ETL Pipeline: AWS S3 → AWS RDS (MySQL)

Project Overview

This project demonstrates a realistic scenario **ETL (Extract, Transform, Load)** pipeline using AWS services and Python. The workflow simulates how raw CSV data uploaded by a client to an **S3 bucket** is processed (cleaned and validated) and then stored in an **AWS RDS MySQL** database.

Use Case

A company receives customer registration data via CSV uploads from their client. The task is to:

- Clean invalid phone numbers and missing fields.
- Upload only valid records to RDS MySQL.
- Store skipped rows in a separate file for manual review.

The Objective of the Project

1. Company receives the file in the S3 Bucket using SFTP

2. Using python script upload the file in Database

3. Invalid rows will be available in a separate file and the file will get downloaded in the local system download section.

4. Now to clean the invalid data , validate it and upload it to the database using python script.

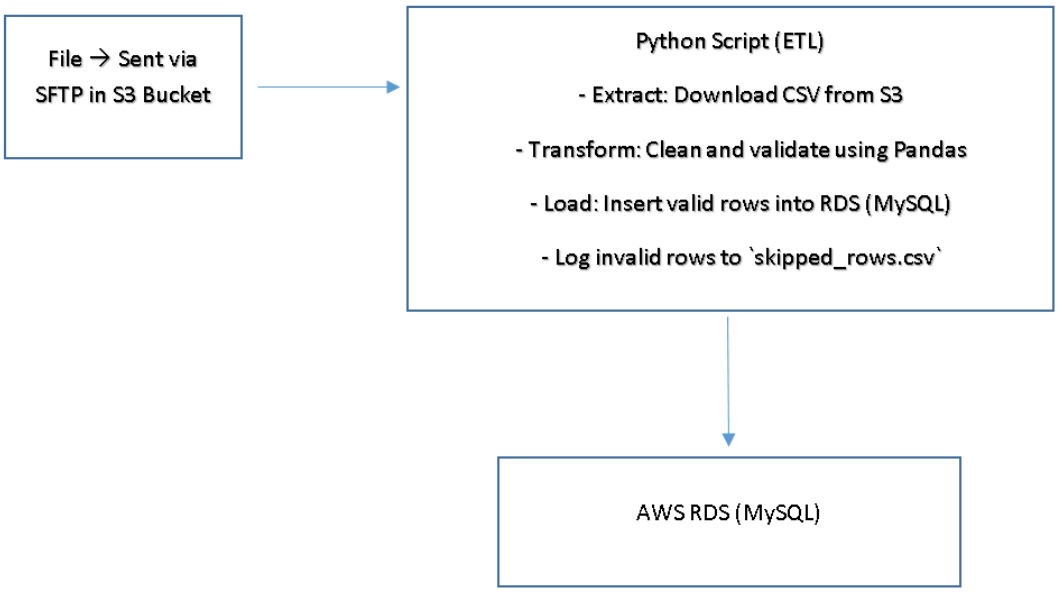
Technologies Used

AWS S3 – For file storage

AWS RDS (MySQL) – For cleaned data storage

Python (pandas, boto3, mysql-connector-python) – For ETL processing

Architecture & Workflow



Features

1. Auto-downloads latest CSV from S3

2. Validates rows (e.g., checks if phone number has 10 digits)

3. Inserts valid data into RDS

4. Logs invalid rows into skipped_rows.csv

Sample Output

- Valid Data: Inserted into users table in RDS
- Invalid Data: Written to skipped_rows.csv for review

Future Improvements -

- Automate using AWS Lambda + S3 Trigger
- Add email notification for skipped rows
- Build a dashboard to track uploads and error stats