Script editor
Write or upload your own code.

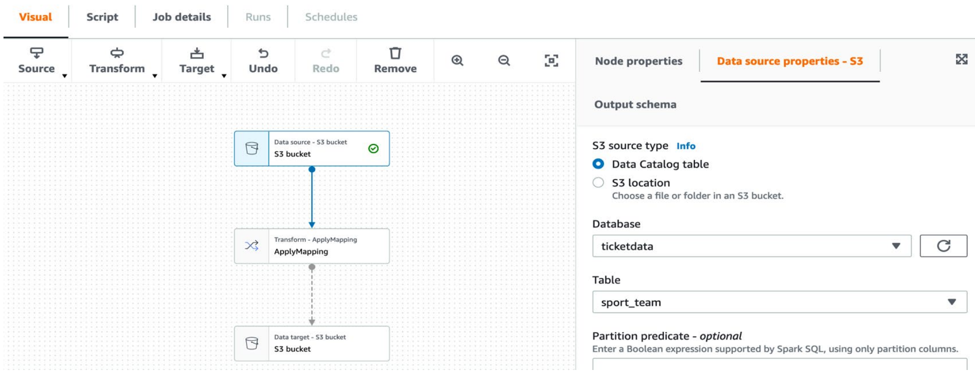
Source: Amazon S3 — JSON, CSV, or Parquet files stored in S3.

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

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.



S3 source type Info

● Data Catalog table

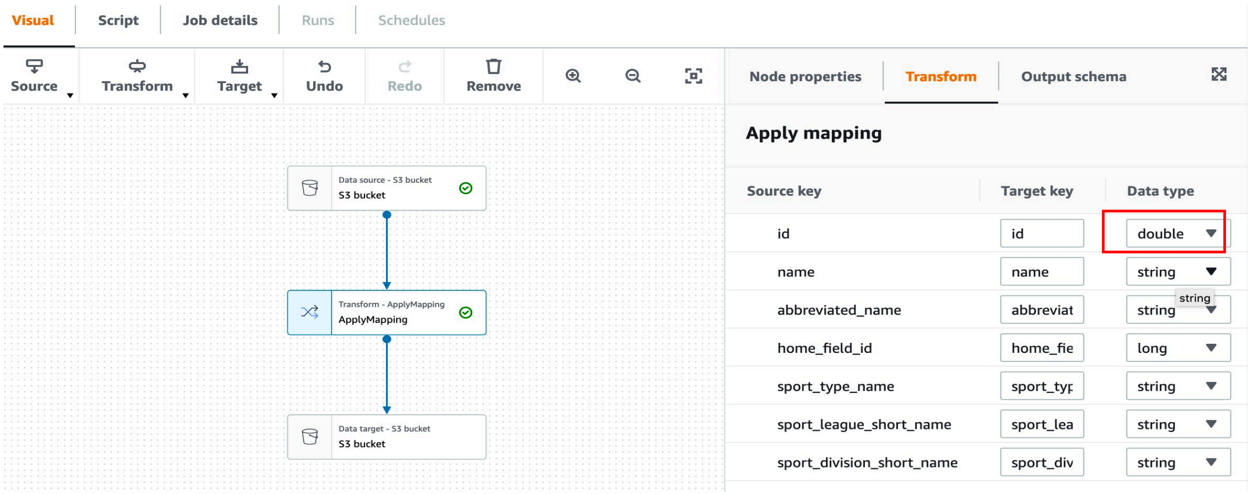
○ S3 location
Choose a file or folder in an S3 bucket.

Database: ticketdata

Table: sport_team

Partition predicate - *optional*
Enter a Boolean expression supported by Spark SQL, using only partition columns.

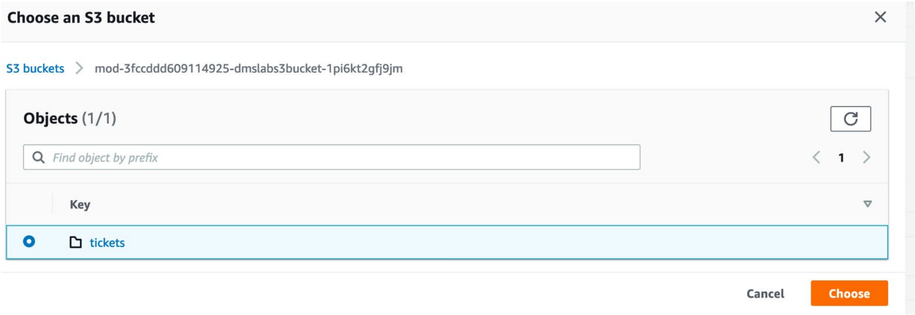
7. Select the “ApplyMapping” node. In the Transform panel on the right and change the data type of “id” column to double in the dropdown.



Apply mapping

Source key	Target key	Data type
id	id	double
name	name	string
abbreviated_name	abbreviat	string
home_field_id	home_fie	long
sport_type_name	sport_typ	string
sport_league_short_name	sport_lea	string
sport_division_short_name	sport_div	string

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

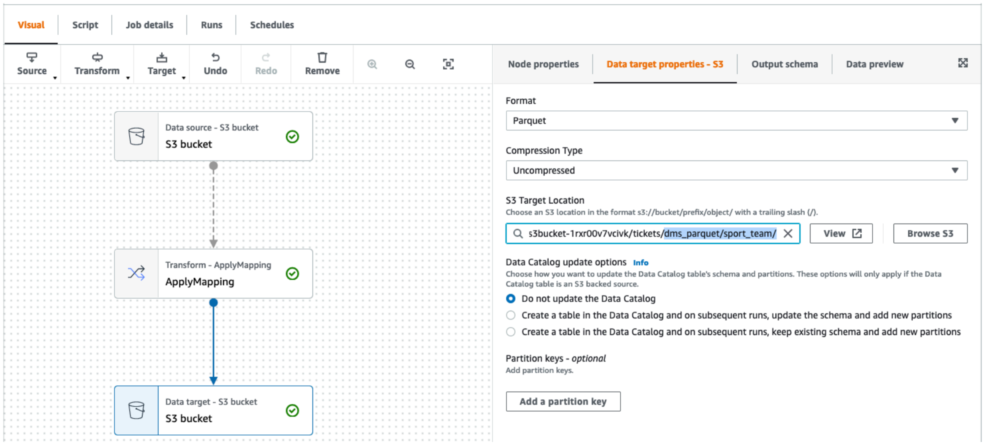

Choose an S3 bucket

S3 buckets > mod-3fccddd609114925-dmslabs3bucket-1pi6kt2gfj9jm

Objects (1/1)

● tickets

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



Format: Parquet

Compression Type: Uncompressed

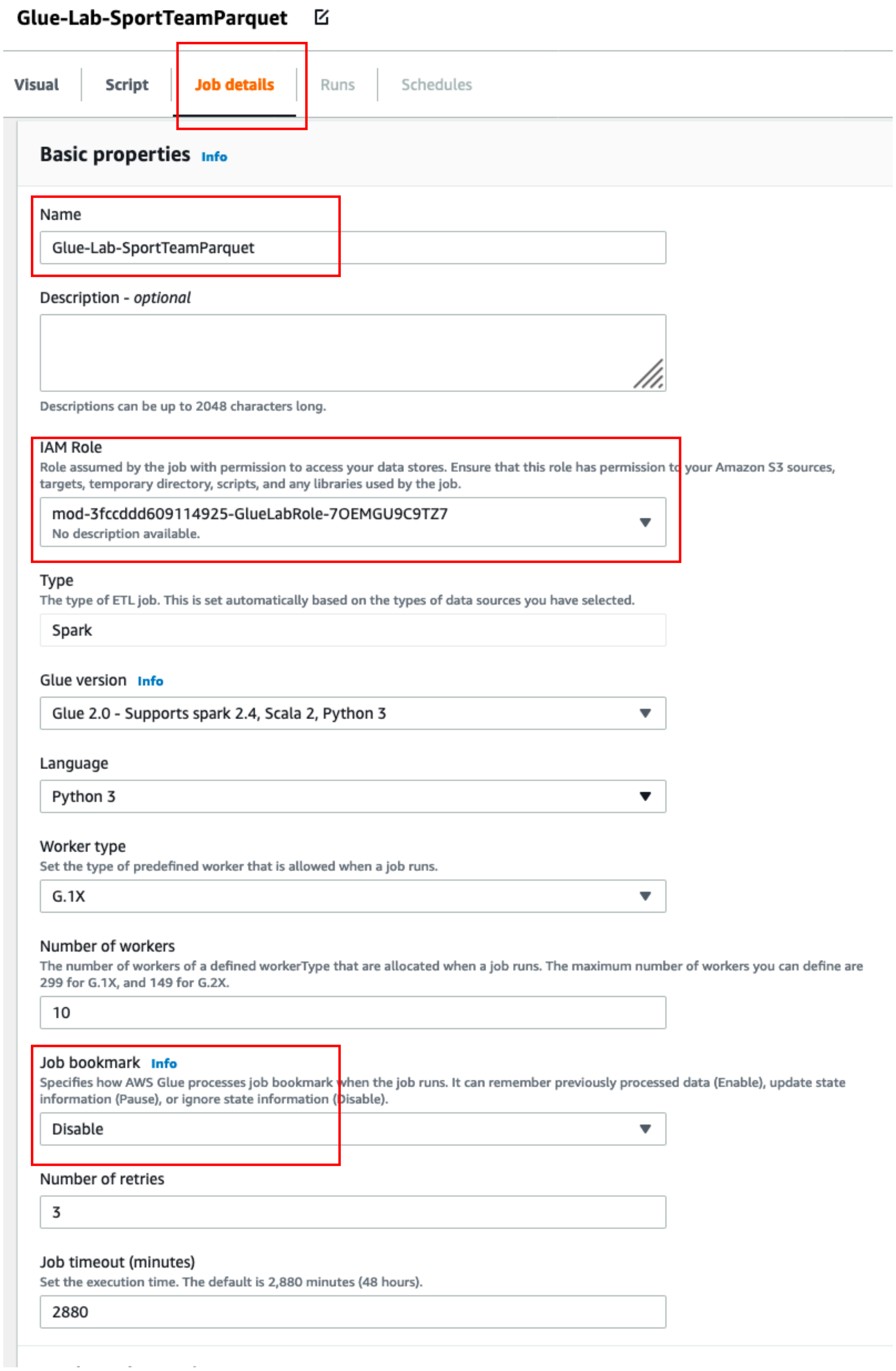
S3 Target Location
Choose an S3 location in the format s3://bucket/prefix/object/ with a trailing slash (/).
s3bucket-1rxr00v7vcivk/tickets/dms_parquet/sport_team/

Data catalog update options Info
Choose how you want to update the Data Catalog table's schema and partitions. These options will only apply if the Data Catalog table is an S3 backed source.
● Do not update the Data Catalog
○ Create a table in the Data Catalog and on subsequent runs, update the schema and add new partitions
○ Create a table in the Data Catalog and on subsequent runs, keep existing schema and add new partitions

Partition keys - *optional*
Add partition keys.

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.



Glue-Lab-SportTeamParquet

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

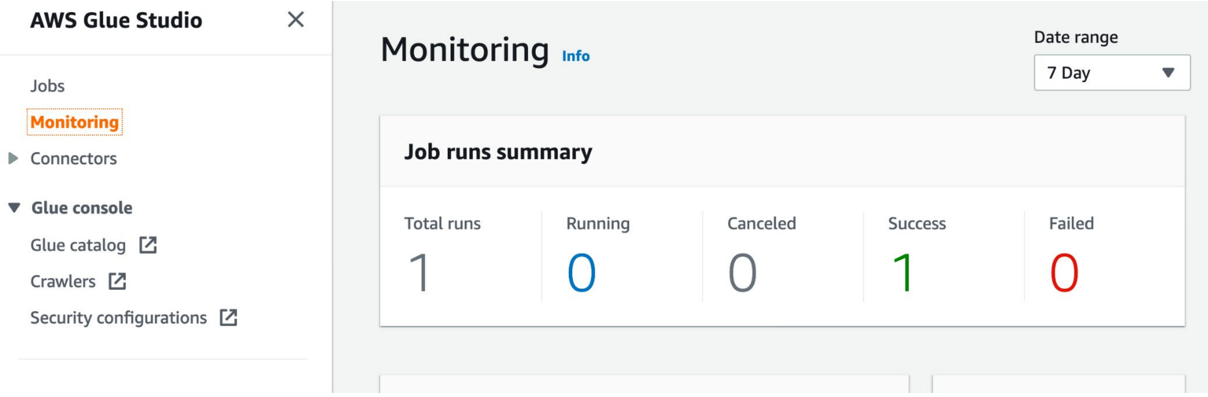
Job bookmark
Specifies how AWS Glue processes job bookmark when the job runs. It can remember previously processed data (Enable), update state information (Pause), or ignore state information (Disable).
Disable

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.

