



## Perform ETL in Glue

ETL jobs in AWS Glue refer to **Extract, Transform, and Load (ETL)** processes that help automate the movement and transformation of data between different data sources and destinations. AWS Glue is a fully managed serverless service that simplifies the creation and execution of ETL jobs.

Here's a breakdown of the ETL process in AWS Glue:

1. **Extract:** Data is extracted from various sources, which could include databases, data lakes (like Amazon S3), or streaming data sources. AWS Glue supports multiple data formats like JSON, CSV, Parquet, and more.
2. **Transform:** The extracted data is processed and transformed to fit the target system's requirements. Common transformations include cleaning the data (e.g., handling missing values), reformatting data, enriching it, or applying business logic.
3. **Load:** The transformed data is then loaded into a target destination, such as a data warehouse (e.g., Amazon Redshift), a different S3 bucket, or a database (e.g., Amazon RDS).

### Features of AWS Glue ETL Jobs:

- **Automatic Schema Discovery:** Glue can automatically detect the schema of your data.
- **Serverless:** No need to manage infrastructure; AWS handles it.
- **Job Scheduler:** You can schedule ETL jobs to run at specific intervals.
- **Glue Data Catalog:** Stores metadata for your datasets, making them easy to manage and query.

AWS Glue ETL jobs are ideal for automating data pipelines in analytics, data warehousing, and data integration tasks.

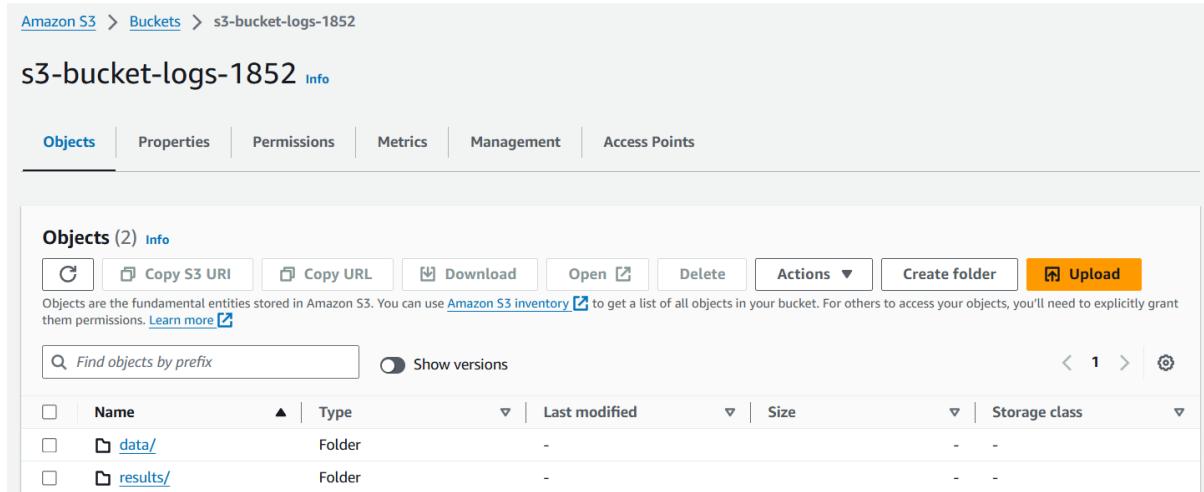
**In this lab, we perform an ETL job using AWS Glue to process data from a CSV file stored in an S3 bucket. The steps involve creating a database and table using a crawler in AWS Glue to extract data from the CSV file. The crawler scans the file, creates a table, and stores the metadata in the database.**

**Next, we use the AWS Glue Script Editor to write and run an ETL job. The script reads data from the created table, performs transformations (such as grouping by province and city, summing confirmed cases, and identifying the maximum confirmed cases), and then writes the processed data back to another S3 bucket in CSV format.**

**The end goal is to automate the data extraction, transformation, and loading process, resulting in a cleaned, aggregated dataset that can be easily accessed for further analysis or reporting.**

## To begin with the Lab:

1. In this lab we are going to see how we can perform ETL jobs in AWS Glue.
2. Here we also have a CSV file named Case which we are going to upload in S3 Bucket.
3. In your S3 bucket you need to create 2 folders with the same name and in the data folder, you need to upload the CSV file.



