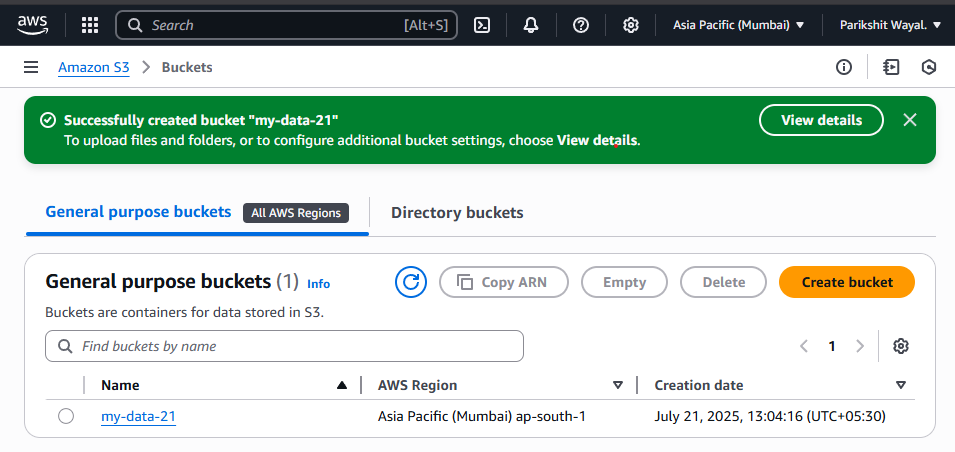
Project: 3

Title: Data Ingestion from S3 to RDS with Fallback to AWS Glue using Dockized Python Application

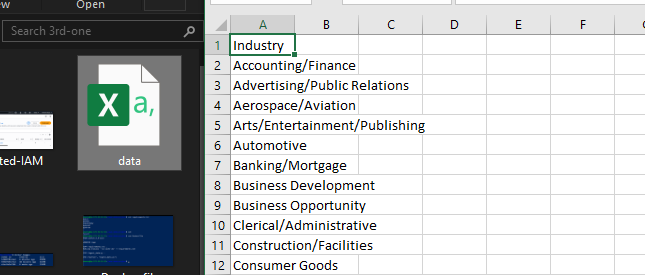
**✅ Step-by-Step Implementation Process**

**1. S3 Bucket Setup**

* I created an **S3 bucket** named my-data-21 in the **ap-south-1** (Mumbai) region using the AWS Console.
* Then I downloaded a simple csv file like example industry one named it **data**.

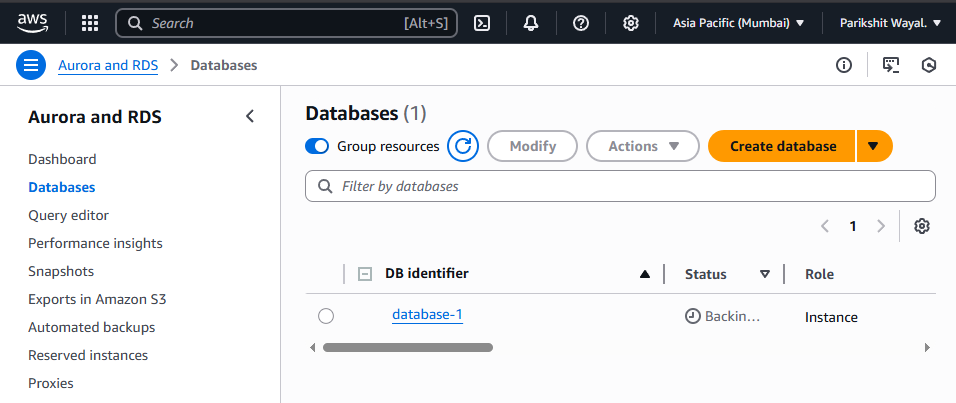


* I uploaded this file into the S3 bucket **data.csv**.



**2. RDS Database Setup**

* I created a **MySQL RDS instance** on AWS (free tier):
  + DB name: database-1
  + Instance class: db.t4g.micro
  + Username: admin
  + Enabled public access.

