# Project 3

## Steps:

***Step 1:*** In the terminal write the command **docker-compose up postgres** which will pull the postgres

***Step 2:*** Then write the command **docker exec -it postgres psql -U myuser mydb** to go to the database

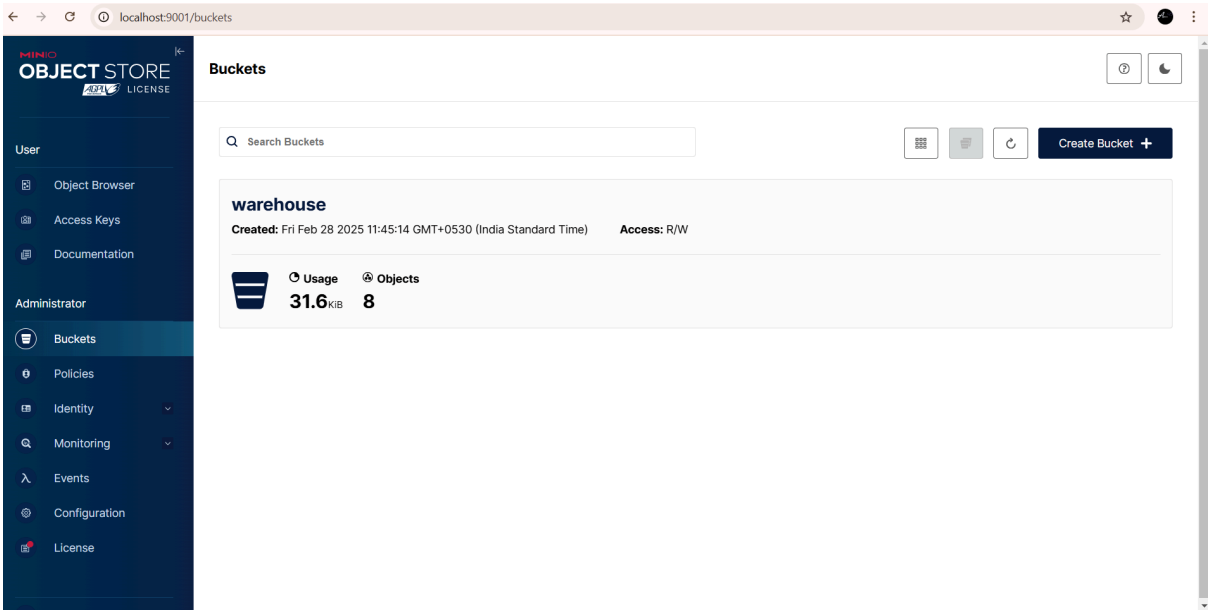
***Step 3:*** Then create a table named sales_data using the SQL command **create table sales_data (id int, product_name varchar(255), category varchar(50), sales_amount numeric(10,2), sales_date date);**

***Step 4:*** Insert the values using the SQL command **insert into sales_data(id, product_name, category, sales_amount, sales_date) values (1,'Product A', 'Electronics', 1000.50, 2024-03-01), (2,'Product B', 'Clothing', 750.25, 2024-03-02), (3,'Product C', 'Home Goods', 1200.75, 2024-03-03), (4,'Product D', 'Electronics', 900.00, 2024-03-04), (5,'Product E', 'Clothing', 600.50, 2024-03-05);**