Amazon S3 > Buckets > s3-bucket-logs-1852

s3-bucket-logs-1852 [Info](#)

Objects Properties Permissions Metrics Management Access Points

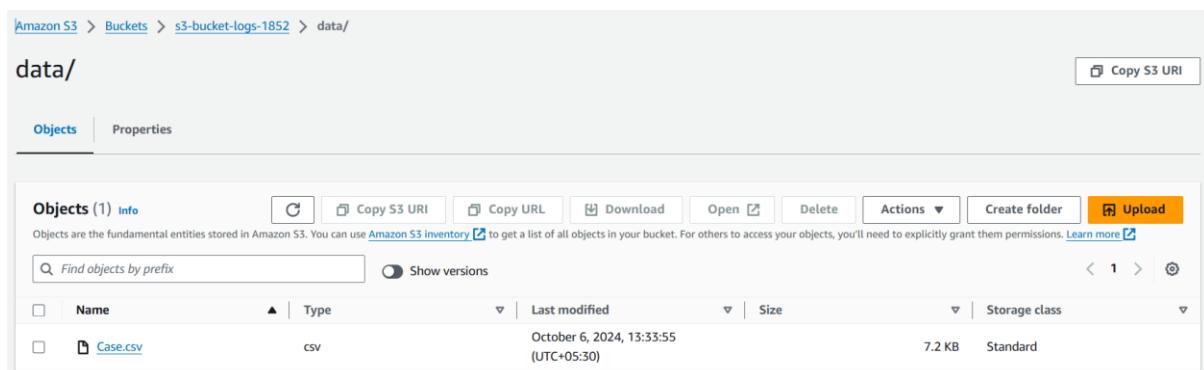
**Objects (2) [Info](#)**

[C](#) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#) [Upload](#)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix  Show versions

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	<a href="#">data/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">results/</a>	Folder	-	-	-



Amazon S3 > Buckets > s3-bucket-logs-1852 > data/

**data/**

[Copy S3 URI](#)

Objects Properties

**Objects (1) [Info](#)**

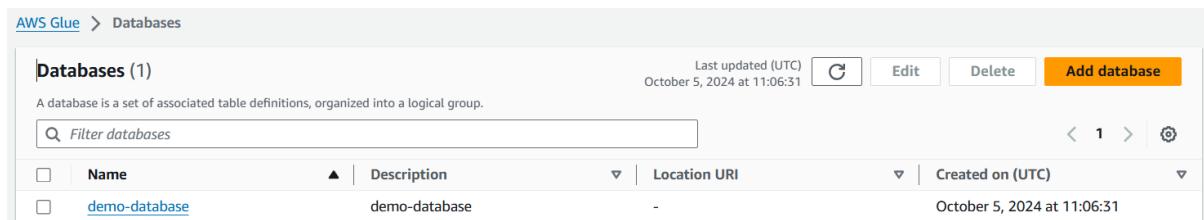
[C](#) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#) [Upload](#)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix  Show versions

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	<a href="#">Case.csv</a>	csv	October 6, 2024, 13:33:55 (UTC+05:30)	7.2 KB	Standard

4. Now come to AWS Glue, here you need to go to Databases and create one. For that, you just need to give a name to your Database and create it.



