<u>Use Case:</u> Data Lake and Analytics Pipeline for a paint manufacturing FMCG company

<u>Scenario:</u>

In the FMCG and paint manufacturing sector, **demand forecasting plays a critical role** in raw material procurement, production planning, and distribution. However, demand signals such as **sales orders, distributor reports, and SAP ERP data** are often stored in **siloed systems**, making it difficult for stakeholders to gain **real-time visibility into demand vs. supply**.

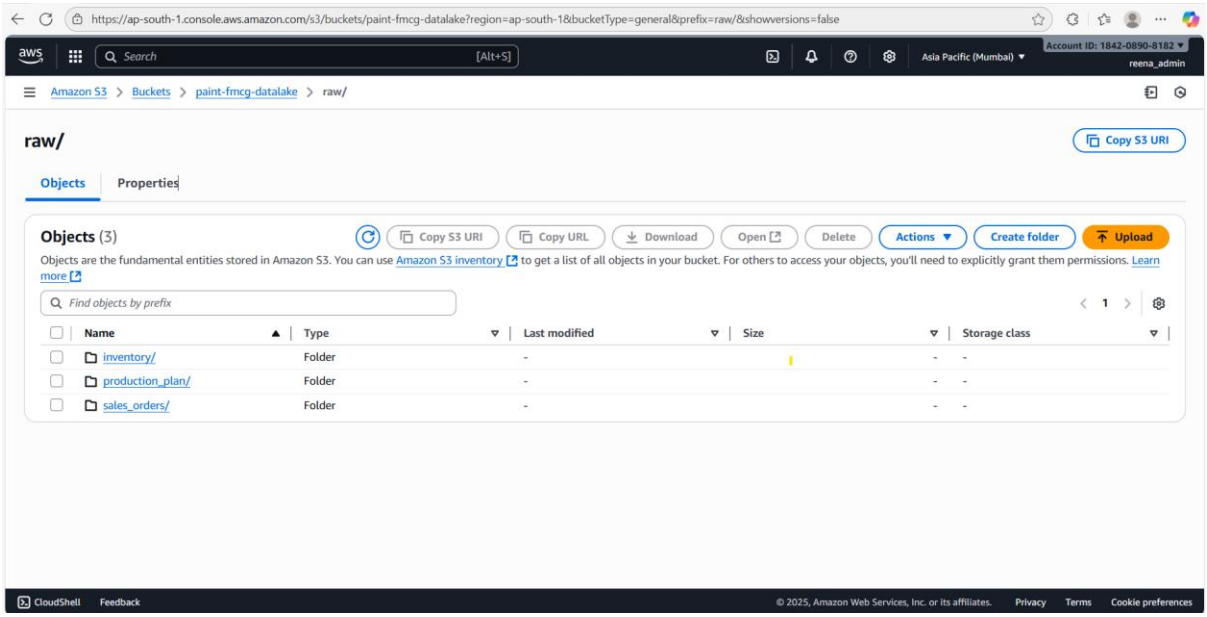
This lack of integration leads to challenges such as:

- Delayed identification of **inventory shortages**
- Inefficient **production planning**
- Missed opportunities to align **distribution with actual demand**

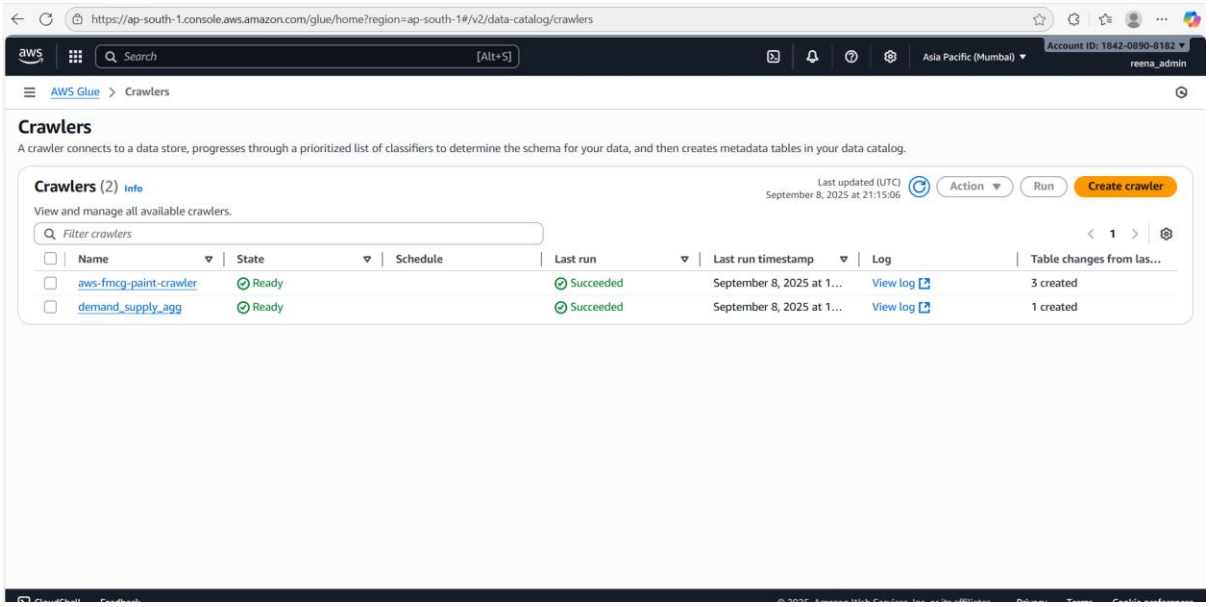
To address this, a **centralized data platform** is needed where sales, inventory, and production data are integrated, enabling **real-time analysis and reporting**. This empowers business teams to monitor **sales performance, inventory health, and potential shortages** proactively, thereby improving **service levels and reducing stockouts**.

