# Retail Customer Journey Analytics Pipeline

This project extends our earlier work by transforming it into a comprehensive end-to-end data engineering solution leveraging cloud infrastructure. It demonstrates the full pipeline from data ingestion and transformation to analysis and visualization in a scalable, cloud-native environment.

**Dataset**: https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce

We will build a cloud-based pipeline that integrates **Amazon S3**, **Amazon EC2**, **Amazon Redshift**, and **Metabase** for data analysis and visualization.

**Steps**:

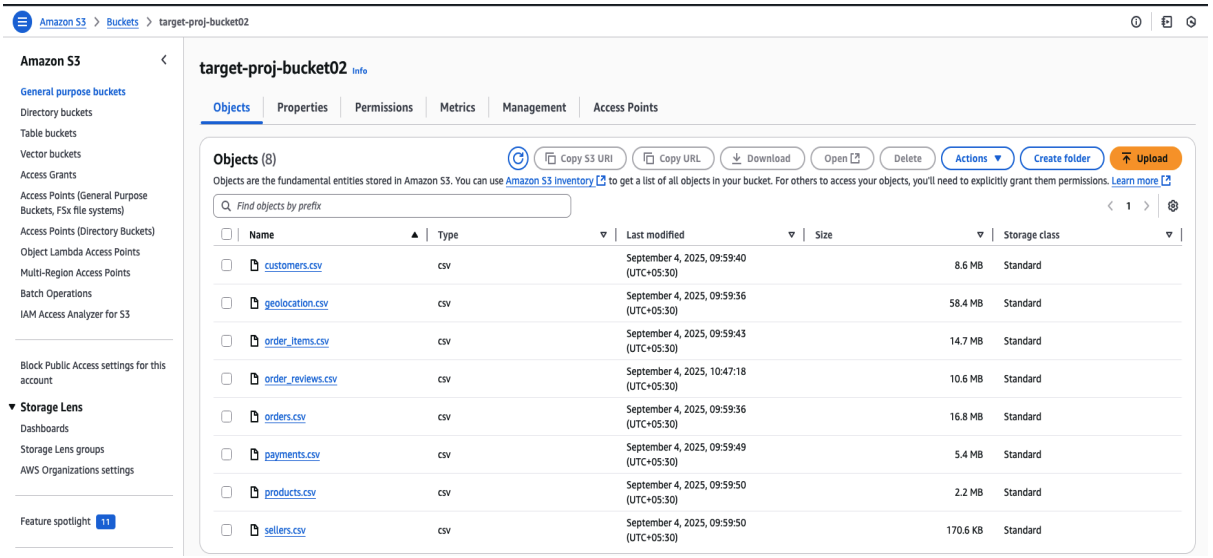
1. Start with raw CSV files from the Olist dataset as the data source.
2. Ingest data into cloud storage using Amazon S3.
3. Use Amazon EC2 instances to perform data processing and transformations.
4. Load the transformed data into Amazon Redshift for efficient querying and analytics.
5. Utilize Metabase to create dashboards and visualize data insights.
6. Integrate these components into a seamless cloud-based data pipeline for end-to-end data engineering and analysis.

1. **Download** the csv files
    a. **Dataset**: https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce

—

2. **Upload** csv files to S3 bucket.

