AWS GLUE CRAWLER: A COMPLETE SETUP GUIDE



In this tutorial, we discuss what a crawler is in Amazon Web Services (AWS), and show you how to make your own Amazon Glue crawler.

A fully managed service from Amazon, AWS Glue handles data operations like ETL (extract, transform, load) to get the data prepared and loaded for analytics activities. Glue can crawl S3, DynamoDB, and JDBC data sources.

What is a crawler?

A crawler is an automated software app that "crawls" through the content of web pages, following the links on those pages to find and crawl additional pages. Sometimes also called spiders or bots, these software programs crawl with a purpose. They may crawl through web pages to scan and index content for search engines. They may also extract large sets of data for content aggregation, identifying trends, conducting sentiment analysis, or monitoring feature sets and prices. Internet archiving sites use crawlers to make an historical record of web pages for future reference. Crawlers are also useful for link checking and SEO audits.

What is the AWS Glue crawler?

The AWS Glue crawler is a tool from Amazon that automates how you discover, catalog, and organize data scraped from various sources. For example, AWS Glue crawler can be used to crawl the Amazon Simple Storage Service (S3) relational database or other sources of data, such as

DynamoDB, JDBC, and MongoDB.

The Amazon Glue crawler scans and catalogs data to detect changes in the structure of data or schema. It can create an updated data catalog and conduct extract, transform, and load (ETL) operations for fast processing and analytics. It also integrates with AWS analytics services. The AWS Glue crawler populates Amazon S3 buckets—scalable, secure, and durable storage containers for large volumes of files and data—along with accompanying metadata.

Learn more about the innovation of AWS cloud databases.

Understanding the AWS Glue architecture and workflow

The AWS Glue architecture and workflow starts with connecting to and crawling a data store, which could be a bucket like Amazon S3, or a database like Amazon RDS, Amazon Redshift, Amazon DynamoDB, or JDBC.

Next, AWS Glue determines the structure and format of the data using classifiers, which are sets of rules that identify Glue data types, schemas, file formats, and a variety of types of metadata.

Lastly, AWS Glue writes the metadata into a centralized repository called a data catalog. This contains information about the source of the data, the data schema, and other data descriptions.

How to create a crawler in AWS Glue

Now that we understand the key components of the Amazon Glue crawler—from the architecture, data stores, and data catalog, to the interaction between each component—let's discuss how to create a crawler in AWS Glue.

1. Download sample JSON data

We need some sample data. Because we want to show how to join data in Glue, we need to have two data sets that have a common element.

In our AWS Glue crawler example, we're using data from IMDB. We have selected a small subset (24 records) of that data and put it into JSON format. (Specifically, they have been formatted to load into DynamoDB, which we will do later.)

One file has the description of a movie or TV series. The other has ratings on that series or movie. Since the data is in two files, it is necessary to join that data in order to get ratings by title. Glue can do that.

Download these two JSON data files:

- Download title data here.
- Download ratings data here.

2. Upload the data to Amazon S3

Create these buckets in S3 using the Amazon AWS command line client. (Don't forget to run **aws configure** to store your private key and secret on your computer so you can access Amazon AWS.)

Below we create the buckets **titles** and **rating** inside **movieswalker**. The reason for this is Glue will

create a separate table schema if we put that data in separate buckets.

(Your top-level bucket name must be unique across all of Amazon. That's an Amazon requirement, since you refer to the bucket by URL. No two customers can have the same URL.)

```
aws s3 mb s3://movieswalker

aws s3 mb s3://movieswalker/titles

aws s3 mb s3://movieswalker/ratings
```

Then copy the title **basics** and **ratings** file to their respective buckets.

```
aws s3 cp 100.basics.json s3://movieswalker/titles

aws s3 cp 100.ratings.tsv.json s3://movieswalker/ratings
```

3. Configure the crawler in Glue

Log into the Glue console for your AWS region. (Mine is European West.)

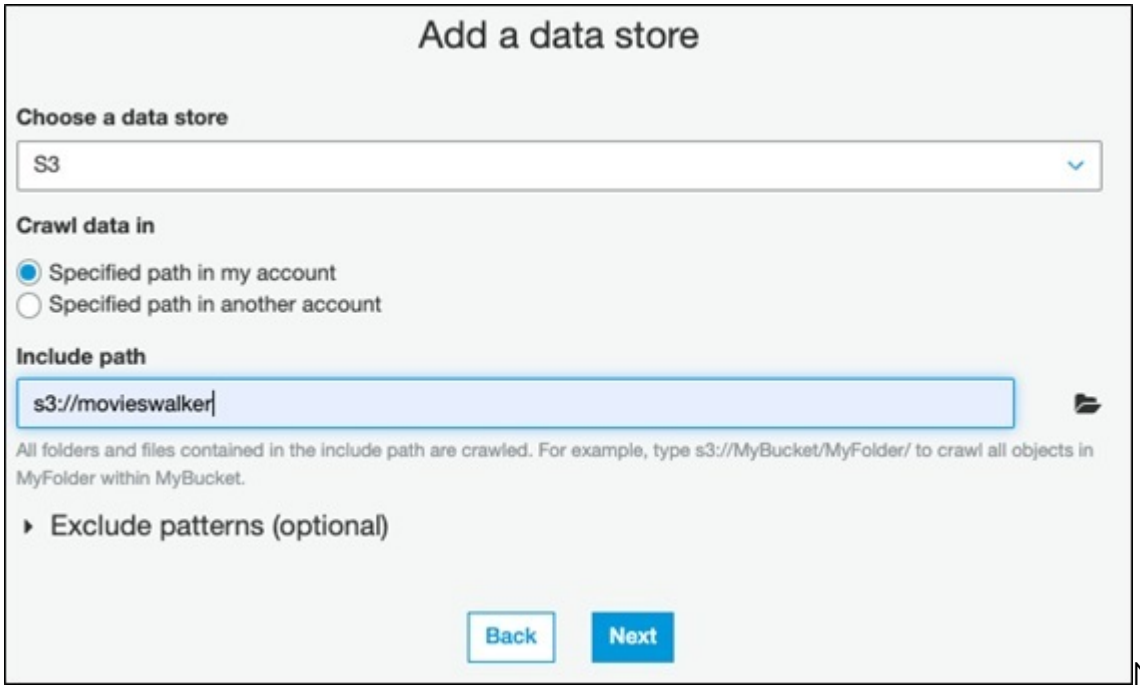
Then go to the crawler screen and add a crawler:



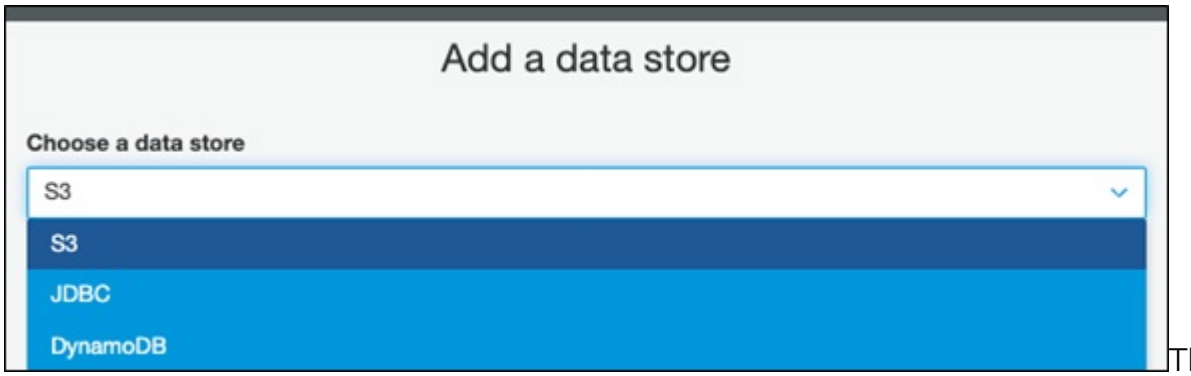
Next, pick a data store. A better name would be **data source**, since we are pulling data from there and storing it in Glue.



Then pick the top-level movieswalker folder we created above.



Notice that the data store can be S3, DynamoDB, or JDBC.



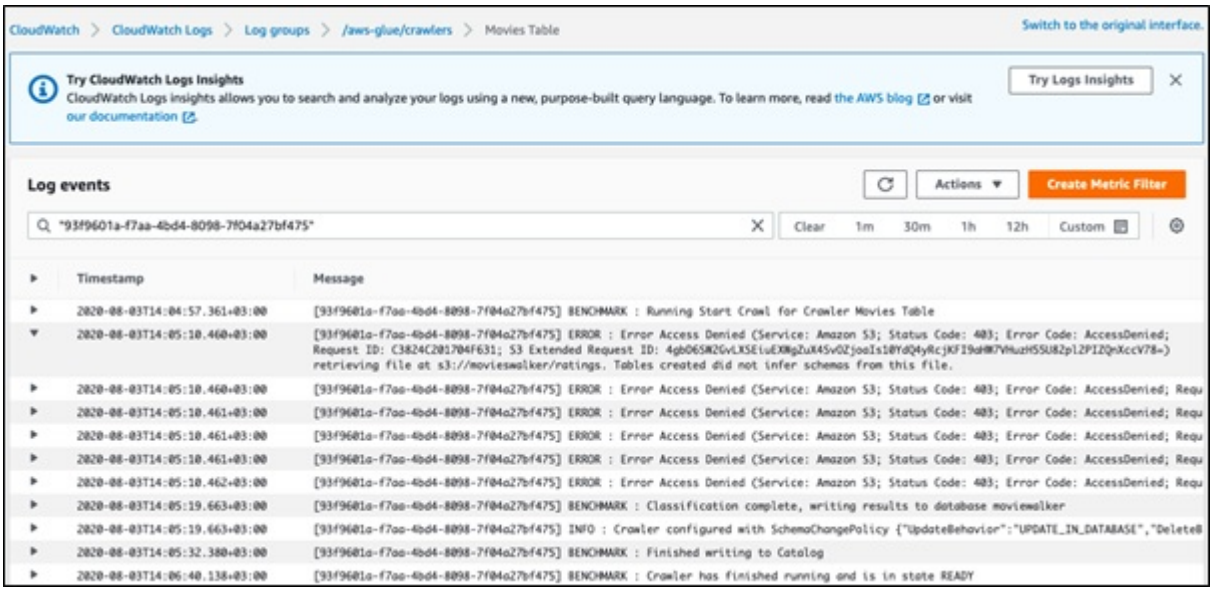
Then start the crawler. When it's done you can look at the logs.

If you get this error it's an S3 policy error. You can make the tables public just for purposes of this tutorial if you don't want to dig into IAM policies. In this case, I got this error because I uploaded the files as the Amazon root user while I tried to access it using a user created with IAM.

```
ERROR : Error Access Denied (Service: Amazon S3; Status Code: 403; Error
Code: AccessDenied; Request ID: 16BA170244C85551; S3 Extended Request ID:
```

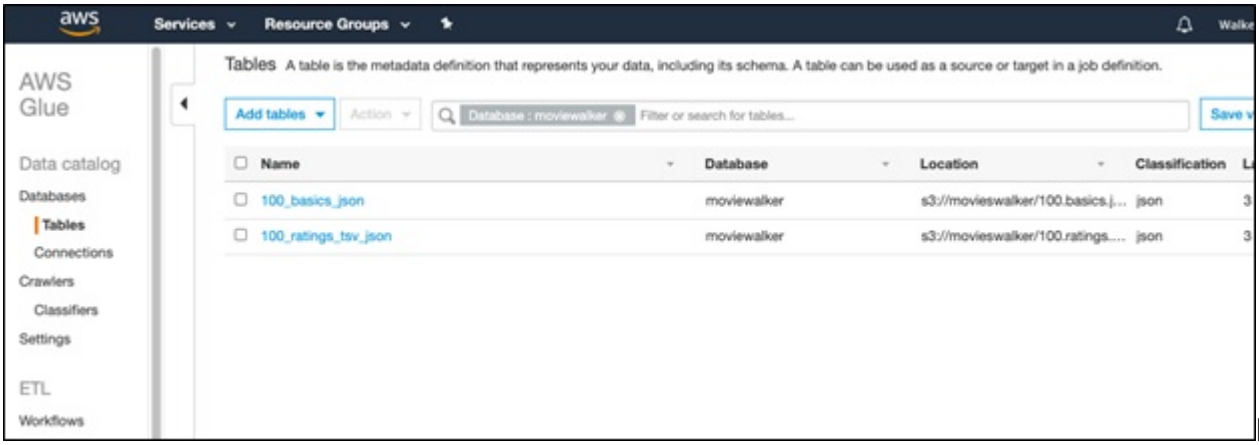
```
y/JBUpMqsdtf/vnugyFZp8k/DK2cr2hldoXP2JY19NkD39xiTEFp/R8M+UkdO5X1SjrYXuJOnXA=)
retrieving file at s3://movieswalker/100.basics.json. Tables created did not
infer schemas from this file.
```

View the crawler log. Here you can see each step of the process.

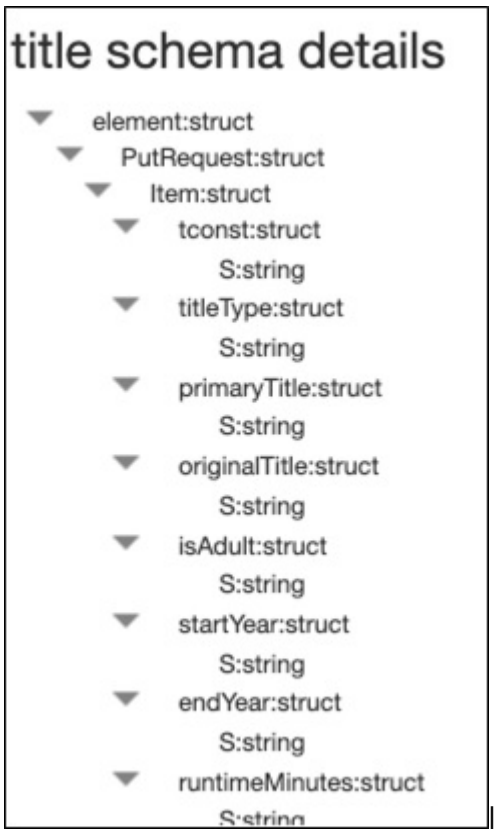


4. View tables created in AWS Glue

Here are the tables created in Glue.



If you click on them you can see the schema.



It has these properties. The item of interest to note here is it stored the data in Hive format, meaning it must be using Hadoop to store that.

AWS Glue crawler tutorial: Key steps summarized

1. Upload data to Amazon S3.
2. Create and configure a crawler with the right access permissions, identifying the data source and path.
3. Configure the classifiers that your crawler will use to interpret the data structure.
4. Specify an output location in the data catalog. You are now ready to trigger and run the crawler.

Using AWS Glue is a great way to automate data discovery, querying, and processing. To learn more about other AWS management tools, visit our blog at AWS Management Tools.