Summary Report

Project Title

Data Ingestion from S3 to RDS with Fallback to AWS Glue using Dockerized Python Application

Author

Samarth Nawale

Date

July 2025

Objective

The goal of this project was to build a robust and fault-tolerant data ingestion pipeline that:

- Reads a CSV file from an Amazon S3 bucket

- Uploads the data into an Amazon RDS MySQL-compatible database

- If the RDS write fails (due to access, connectivity, or permission issues), it falls back to AWS Glue by:

  - Creating a table in the Glue Data Catalog

  - Registering the data from S3

All of this is packaged in a Dockerized Python application, making it portable and easy to deploy.

Data Flow

1. Input Source: CSV file (e.g., sample_data.csv) in S3 bucket 'data-pipeline-samarth'.

2. Primary Path:

   - Read with pandas.

   - Insert into RDS table 'students' in DB 'ingestdb'.

3. Fallback Path:

   - On failure, create a Glue table.

   - Register path s3://data-pipeline-samarth/fallback/ in Glue DB 'fallback_db'.

Summary Report

AWS Services Used

- Amazon S3: Source file storage

- Amazon RDS: MySQL-compatible database

- AWS Glue: Metadata fallback for failed inserts

- IAM: Secure access management

- EC2: Host environment for running Docker

Docker Setup

- Python 3.9 image with required libraries

- Dockerfile installs boto3, pandas, sqlalchemy, pymysql

- Runs main.py on startup with environment variables

- Hosted and executed via Ubuntu EC2 instance

Challenges and Solutions

- Docker Desktop didn't work on Windows 11 Home -> Used EC2 Ubuntu instance

- RDS user access denied -> Created new 'ingestuser' with correct grants

- Glue fallback failed due to region error -> Added AWS_DEFAULT_REGION env

- Repeated Glue table error -> Handled with condition checks and overwrite

- SCP errors -> Fixed PEM file path and permissions

Conclusion

Successfully demonstrated integration of multiple AWS services using a Dockerized Python app.

Ensured robust fallback and data handling. Ready for real-world ETL, batch ingestion, and data pipeline use cases.