***Step 5:*** Pull the other images **docker-compose up spark nessie minio dremio**

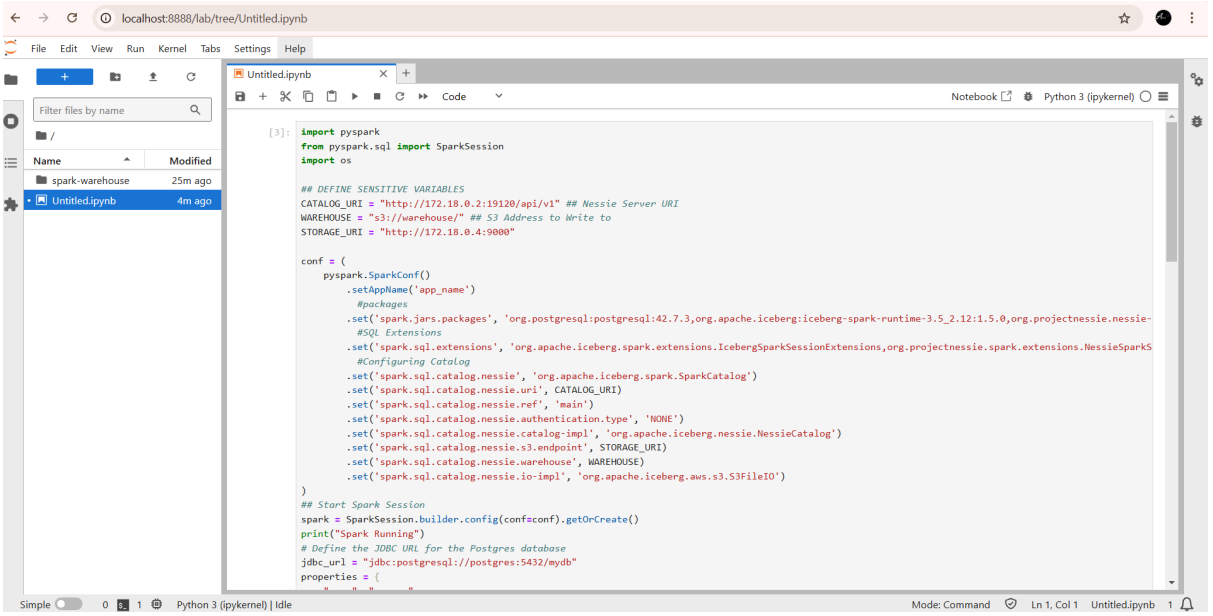
***Step 6:*** Create a bucket in minio:

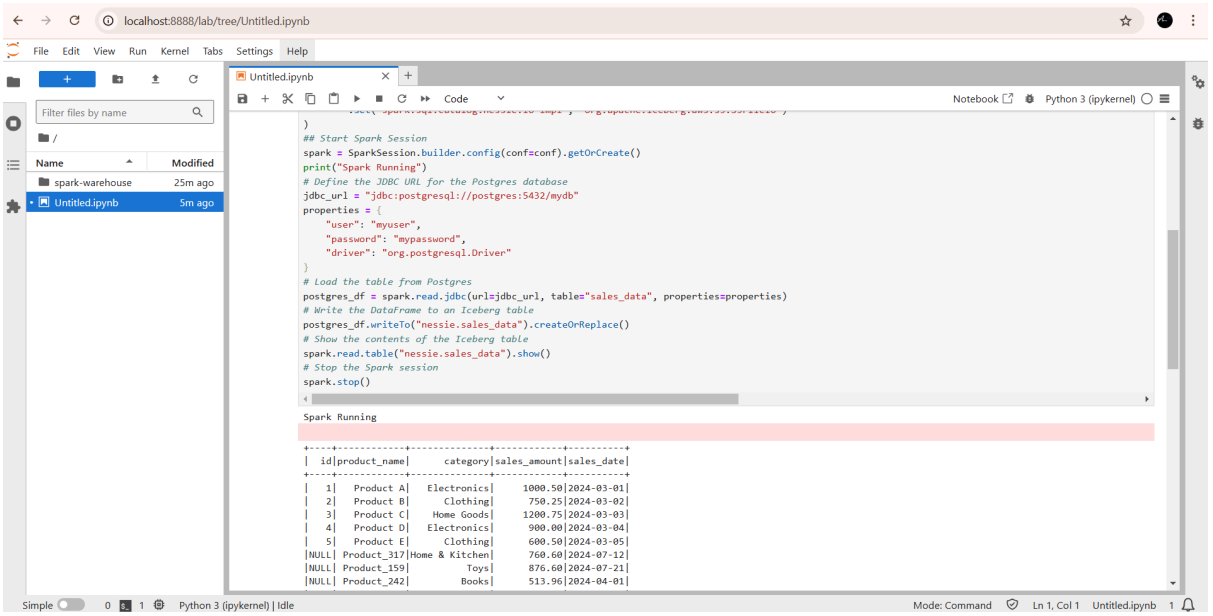
a. Go to the browser and type **localhost:9001** to open it
b. Username: admin
c. Password: password
d. Click on create bucket in the object store and name it warehouse