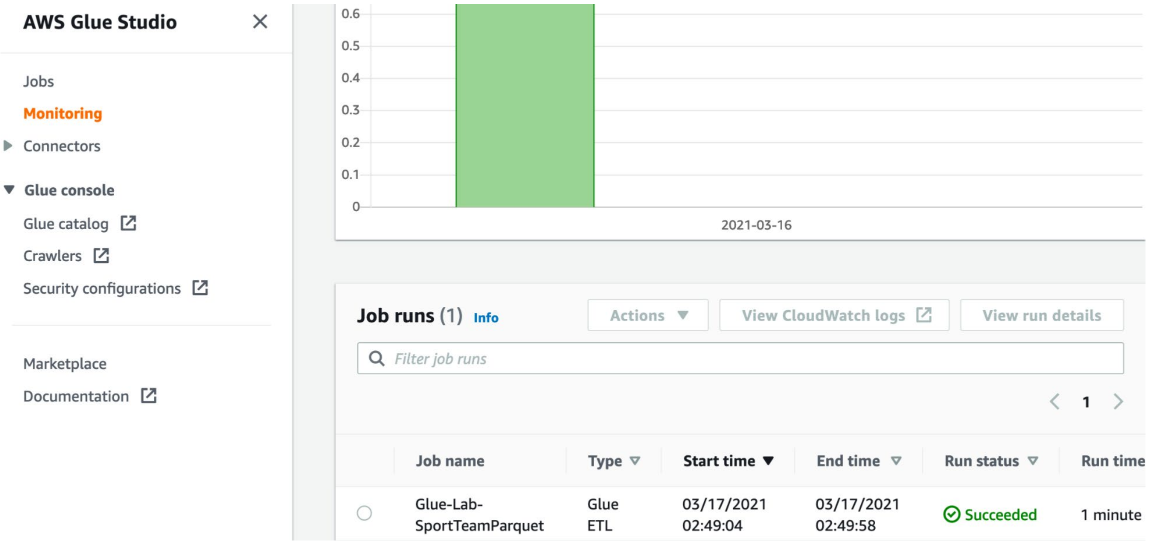
15. Once you see the “**Successfully created job**” message in the banner, click the “**Run**” button to start the job.
16. Select “**Jobs**” from the navigation panel on the left-hand side to see a list of your jobs.
17. Select “**Monitoring**” from the navigation panel on the left-hand side to view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



Job runs summary

Total runs	Running	Canceled	Success	Failed
1	0	0	1	0

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.

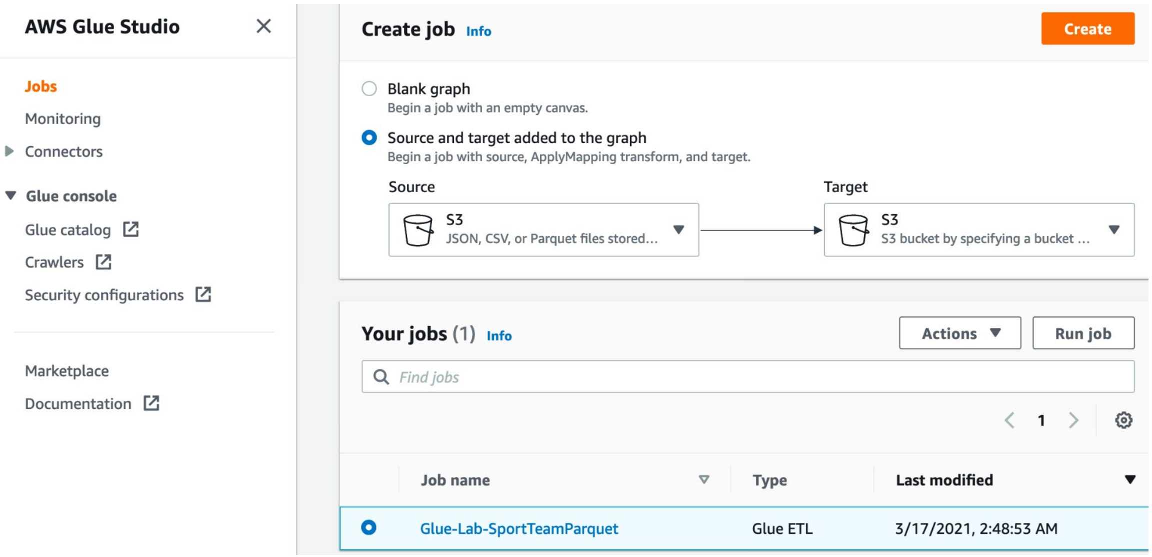


	Job name	Type	Start time	End time	Run status	Run time
○	Glue-Lab-SportTeamParquet	Glue ETL	03/17/2021 02:49:04	03/17/2021 02:49:58	Succeeded	1 minute

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.



AWS Glue Studio

Jobs
Monitoring
Connectors
Glue console
Glue catalog
Crawlers
Security configurations
Marketplace
Documentation

Create job Info | Create

○ Blank graph
Begin a job with an empty canvas.

● Source and target added to the graph
Begin a job with source, ApplyMapping transform, and target.

Source: S3 — JSON, CSV, or Parquet files stored…
Target: S3 — S3 bucket by specifying a bucket …

Your jobs (1) Info | Actions | Run job

Find jobs

	Job name	Type	Last modified
●	Glue-Lab-SportTeamParquet	Glue ETL	3/17/2021, 2:48:53 AM

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

1. Sport_Location:

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

Task / Action	Attribute	Values
“Data source - S3 bucket” node	Database	ticketdata
	Table	sport_location
“Transform - ApplyMapping” node	Schema transformations	None
“Data target - S3 bucket” node	Format	Parquet
	Compression Type	Uncompressed
	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:

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
	Compression Type	Uncompressed
	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

Task / Action	Attribute	Values
	Compression Type	Uncompressed
	S3 target path	`tickets/dms_parquet/sporting_event_ticket/`
“Job details tab”	Job Name	`Glue-Lab-SportingEventTicketParquet`
	IAM Role	xxx-GlueLabRole-xxx
	Job bookmark	Disable

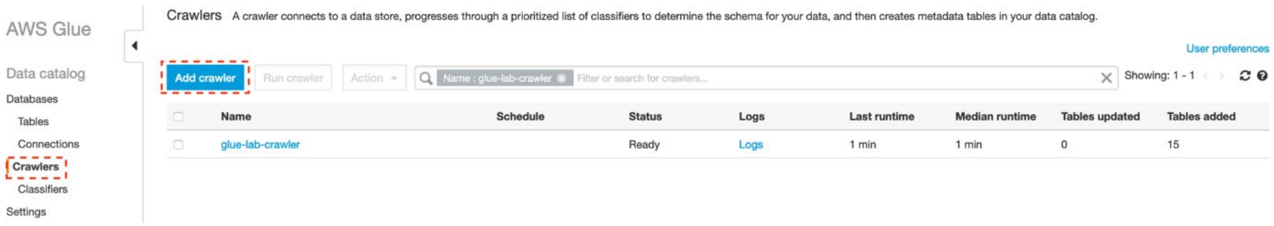
4. Person:

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
	Compression Type	Uncompressed
	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

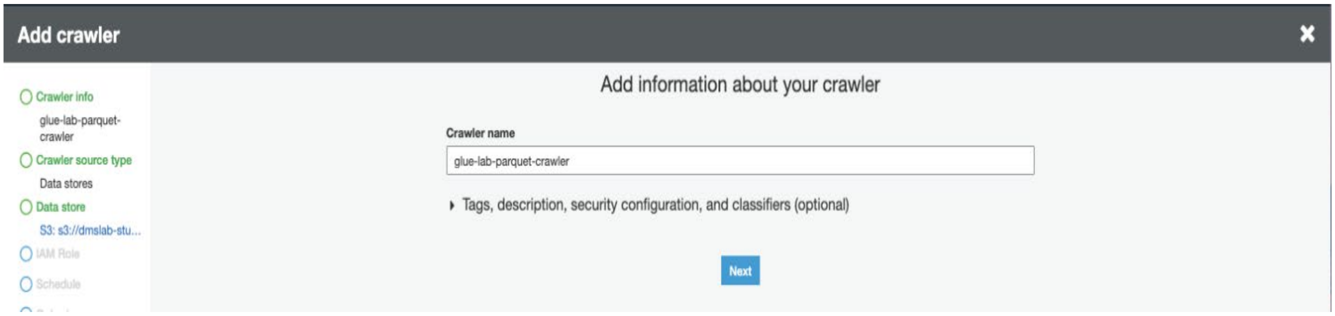
1. In the Glue Studio naviation menu, select **Crawlers** to open the Glue Crawlers page in a new tab. Click **Add crawler**.



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.

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

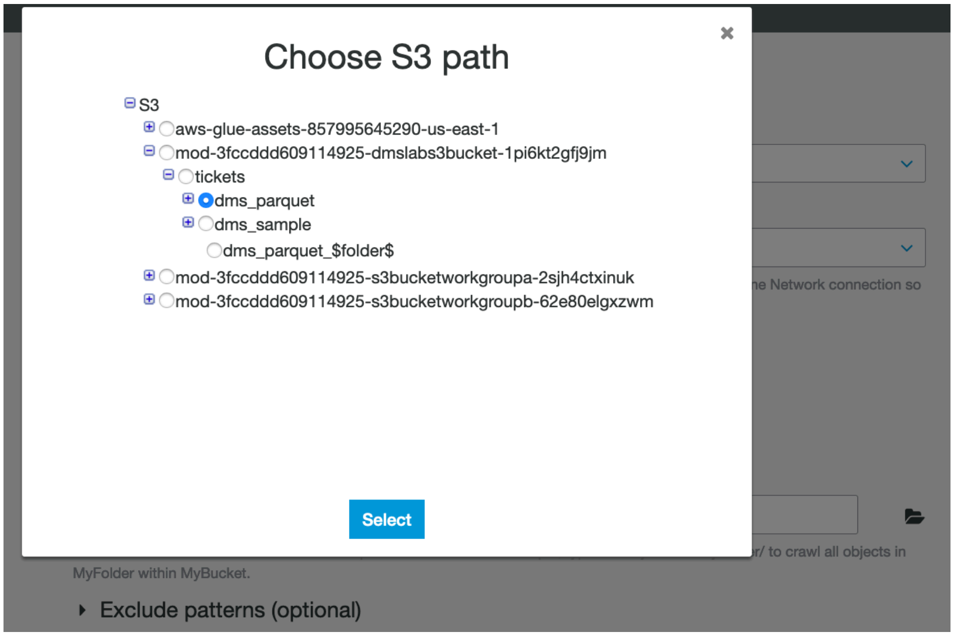
2. For **Crawler name**, type **glue-lab-parquet-crawler** and Click **Next**.



Add information about your crawler

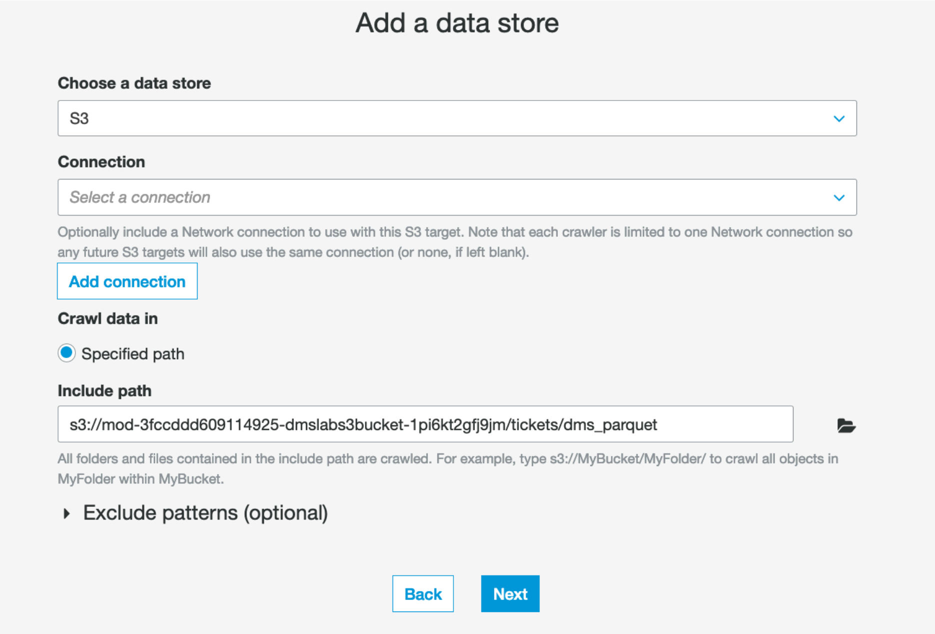
Crawler name: glue-lab-parquet-crawler

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**.



Choose S3 path

- S3
  - ○ aws-glue-assets-857995645290-us-east-1
  - ○ mod-3fccddd609114925-dmslabs3bucket-1pi6kt2gfj9jm
    - ○ tickets
      - ● dms_parquet
      - ○ dms_sample
      - ○ dms_parquet_$folder$
  - ○ mod-3fccddd609114925-s3bucketworkgroupa-2sjh4ctxinuk
  - ○ mod-3fccddd609114925-s3bucketworkgroupb-62e80elgxzwm



Add a data store

Choose a data store

S3

Connection

Select a connection

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).

Crawl data in

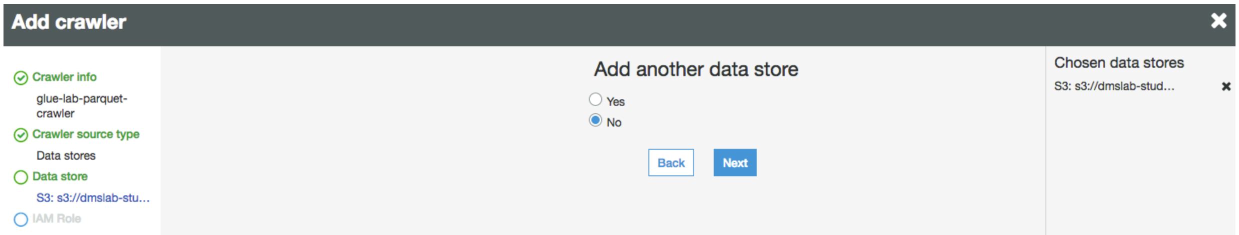
● Specified path

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.

