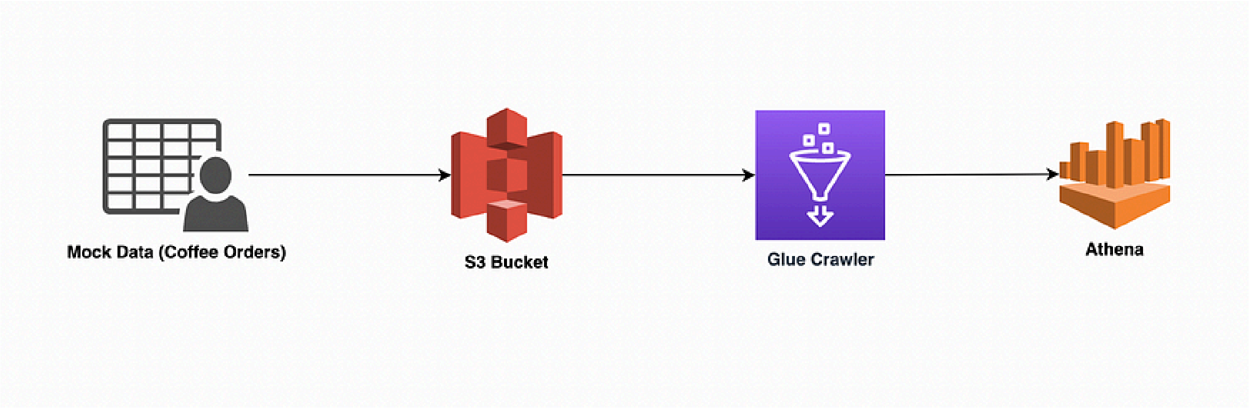
Create an ETL Job using AWS Glue Studio, S3, & Athena

Background



In this blog post, I will walk us through on how to create an ETL job using the following AWS Cloud Services:

- **Amazon S3** *(for intermediary data storage)*
- **AWS Glue Studio** *(main engine of the ETL job)*
- **Amazon Athena** *(query tool)*

ETL stands for **Extract**, **Transform**, and **Load**.

In almost all Enterprise IT settings, it is very common to have huge volumes of Analytical data. In order for that data to be consumable and used by business areas to generate insights and forecast business goals, the data needs to be:

1. Extracted from where it is stored/originated,
2. Transformed to fit a certain format (usually involves cleaning the data, removing duplicates, etc.)

and then,

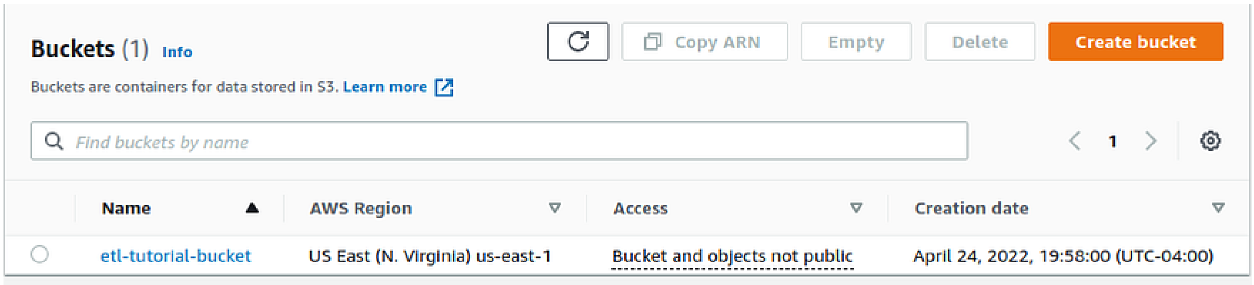
3. Loaded into the target destination source.

Very often, there is also the presence of a large data warehouse, and multiple databases/storage layers feeding data into the larger warehouse for general business consumption.

Without further ado, we will now create an ETL Job using AWS Glue (an event-driven, serverless computing service).

Step 1 — Create an S3 Bucket

This S3 Bucket will store our data. We're basically just building a database within the S3 Object storage service.



Buckets (1)

Buckets are containers for data stored in S3. Learn more

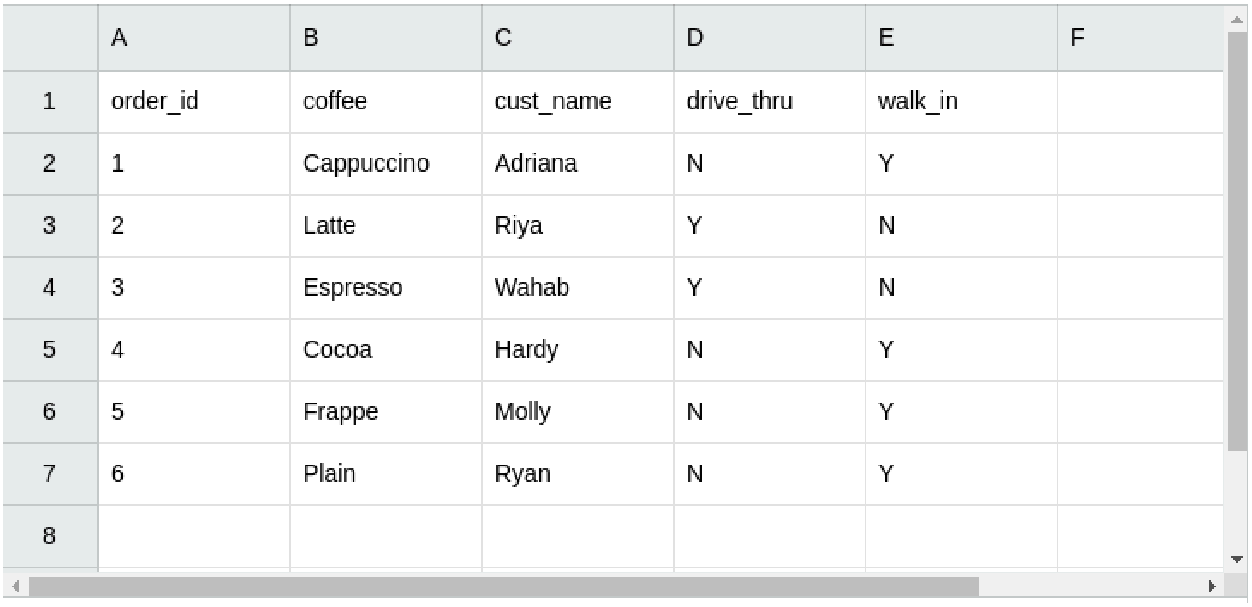
	Name	AWS Region	Access	Creation date
○	etl-tutorial-bucket	US East (N. Virginia) us-east-1	Bucket and objects not public	April 24, 2022, 19:58:00 (UTC-04:00)

S3 Bucket created (keep default options when creating it)

Step 2 — Create some Mock Data

For this step, I just created a quick .csv file with data about coffee orders.