AWS Glue > Databases

**Databases (1)**

Last updated (UTC)  
October 5, 2024 at 11:06:31

[C](#) [Edit](#) [Delete](#) [Add database](#)

Filter databases

<input type="checkbox"/>	Name	Description	Location URI	Created on (UTC)
<input type="checkbox"/>	<a href="#">demo-database</a>	demo-database	-	October 5, 2024 at 11:06:31

5. Then go to Tables and choose Add tables using crawlers.

AWS Glue > Tables

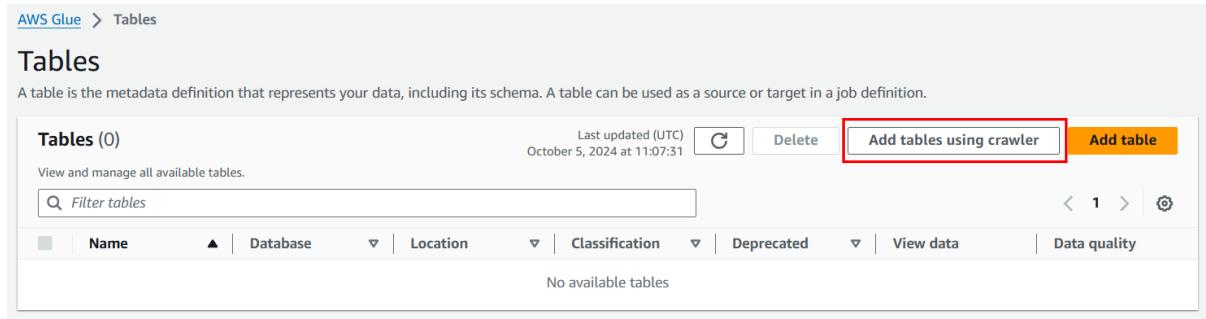
## Tables

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.

Tables (0) Last updated (UTC) October 5, 2024 at 11:07:31 Delete Add tables using crawler Add table

View and manage all available tables. Filter tables □ ▲ Name Database ▼ Location ▼ Classification ▼ Deprecated ▼ View data | Data quality

No available tables



6. On step one, give a name and description to your crawler.

AWS Glue > Crawlers > Add crawler

### Set crawler properties

Step 1 Set crawler properties

Step 2 Choose data sources and classifiers

Step 3 Configure security settings

Step 4 Set output and scheduling

Step 5 Review and create

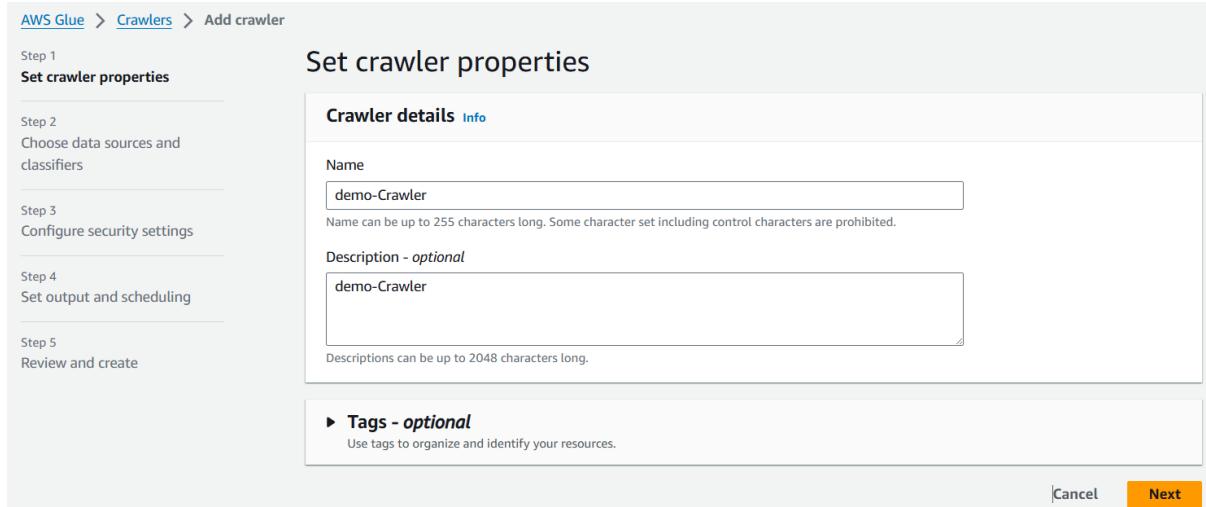
**Crawler details** Info

Name: demo-Crawler  
Name can be up to 255 characters long. Some character set including control characters are prohibited.