<u>Solution:</u>

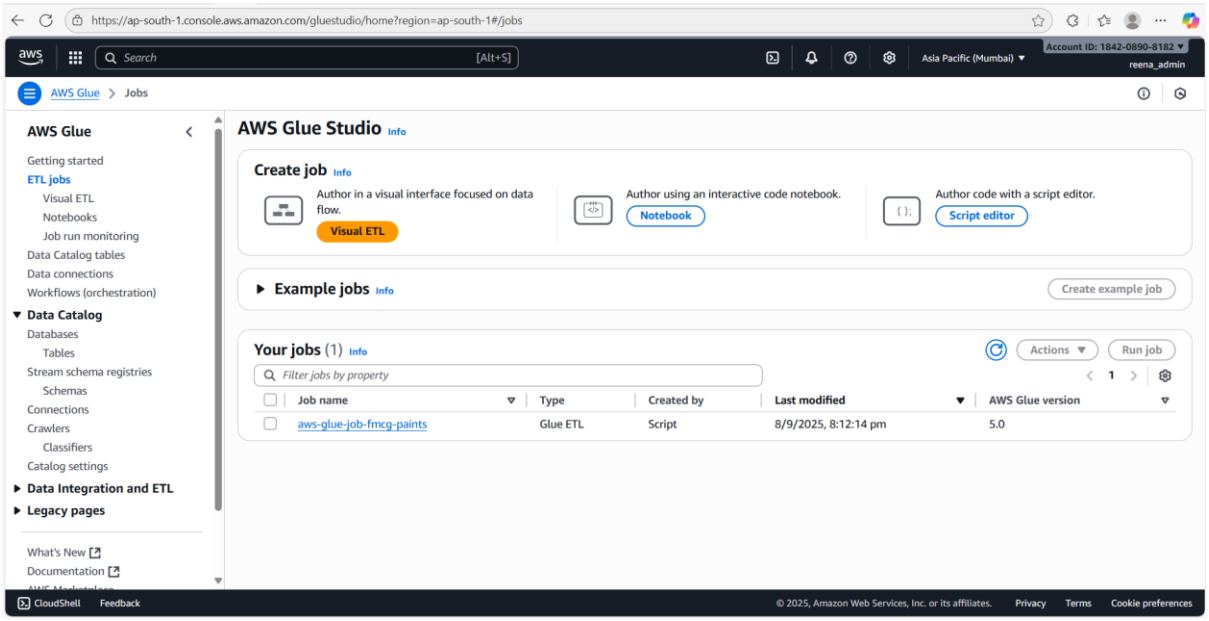
Step 1: Raw data files uploaded to S3

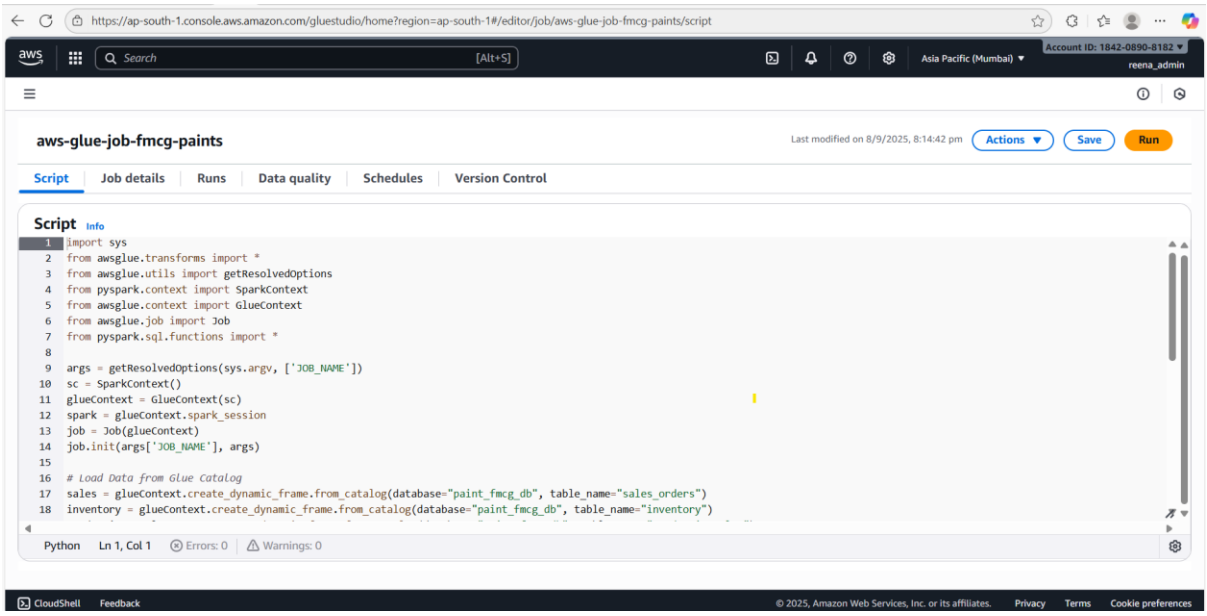


Step 2: Data cataloged using AWS Glue Crawlers