## ***Step 7:*** Open the Spark notebook

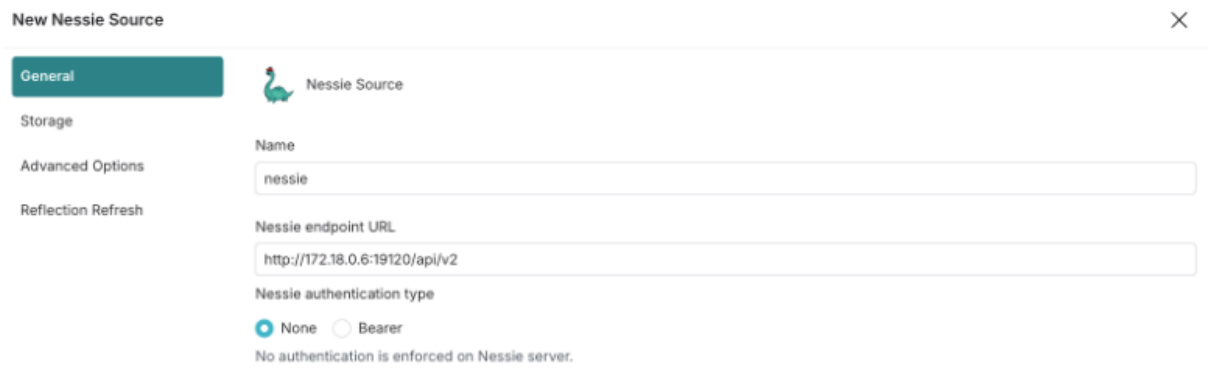
a. Type **localhost:8888** on the browser
b. Create a new Python notebook and write the Python code mentioned below and change the IP address of the minio to the address of the local computer. (**NOTE:** To get the IP address use **docker inspect minio**)
c. Run the Python code