    **Eg**: aws s3 cp customerscsv s3://olist-de-raw-useast1-dev/olist_customers_dataset_csv/customers.csv



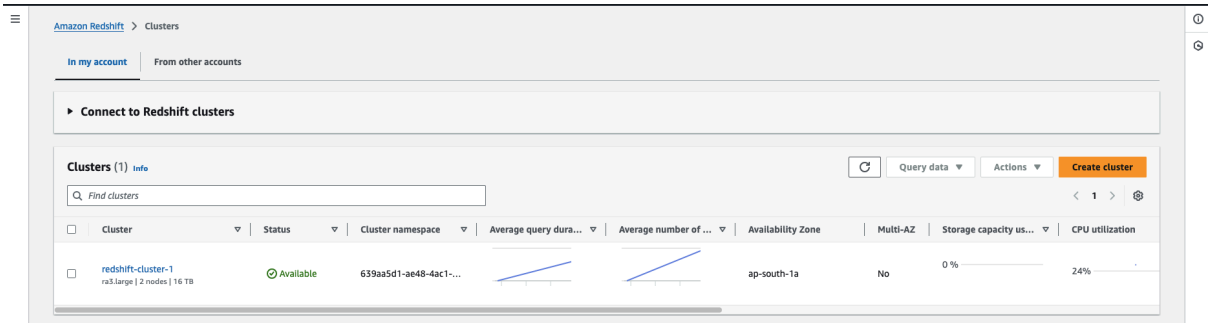

3. **Use Amazon EC2 instances** to perform data processing and transformations.
    a. Skip this as we don't have data processing to do here.

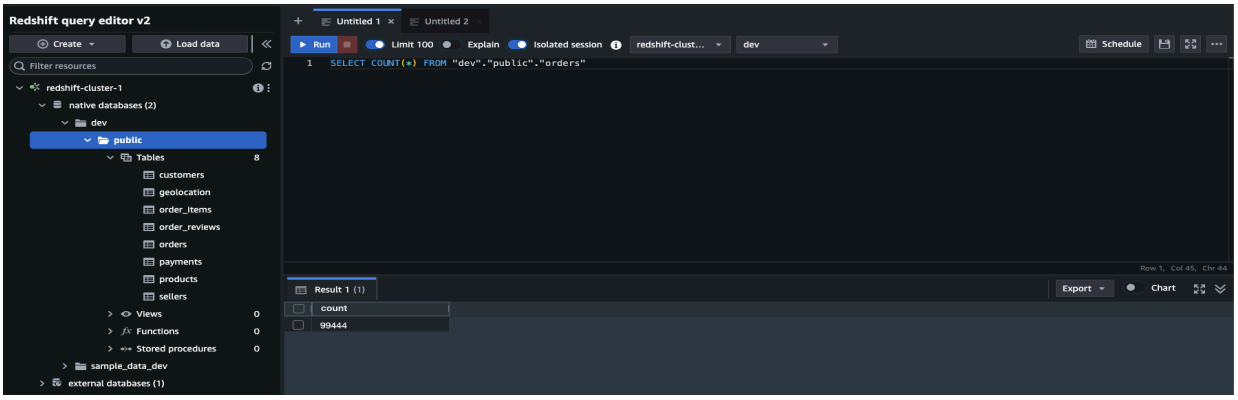
—

4. **Load** the transformed data into Amazon Redshift for efficient querying and analytics.

    Eg: Below is a sample create query to create the ***'orders'*** table in Redshift.

```
CREATE TABLE orders (
    order_id VARCHAR(1000),
    customer_id VARCHAR(1000),
    order_status VARCHAR(1000),
    order_purchase_timestamp TIMESTAMP,
    order_approved_at TIMESTAMP,
    order_delivered_carrier_date TIMESTAMP,
    order_delivered_customer_date TIMESTAMP,
    order_estimated_delivery_date TIMESTAMP
);
```




All table schema is created in Redshift and all data from S3 is copied into the database.

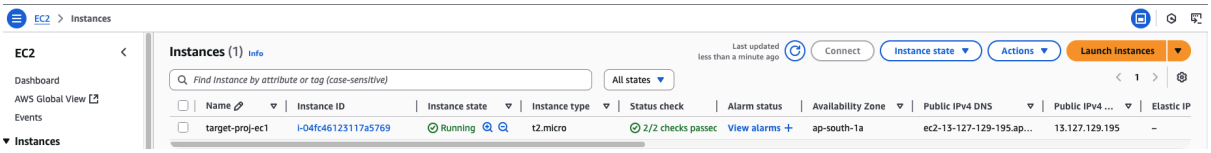
Eg:

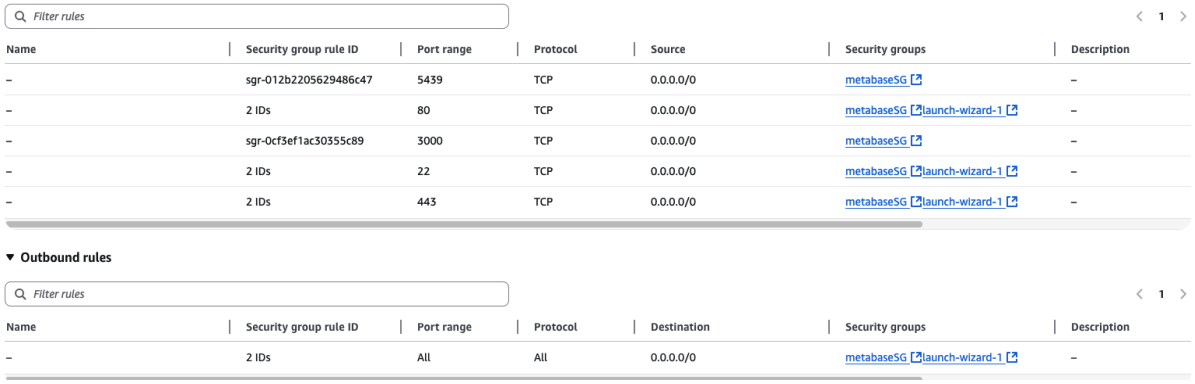
```
COPY orders
FROM 's3://target-proj-bucket02/orders.csv'
IAM_ROLE 'arn:aws:iam::574816783308:role/myRedshiftRole'
CSV
DELIMITER ','
IGNOREHEADER 1;
```

—

5. **Use Amazon EC2 instances** to set up the metabase.

Create an EC2 instance, add it to security groups for access so that it has access to Redshift.





Filter rules

| Name | Security group rule ID | Port range | Protocol | Source | Security groups | Description |
|---|---|---|---|---|---|---|
| – | sgr-012b2205629486c47 | 5439 | TCP | 0.0.0.0/0 | metabaseSG | – |
| – | 2 IDs | 80 | TCP | 0.0.0.0/0 | metabaseSG launch-wizard-1 | – |
| – | sgr-0cf3ef1ac30355c89 | 3000 | TCP | 0.0.0.0/0 | metabaseSG | – |
| – | 2 IDs | 22 | TCP | 0.0.0.0/0 | metabaseSG launch-wizard-1 | – |
| – | 2 IDs | 443 | TCP | 0.0.0.0/0 | metabaseSG launch-wizard-1 | – |



| Name | Security group rule ID | Port range | Protocol | Destination | Security groups | Description |
|---|---|---|---|---|---|---|
| – | 2 IDs | All | All | 0.0.0.0/0 | metabaseSG launch-wizard-1 | – |

Connect to EC2 instance via SSH

**Eg**: ssh -i target-prj-ec1.pem ec2-user@65.2.125.225

- Install Docker and Metabase
    - `sudo yum update -y`
    - `sudo amazon-linux-extras install docker`
    - `sudo service docker start`
    - `sudo systemctl enable docker`
- **Run Metabase Docker Container**
    - `docker pull metabase/metabase`
    - `docker run -d -p 3000:3000 --name metabase metabase/metabase`
- **Access Metabase via your browser.**

  http://13.127.129.195:3000
- **Connect Metabase to Redshift**

  Configure Database Connection in Metabase:
    - Go to Admin Settings > Databases > Add Database.
    - Enter the connection details for your database (e.g., Amazon Redshift):
    - Database type
    - Host — end point in redshift cluster
    - Port — default: 5439
    - Database name
    - Username
    - Password

4

## Add your data

Are you ready to start exploring your data? Add it below **(optional)**.