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



Add crawler

Add another data store

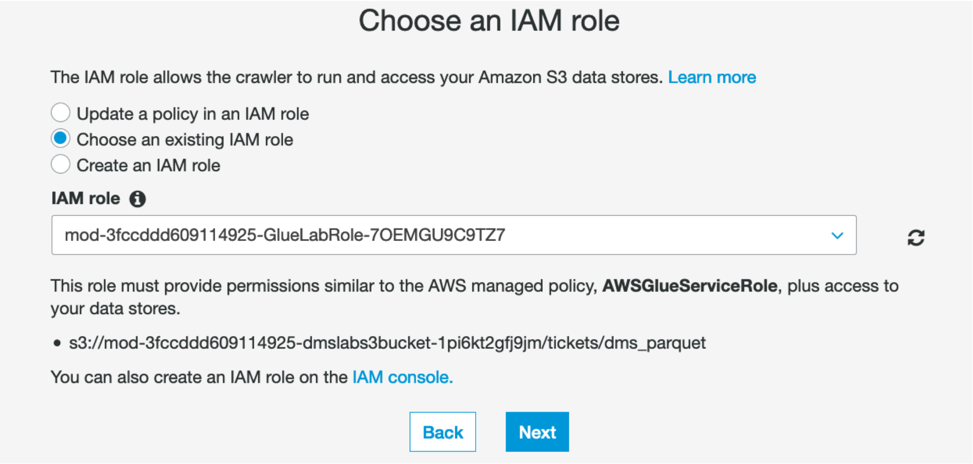
○ Yes

● No

Chosen data stores

S3: s3://dmslab-stud...

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



Choose an IAM role

The IAM role allows the crawler to run and access your Amazon S3 data stores. Learn more

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

IAM role

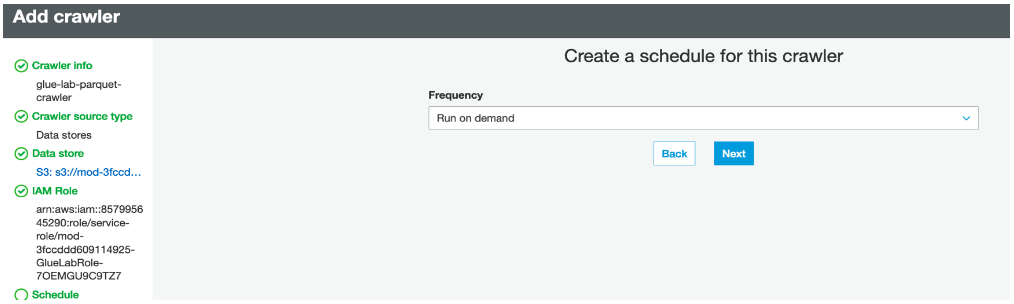
mod-3fccddd609114925-GlueLabRole-7OEMGU9C9TZ7

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

- s3://mod-3fccddd609114925-dmslabs3bucket-1pi6kt2gfj9jm/tickets/dms_parquet

You can also create an IAM role on the IAM console.

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



Add crawler

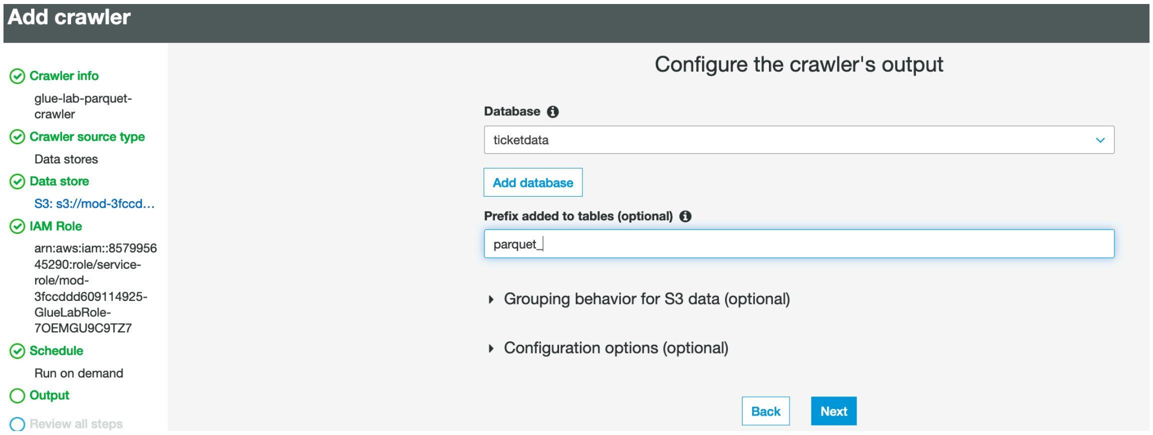
✓ Crawler info
glue-lab-parquet-crawler
✓ Crawler source type
Data stores
✓ Data store
S3: s3://mod-3fccd...
✓ IAM Role
arn:aws:iam::857995645290:role/service-role/mod-3fccddd609114925-GlueLabRole-7OEMGU9C9TZ7
○ Schedule

Create a schedule for this crawler

Frequency
Run on demand

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

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



Add crawler

Configure the crawler's output

- Crawler info: glue-lab-parquet-crawler
- Crawler source type: Data stores
- Data store: S3: s3://mod-3fccd…
- IAM Role: arn:aws:iam::857995645290:role/service-role/mod-3fccddd609114925-GlueLabRole-7OEMGU9C9TZ7
- Schedule: Run on demand
- Output
- Review all steps

Database: ticketdata

Add database

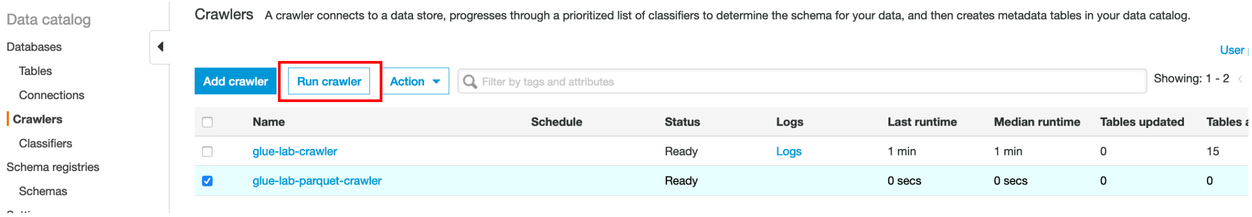
Prefix added to tables (optional): parquet_

▸ Grouping behavior for S3 data (optional)

▸ Configuration options (optional)

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.



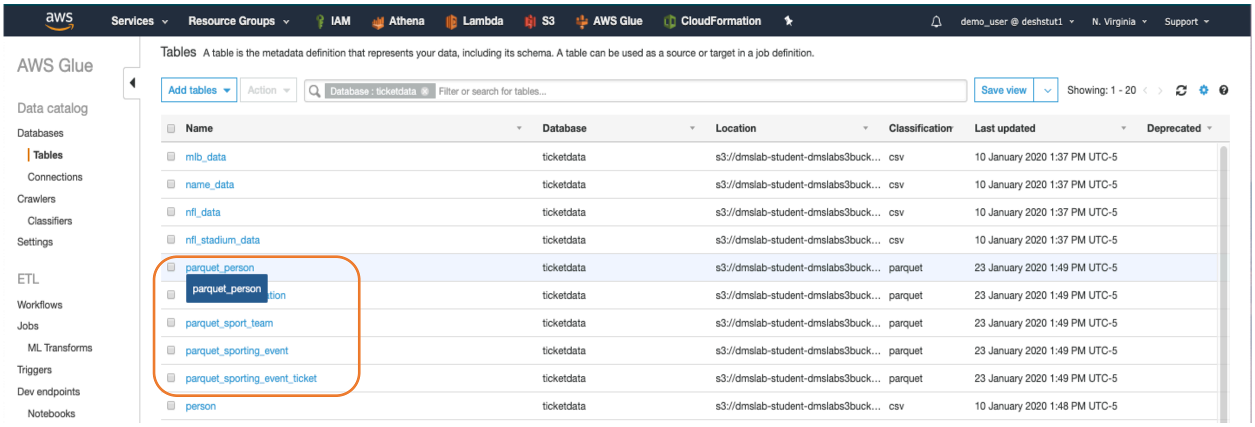
Crawlers A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

Showing: 1 - 2

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

Confirm you can see the tables:

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



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

Showing: 1 - 20

□	Name	Database	Location	Classification	Last updated	Deprecated
□	mlb_data	ticketdata	s3://dmslab-student-dmslabs3buck…	csv	10 January 2020 1:37 PM UTC-5	
□	name_data	ticketdata	s3://dmslab-student-dmslabs3buck…	csv	10 January 2020 1:37 PM UTC-5	
□	nfl_data	ticketdata	s3://dmslab-student-dmslabs3buck…	csv	10 January 2020 1:37 PM UTC-5	
□	nfl_stadium_data	ticketdata	s3://dmslab-student-dmslabs3buck…	csv	10 January 2020 1:37 PM UTC-5	
□	parquet_person	ticketdata	s3://dmslab-student-dmslabs3buck…	parquet	23 January 2020 1:49 PM UTC-5	
□	tion	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	

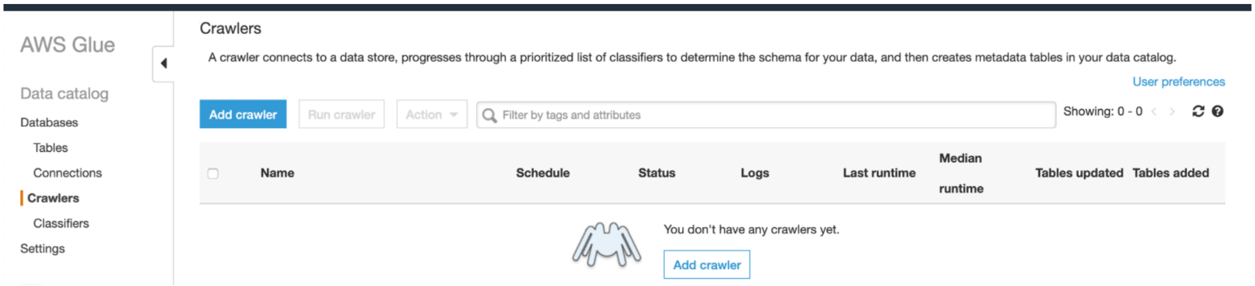
PART B: Glue Job Bookmark (Optional):

****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.



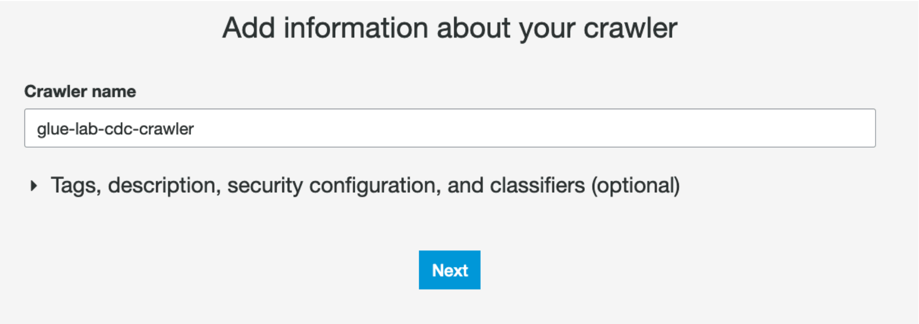
Crawlers

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.

	Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added

You don't have any crawlers yet.

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

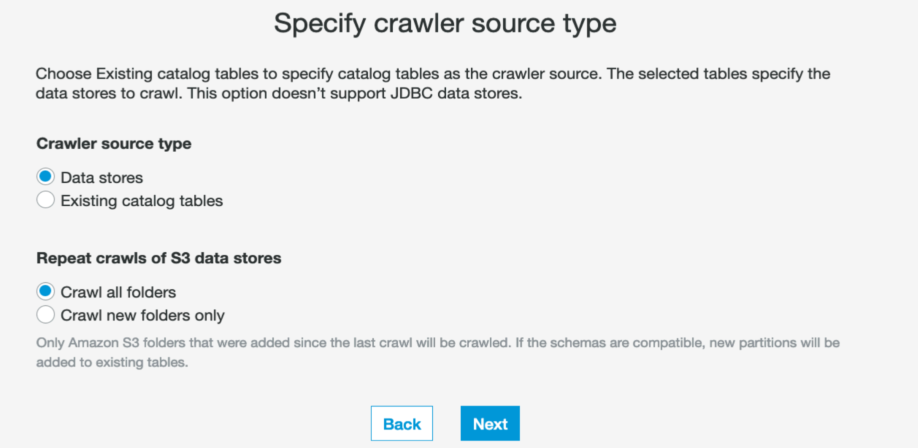


Add information about your crawler

Crawler name: glue-lab-cdc-crawler

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

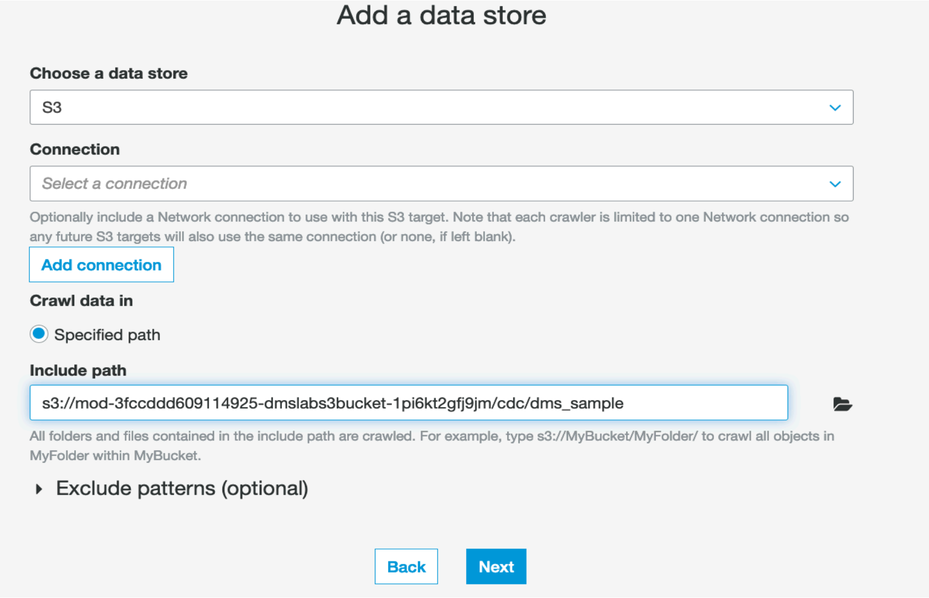
● Data stores
○ Existing catalog tables

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.