Description - optional: demo-Crawler  
Descriptions can be up to 2048 characters long.

Tags - optional  
Use tags to organize and identify your resources.

Cancel Next



7. Then in step 2 you need to add a data source which is your S3 bucket. Choose the same options as shown in the snapshot.

AWS Glue > Crawlers > Add crawler

### Choose data sources and classifiers

Step 1 Set crawler properties

Step 2 Choose data sources and classifiers

Step 3 Configure security settings

Step 4 Set output and scheduling

Step 5 Review and create

**Data source configuration**

Is your data already mapped to Glue tables?

Not yet Select one or more data sources to be crawled.

Yes Select existing tables from your Glue Data Catalog.

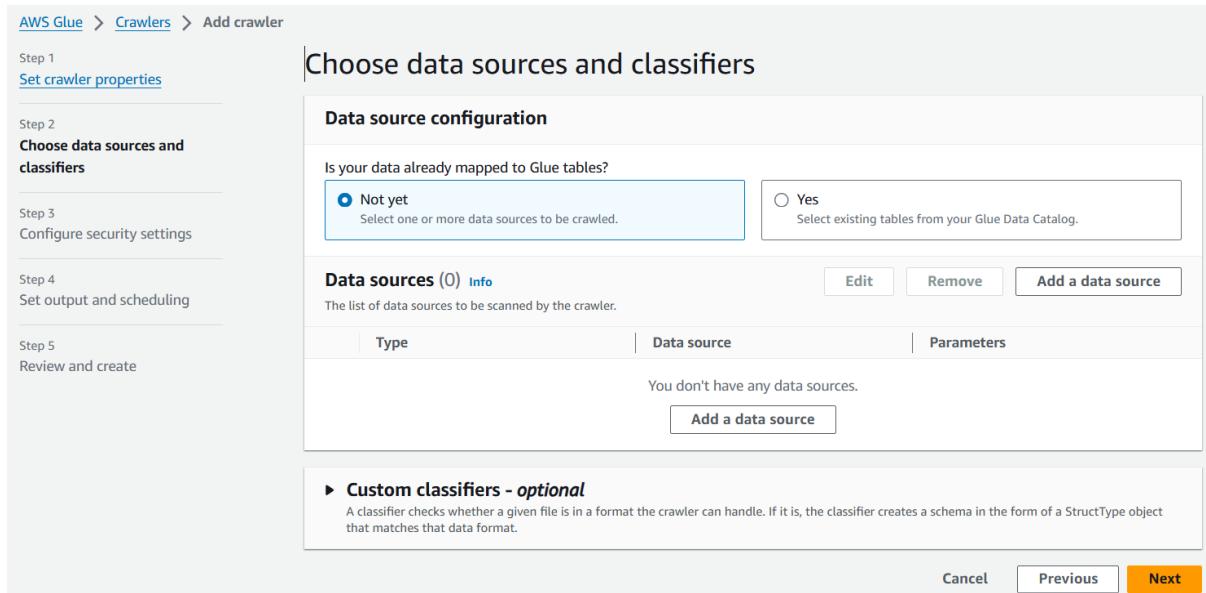
**Data sources (0)** Info Edit Remove Add a data source

The list of data sources to be scanned by the crawler.

Type	Data source	Parameters
You don't have any data sources. <a href="#">Add a data source</a>		

Custom classifiers - optional  
A classifier checks whether a given file is in a format the crawler can handle. If it is, the classifier creates a schema in the form of a StructType object that matches that data format.