**Amazon Redshift**

**Connection string (optional)**

jdbc:redshift://redshift-cluster-1.cyhjrljxgyvk.ap-south-1.redshift.amazonaws.com:5439/dev

You can use a connection string to pre-fill the details below.

**Display name**

dev

**Host**

redshift-cluster-1.cyhjrljxgyvk.ap-south-1.redshift.amazonaws.com

**Port**

5439

**Database name**

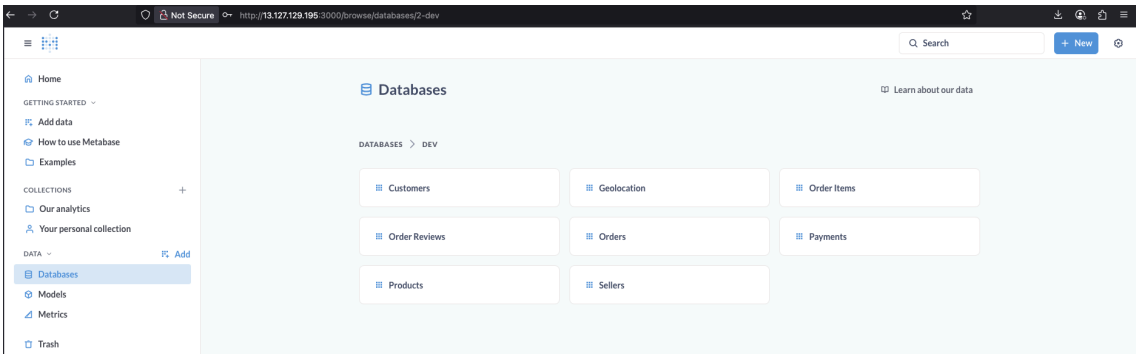
dev

**Schemas**

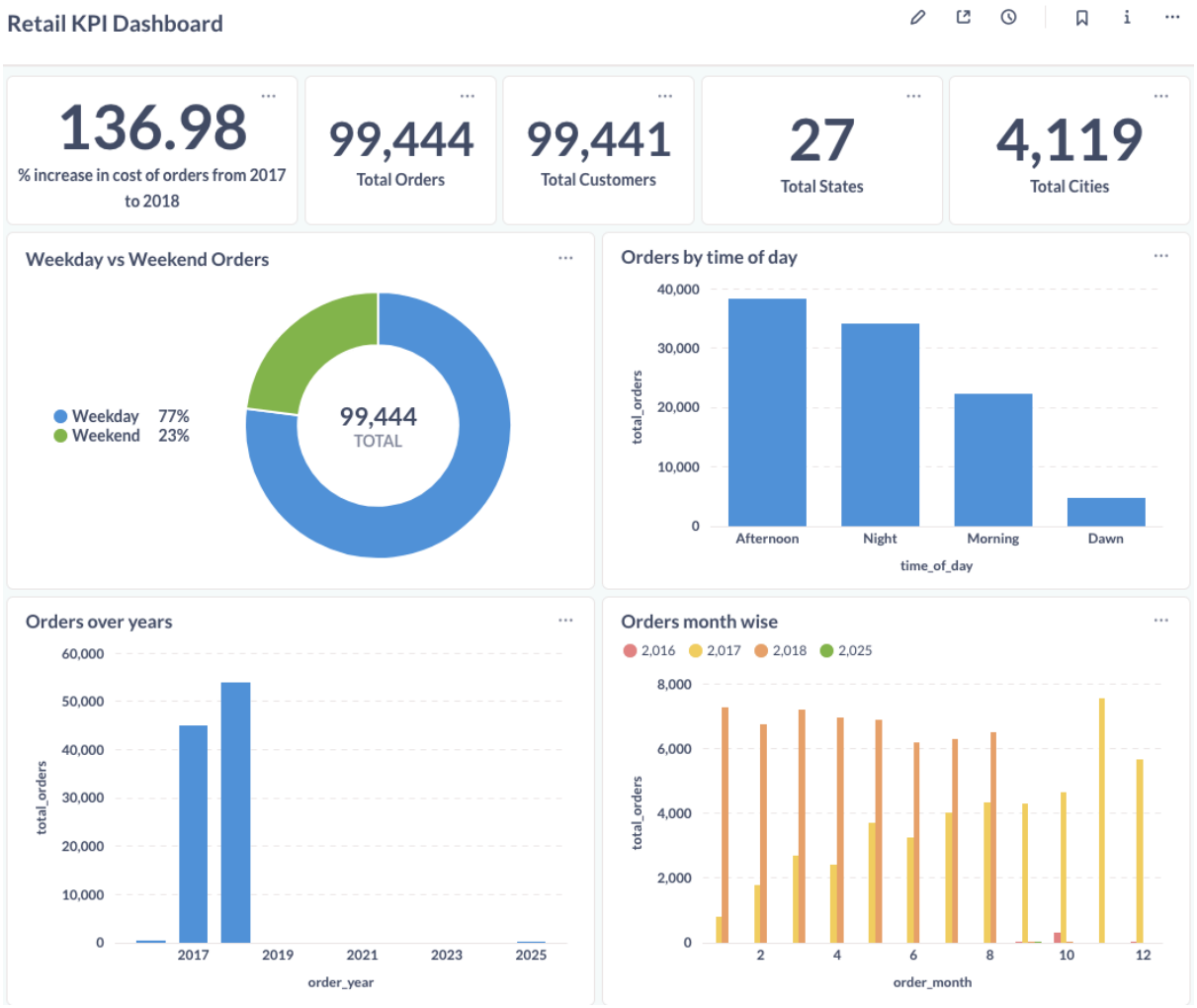
All

**Username**

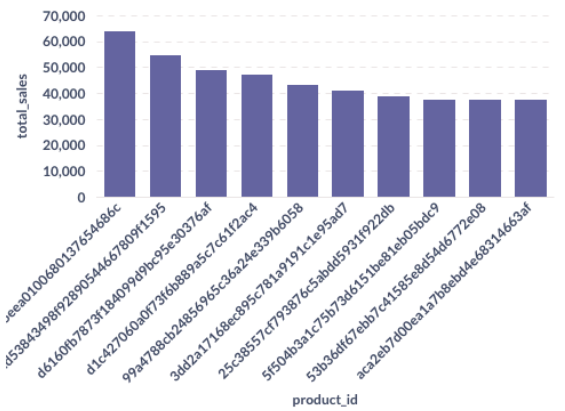
Skip | Connect database



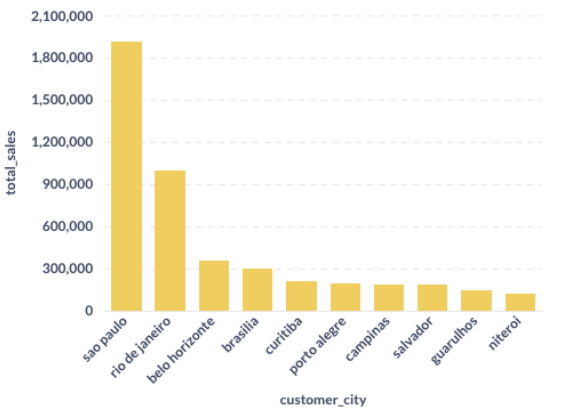
Lets create a simple KPI dashboard similar to our SQL queries.