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**.



Add a data store

Choose a data store

S3

Connection

Select a connection

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).

Crawl data in

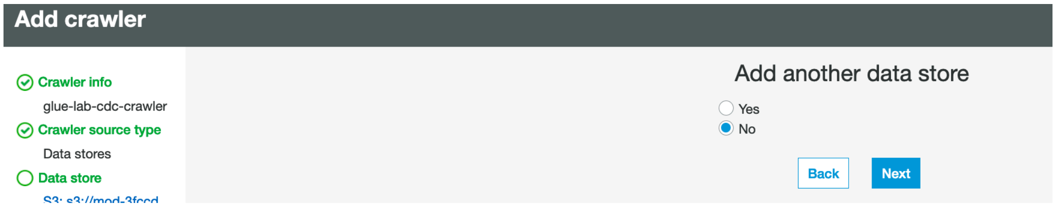
● Specified path

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.

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://mod-3fccd

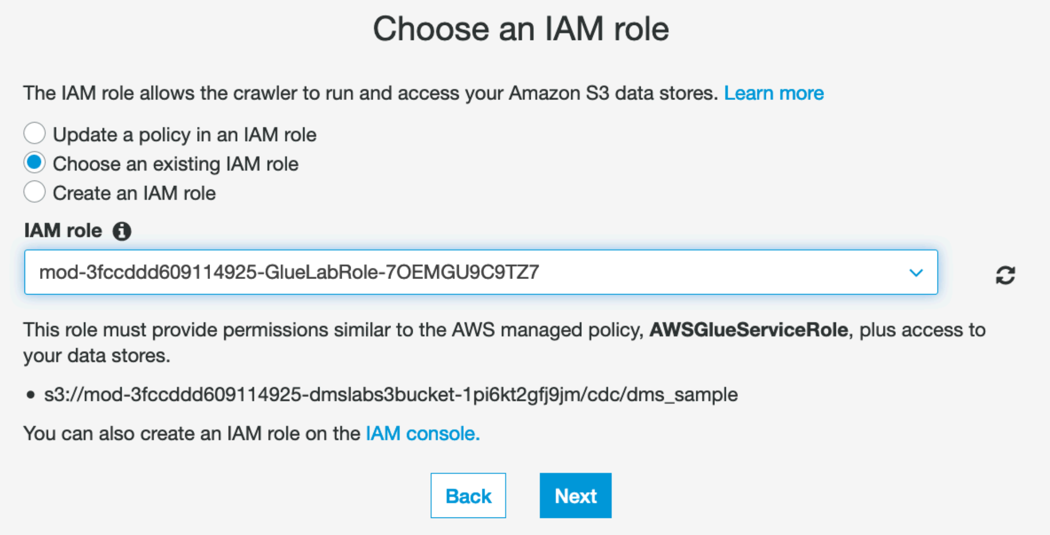
Add another data store

○ Yes
● No

9. On the **Choose an IAM role** page, make the following selections:

a. Select **Choose an existing IAM role**.
b. **For IAM role**, select **xxx-GlueLabRole-xxx**. E.g. “dmslab-student-GlueLabRole-ZOQDII7JTBUM”

10. Click **Next**.



Choose an IAM role

The IAM role allows the crawler to run and access your Amazon S3 data stores. Learn more

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

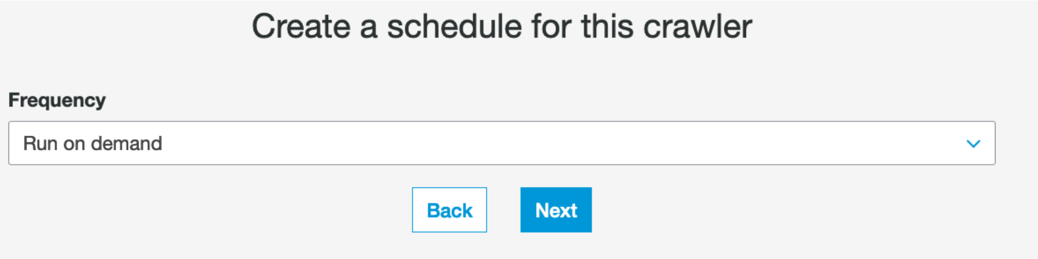
IAM role: mod-3fccddd609114925-GlueLabRole-7OEMGU9C9TZ7

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

- s3://mod-3fccddd609114925-dmslabs3bucket-1pi6kt2gfj9jm/cdc/dms_sample

You can also create an IAM role on the IAM console.

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



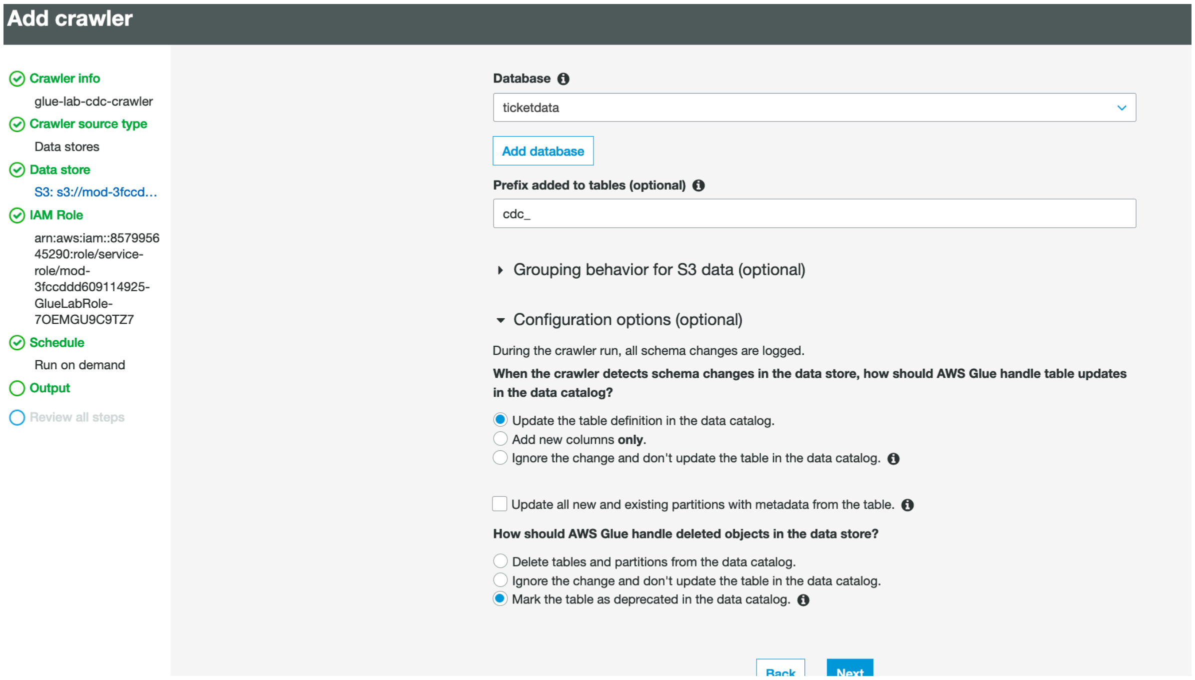
Create a schedule for this crawler

Frequency: Run on demand

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**.



Add crawler

- Crawler info: glue-lab-cdc-crawler
- Crawler source type: Data stores
- Data store: S3: s3://mod-3fccd…
- IAM Role: arn:aws:iam::857995645290:role/service-role/mod-3fccddd609114925-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?

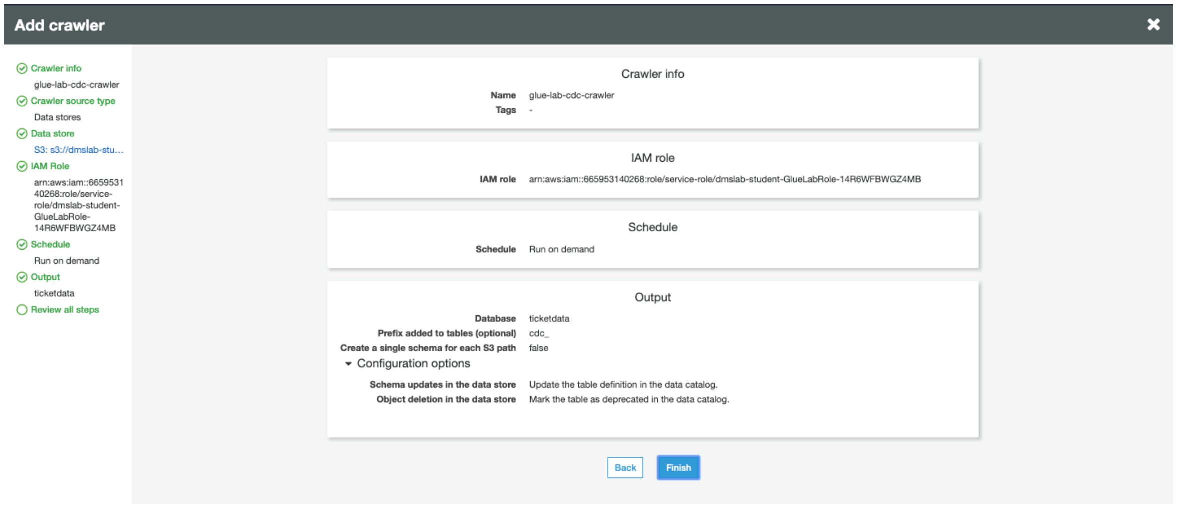
● Update the table definition in the data catalog.
○ Add new columns **only**.
○ Ignore the change and don't update the table in the data catalog.

□ 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-14R6WFBWGZ4MB
- 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-14R6WFBWGZ4MB

Schedule

Schedule	Run on demand

Output

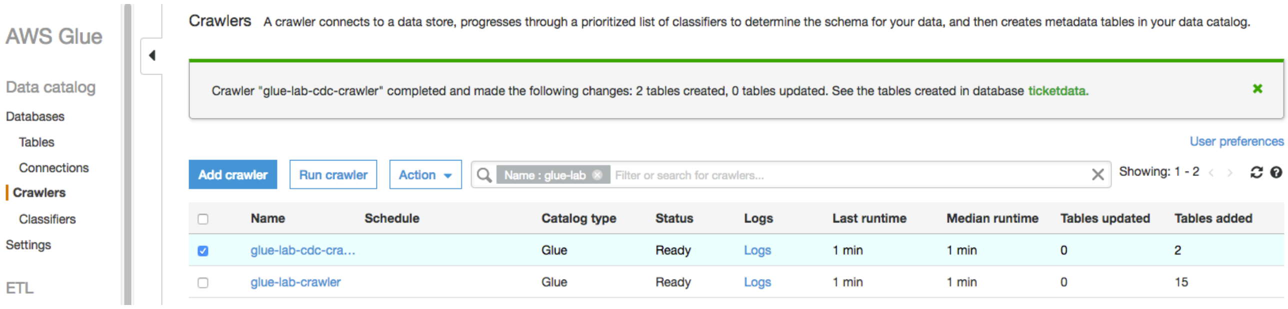
Database	ticketdata
Prefix added to tables (optional)	cdc_
Create a single schema for each S3 path	false

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.

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**.

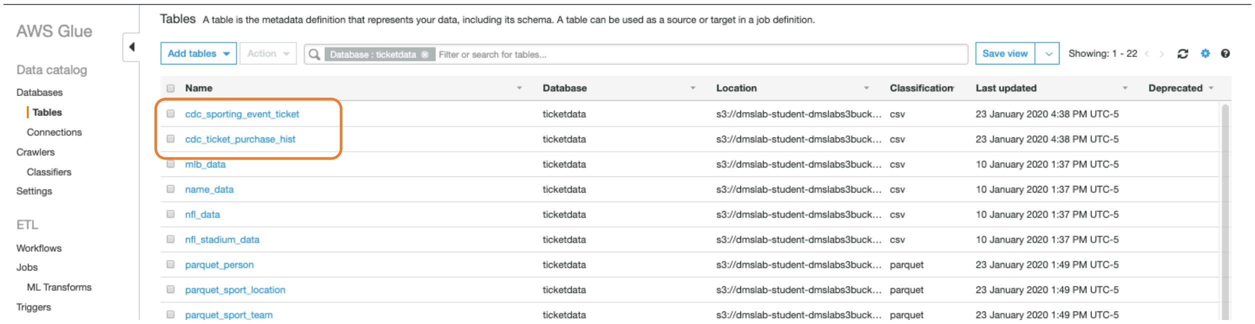


Crawlers A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

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.