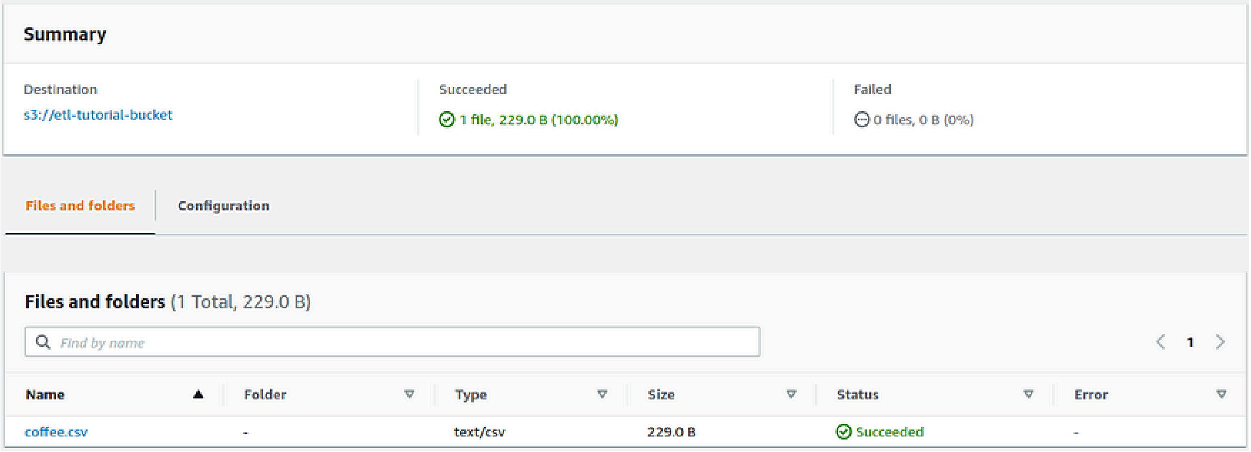
This data file will be uploaded to the etl-tutorial-bucket from the previous step.



	A	B	C	D	E	F
1	order_id	coffee	cust_name	drive_thru	walk_in	
2	1	Cappuccino	Adriana	N	Y	
3	2	Latte	Riya	Y	N	
4	3	Espresso	Wahab	Y	N	
5	4	Cocoa	Hardy	N	Y	
6	5	Frappe	Molly	N	Y	
7	6	Plain	Ryan	N	Y	
8						

coffee.csv

Step 3 — Upload the coffee data file to the S3 Bucket



Summary

Destination
s3://etl-tutorial-bucket

Succeeded
✓ 1 file, 229.0 B (100.00%)

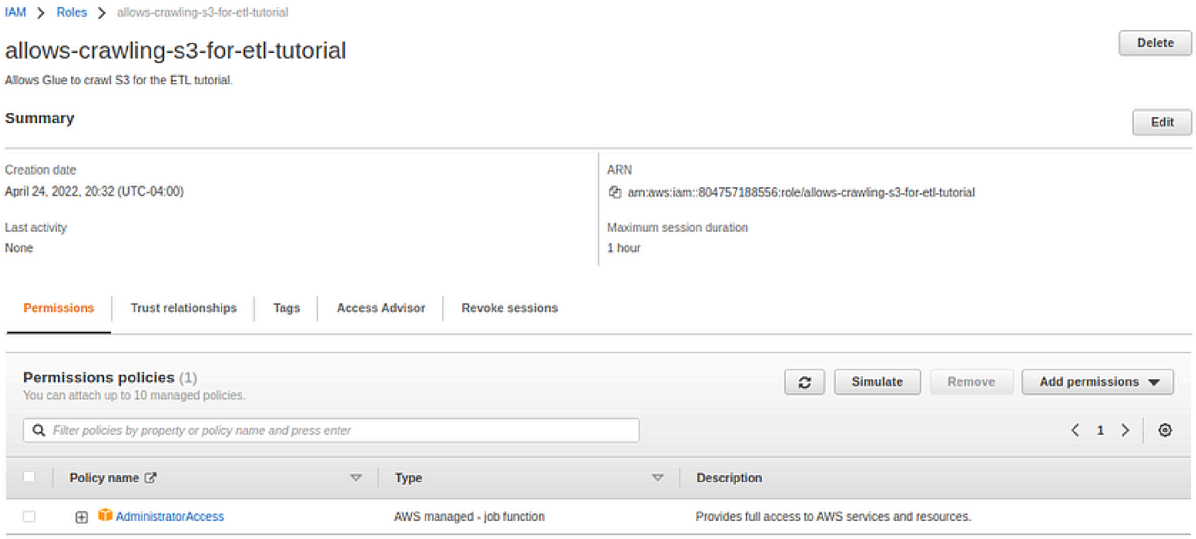
Failed
⊖ 0 files, 0 B (0%)

Files and folders (1 Total, 229.0 B)

Name	Folder	Type	Size	Status	Error
coffee.csv	-	text/csv	229.0 B	✓ Succeeded	-

coffee.csv uploaded to S3

Step 4 — Create this IAM Role for Glue with the Required permissions



allows-crawling-s3-for-etl-tutorial

Allows Glue to crawl S3 for the ETL tutorial.

Summary

Creation date
April 24, 2022, 20:32 (UTC-04:00)

Last activity
None

ARN
arn:aws:iam::804757188556:role/allows-crawling-s3-for-etl-tutorial

Maximum session duration
1 hour

Permissions policies (1)
You can attach up to 10 managed policies.

Policy name	Type	Description
AdministratorAccess	AWS managed - job function	Provides full access to AWS services and resources.

Step 5— Define the Glue Data Crawler in AWS Glue

AWS Glue was launched in 2017. It is a fully-managed, serverless, and AWS Cloud-optimized ETL Service offering.

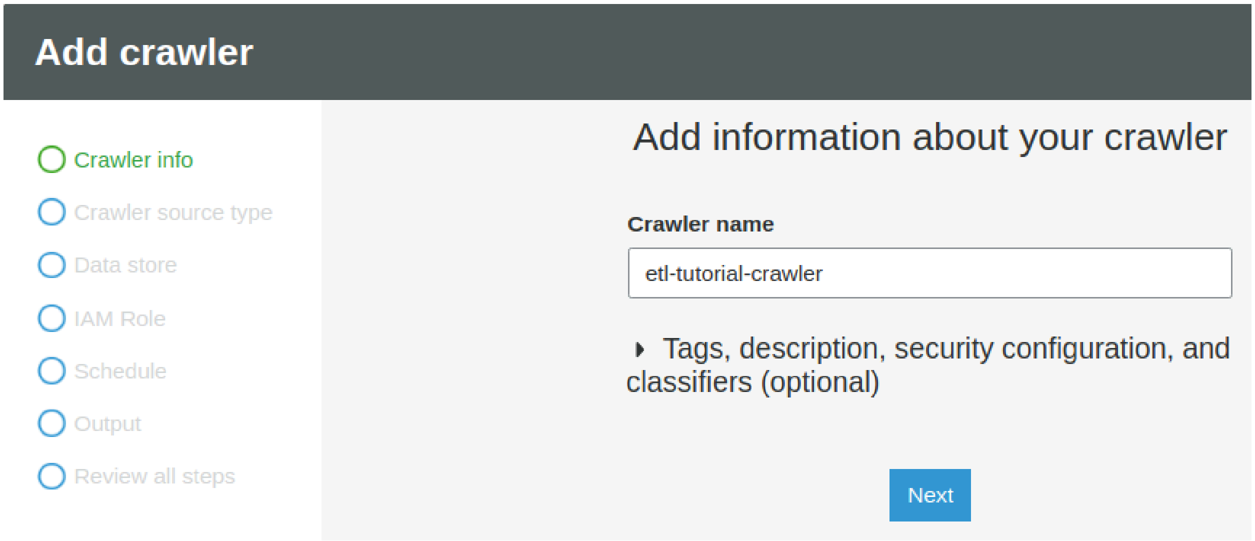
Using Glue, there are numerous ways we can connect to our data store in S3. However, for this tutorial, I am going to demo this using Glue Crawler.

Glue Crawler is one of the most widely-used method among Data Engineers.

Basically, this Crawler will “crawl” the data file in our S3 bucket, and using the data available in the file in S3 will create a table schema in Glue.

To create and add the Crawler to our S3 data store:

5a. Gave my crawler the name `etl-tutorial-crawler` then click **Next**.