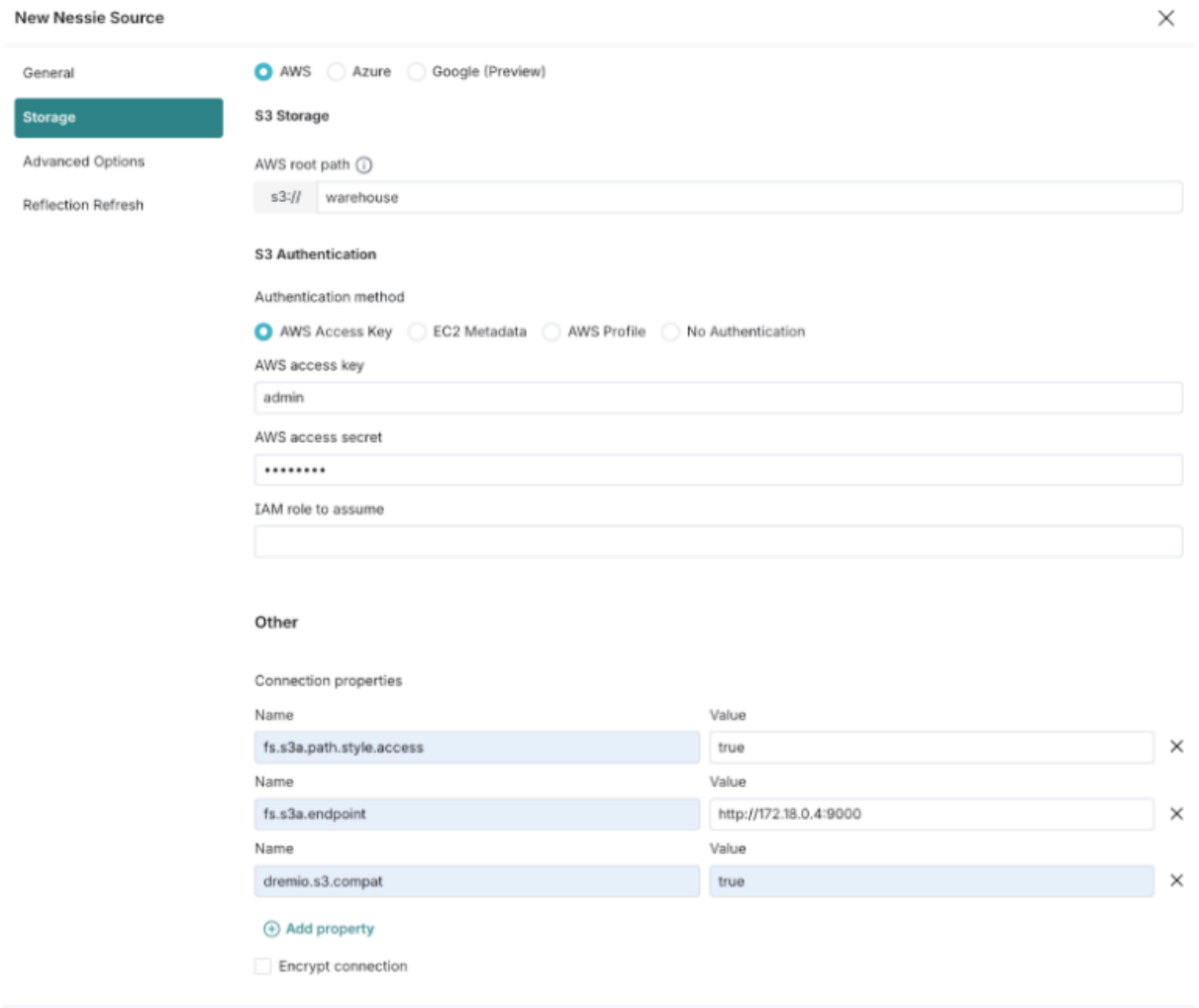


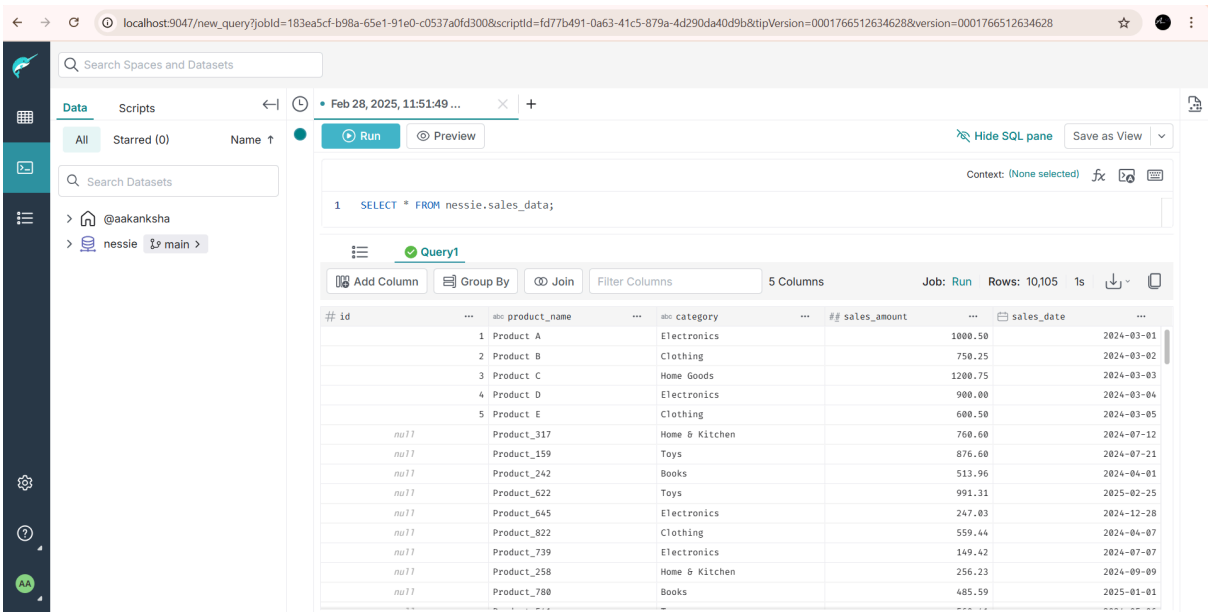



## ***Step 8:*** Open the dremio for connection.

a. **localhost:9047**
b. Create username and password and do not include any special characters in the password.
c. Click on Add Source => Select nessie as the source