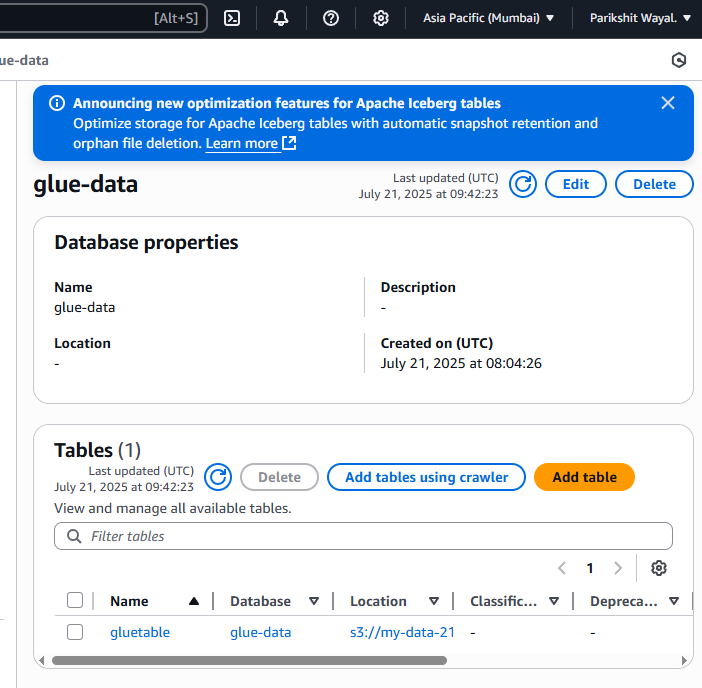


* On my EC2 machine, I installed the **MySQL client** and connected to the RDS endpoint.
* Then I configure connection and :

CREATED DATABASE mydata

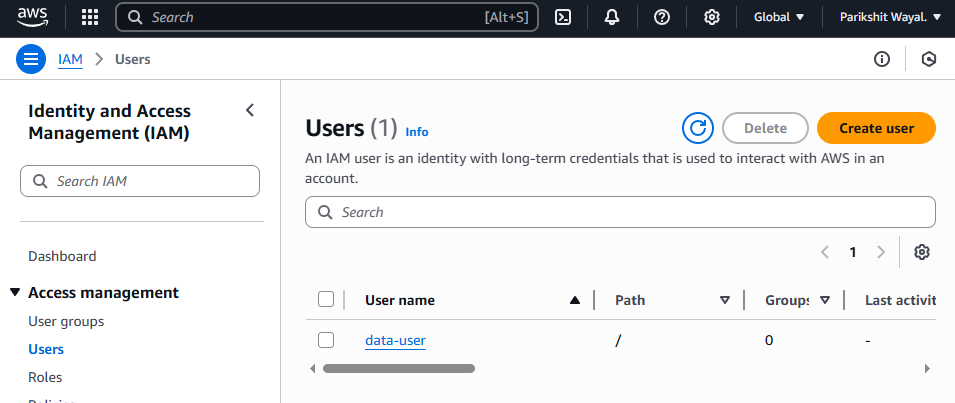
**3. AWS Glue Setup (Fallback Support)**

* In the AWS Glue Console, I created a new **database** named glue-data.
* I also created an **S3 folder** path inside my bucket: s3://my-data-21/glue-data/

  
This folder is used to store fallback data in case RDS fails.

**4. IAM Credentials Setup**

* I created an IAM user named **data-user** with **programmatic access**.



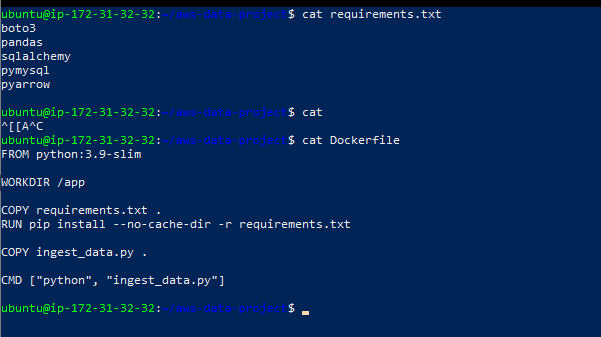
* I attached policies that allow:
  + Full Access to **S3**, **RDS**, and **Glue**.

**5. Code Development (Python + Docker)**

* On my EC2 instance, I made a folder:

**mkdir ~/aws-data-project**

* Inside it, I wrote the Python script: **ingest\_data.py**  
  It:
  + Reads the CSV file from S3 using boto3
  + Parses it using pandas
  + Tries to upload to RDS using SQLAlchemy + PyMySQL
  + If RDS fails → it falls back and:
    - Converts the data to **Parquet**
    - Uploads it to S3
    - Registers a table in AWS **Glue Data Catalog**
* I created **requirements.txt** with:



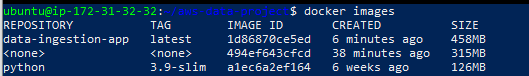
* I also created a **Dockerfile** to:
  + Use python:3.9-slim as the base
  + Install all dependencies
  + Copy and run the script on container startup

**6. Docker Build and Fixes**

* I faced some issues initially:
  + **Docker permission error**: Solved by adding user to Docker group.
  + **Disk space error**: Solved by using docker system prune and increasing EBS volume.
  + **Typo in filename**: Fixed inegst\_data.py → ingest\_data.py.
* Finally, I built the Docker image:

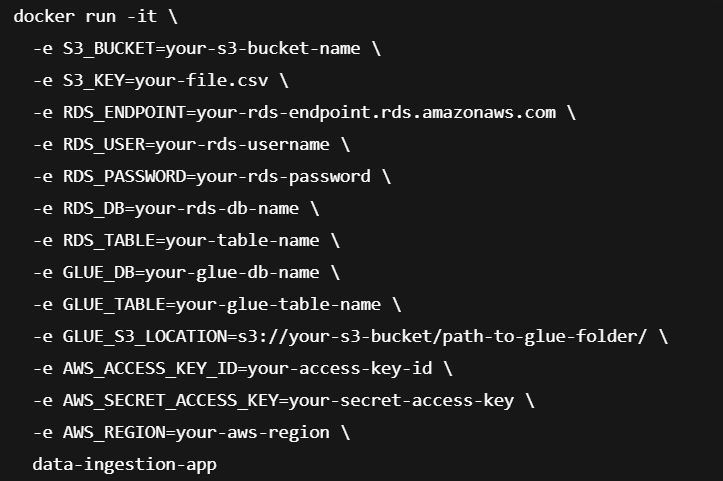
bash

**docker build -t data-ingestion-app .**

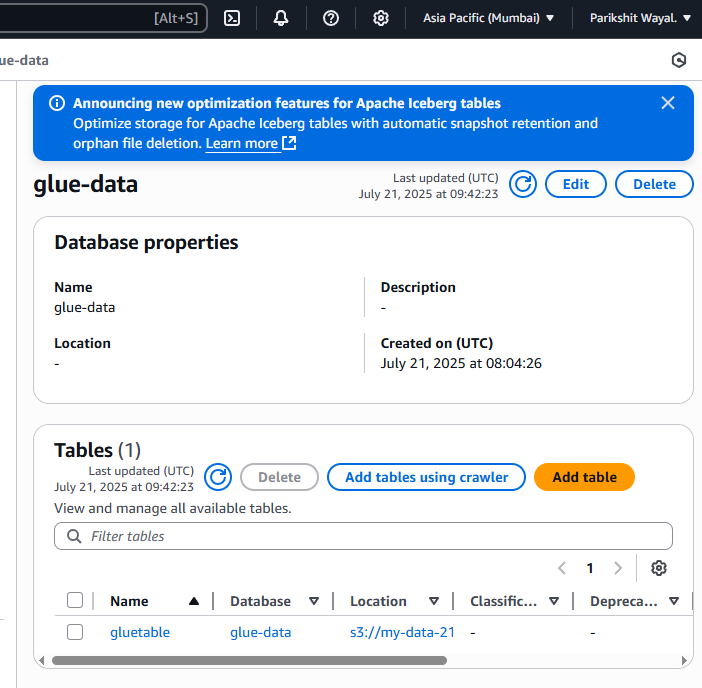
****

**7. Testing and Execution**

* I ran the Docker container using docker run, passing all necessary environment variables:
  + S3 bucket/key
  + RDS endpoint, user, password
  + Glue DB/table name and S3 fallback location
  + AWS keys and region



* **Test Result:**
  + ✅ If the RDS password was correct → **data successfully uploaded to RDS**.
  + ❌ If the password was wrong → **RDS upload failed**, but fallback worked:
    - Data converted to Parquet
    - Saved to S3
    - Table created in AWS Glue

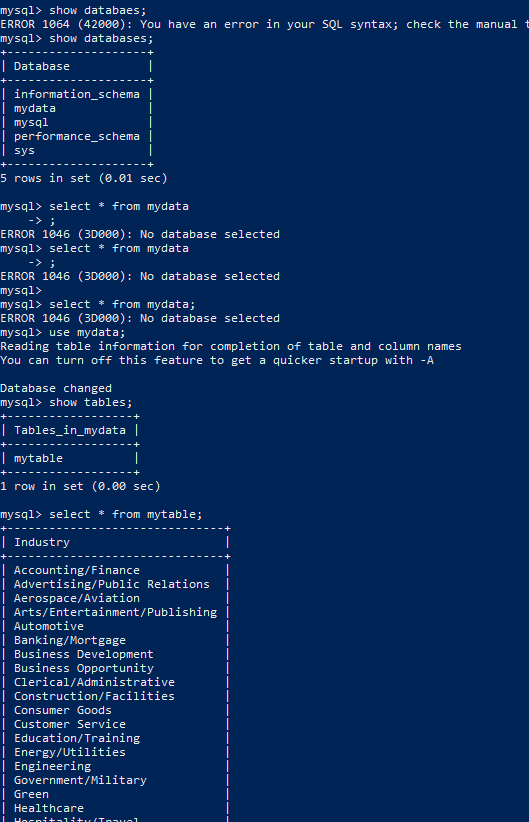


**8. Verification and Cleanup**

* I connected to RDS with MySQL client and verified:

**USE mydata;**

**SELECT \* FROM mytable;**

****

* I checked AWS Glue Console → confirmed the table with correct schema.
* For cleanup:
  + I stopped RDS to avoid cost.
  + Deleted files in S3 and the Glue table.
  + Pruned Docker images (docker system prune -a).