Add crawler

- ○ Crawler info
- ○ Crawler source type
- ○ Data store
- ○ IAM Role
- ○ Schedule
- ○ Output
- ○ Review all steps

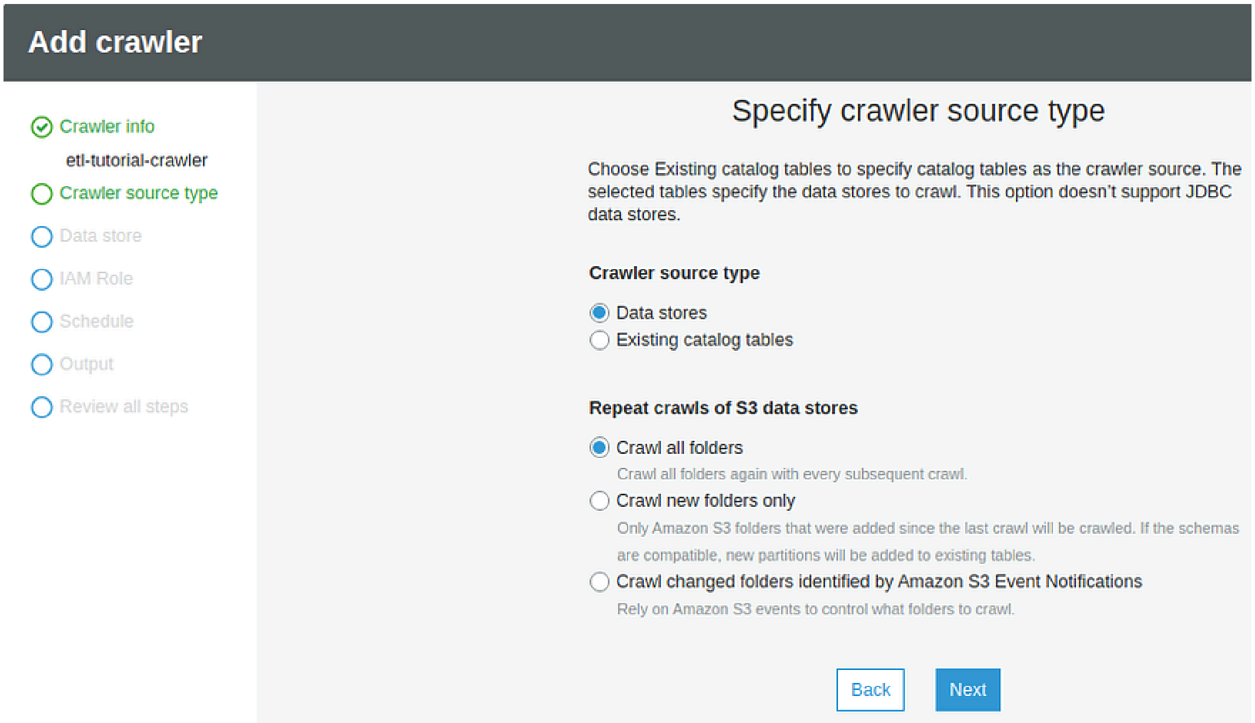
Add information about your crawler

Crawler name

etl-tutorial-crawler

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

5b. You can keep the default settings as is for the Crawler. Click **Next**.



Add crawler

- ✓ Crawler info
  etl-tutorial-crawler
- ○ Crawler source type
- ○ Data store
- ○ IAM Role
- ○ Schedule
- ○ Output
- ○ Review all steps

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.

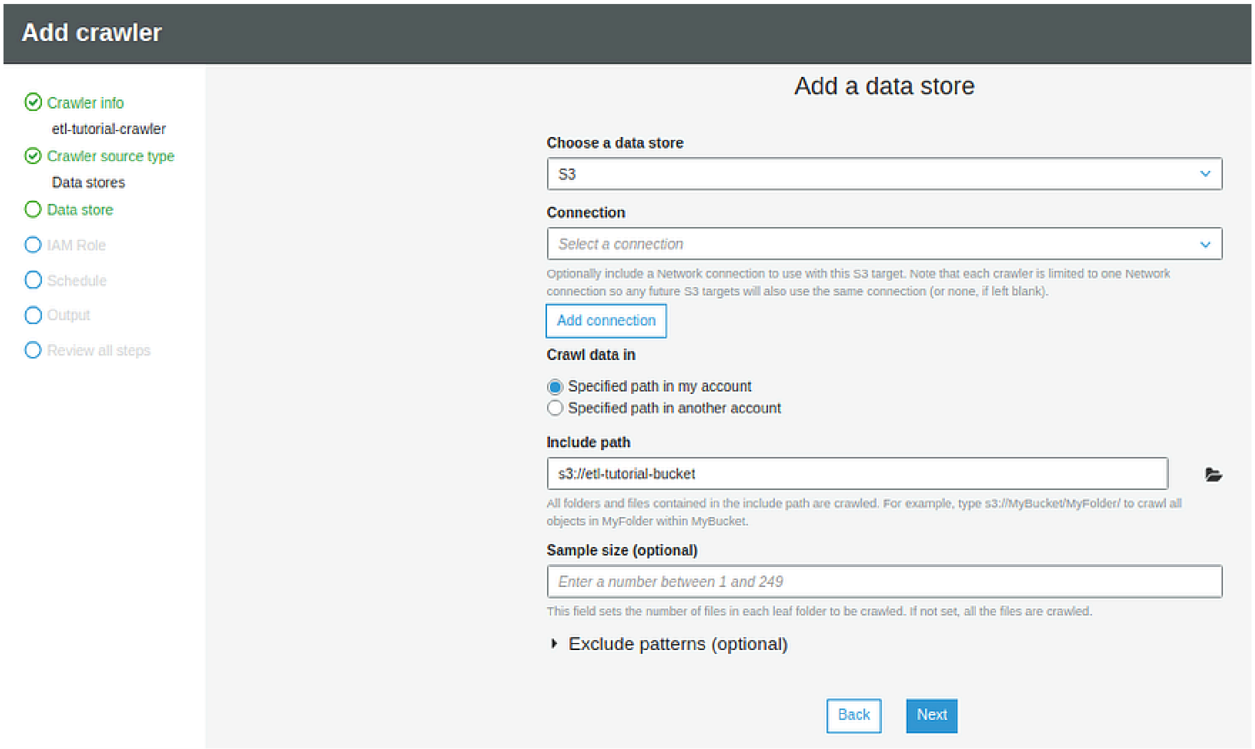
Crawler source type

● Data stores
○ Existing catalog tables

Repeat crawls of S3 data stores

● Crawl all folders
Crawl all folders again with every subsequent crawl.
○ 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.
○ Crawl changed folders identified by Amazon S3 Event Notifications
Rely on Amazon S3 events to control what folders to crawl.

5c. Select **S3** for the **data store**, and add the name of your S3 bucket. The click **Next**.



Add crawler

- Crawler info: etl-tutorial-crawler
- Crawler source type: Data stores
- Data store
- IAM Role
- Schedule
- Output
- Review all steps

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).