New Nessie Source
General
Storage
Advanced Options
Reflection Refresh
AWS
Azure
Google (Preview)
S3 Storage
AWS root path
s3://
warehouse
S3 Authentication
Authentication method
AWS Access Key
EC2 Metadata
AWS Profile
No Authentication
AWS access key
admin
AWS access secret
••••••••
IAM role to assume
Other
Connection properties
Name
Value
fs.s3a.path.style.access
true
Name
Value
fs.s3a.endpoint
http://172.18.0.4:9000
Name
Value
dremio.s3.compat
true
Add property
Encrypt connection

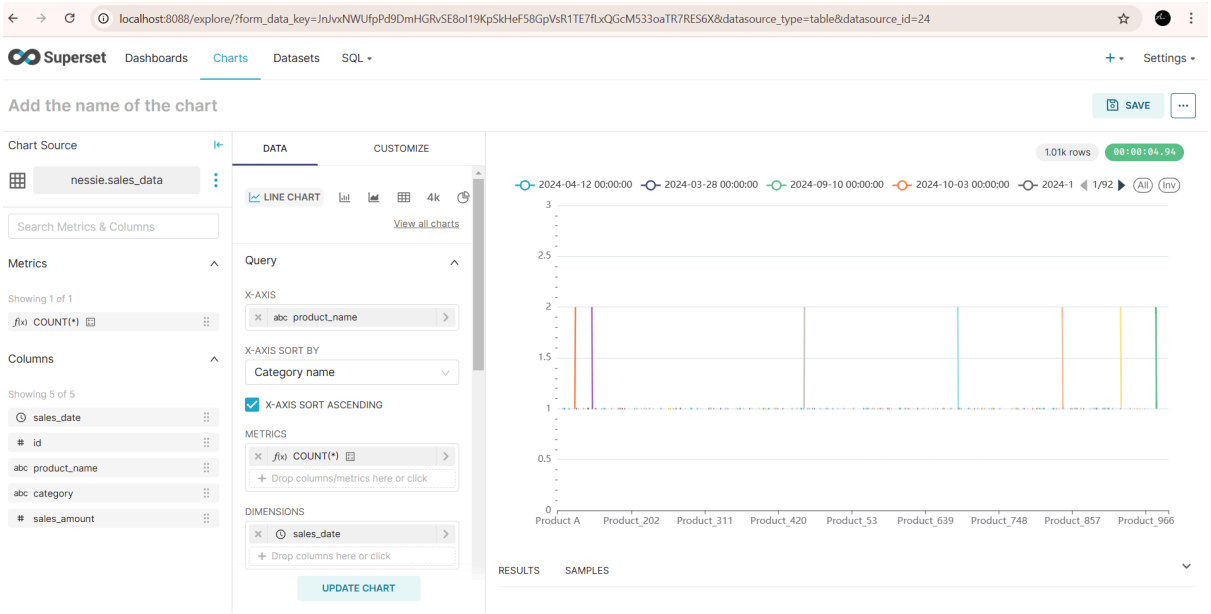

localhost:9047/new_query?jobId=183ea5cf-b98a-65e1-91e0-c0537a0fd300&scriptId=fd77b491-0a63-41c5-879a-4d290da40d9b&tipVersion=0001766512634628&version=0001766512634628
Search Spaces and Datasets
Data
Scripts
All
Starred (0)
Name
Search Datasets
@aakanksha
nessie
main
Feb 28, 2025, 11:51:49 ...
Run
Preview
Hide SQL pane
Save as View
Context: (None selected)
1 SELECT * FROM nessie.sales_data;
Query1
Add Column
Group By
Join
Filter Columns
5 Columns
Job: Run
Rows: 10,105
1s
id
product_name
category
sales_amount
sales_date
1 Product A Electronics 1000.50 2024-03-01
2 Product B Clothing 750.25 2024-03-02
3 Product C Home Goods 1200.75 2024-03-03
4 Product D Electronics 900.00 2024-03-04
5 Product E Clothing 600.50 2024-03-05
null Product_317 Home & Kitchen 760.60 2024-07-12
null Product_159 Toys 876.60 2024-07-21
null Product_242 Books 513.96 2024-04-01
null Product_622 Toys 991.31 2025-02-25
null Product_645 Electronics 247.03 2024-12-28
null Product_822 Clothing 559.44 2024-04-07
null Product_739 Electronics 149.42 2024-07-07
null Product_258 Home & Kitchen 256.23 2024-09-09
null Product_780 Books 485.59 2025-01-01

***Step 9:*** Pull the superset using the command **docker-compose up superset**

***Step 10:*** Then execute the docker image by the command **docker exec -it superset superset init**

***Step 11:*** Open Superset

a. **localhost:8088**
b. Password: admin
c. Click on ‘+’ => Connect => Add database
    i. In the display name select **Other**
    ii. In the SQL ALCHEMY URI type dremio+flight://username:password@dremio:32010/?UseEncryption=false (NOTE: username and password of dremio)
    iii. Click on Connect




**NOTE:**
To generate random data in postgreSQL use the command:

```
INSERT INTO sales_data (product_name, category, sales_amount, sales_date)
SELECT  'Product_' || floor(random() * 1000)::int, (ARRAY['Electronics', 'Clothing', 'Home & Kitchen', 'Books', 'Toys'])[floor(random() * 5) + 1], round((random() * 990 + 10)::numeric, 2),    now() - (random() * interval '365 days') FROM generate_series(1, 10000);
```

# About the technologies and their impact

## 1. Nessie (Catalog Server)

### What it does?

Nessie is like a **version control system (like Git) for data**. It helps in tracking changes made to datasets over time. This means you can roll back to a previous version of your data if needed.

### Impact if missing?

Without Nessie, **data versioning** will be difficult. If someone makes a mistake in the data, there will be no easy way to revert it, leading to data corruption or inconsistencies.

## 2. MinIO (Storage Server)

### What it does?

MinIO is a **cloud storage service**, similar to Amazon S3, but you can run it on your own servers. It stores large amounts of structured and unstructured data.

### Impact if missing?

Without MinIO, there will be **no centralized storage** for the data. Other services (like Spark and Dremio) that need to process or analyze data will not have a place to fetch it from.

## 3. Dremio (Query Engine)

### What it does?

Dremio is a **high-performance SQL engine** that allows users to query data from different sources (like MinIO, PostgreSQL, or Spark) without moving the data.

### Impact if missing?

Without Dremio, **SQL queries on distributed data will be slow** and difficult to manage. You would need separate tools for each data source, leading to inefficiency.

## 4. Apache Spark (Big Data Processing)

### What it does?

Spark is a powerful **data processing framework** that can handle large-scale data analysis, transformations, and machine learning tasks.

**Impact if missing?**

Without Spark, **processing big data would be very slow**. Tasks like data cleaning, ETL (Extract, Transform, Load), and running ML models will take longer and require more resources.

## 5. PostgreSQL (Database)

**What it does?**

PostgreSQL (or Postgres) is a **relational database** used to store structured data in tables with relationships.

**Impact if missing?**

Without Postgres, **storing structured data would be difficult**. You would have to rely only on file-based storage (MinIO), which is less efficient for structured queries.

## 6. Superset (Data Visualization)

**What it does?**

Superset is a **business intelligence (BI) tool** that helps create dashboards and visualizations based on data from different sources.

**Impact if missing?**

Without Superset, **data insights would be harder to obtain**. Users would need to manually run SQL queries and analyze raw data instead of seeing easy-to-understand charts and dashboards.

## Overall Impact if Any Service is Missing

- **Data inconsistency** (without Nessie)
- **No storage for big data** (without MinIO)
- **Slow and inefficient queries** (without Dremio)
- **Difficult data processing** (without Spark)
- **No structured data storage** (without PostgreSQL)
- **No easy way to visualize data** (without Superset)

Together, these technologies form a **complete data engineering ecosystem**, ensuring efficient storage, processing, and analysis of data.