# QuickBite ETL Pipeline – Project Summary

**Project Objective**: The purpose of this project was to build a serverless data pipeline for the fictional restaurant chain QuickBite, utilizing AWS Glue and Apache Airflow (MWAA) to automate ETL processes and data aggregation.

**Technology Stack:**

* **AWS Glue** – for ETL job execution and transformation logic
* **Amazon S3** – as both the raw data lake and trusted destination storage
* **Apache Airflow (MWAA)** – for orchestration and DAG scheduling
* **Python & PySpark** – for transformation logic inside the Glue job
* **PostgreSQL / Redshift (optional)** – for downstream analytics
* **IAM Roles & Policies** – for securing access between services

**ETL WORKFLOW**

1. **INPUT SOURCE:**

Raw transaction data (fact\_order.csv) is stored in:

* s3://quickbite-data-suraj/Trusted/transactions/year=2025/month=5/

1. **Glue ETL Job:**

A Python-based AWS Glue job (glue\_job\_total\_sales) reads the source data and:

* Casts order\_amount to double
* Groups the data by region
* Calculates total sales per region

1. **Output Destination:**

The aggregated results are written to:

* s3://quickbite-data-suraj/Trusted/aggregates/sales\_by\_region/

1. **Airflow DAG**

* Created as quickbite\_etl\_pipeline.py
* Runs daily (@daily schedule)
* Orchestrates the Glue job and optionally a Redshift SQL task

1. **Testing & Validation**

* **DAG Triggered in MWAA**
* **Glue Job ran successfully**
* **Output files (.csv) confirmed in S3 output path**
* **Airflow logs and task run history captured**

1. **Troubleshooting**

* Initially encountered AccessDeniedException when triggering the Glue job via Airflow.
* Resolved by attaching a custom IAM policy allowing:
* "Action": "glue: StartJobRun"
* Later DAG runs succeeded and were validated through Airflow UI and S3 outputs.

1. **Deliverables**

* quickbite\_etl\_pipeline.py – Airflow DAG file
* glue\_job\_total\_sales.py – (if using custom script)
* S3 Output Files Screenshot
* Airflow Success Screenshot
* README.md
* Final Folder Structure with all assets

1. **Conclusion**

This project successfully demonstrates an end-to-end cloud-based ETL solution using AWS services. It highlights serverless orchestration, secure role-based access, and automated daily transformations, providing a solid foundation for real-world data pipelines.