Add connection

Crawl data in

● Specified path in my account
○ Specified path in another account

Include path: s3://etl-tutorial-bucket

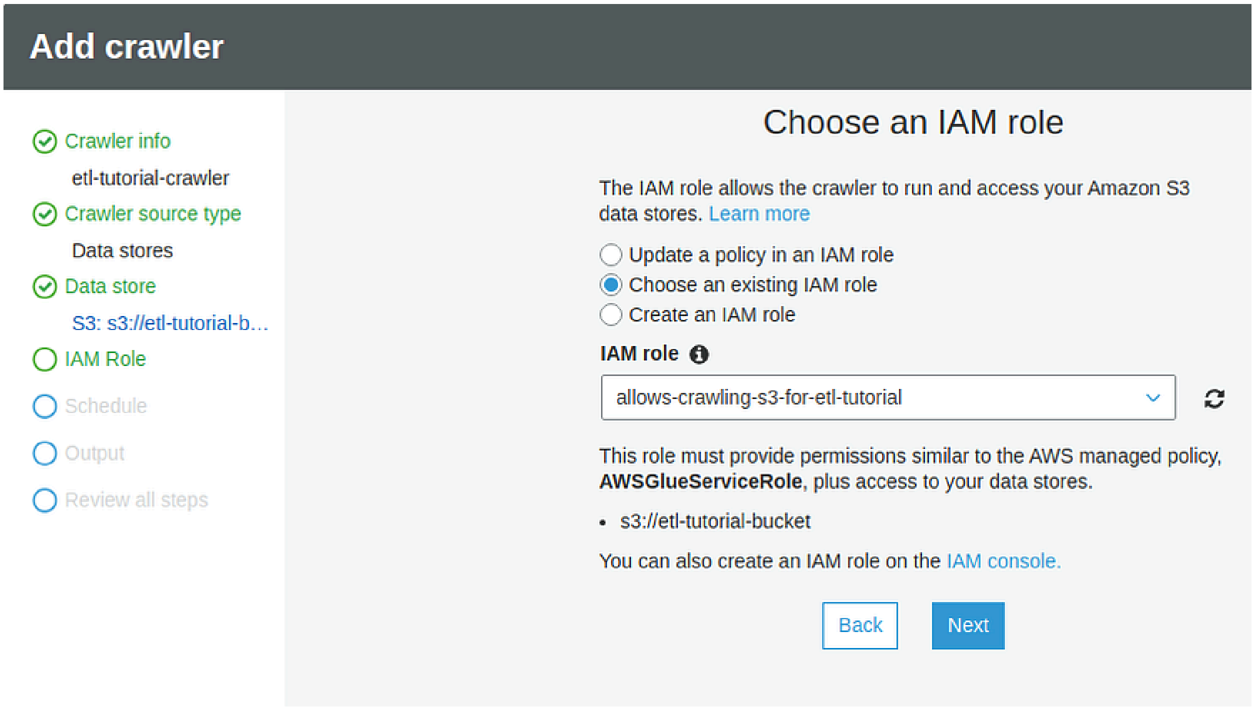
All folders and files contained in the include path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

Sample size (optional): Enter a number between 1 and 249

This field sets the number of files in each leaf folder to be crawled. If not set, all the files are crawled.

▸ Exclude patterns (optional)

5d. Select the **Choose an existing IAM Role** radio button, and select the IAM Role we created in Step 4. Then click **Next**.



Add crawler

✓ Crawler info
etl-tutorial-crawler
✓ Crawler source type
Data stores
✓ Data store
S3: s3://etl-tutorial-b...
○ 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

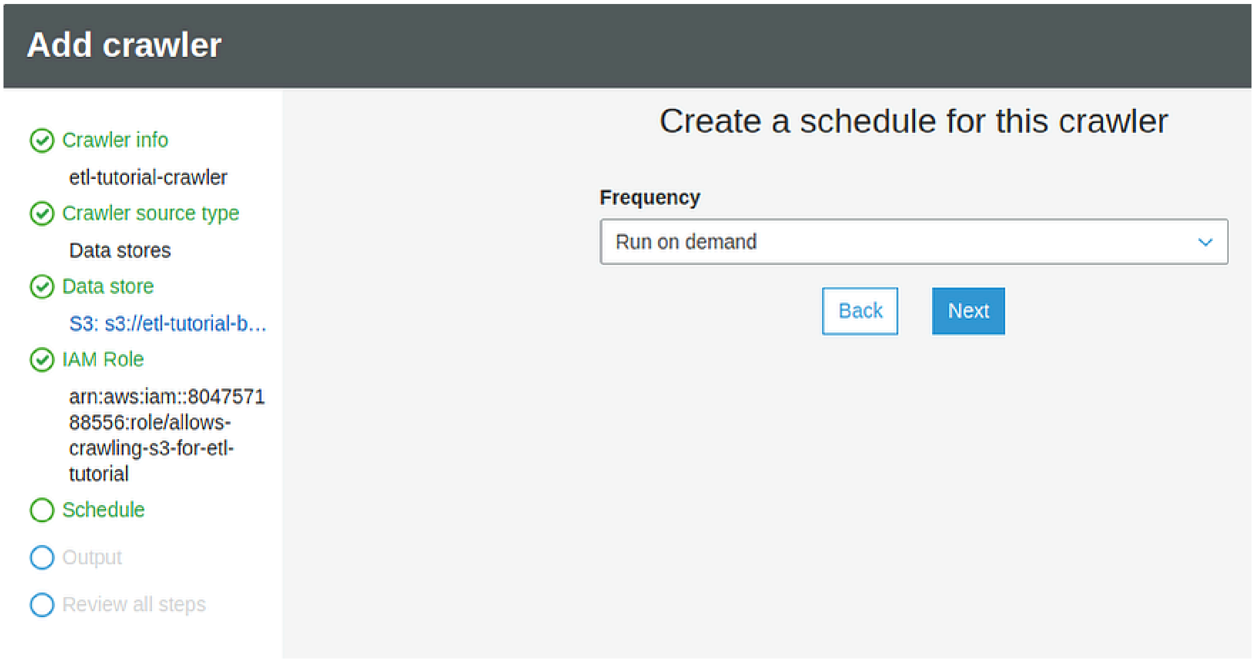
allows-crawling-s3-for-etl-tutorial

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

- s3://etl-tutorial-bucket

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

5e. Set the frequency to **Run on Demand**. Then click **Next**. (Note: if required, you can always set customized scheduling for your Crawler to run on an hourly, daily, weekly, o monthly cadence).