Cancel Previous Next



## Add data source

X

### Data source

Choose the source of data to be crawled.

S3



### Network connection - *optional*

Optionally include a Network connection to use with this S3 target. Note that each crawler is limited to one Network connection so any other S3 targets will also use the same connection (or none, if left blank).

▼ C

[Clear selection](#)

[Add new connection](#)

### Location of S3 data

- In this account
- In a different account

### S3 path

Browse for or enter an existing S3 path.

s3://s3-bucket-logs-1852



[View](#)

[Browse S3](#)

All folders and files contained in the S3 path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

### Subsequent crawler runs

This field is a global field that affects all S3 data sources.

- Crawl all sub-folders

Crawl all folders again with every subsequent crawl.

- Crawl new sub-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 based on events

Rely on Amazon S3 events to control what folders to crawl.

[Cancel](#)

[Add an S3 data source](#)

8. Now on step 3 you need to choose an IAM role if you have, or you can click on Create new IAM role and create a new one. For the IAM role permission for simplicity you can add Administrator Access.

AWS Glue > Crawlers > Add crawler

Step 1 Set crawler properties

Step 2 Choose data sources and classifiers

**Step 3 Configure security settings**

Step 4 Set output and scheduling

Step 5 Review and create

## Configure security settings

**IAM role** [Info](#)

Existing IAM role  
gluelab [View](#)

[Create new IAM role](#) [Update chosen IAM role](#)

Only IAM roles created by the AWS Glue console and have the prefix "AWSGlueServiceRole-" can be updated.

**Lake Formation configuration - optional**

Allow the crawler to use Lake Formation credentials for crawling the data source. [Learn more.](#)

Use Lake Formation credentials for crawling S3 data source  
Checking this box will allow the crawler to use Lake Formation credentials for crawling the data source. If the data source is registered in another account, you must provide the registered account ID. Otherwise, the crawler will crawl only those data sources associated to the account. Only applicable to S3, Glue Catalog, Iceberg, and Hudi data sources.

**Security configuration - optional**

Enable at-rest encryption with a security configuration.

[Cancel](#) [Previous](#) [Next](#)

9. In step 4, choose your target database and for the crawler schedule choose on demand. Then just move ahead and create your crawler.

Step 1 Set crawler properties

Step 2 Choose data sources and classifiers

**Step 3 Configure security settings**

Step 4 Set output and scheduling

Step 5 Review and create

## Set output and scheduling

**Output configuration** [Info](#)

Target database  
demo-database [View](#)

[Clear selection](#) [Add database](#)

Table name prefix - *optional*

Maximum table threshold - *optional*  
This field sets the maximum number of tables the crawler is allowed to generate. In the event that this number is surpassed, the crawl will fail with an error. If not set, the crawler will automatically generate the number of tables depending on the data schema.

**Advanced options**

**Crawler schedule**  
You can define a time-based schedule for your crawlers and jobs in AWS Glue. The definition of these schedules uses the Unix-like cron syntax. [Learn more.](#)

Frequency  
On demand

10. Below you can see that your crawler is ready, so click on run crawler and wait for it to get completed.

AWS Glue > Crawlers > demo-Crawler

## demo-Crawler

Last updated (UTC)  
October 5, 2024 at 11:11:47

[Run crawler](#) [Edit](#) [Delete](#)

Crawler properties			
Name demo-Crawler	IAM role <a href="#">gluelab</a>	Database demo-database	State READY
Description demo-Crawler	Security configuration -	Lake Formation configuration -	Table prefix -
Maximum table threshold -			
<a href="#">► Advanced settings</a>			

11. You can see that our crawler run has been completed. Now if you to tables.

Crawler runs | Schedule | Data sources | Classifiers | Tags

### Crawler runs (1)

The list of crawler runs for this crawler.