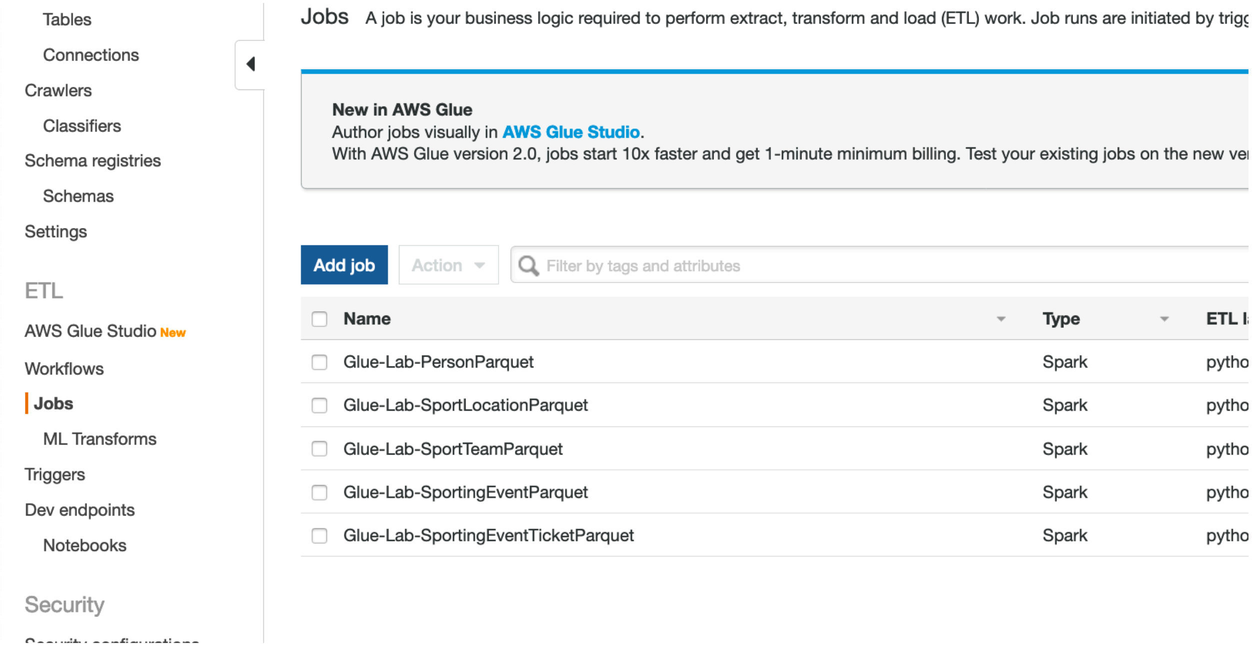


Tables A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

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

Step 2: Create a Glue Job with Bookmark Enabled

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



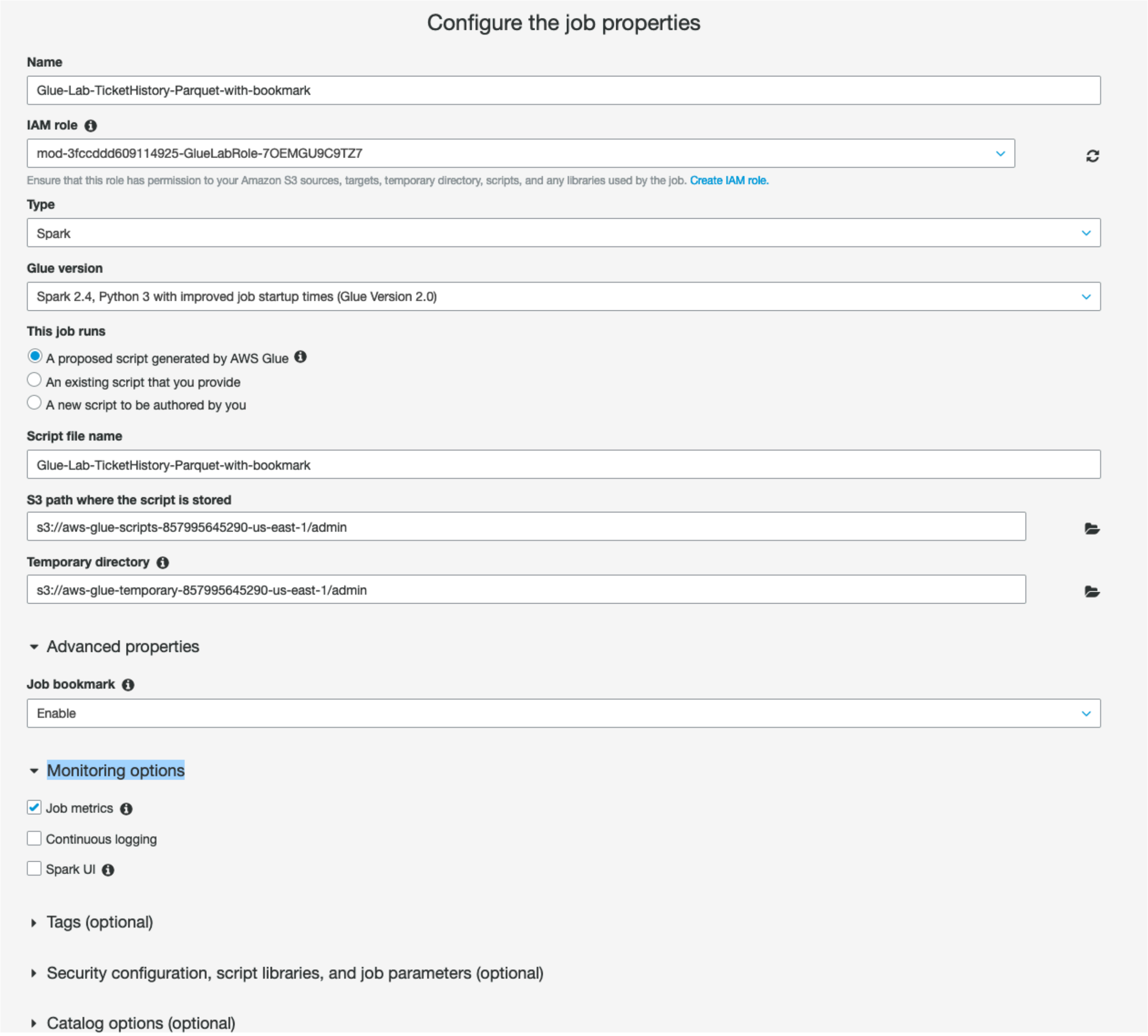
Jobs A job is your business logic required to perform extract, transform and load (ETL) work. Job runs are initiated by trigg

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

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

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**



Configure the job properties

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

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

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

Type: Spark

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

This job runs

● A proposed script generated by AWS Glue
○ An existing script that you provide
○ A new script to be authored by you

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

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

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

▾ Advanced properties

Job bookmark: Enable

▾ Monitoring options

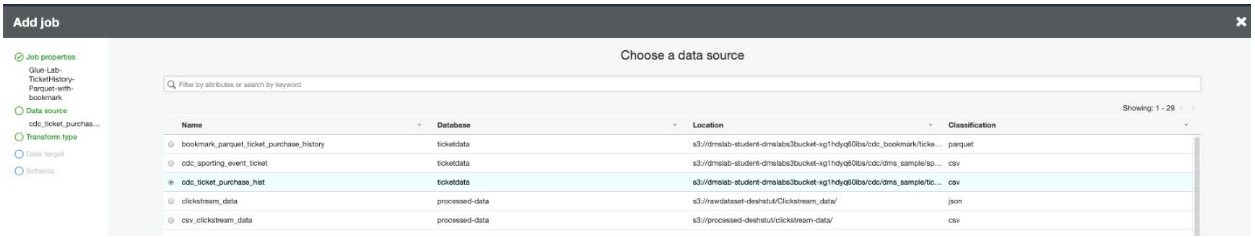
☑ Job metrics
☐ Continuous logging
☐ Spark UI

▸ Tags (optional)

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

▸ Catalog options (optional)

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



Add job

Job properties: Glue-Lab-TicketHistory-Parquet-with-bookmark
Data source: cdc_ticket_purchas...
Transform type
Data target
Schema

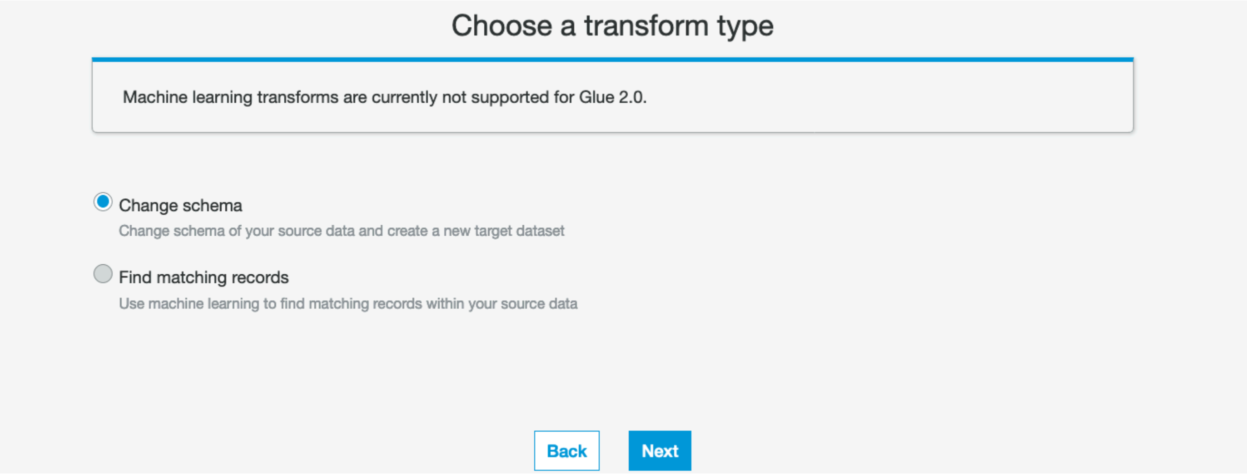
Choose a data source

Filter by attributes or search by keyword

Showing: 1 - 29

	Name	Database	Location	Classification
○	bookmark_parquet_ticket_purchase_history	ticketdata	s3://dmslab-student-dmslabs3bucket-xg1hdyq60ibs/cdc_bookmark/ticke...	parquet
○	cdc_sporting_event_ticket	ticketdata	s3://dmslab-student-dmslabs3bucket-xg1hdyq60ibs/cdc/dms_sample/sp...	csv
●	cdc_ticket_purchase_hist	ticketdata	s3://dmslab-student-dmslabs3bucket-xg1hdyq60ibs/cdc/dms_sample/tic...	csv
○	clickstream_data	processed-data	s3://rawdataset-deshstut/Clickstream_data/	json
○	csv_clickstream_data	processed-data	s3://processed-deshstut/clickstream-data/	csv