Add crawler

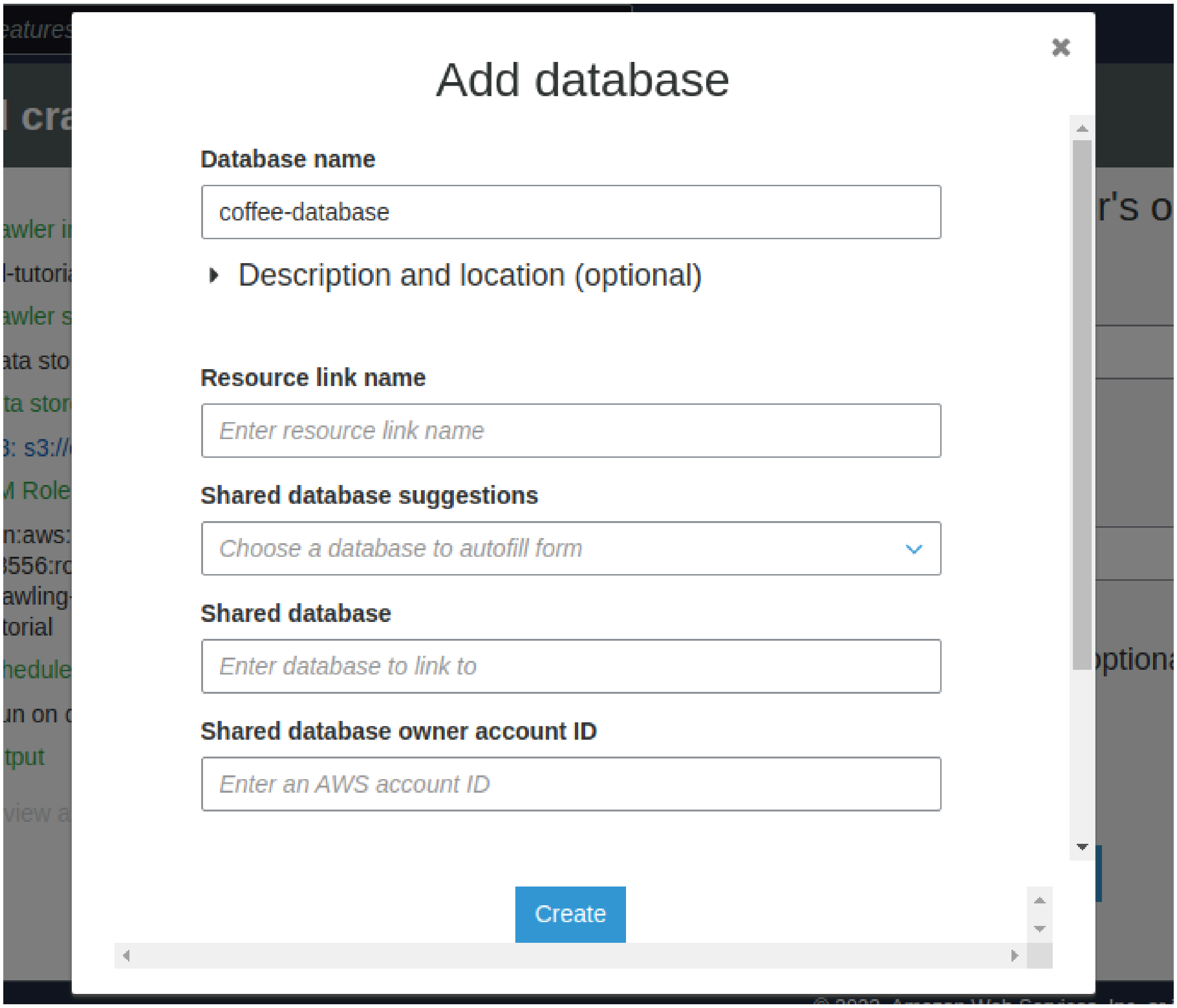
✓ Crawler info
etl-tutorial-crawler
✓ Crawler source type
Data stores
✓ Data store
S3: s3://etl-tutorial-b...
✓ IAM Role
arn:aws:iam::804757188556:role/allows-crawling-s3-for-etl-tutorial
○ Schedule
○ Output
○ Review all steps

Create a schedule for this crawler

Frequency

Run on demand

5f. For storing the output from the Crawler, I created a database `coffee-database` and then clicked **Next**.



Add database

Database name: coffee-database

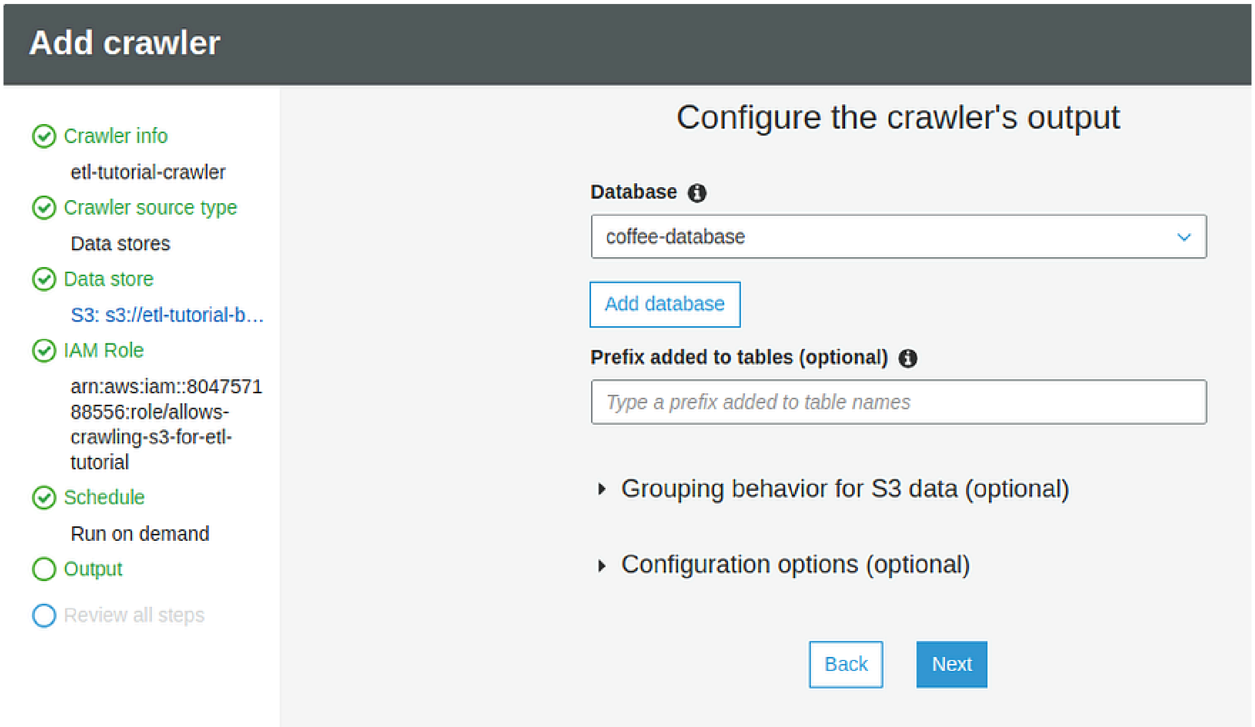
Description and location (optional)

Resource link name: Enter resource link name

Shared database suggestions: Choose a database to autofill form

Shared database: Enter database to link to

Shared database owner account ID: Enter an AWS account ID



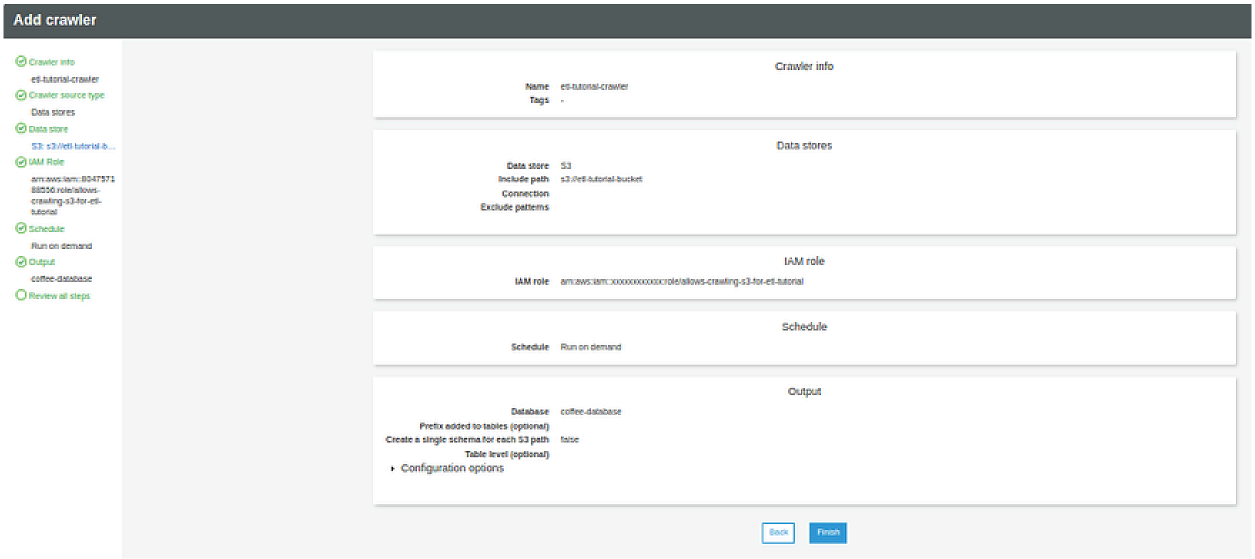
Configure the crawler's output

Crawler info: etl-tutorial-crawler
Crawler source type: Data stores
Data store: S3: s3://etl-tutorial-b…
IAM Role: arn:aws:iam::804757188556:role/allows-crawling-s3-for-etl-tutorial
Schedule: Run on demand

Database: coffee-database

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

5g. Lastly, review all the settings that we've configured and click **Finish** to create our Crawler.