Start time (UTC)	End time (UTC)	Current/last duration	Status	DPU hours	Table changes
October 5, 2024 at 11:13:18	October 5, 2024 at 11:14:35	01 min 17 s	<a href="#">Completed</a>	-	-

[Filter data](#) [Filter by a date and time range](#)

12. You will see a table has been created.

AWS Glue > 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.

Tables (1)

Last updated (UTC)  
October 5, 2024 at 11:19:12

[Delete](#) [Add tables using crawler](#) [Add table](#)

Name	Database	Location	Classification	Deprecated	View data	Data quality
s3_bucket_logs_1852	demo-database	s3://s3-bucket-logs-1	CSV	-	<a href="#">Table data</a>	<a href="#">View data quality</a>

13. Also, if you go inside your database, you can see your table there too.

AWS Glue > Databases > demo-database

## demo-database

Last updated (UTC)  
October 6, 2024 at 08:09:29

[Edit](#) [Delete](#)

Database properties			
Name demo-database	Description demo-database	Location -	Created on (UTC) October 6, 2024 at 08:06:06

Tables (1)

Last updated (UTC)  
October 6, 2024 at 08:09:31

[Delete](#) [Add tables using crawler](#) [Add table](#)

Name	Database	Location	Classification	Deprecated	View data	Data quality
s3_bucket_logs_1852	demo-database	s3://s3-bucket-logs-1	CSV	-	<a href="#">Table data</a>	<a href="#">View data quality</a>

14. Now to create an ETL job, from the left pane choose ETL jobs. Then click on Script Editor.

AWS Glue Studio Jobs

Create job [Info](#)

Author in a visual interface focused on data flow. **Visual ETL**

Author using an interactive code notebook. **Notebook**

Author code with a script editor. **Script editor**

Example jobs [Info](#) Create example job

Your jobs (0) [Info](#)

No jobs

You have not created a job yet.

Create job from a blank graph

15. Then choose Spark as your Engine and choose Upload script to upload your script which is given to you with the lab and click on create script.

Script

Engine

Spark

Options

Start fresh

Upload script

**Choose file**

Limited to Python (\*.py, \*.py3) files only.

script.py  
1.53 KiB  
October 04, 2024

Cancel **Create script**

16. Once your script has been uploaded, in lines 16, 17, and 19 you need to mention the name of your glue database, glue table, and the s3 bucket name as you can see below.

## Script Info

```
9
10 spark_context = SparkSession.builder.getOrCreate()
11 #spark_context = SparkContext.getOrCreate()
12
13 glue_context = GlueContext(spark_context)
14 session = glue_context.spark_session
15
16 glue_db = "demo-database"
17 glue_tbl = "s3_bucket_logs_1852"
18 # s3://aws-ml-13123123/Case.csv
19 s3_write_path = "s3://s3-bucket-logs-1852"
20
21 dt_start = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
22 print("Start time:", dt_start)
```

17. Then go to job details and give a name to your job, choose the same IAM role that you have chosen for the crawlers.

Script Job details Runs Data quality Schedules Version Control

### Basic properties Info

Name  
etl-job

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.  
gluelab

Type  
The type of ETL job. This is set automatically based on the types of data sources you have selected.  
Spark

18. After that scroll down and enable Automatically scale the number of workers and in the maximum number of workers write 5.
19. Then just keep everything to default and save your job.

#### Automatically scale the number of workers

- AWS Glue will optimize costs and resource usage by dynamically scaling the number of workers up and down throughout the job run. Requires Glue 3.0 or later.

#### Maximum number of workers

The number of workers you want AWS Glue to allocate to this job.