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



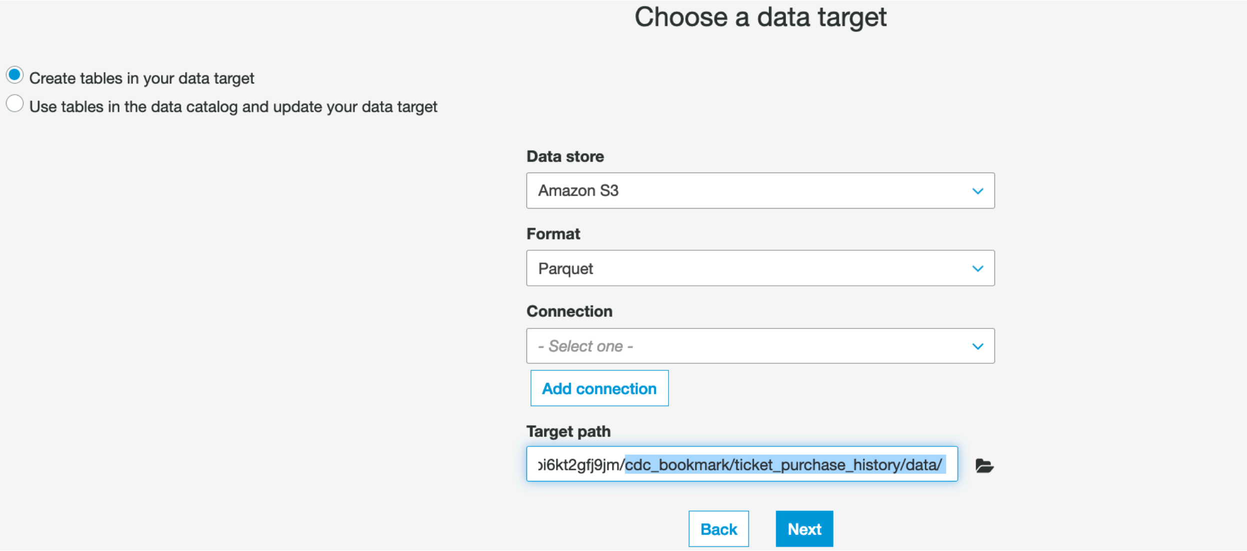
Choose a transform type

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

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

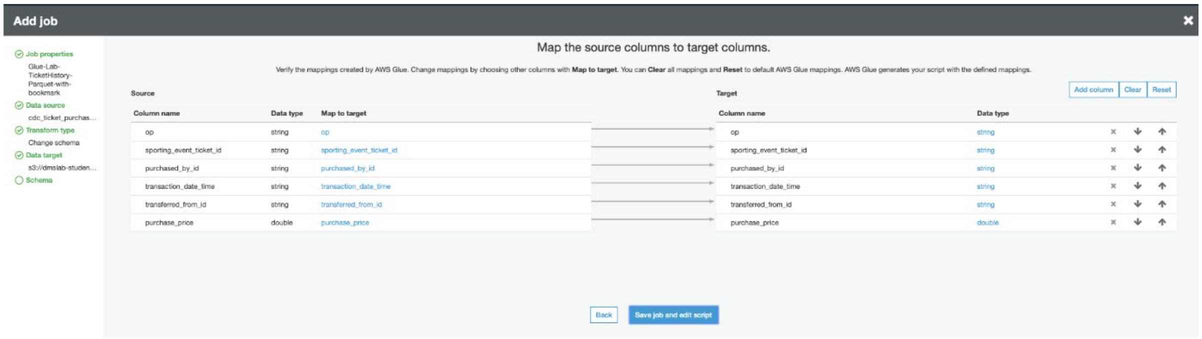
Data store: Amazon S3

Format: Parquet

Connection: - Select one -

Target path: ›i6kt2gfj9jm/cdc_bookmark/ticket_purchase_history/data/

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

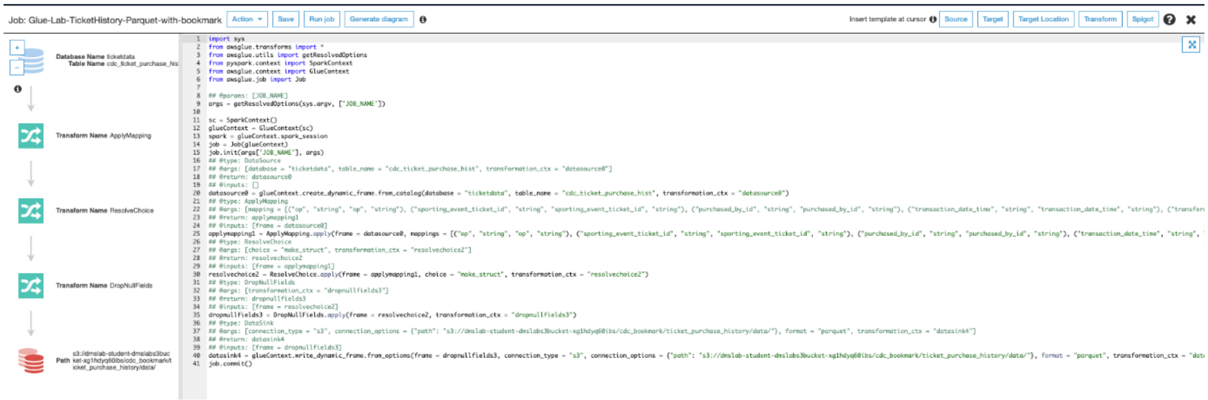


Map the source columns to target columns.

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 Column name	Data type	Map to target	Target Column name	Data type
op	string	op	op	string
sporting_event_ticket_id	string	sporting_event_ticket_id	sporting_event_ticket_id	string
purchased_by_id	string	purchased_by_id	purchased_by_id	string
transaction_date_time	string	transaction_date_time	transaction_date_time	string
transferred_from_id	string	transferred_from_id	transferred_from_id	string
purchase_price	double	purchase_price	purchase_price	double

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

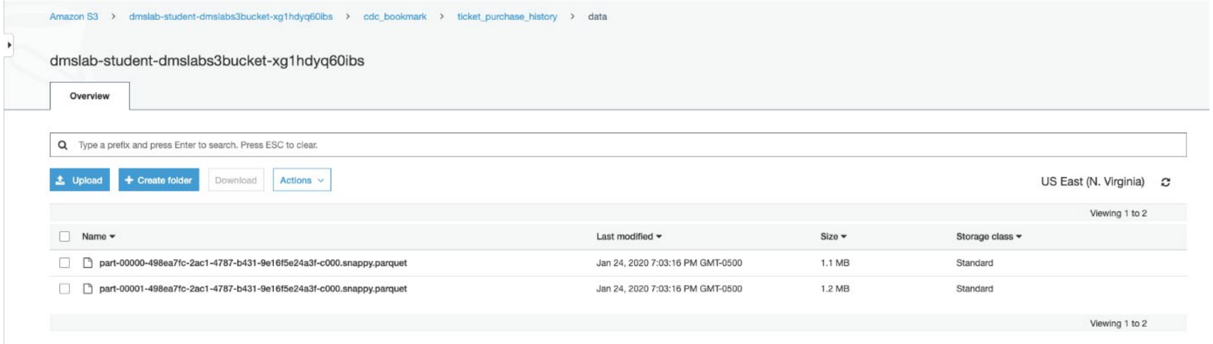

Job: Glue-Lab-TicketHistory-Parquet-with-bookmark
Action
Save
Run job
Generate diagram
Insert template at cursor
Source
Target
Target Location
Transform
Spigot
Database Name ticketdata
Table Name cdc_ticket_purchase_his
Transform Name ApplyMapping
Transform Name ResolveChoice
Transform Name DropNullFields
Path s3://dmslab-student-dmslabs3bucket-xg1hdyq60ibs/cdc_bookmark/ticket_purchase_history/data/

```
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job

## @params: [JOB_NAME]
args = getResolvedOptions(sys.argv, ['JOB_NAME'])

sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args['JOB_NAME'], args)
## @type: DataSource
## @args: [database = "ticketdata", table_name = "cdc_ticket_purchase_hist", transformation_ctx = "datasource0"]
## @return: datasource0
## @inputs: []
datasource0 = glueContext.create_dynamic_frame.from_catalog(database = "ticketdata", table_name = "cdc_ticket_purchase_hist", transformation_ctx = "datasource0")
## @type: ApplyMapping
## @args: [mapping = [("op", "string", "op", "string"), ("sporting_event_ticket_id", "string", "sporting_event_ticket_id", "string"), ("purchased_by_id", "string", "purchased_by_id", "string"), ("transaction_date_time", "string", "transaction_date_time", "string"), ("transferr
## @return: applymapping1
## @inputs: [frame = datasource0]
applymapping1 = ApplyMapping.apply(frame = datasource0, mappings = [("op", "string", "op", "string"), ("sporting_event_ticket_id", "string", "sporting_event_ticket_id", "string"), ("purchased_by_id", "string", "purchased_by_id", "string"), ("transaction_date_time", "string",
## @type: ResolveChoice
## @args: [choice = "make_struct", transformation_ctx = "resolvechoice2"]
## @return: resolvechoice2
## @inputs: [frame = applymapping1]
resolvechoice2 = ResolveChoice.apply(frame = applymapping1, choice = "make_struct", transformation_ctx = "resolvechoice2")
## @type: DropNullFields
## @args: [transformation_ctx = "dropnullfields3"]
## @return: dropnullfields3
## @inputs: [frame = resolvechoice2]
dropnullfields3 = DropNullFields.apply(frame = resolvechoice2, transformation_ctx = "dropnullfields3")
## @type: DataSink
## @args: [connection_type = "s3", connection_options = {"path": "s3://dmslab-student-dmslabs3bucket-xg1hdyq60ibs/cdc_bookmark/ticket_purchase_history/data/"}, format = "parquet", transformation_ctx = "datasink4"]
## @return: datasink4
## @inputs: [frame = dropnullfields3]
datasink4 = glueContext.write_dynamic_frame.from_options(frame = dropnullfields3, connection_type = "s3", connection_options = {"path": "s3://dmslab-student-dmslabs3bucket-xg1hdyq60ibs/cdc_bookmark/ticket_purchase_history/data/"}, format = "parquet", transformation_ctx = "dat
job.commit()
```

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

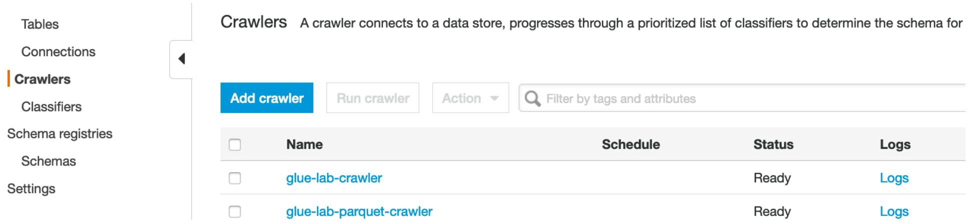

Amazon S3 › dmslab-student-dmslabs3bucket-xg1hdyq60ibs › cdc_bookmark › ticket_purchase_history › data
Overview
Type a prefix and press Enter to search. Press ESC to clear.
Upload
Create folder
Download
Actions
US East (N. Virginia)
Viewing 1 to 2

dmslab-student-dmslabs3bucket-xg1hdyq60ibs

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

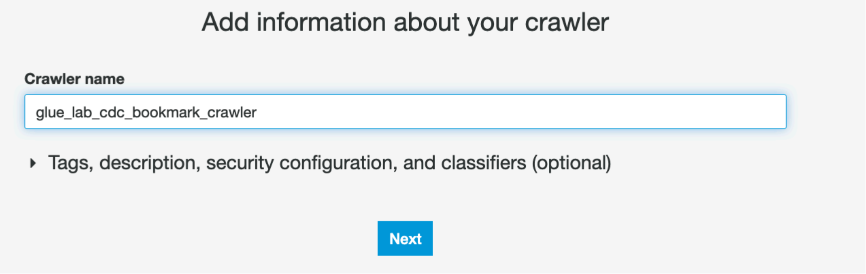
1. Once you have the data in S3 bucket, navigate to **Glue Console** and now we will crawl the parquet data in S3 to create data catalog.
2. Click on **Add crawler**



Crawlers A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for

	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**.

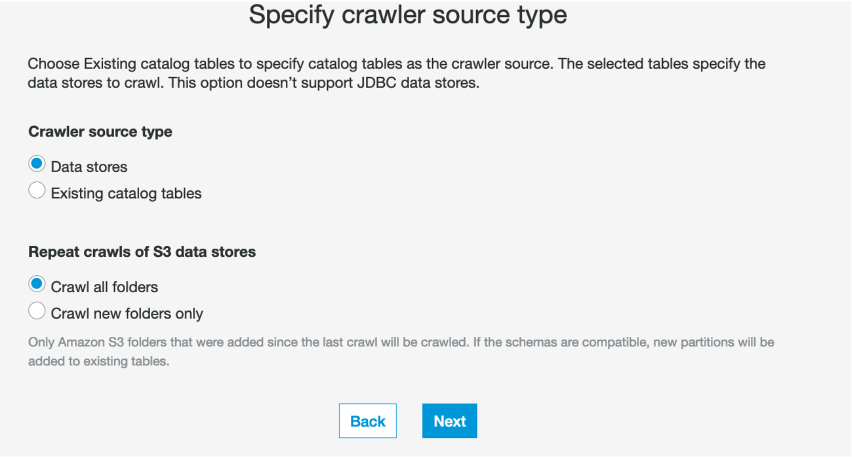


Add information about your crawler

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

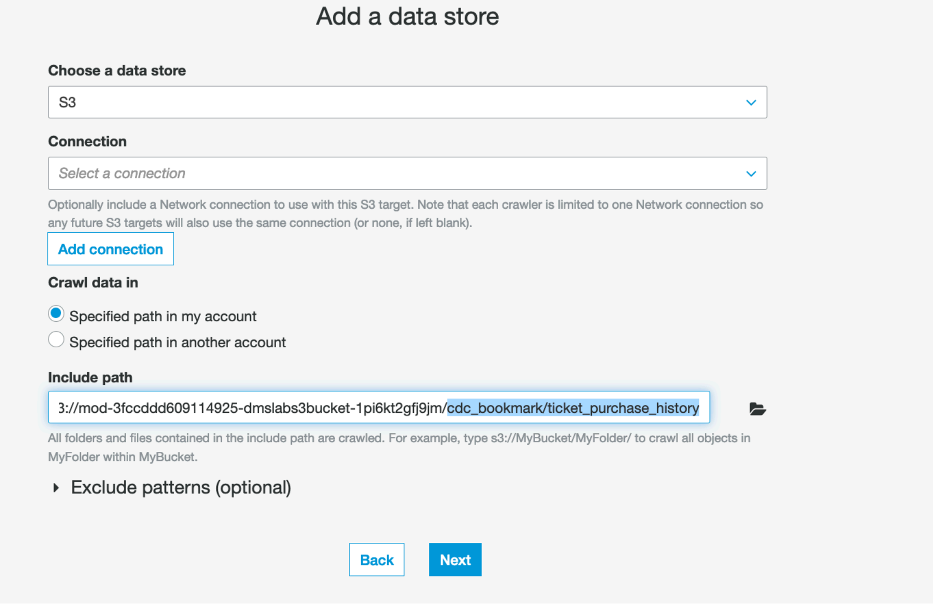
● Data stores
○ Existing catalog tables

Repeat 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.

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**



Add a data store

Choose a data store

S3

Connection

Select a connection

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).