Crawler info: etl-tutorial-crawler
Crawler source type: Data stores
Data store: S3: s3://etl-tutorial-b…
IAM Role: arn:aws:iam::8047571 88556:role/allows-crawling-s3-for-etl-tutorial
Schedule: Run on demand
Output: coffee-database

Crawler info

Name etl-tutorial-crawler
Tags -

Data stores

Data store S3
Include path s3://etl-tutorial-bucket
Connection
Exclude patterns

IAM role

IAM role arn:aws:iam::xxxxxxxxxxxx:role/allows-crawling-s3-for-etl-tutorial

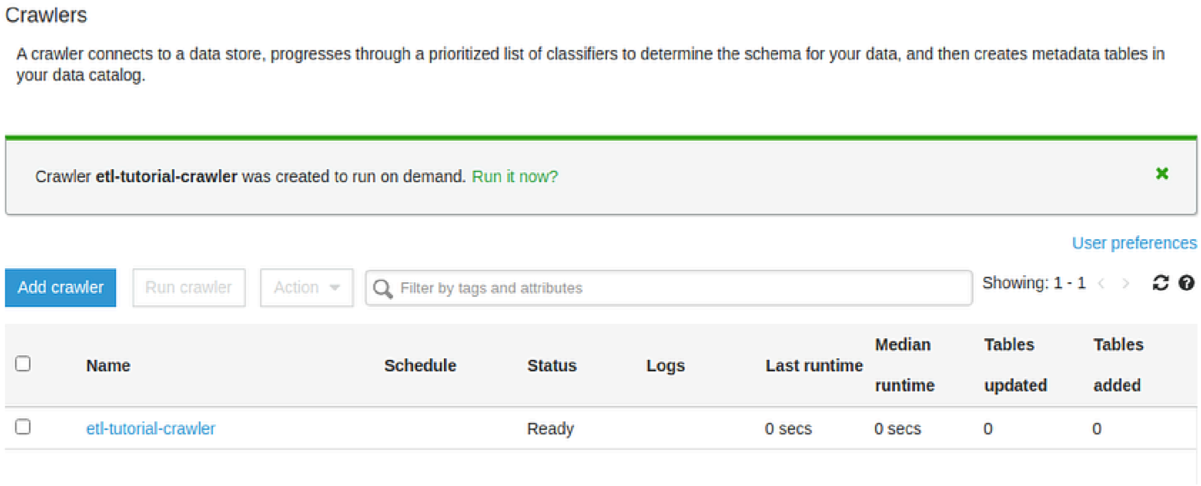
Schedule

Schedule Run on demand

Output

Database coffee-database
Prefix added to tables (optional)
Create a single schema for each S3 path false
Table level (optional)

5h. Once the Crawler has been created, we should see this:



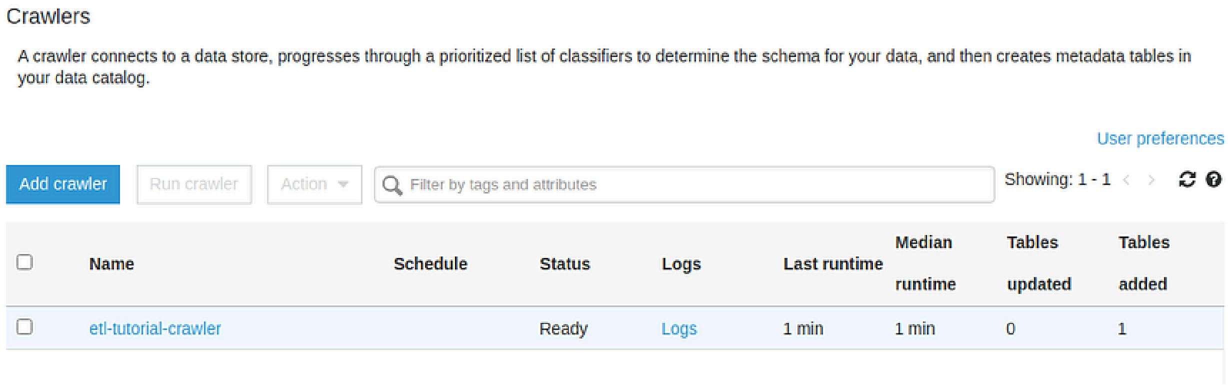
Crawlers

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 **etl-tutorial-crawler** was created to run on demand. Run it now?

□	Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
□	etl-tutorial-crawler		Ready		0 secs	0 secs	0	0

Try running the Crawler. If the job status updates from starting to stopping, and then to ready, that tells that the Crawler job was successful.



Crawlers

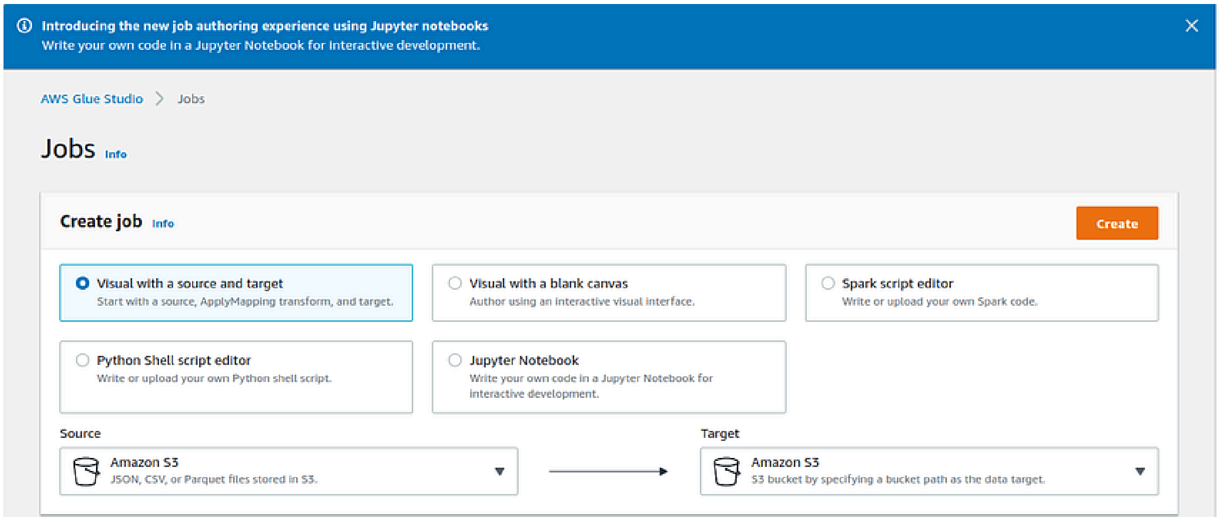
A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

□	Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
□	etl-tutorial-crawler		Ready	Logs	1 min	1 min	0	1

Step 6 — Defining our Glue Job

Now that we have finished setting up the Glue Crawler to point and recognize the data file in S3, we will define our Glue job to perform the ETL portion of this tutorial.

6a. From the AWS Console, go to AWS Glue Studio:



Introducing the new job authoring experience using Jupyter notebooks
Write your own code in a Jupyter Notebook for interactive development.

AWS Glue Studio > Jobs

Jobs Info

Create job Info

● 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.

○ Spark script editor
Write or upload your own Spark code.

○ Python Shell script editor
Write or upload your own Python shell script.