5

20. Below you can see that your job has been saved successfully but you need to run it manually.



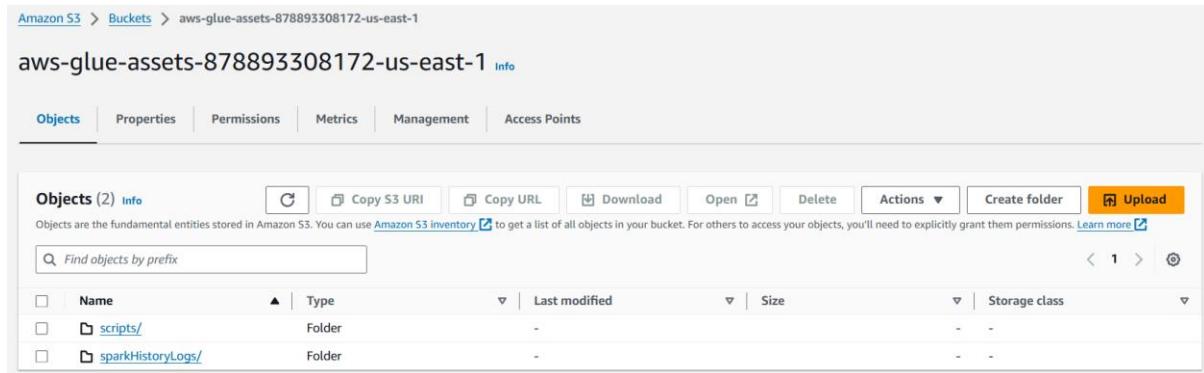
21. And if you go to runs you will see that your job has been running.

A screenshot of the AWS Glue job runs page for the "etl-job" job. The "Runs" tab is selected. A table shows one job run: "Job runs (1/1) Info". The run status is "Running", start time is "10/06/2024 13:49:52", end time is "-", duration is "0 s", capacity is "5 DPU", worker type is "G.1X", and glue version is "4.0". Below the table, a "Run details" card provides more information: Job name "etl-job", Start time "10/06/2024 13:49:52", End time "10/06/2024 13:51:15", Duration "1 m 14 s", Capacity "5 DPU", Worker type "G.1X", and Glue version "4.0". Other details include Log group name "/aws-glue/jobs" and Number of workers "5".

22. Below you can see that the run status has been changed to succeed.

A screenshot of the AWS Glue job runs page for the "etl-job" job. The "Runs" tab is selected. A table shows one job run: "Job runs (1/1) Info". The run status is "Succeeded", start time is "10/06/2024 13:49:52", end time is "10/06/2024 13:51:15", duration is "1 m 14 s", capacity is "5 DPU", worker type is "G.1X", and glue version is "4.0". Below the table, a "Run details" card provides more information: Job name "etl-job", Start time "10/06/2024 13:49:52", End time "10/06/2024 13:51:15", Duration "1 m 14 s", Capacity "5 DPU", Worker type "G.1X", and Glue version "4.0". Other details include Log group name "/aws-glue/jobs" and Number of workers "5". The "Run details" card also includes sections for "Input arguments (11)", "Continuous logs", "Run insights", "Metrics", and "Spark UI".

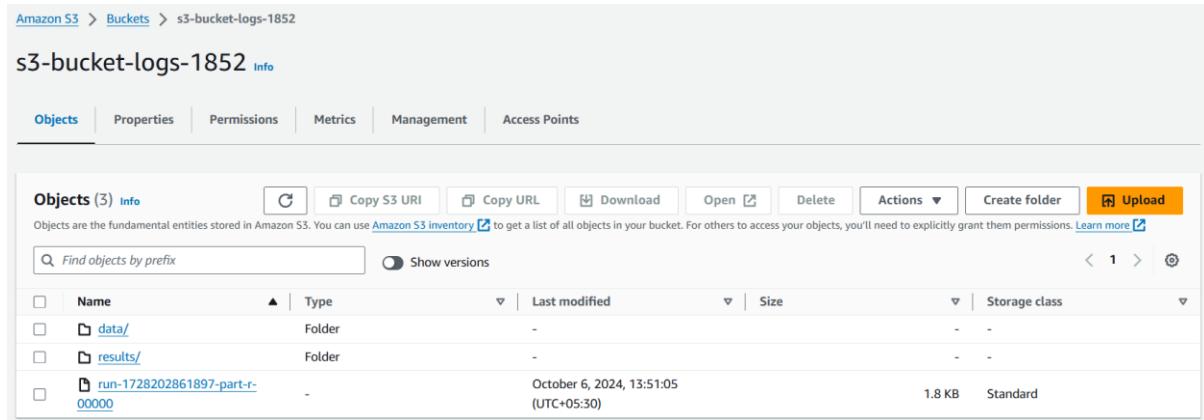
23. Now if you go to S3 you will see that a new bucket has been created where your script was saved and the logs that the ETL job has created. You can go inside these folders and check out the files.



The screenshot shows the AWS S3 console interface. The top navigation bar shows 'Amazon S3 > Buckets > aws-glue-assets-878893308172-us-east-1'. Below the navigation is a header with tabs: Objects (highlighted), Properties, Permissions, Metrics, Management, and Access Points. Under the 'Objects' tab, there is a sub-header 'Objects (2) Info'. A toolbar below the sub-header includes actions like Copy S3 URI, Copy URL, Download, Open, Delete, Actions (with a dropdown arrow), Create folder, and Upload. A note below the toolbar states: 'Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions.' There is a search bar labeled 'Find objects by prefix'. The main table lists two objects:

Name	Type	Last modified	Size	Storage class
scripts/	Folder	-	-	-
sparkHistoryLogs/	Folder	-	-	-

24. Navigate to the bucket that you have mentioned to store the data, here you will see that a new object has been added to your bucket. Now you should download this object.



The screenshot shows the AWS S3 console interface. The top navigation bar shows 'Amazon S3 > Buckets > s3-bucket-logs-1852'. Below the navigation is a header with tabs: Objects (highlighted), Properties, Permissions, Metrics, Management, and Access Points. Under the 'Objects' tab, there is a sub-header 'Objects (3) Info'. A toolbar below the sub-header includes actions like Copy S3 URI, Copy URL, Download, Open, Delete, Actions (with a dropdown arrow), Create folder, and Upload. A note below the toolbar states: 'Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions.' There is a search bar labeled 'Find objects by prefix' and a 'Show versions' button. The main table lists three objects:

Name	Type	Last modified	Size	Storage class
data/	Folder	-	-	-
results/	Folder	-	-	-
run-1728202861897-part-r-00000	File	October 6, 2024, 13:51:05 (UTC+05:30)	1.8 KB	Standard

25. Open this file in Excel, if you cannot open this file in Excel then you can rename the file and add a .csv extension.
26. You can also compare the file to see what has changed. Also, you must read the script because the changes were written in the script.

	A	B	C	D
1	province	city	TotalConfirmed	MaxFromOneConfirmedCase
2	Gyeongsangnam-do	-	39	18
3	Seoul	Seongdong-gu	13	13
4	Busan	Suyeong-gu	5	5
5	Daejeon	Seo-gu	3	3
6	Gyeonggi-do	Uijeongbu-si	50	50
7	Gangwon-do	Wonju-si	3	3
8	Chungcheongbuk-do	from other city	6	6
9	Chungcheongnam-do	-	25	14
10	Jeju-do	from other city	1	1
11	Chungcheongbuk-do	Goesan-gun	11	11
12	Jeollabuk-do	from other city	1	1
13	Gyeongsangbuk-do	-	336	192
14	Jeju-do	-	13	9
15	Seoul	Eunpyeong-gu	14	14
16	Seoul	Jung-gu	7	7
17	Busan	Dongnae-gu	39	39
18	Gyeongsangbuk-do	from other city	615	566
19	Seoul	from other city	8	8
20	Busan	from other city	13	12
21	Daegu	Dong-gu	37	37
22	Gyeonggi-do	Seongnam-si	94	72
23	Gyeonggi-do	Suwon-si	10	10

27. Once you are done then delete all the resources. Start with database, crawlers, tables and ETL jobs.