

Project Title: Serverless Data Pipeline using AWS Lambda and AWS Glue for CSV Transformation

Project Scenario:

A user uploads raw CSV files into an S3 bucket. These files may contain duplicate records or inconsistent formatting. The goal is to automate a transformation pipeline such that:

- A Lambda function is triggered when a new CSV file is uploaded.
 - The Lambda function reads the CSV, removes duplicates, and stores the cleaned CSV in another S3 folder.
 - An AWS Glue job automatically reads the cleaned CSV, performs further transformations (e.g., uppercasing name fields), and stores the final output in a third S3 folder.
-

Solution Architecture:

1. S3 Bucket Structure:

- `lambda_folder/input_folder/` – Raw CSV files uploaded here
- `lambda_folder/output_folder/` – Cleaned CSV output from Lambda
- `glue/final_output/` – Final transformed CSV output from Glue

2. Lambda Configuration:

- **Trigger:** S3 event notification on `lambda_folder/input_folder/`
- **Functionality:**
 - Read CSV file from S3
 - Remove duplicate rows
 - Save cleaned CSV to `lambda_folder/output_folder/`
- **Key Fixes:**
 - Used `.decode('utf-8-sig')` to remove BOM
 - Used `csv.reader` and `csv.writer` with deduplication logic

3. Lambda Code Summary:

- Reads file from S3 input folder
- Deduplicates rows
- Writes clean data as CSV back to S3

4. Glue Configuration:

- **Trigger:** Manual run or triggered via EventBridge
- **Input Path:** lambda_folder/output_folder/
- **Output Path:** glue/final_output/
- **Transformations:**
 - Reads CSV with .option("header", "true")
 - Applies transformation: uppercasing “name” column (if it exists)
 - Writes final CSV with header to output folder

5. Glue Code Summary:

```
spark.read.option("header", "true").option("inferSchema",  
"true").csv(input_path)
```

- Ensures header recognition
 - Applies transformation
 - Uses .write.mode("overwrite").option("header", "true").csv(output_path)
-

Common Issues & Fixes:

Issue	Cause	Fix
AnalysisException / File Not Found	BOM issues, missing files, malformed rows	Cleaned with Lambda; verified file presence
IAM Permission Errors	IAM role lacked s3:GetObject, s3:PutObject	Added correct IAM policies to Lambda and Glue roles
Format Mismatch	Initially tried JSON	Switched to CSV for simplicity

Test Dataset Used:

Book1.csv:

```
id,name,email,age  
1,Alice,alice@example.com,28  
2,Bob,bob@example.com,34  
3,Charlie,charlie@example.com,30  
2,Bob,bob@example.com,34
```

Final Result:

- Fully automated, serverless CSV processing pipeline
- Cleaned duplicate rows using Lambda

- Uppercased name field using Glue
 - Final transformed CSVs available in S3 output folder
-

💡 Next Steps (Optional Enhancements):

- Add timestamp or UUID to file names for uniqueness
 - Trigger Glue via EventBridge automatically
 - Integrate Athena or QuickSight for reporting and querying
-

📁 Workspace Output (Paste Your Results Below)

📝 Paste screenshots, logs, and outputs:

[S3 structure screenshots]

The screenshot shows the Amazon S3 console interface. The top navigation bar includes 'Amazon S3', 'Buckets', and 'my-data-pipeline-bucky'. Below the navigation is a breadcrumb trail: 'my-data-pipeline-bucky'. The main area is titled '-data-pipeline-bucky' with an 'Info' link. A navigation bar at the top of the main area includes 'Objects', 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. Below this is a toolbar with actions like 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', 'Create folder', and 'Upload'. A search bar labeled 'Find objects by prefix' is present. A table lists objects under 'Objects (2)'. The columns are 'Name', 'Type', 'Last modified', 'Size', and 'Storage class'. The objects listed are 'glue/' and 'lambda_folder/'.

[Lambda execution logs]

The screenshot shows the CloudWatch Logs console. The left sidebar has a tree view with 'CloudWatch' selected, followed by 'Favorites and recents', 'Dashboards', 'Logs' (selected), 'Log groups' (selected), 'Log Anomalies', 'Live Tail', 'Logs Insights' (selected), 'Contributor Insights', 'Metrics', 'Application Signals' (selected), 'Network Monitoring', and 'Insights'. The main area is titled 'Log streams (13)' and shows a list of log streams with their last event time. The streams are listed as follows:

Last event time
2025-06-21 09:46:55 (UTC)
2025-06-21 09:19:43 (UTC)
2025-06-21 09:13:01 (UTC)
2025-06-21 09:08:28 (UTC)
2025-06-21 08:57:17 (UTC)
2025-06-21 05:33:13 (UTC)
2025-06-21 05:16:28 (UTC)
2025-06-21 05:09:22 (UTC)
2025-06-21 05:09:09 (UTC)
2025-06-21 04:57:05 (UTC)
2025-06-21 04:38:53 (UTC)
2025-06-21 04:27:51 (UTC)
2025-06-21 04:04:57 (UTC)

[Glue job logs]

Run status	Retries	Start time (Local)	End time (Local)	Duration	Capacity (DPUs)	Worker type	Glue version
Succeeded	0	06/21/2025 15:21:11	06/21/2025 15:22:21	1 m 4 s	10 DPUs	G.1X	4.0
Failed	0	06/21/2025 15:17:26	06/21/2025 15:18:13	40 s	10 DPUs	G.1X	4.0
Failed	0	06/21/2025 14:50:38	06/21/2025 14:51:38	50 s	10 DPUs	G.1X	4.0
Failed	0	06/21/2025 14:43:28	06/21/2025 14:44:46	1 m 4 s	10 DPUs	G.1X	4.0
Failed	0	06/21/2025 14:38:53	06/21/2025 14:40:24	1 m 15 s	10 DPUs	G.1X	4.0
Failed	0	06/21/2025 14:28:18	06/21/2025 14:29:26	1 m	10 DPUs	G.1X	4.0
Failed	0	06/21/2025 14:17:14	06/21/2025 14:18:34	1 m 13 s	10 DPUs	G.1X	4.0
Failed	0	06/21/2025 14:12:21	06/21/2025 14:13:29	1 m	10 DPUs	G.1X	4.0
Failed	0	06/21/2025 14:08:18	06/21/2025 14:09:54	1 m 21 s	10 DPUs	G.1X	4.0
Failed	0	06/21/2025 11:14:38	06/21/2025 11:15:44	59 s	10 DPUs	G.1X	4.0

[Before & After CSV samples]

CSV Before (Raw CSV - uploaded to input_folder/)

```
id,name,email,age  
1,Alice,alice@example.com,28  
2,Bob,bob@example.com,34  
3,Charlie,charlie@example.com,30  
2,Bob,bob@example.com,34
```

CSV After Lambda (Cleaned CSV - stored in output_folder/)

```
id,name,email,age  
1,Alice,alice@example.com,28  
2,Bob,bob@example.com,34  
3,Charlie,charlie@example.com,30
```

Author & Date:

Name: Atmakuru Siva Sandeep

Date: June 21, 2025