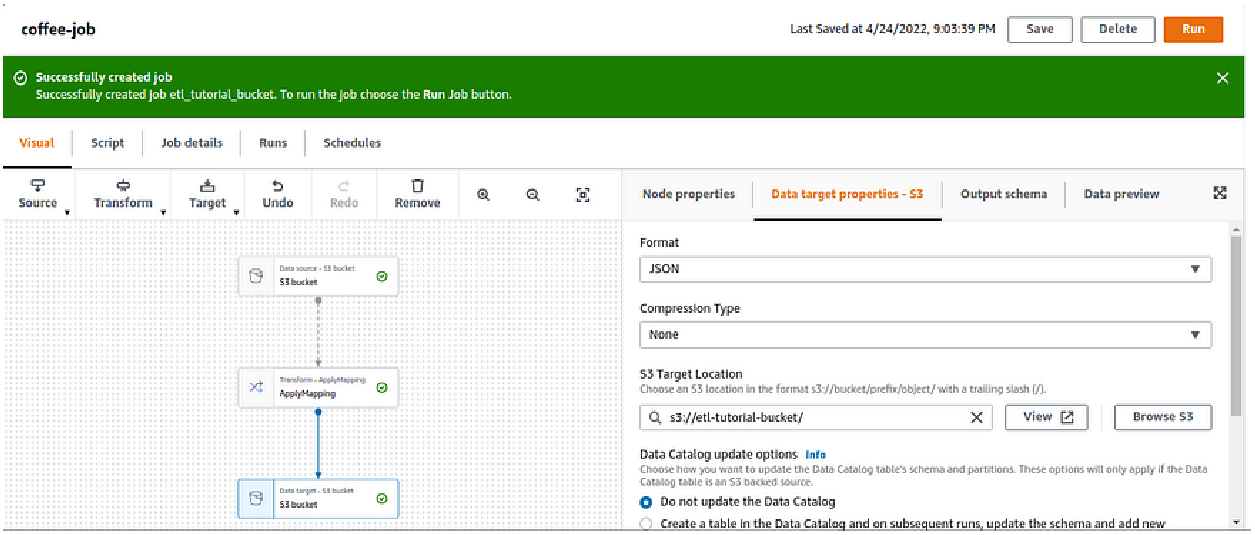
○ Jupyter Notebook
Write your own code in a Jupyter Notebook for interactive development.

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.

You will want to select Amazon S3 as both the Source and Target. Click **Create**.

Set up your Glue Job to look like this:



coffee-job

Last Saved at 4/24/2022, 9:03:39 PM

Successfully created job
Successfully created job etl_tutorial_bucket. To run the job choose the Run Job button.

Data source - S3 bucket: S3 bucket

Transform - ApplyMapping: ApplyMapping

Data target - S3 bucket: S3 bucket

Format: JSON

Compression Type: None

S3 Target Location
Choose an S3 location in the format s3://bucket/prefix/object/ with a trailing slash (/).

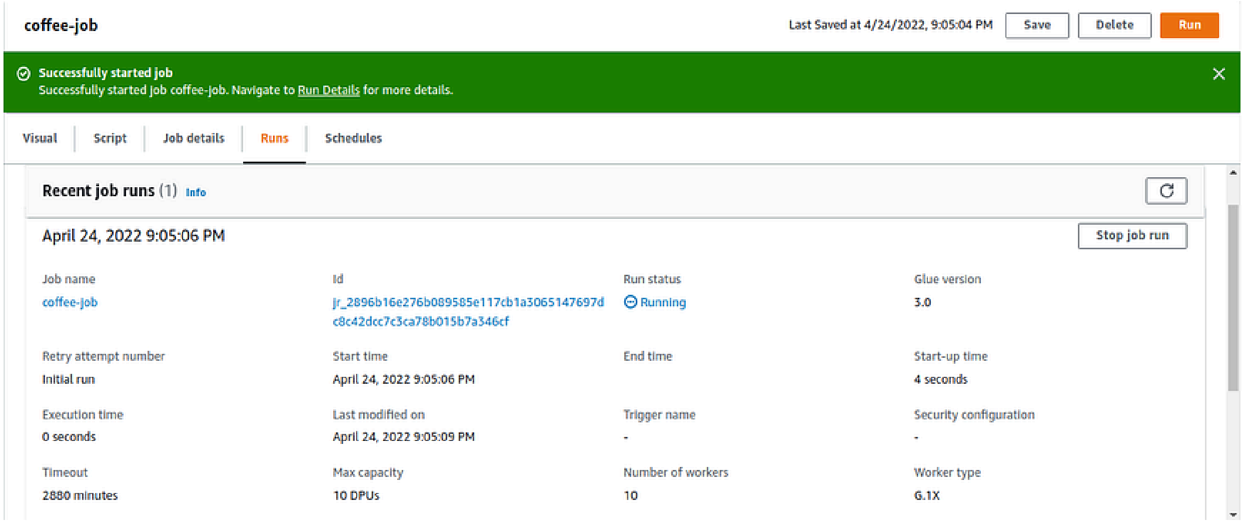
s3://etl-tutorial-bucket/

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

6b. Run the job.

While it’s running, you should see an output like this:



Successfully started job
Successfully started job coffee-job. Navigate to Run Details for more details.

April 24, 2022 9:05:06 PM

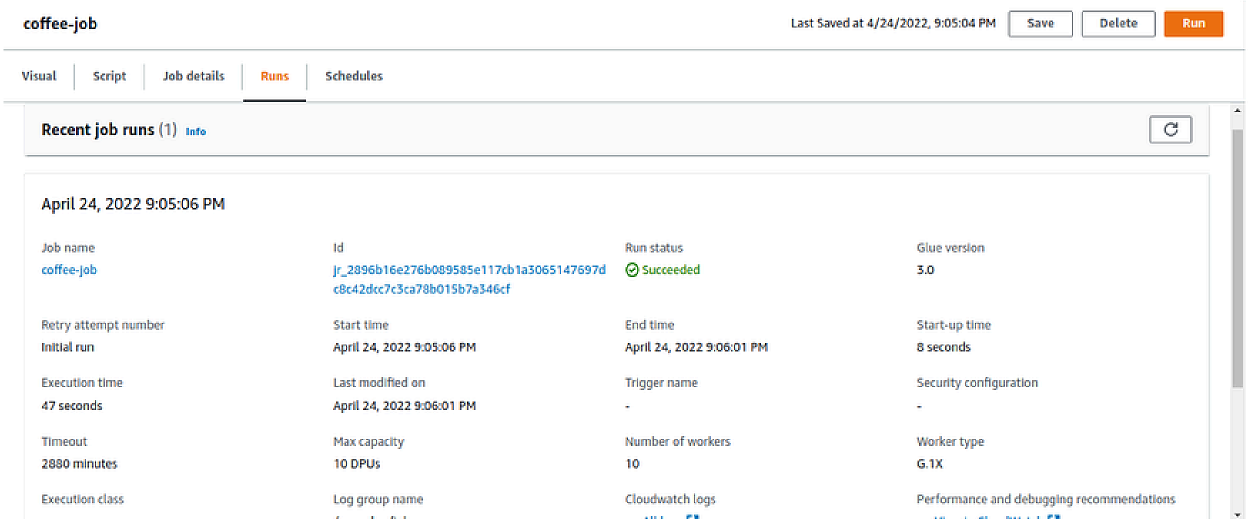
Job name: coffee-job
Id: jr_2896b16e276b089585e117cb1a3065147697dc8c42dcc7c3ca78b015b7a346cf
Run status: Running
Glue version: 3.0

Retry attempt number: Initial run
Start time: April 24, 2022 9:05:06 PM
End time:
Start-up time: 4 seconds

Execution time: 0 seconds
Last modified on: April 24, 2022 9:05:09 PM
Trigger name: -
Security configuration: -