## Top 10 products by sales




## Top 10 cities by sales

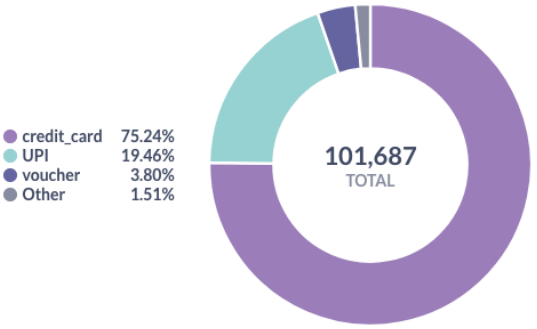



## Count Cities & States of Customers who ordered

| customer_city | customer_state | order_cnt |
|---|---|---|
| sao paulo | SP | 15,540 |
| rio de janeiro | RJ | 6,882 |
| belo horizonte | MG | 2,773 |
| brasilia | DF | 2,131 |
| curitiba | PR | 1,521 |
| campinas | SP | 1,444 |
| porto alegre | RS | 1,379 |

2,000 rows

## Orders by payment type

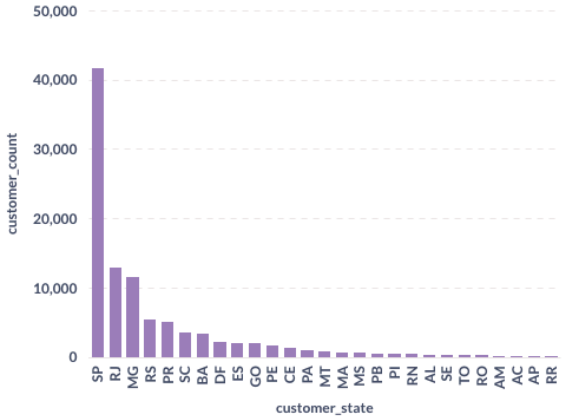


## MoM order placed by state

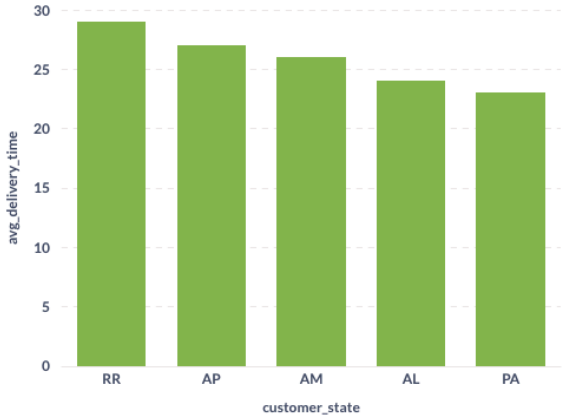
| order_year | order_month | customer_state | total_orders |
|---|---|---|---|
| 2,016 | 9 | RR | 1 |
| 2,016 | 9 | RS | 1 |
| 2,016 | 9 | SP | 2 |
| 2,016 | 10 | AL | 2 |
| 2,016 | 10 | BA | 4 |
| 2,016 | 10 | CE | 8 |
| 2,016 | 10 | DF | 6 |
| 2,016 | 10 | ES | 4 |

566 rows

## Customer Distribution by State




## Top 5 states with high avg delivery time




## Total & Avg value of order price by state

| customer_state | total_order_price | average_order_price |
|---|---|---|
| SP | 5,203,205.05 | 109.65 |
| RJ | 1,824,092.67 | 125.12 |
| MG | 1,585,308.03 | 120.75 |
| RS | 750,304.02 | 120.34 |
| PR | 683,083.76 | 119 |
| SC | 520,553.34 | 124.65 |
| BA | 511,349.99 | 134.6 |
| DF | 302,603.94 | 125.77 |

27 rows

### Top 5 states with high avg freight value


### Top 5 states with low avg delivery time


### Total & Avg value of Freight by State

| customer_state | total_freight_value | average_freight_value |
|---|---|---|
| SP | 718,723.07 | 15.15 |
| RJ | 305,589.31 | 20.96 |
| MG | 270,853.46 | 20.63 |
| RS | 135,522.74 | 21.74 |
| PR | 117,851.68 | 20.53 |
| BA | 100,156.68 | 26.36 |
| SC | 89,660.26 | 21.47 |
| PE | 59,449.66 | 32.92 |

27 rows

### Low 5 states with avg freight value


### Difference between avg of actual vs estimated delivery time