Crawl data in

● Specified path in my account

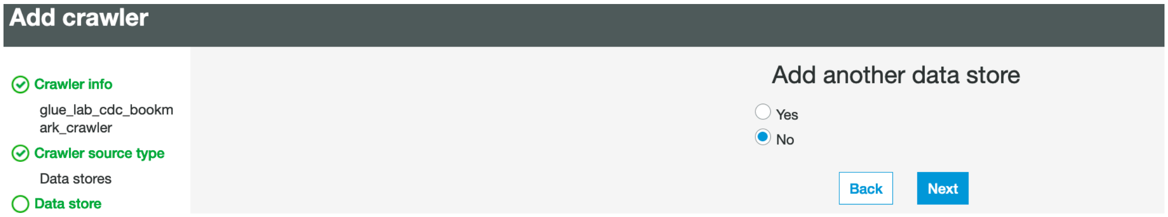
○ Specified path in another account

Include path

3://mod-3fccddd609114925-dmslabs3bucket-1pi6kt2gfj9jm/cdc_bookmark/ticket_purchase_history

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

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



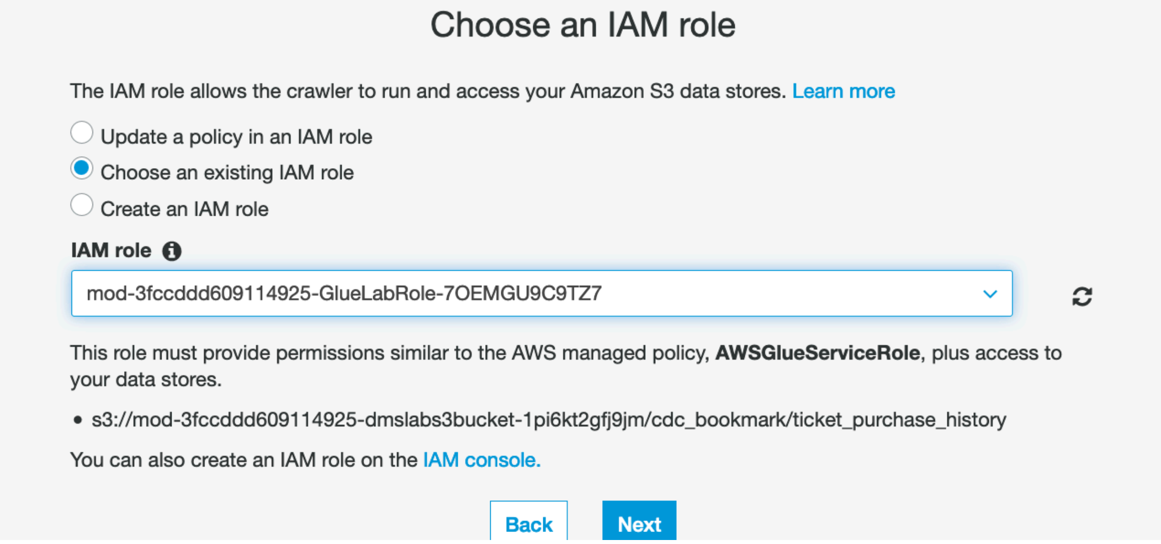
Add crawler

Add another data store

○ Yes

● No

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



Choose an IAM role

The IAM role allows the crawler to run and access your Amazon S3 data stores. Learn more

○ Update a policy in an IAM role

● Choose an existing IAM role

○ Create an IAM role

IAM role

mod-3fccddd609114925-GlueLabRole-7OEMGU9C9TZ7

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

- s3://mod-3fccddd609114925-dmslabs3bucket-1pi6kt2gfj9jm/cdc_bookmark/ticket_purchase_history

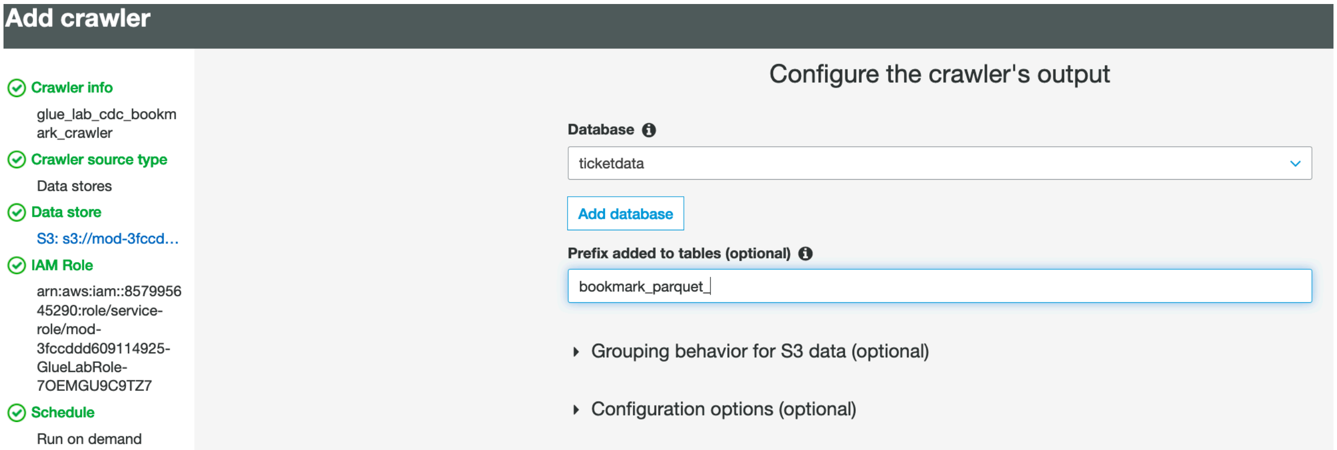
You can also create an IAM role on the IAM console.

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

10. For the crawler's output:
    a. For Database, select "**ticketdata**" database.

b. Optionally, add prefix to the newly created tables for easy identification. Provide the prefix as **bookmark_parquet_**

c. Click **Next**



Add crawler

- Crawler info: glue_lab_cdc_bookmark_crawler
- Crawler source type: Data stores
- Data store: S3: s3://mod-3fccd…
- IAM Role: arn:aws:iam::857995645290:role/service-role/mod-3fccddd609114925-GlueLabRole-7OEMGU9C9TZ7
- Schedule: Run on demand

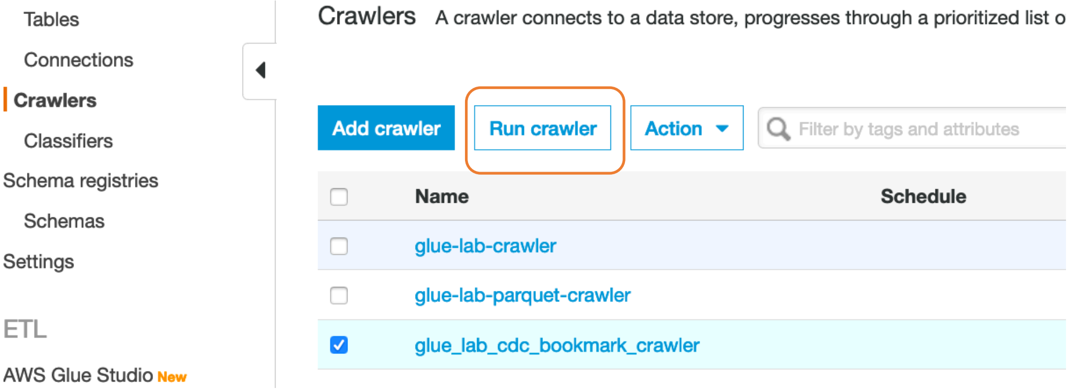
Configure the crawler's output

Database: ticketdata

Prefix added to tables (optional): bookmark_parquet_

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

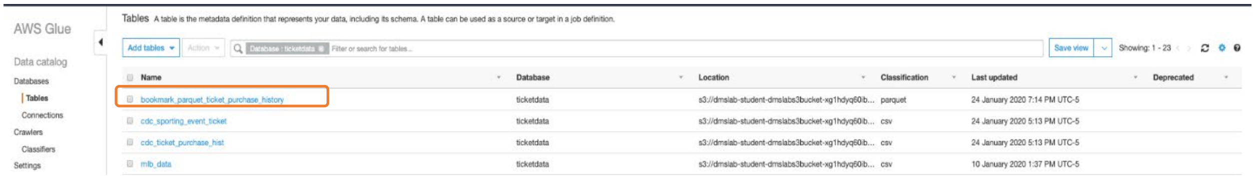
11. Review all the details and click on **Finish**. Then **Run crawler**.



Crawlers A crawler connects to a data store, progresses through a prioritized list o

	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**"

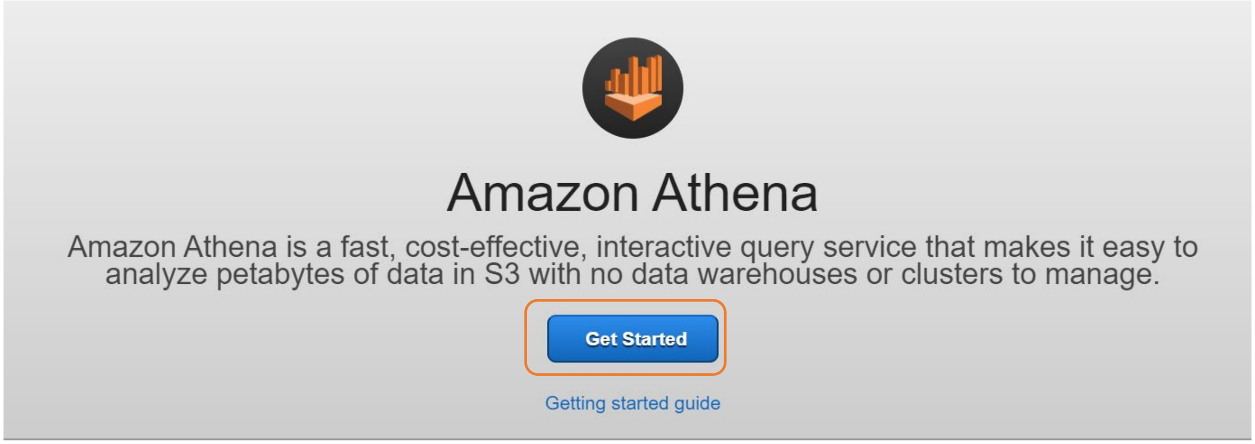


Tables A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

Name	Database	Location	Classification	Last updated	Deprecated
bookmark_parquet_ticket_purchase_history	ticketdata	s3://dmslab-student-dmslabs3bucket-xg1hdyq60ib...	parquet	24 January 2020 7:14 PM UTC-5	
cdc_sporting_event_ticket	ticketdata	s3://dmslab-student-dmslabs3bucket-xg1hdyq60ib...	csv	24 January 2020 5:13 PM UTC-5	
cdc_ticket_purchase_hist	ticketdata	s3://dmslab-student-dmslabs3bucket-xg1hdyq60ib...	csv	24 January 2020 5:13 PM UTC-5	
mlb_data	ticketdata	s3://dmslab-student-dmslabs3bucket-xg1hdyq60ib...	csv	10 January 2020 1:37 PM UTC-5	

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

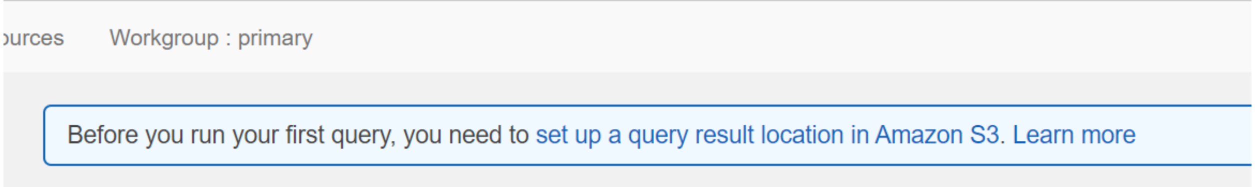
If it's the first time you are using Athena in your AWS Account, click **Get Started**



Amazon Athena

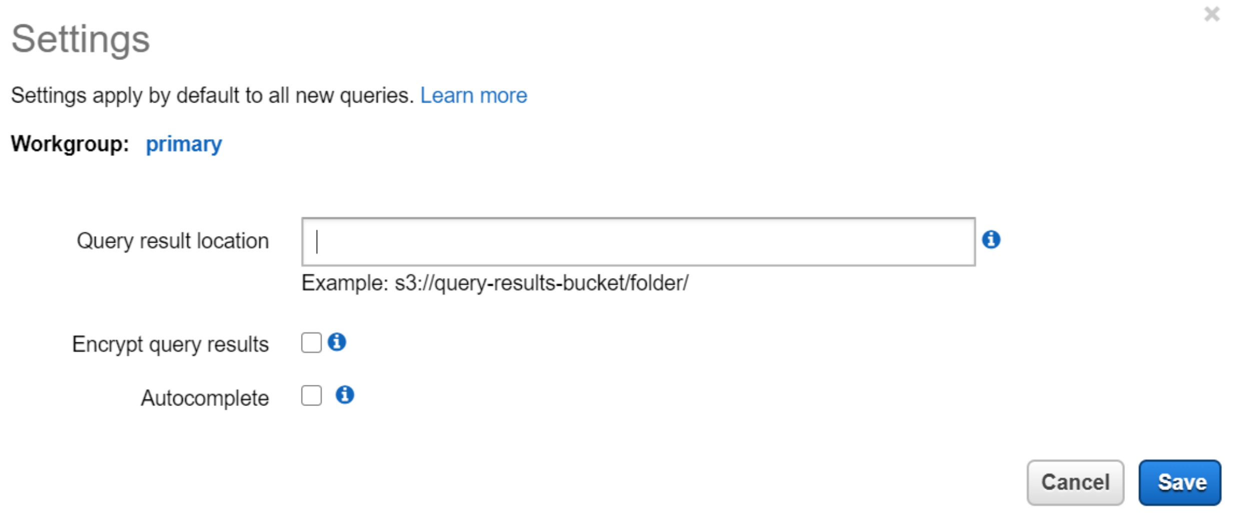
Amazon Athena is a fast, cost-effective, interactive query service that makes it easy to analyze petabytes of data in S3 with no data warehouses or clusters to manage.

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



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

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



Settings

Settings apply by default to all new queries. Learn more

Workgroup: primary

Query result location: |

Example: s3://query-results-bucket/folder/

Encrypt query results □

Autocomplete □

To select some rows from the table, try running:

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

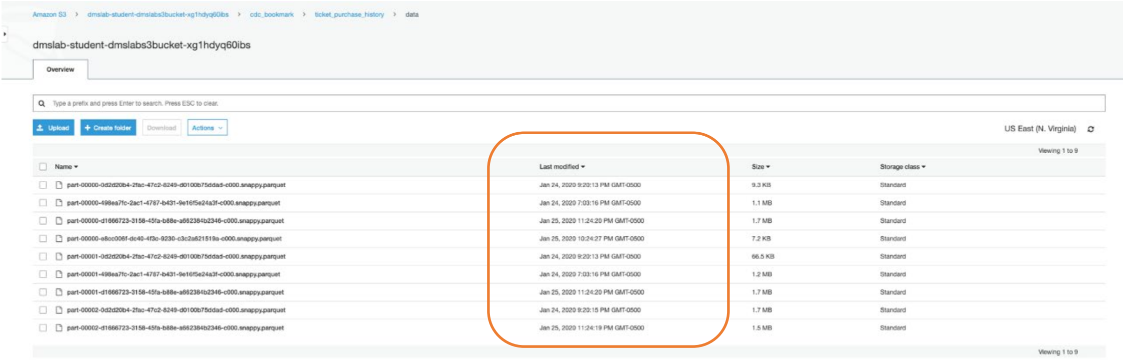
To get a row count, run:

Before moving on to next step, note the rowcount.

Step 4: Generate CDC data and to observe bookmark functionality

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

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



Amazon S3 › dmslab-student-dmslabs3bucket-xg1hdyq60ibs › cdc_bookmark › ticket_purchase_history › data

dmslab-student-dmslabs3bucket-xg1hdyq60ibs

Name	Last modified	Size	Storage class
part-00000-0d2d20b4-2fac-47c2-8249-d0100b75ddad-c000.snappy.parquet	Jan 24, 2020 9:20:13 PM GMT-0500	9.3 KB	Standard
part-00000-498ea7fc-2ac1-4787-b431-9e16f5e24a3f-c000.snappy.parquet	Jan 24, 2020 7:03:16 PM GMT-0500	1.1 MB	Standard
part-00000-d1666723-3158-45fa-b88e-a662384b2346-c000.snappy.parquet	Jan 25, 2020 11:24:20 PM GMT-0500	1.7 MB	Standard
part-00000-e8cc006f-dc40-4f3c-9230-c3c2a621519a-c000.snappy.parquet	Jan 25, 2020 10:24:27 PM GMT-0500	7.2 KB	Standard
part-00001-0d2d20b4-2fac-47c2-8249-d0100b75ddad-c000.snappy.parquet	Jan 24, 2020 9:20:13 PM GMT-0500	66.5 KB	Standard
part-00001-498ea7fc-2ac1-4787-b431-9e16f5e24a3f-c000.snappy.parquet	Jan 24, 2020 7:03:16 PM GMT-0500	1.2 MB	Standard
part-00001-d1666723-3158-45fa-b88e-a662384b2346-c000.snappy.parquet	Jan 25, 2020 11:24:20 PM GMT-0500	1.7 MB	Standard
part-00002-0d2d20b4-2fac-47c2-8249-d0100b75ddad-c000.snappy.parquet	Jan 24, 2020 9:20:15 PM GMT-0500	1.7 MB	Standard
part-00002-d1666723-3158-45fa-b88e-a662384b2346-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:

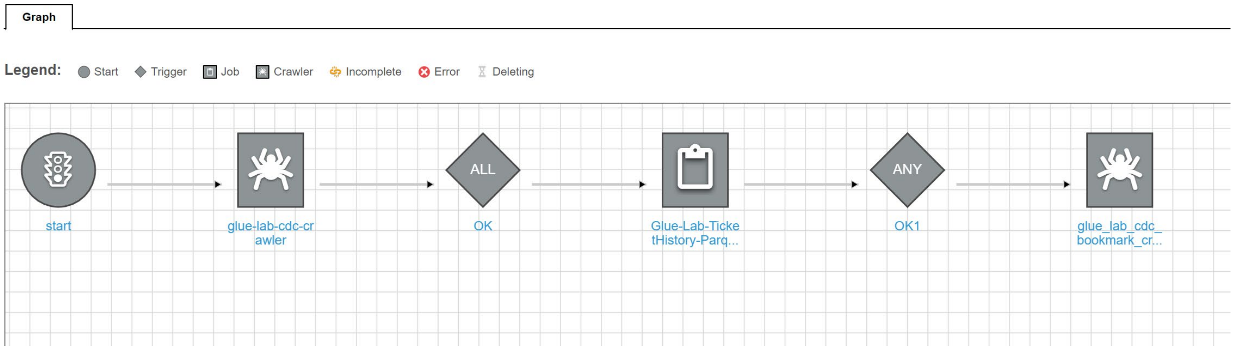
In AWS Glue, you can use workflows to create and visualize complex extract, transform, and load (ETL) activities involving multiple crawlers, jobs, and triggers. Each workflow manages the execution and monitoring of all its components. As a workflow runs each component, it records execution progress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

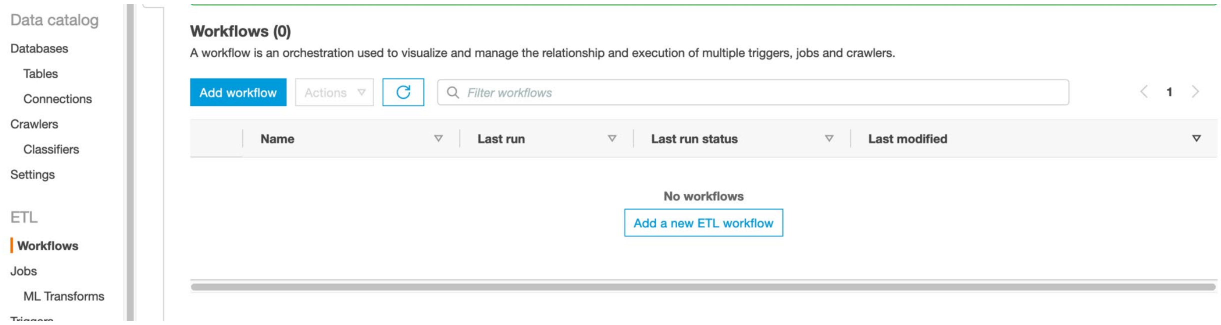
Try creating a new Glue Workflow to string together the two Crawlers and one Job from part B as follows:

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**.



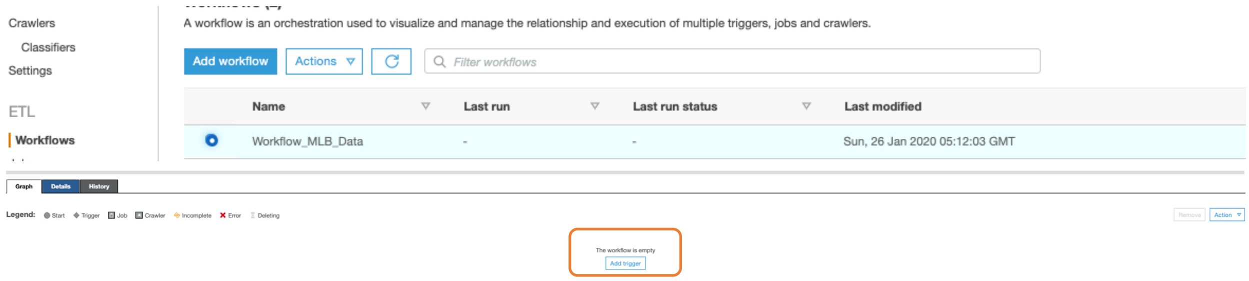
Workflows (0)

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

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.



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

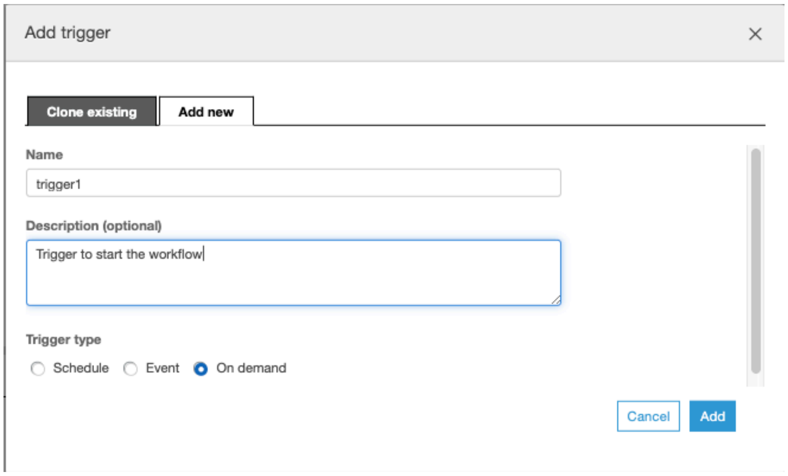
	Name	Last run	Last run status	Last modified
●	Workflow_MLB_Data	-	-	Sun, 26 Jan 2020 05:12:03 GMT

Graph
Details
History
Legend: ● Start ◆ Trigger ▣ Job ▣ Crawler Incomplete ✗ Error Deleting
Remove
Action

The workflow is empty

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



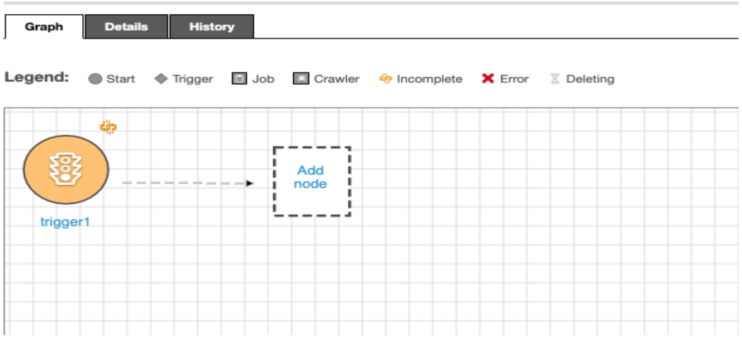
Add trigger

Name: trigger1

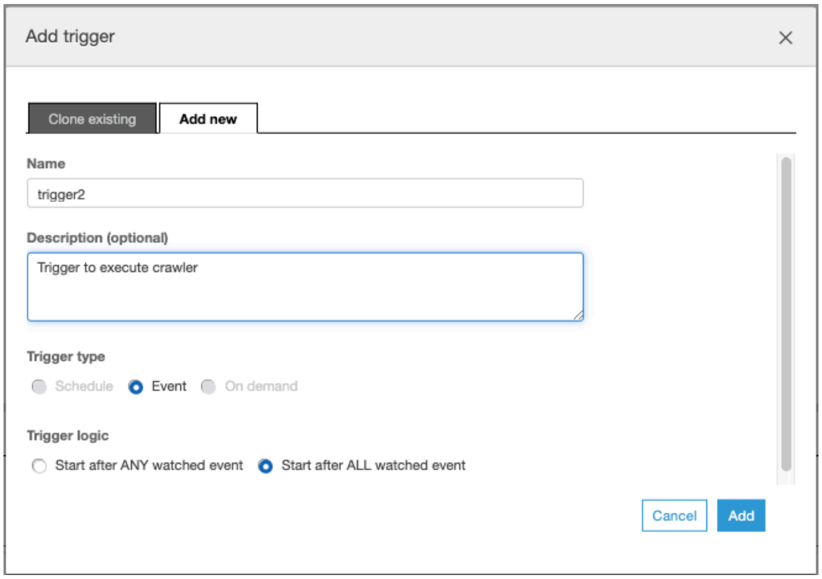
Description (optional): Trigger to start the workflow

Trigger type: ○ Schedule ○ Event ● On demand

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


Graph
Details
History
Legend: ● Start ◆ Trigger ▣ Job ▣ Crawler Incomplete ✖ Error Deleting
trigger1
Add node

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

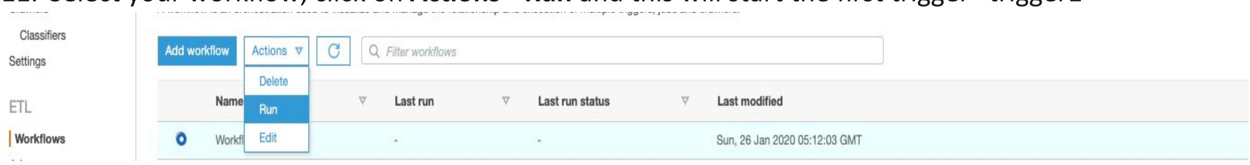
Description (optional): Trigger to execute crawler

Trigger type: ○ Schedule ● Event ○ On demand

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

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

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



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

Congratulations!! You have successfully completed this lab