Timeout: 2880 minutes
Max capacity: 10 DPUs
Number of workers: 10
Worker type: G.1X

After the Glue job is finished running, your output should look like this:



April 24, 2022 9:05:06 PM

Job name: coffee-job
Id: jr_2896b16e276b089585e117cb1a3065147697dc8c42dcc7c3ca78b015b7a346cf
Run status: Succeeded
Glue version: 3.0

Retry attempt number: Initial run
Start time: April 24, 2022 9:05:06 PM
End time: April 24, 2022 9:06:01 PM
Start-up time: 8 seconds

Execution time: 47 seconds
Last modified on: April 24, 2022 9:06:01 PM
Trigger name: -
Security configuration: -

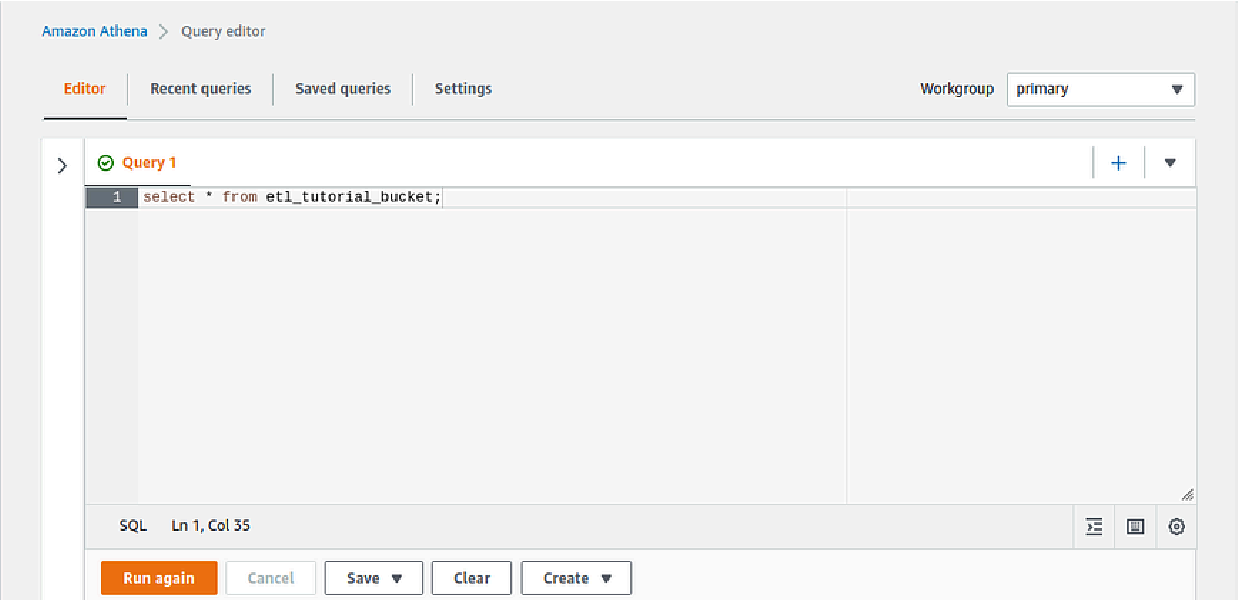
Timeout: 2880 minutes
Max capacity: 10 DPUs
Number of workers: 10
Worker type: G.1X

Execution class
Log group name
Cloudwatch logs
Performance and debugging recommendations

Step 7 — Querying our data using Athena

AWS Athena is a query service offered by AWS that analyzes data in S3 using SQL. For this tutorial, I will run Athena to query the ETL data and run some SQL operations.

Go to the AWS console, and search for Athena. Then run this query as shown below:

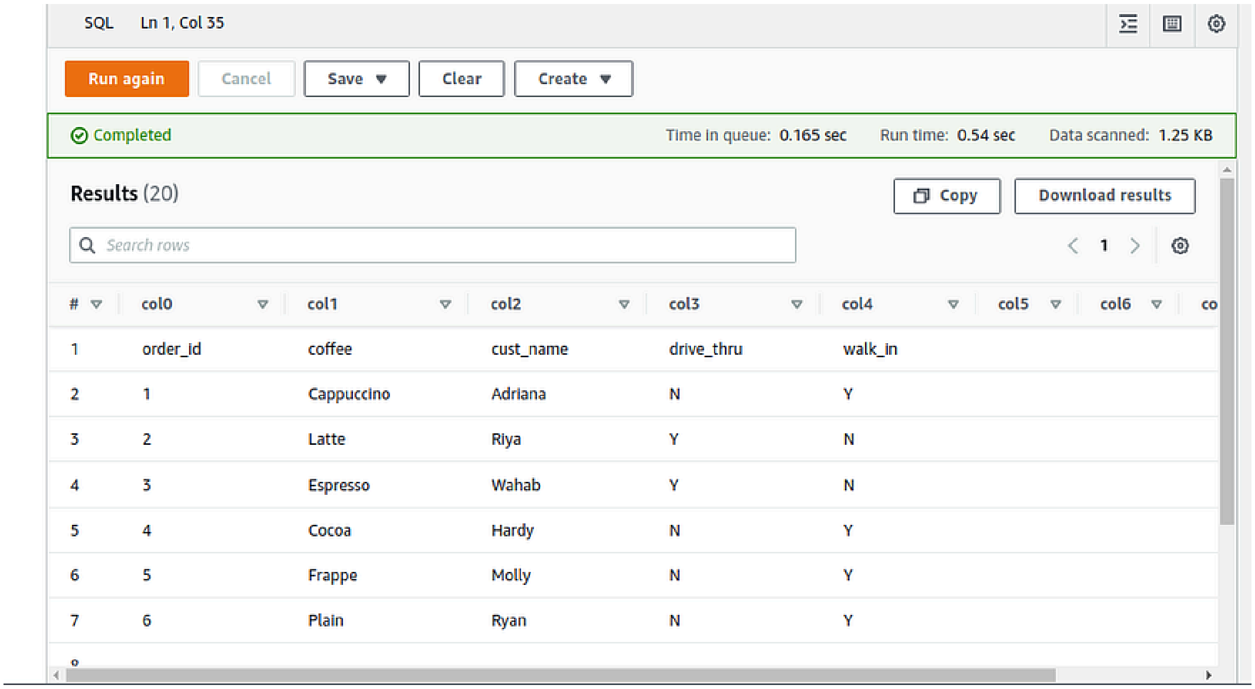


Workgroup: primary

```
select * from etl_tutorial_bucket;
```

Ln 1, Col 35

After running the query, we should see our results. :)



Ln 1, Col 35

Completed Time in queue: 0.165 sec Run time: 0.54 sec Data scanned: 1.25 KB

Results (20)

#	col0	col1	col2	col3	col4	col5	col6	co
1	order_id	coffee	cust_name	drive_thru	walk_in			
2	1	Cappuccino	Adriana	N	Y			
3	2	Latte	Riya	Y	N			
4	3	Espresso	Wahab	Y	N			
5	4	Cocoa	Hardy	N	Y			
6	5	Frappe	Molly	N	Y			
7	6	Plain	Ryan	N	Y			

Overall summary

When it comes to data engineering, in general, ETL operations and pipelines are a huge part of it. Apart from Glue, there are a multitude of other ETL services (i.e. Cloud Air Fusion by Google Cloud, Data Factory offerred by Azure, and many others).

Aside from whichever one you pick, the basic idea is that it should assist with reduction of engineering efforts required by supplying a base (source and target locations and configurations).

We have just completed ETL using AWS Glue. :)

My hope is that this walkthrough helps you with your projects!

In case I missed anything, or if you have any questions/thoughts, please submit them in the comments section below.