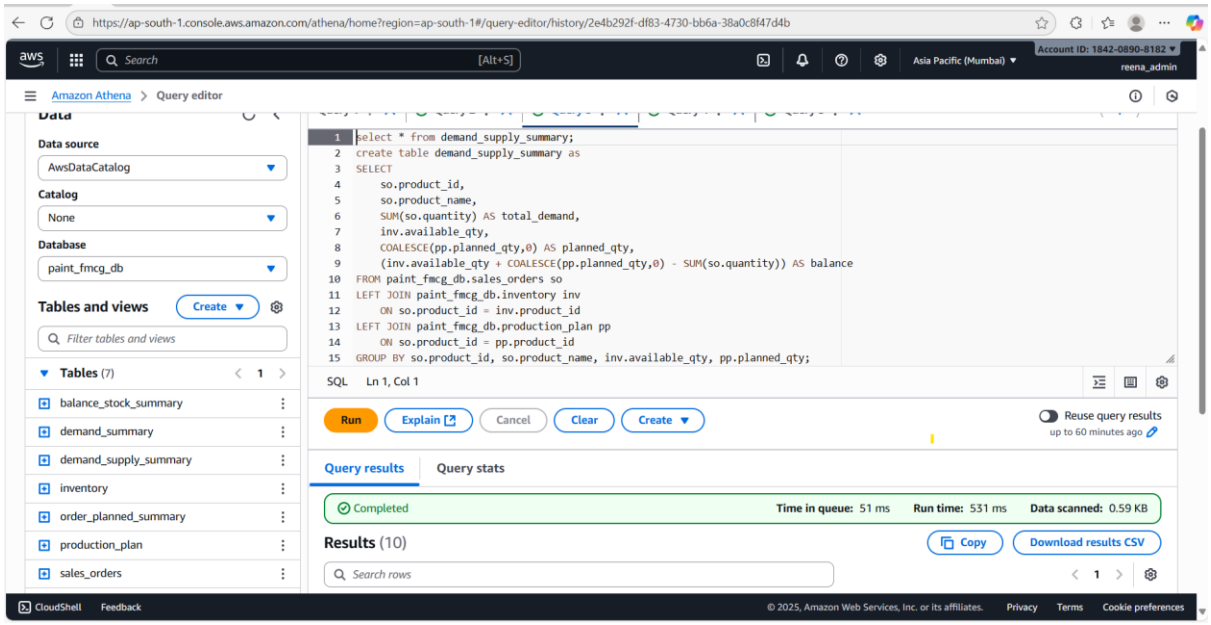


Step 3: ETL pipeline with AWS Glue (PySpark script)





Step 4: Dataset creation in Athena for visualization



Demand Vs Supply:

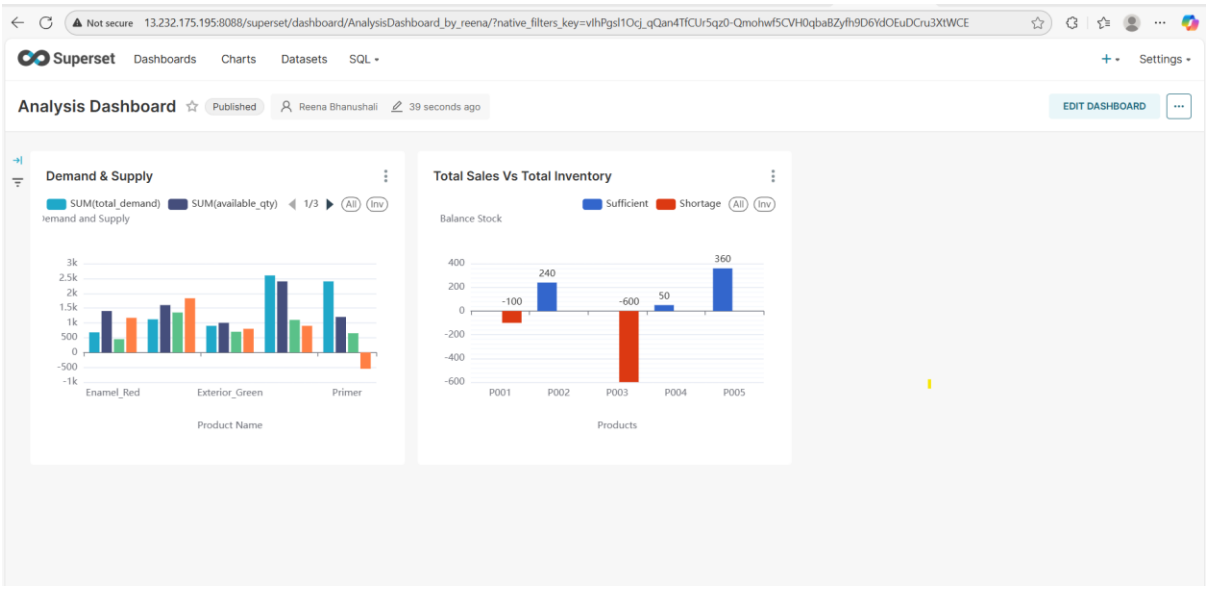
```
SELECT
    so.product_id,
    so.product_name,
    SUM(so.quantity) AS total_demand,
    inv.available_qty,
    COALESCE(pp.planned_qty,0) AS planned_qty,
    (inv.available_qty + COALESCE(pp.planned_qty,0) - SUM(so.quantity)) AS balance
FROM paint_fmcg_db.sales_orders so
```

```
LEFT JOIN paint_fmcg_db.inventory inv
    ON so.product_id = inv.product_id
LEFT JOIN paint_fmcg_db.production_plan pp
    ON so.product_id = pp.product_id
GROUP BY so.product_id, so.product_name, inv.available_qty, pp.planned_qty;
```

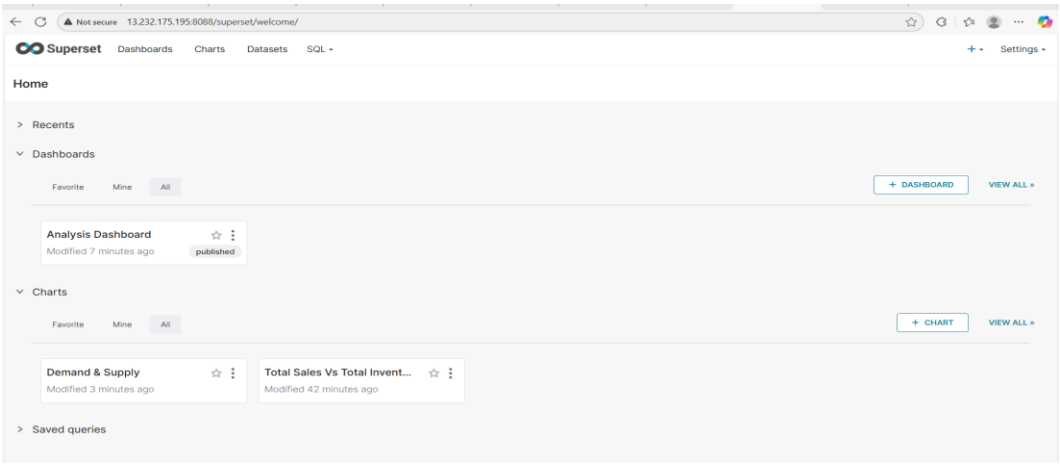
Sales Vs Inventory:

```
SELECT
    s.product_id,
    SUM(s.quantity) AS total_ordered,
    COALESCE(i.available_qty, 0) AS total_inventory,
    COALESCE(i.available_qty, 0) - SUM(s.quantity) AS balance_stock
FROM sales_orders s
LEFT JOIN inventory i
    ON s.product_id = i.product_id
GROUP BY s.product_id, i.available_qty
ORDER BY balance_stock ASC;
```

Step 5: Dashboard built with Apache Superset hosted on EC2 instance



Superset Interface:



EC2 instance where Superset is hosted: