**Data Engineering Project**

**Describtion**

This project entails typical Analytical Data Engineering, involving the ingestion of data from various sources and its loading into the Snowflake data warehouse. Within the warehouse, after undergoing a series of data transformation processes, preparing the data for Business Intelligence (BI) usage. The BI tool Metabases connects to the data warehouse to generate diverse dashboards and reports.

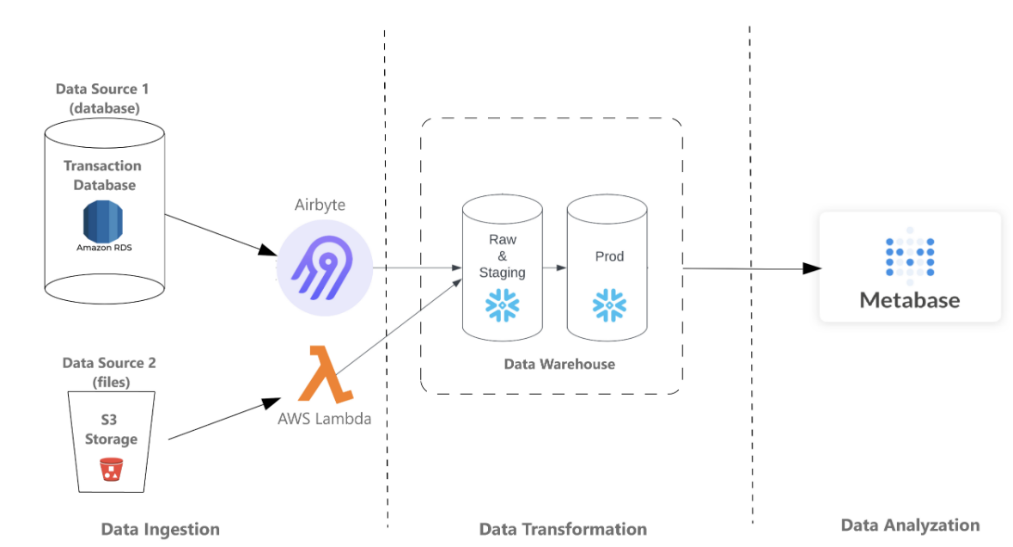


Figure 1: Project Diagram

**Data preview**

The dataset utilized originates from TPCDS, a well-known dataset designed for database testing, with a specific emphasis on Retail Sales. It encompasses sales records from both websites and catalogs, along with detailed information on inventory levels for each item within every warehouse. Moreover, it incorporates 15 dimensional tables containing valuable information about customers, warehouses, items, and more.

**Business Requirement**

The ultimate objective is to generate dashboards and reports using the BI tool Metabase. It is essential to comprehend the requirements for Metabase dashboards and reports:

1. Determine the top and bottom-performing items of the week by analyzing sales amounts and quantities.
2. Show items with low inventory levels on a weekly basis.
3. Identify items with low stock levels, including their associated week and warehouse numbers, marked as "True".

**Project Infrastructure**

The entire infrastructure is constructed in the cloud :

Servers: Create several servers on the AWS cloud.(EC2)

Tools: Install various tools on these servers, including Airbyte for data ingestion, and Metabase as the BI tool for building dashboards.

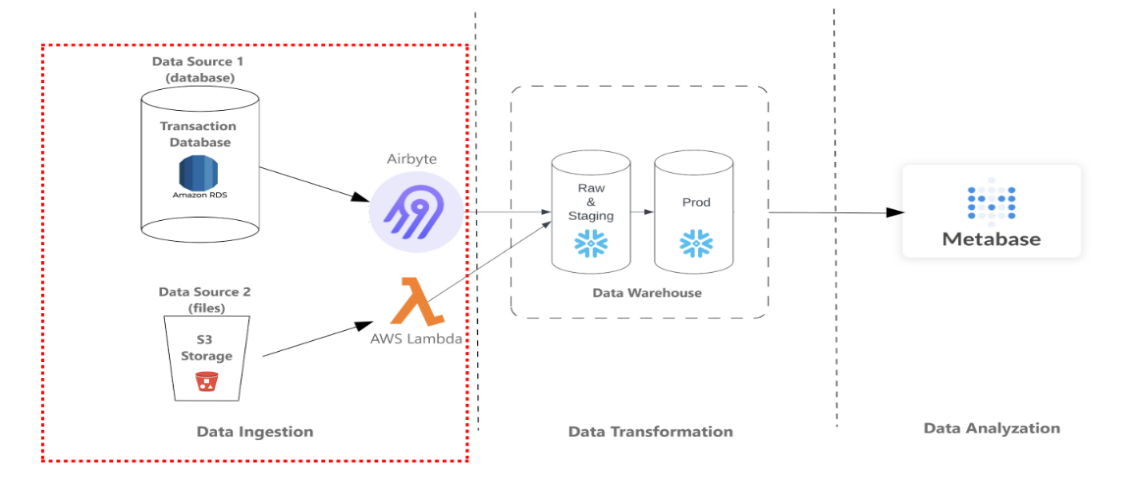
Cloud Data Warehouse: using Snowflake, the cloud data warehouse, to store data and perform data transformation.

- AWS Lambda: Using AWS Lambda, a serverless service, to ingest data from AWS data storage (S3).

**Project Steps**

**1.** **Data Ingestion**

he first part of the project involves Data Ingestion. It entails connecting to two data sources: the Postgres database and the AWS S3 bucket. Utilizing Airbyte, establish a connection to the raw\_st schema of the Postgres database on AWS RDS, and transfer all tables to the Snowflake data warehouse. In addition, leverage AWS Lambda to connect to the AWS S3 bucket and transfer the file named inventory.csv from the S3 bucket to the Snowflake data warehouse.



1.1Create my own database in SnowFlake account TPCDS

A close-up of a computer screen

Description automatically generated

1.2 Create schema called ‘RAW’ and table called ‘INVENTORY’ without date because it will load from lambda function

A screenshot of a inventory

Description automatically generated

1.3 Create 2 EC2 instances. One instance is in t2.small size, another is a t2.large. The small size one will be used to install Metabase, and the t2.large is used to install Airbyte.

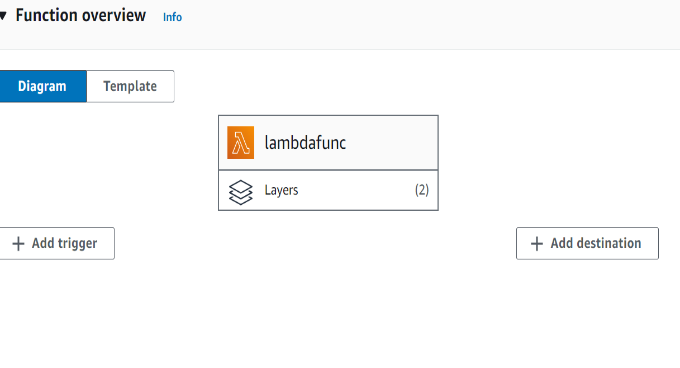
A screenshot of a computer

Description automatically generated

Also I Installed Docker on both the created instances , and that’s for simplifies management, enhances portability, and improves scalability of these applications in cloud environments.

1.4 Lambda Function :

Used for loading the inventory data tableA screenshot of a computer

Description automatically generated

1.5 Airbyte Installation and Configuration

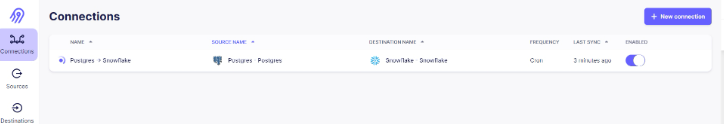
Starting the instance ‘AirbyteINS’ and install ‘Airbyte’

A screenshot of a computer program

Description automatically generated

1.6 connect to AWS RDS **Postgres as source and snowflake as destination**

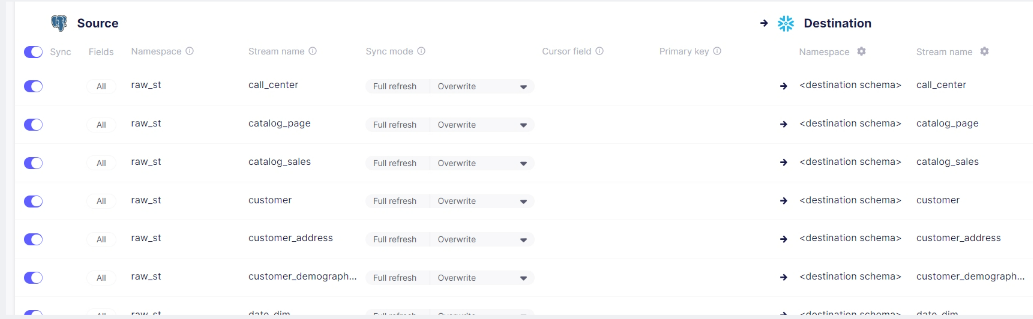
**And setting up the connection**



**A screenshot of a computer

Description automatically generated**

Loading the data from the PostgreSQL to Snowflake Using Airbyte server.



As it is shown here all data are there ,that means that is the connection is complete and worked.

**2. Data Modeling**

Firstly, since we have ingested data from RDS and S3 bucket to Snowflake RAW schema.

2.1 Data Background

The dataset, sourced from TPCDS, is renowned for its application in database testing, specifically oriented towards Retail Sales. Within this dataset, you'll find sales records from both the website and catalog, along with comprehensive inventory levels for each item in every warehouse. Furthermore, there are 15 dimensional tables encompassing details about customers, warehouses, items, and more

The complete dataset is not stored in a single location, rather, it has been divided into two parts:

RDS: All tables, except for the inventory tables, are housed in the Postgres DB on AWS RDS. These tables undergo daily refreshes, ensuring they are consistently updated with the latest data. To access the most recent information, it is necessary to execute the ETL process on a daily basis.

S3 Bucket: The sole Inventory table is hosted in an S3 bucket. Each day, a new file containing the latest data is deposited into the S3 bucket. However, it's important to note that the inventory table typically captures data only at the end of each week. Consequently, on most occasions, I must find only one entry per item in each warehouse for each week. Despite this, it's necessary to ingest the inventory file from the S3 bucket on a daily basis.

2.2 Tables in the Dataset



2.3 Business Requirements and Data Modeling

* sum\_qty\_wk: The sum of sales\_quantity for this week.
* sum\_amt\_wk: The sum of sales\_amount for this week.
* sum\_profit\_wk: The sum of net\_profit for this week.
* avg\_qty\_dy: The average daily sales\_quantity for this week (= sum\_qty\_wk/7).
* inv\_on\_hand\_qty\_wk: The item’s inventory on hand at the end of each week in all warehouses (= The inventory on hand at the end of this week).
* wks\_sply: Weeks of supply, an estimated metric to see how many weeks the inventory can supply the sales (inv\_on\_hand\_qty\_wk/sum\_qty\_wk).
* low\_stock\_flg\_wk: Low stock weekly flag. For example, if there is a single day where (avg\_qty\_dy > 0 && (avg\_qty\_dy > inventory\_on\_hand\_qty\_wk)) in the week, then mark this week's flag as True.

In addition to the fact table, also need to integrate a customer dimension with columns from the tables: Customer, Customer Address, Customer Demographics, Household Demographics, Income Band.

Besides, in order to realize the requirements from the Metabase BI purpose, iam also need the Date\_dim, Warehouse, and Item tables in the Prod schema.

**The DDL code:** 

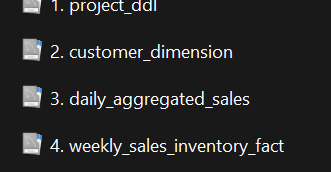
2.4 ETL and Data Loading

I need to create ETL scripts to populate raw table to the newly created data model. You will need:

Develop a merge script to integrate the new Customer dimension table into the existing dimension table within the Analytics schema, following Type 2 methodology.

Develop a merge script to incorporate the newly added daily sales records into the existing daily sales fact table within the Analytics schema.

Create a script to perform a join between the daily sales fact table and the updated inventory table, creating the weekly sales and inventory fact table within the Analytics schema.



I can not show the code (too much lines), But I reference it in my GitHub.

2.5 Scheduling with Snowflake

the above snowflake SQL scripts are only one-time scripts. To update the table daily and weekly, i need to create tasks and stored procedure in Snowflake which updates the daily sales fact table every day and the weekly inventory and sales fact table every week.

I focused on automating daily and weekly data aggregations for Snowflake tables and loading the results into our Snowflake database. The objective is to streamline data processing and enhance analytical capabilities by regularly updating aggregated datasets.

**3. Data Visualization**

I will use Metabase to visualize our data and generate the reports i need.

The business requirements for the Metabase reports are:

1. Identify the highest and lowest performing items of the week by analyzing sales amounts and quantities.

2. Display items with low supply levels for each week.

3. Detect items experiencing low stock levels, along with their corresponding week and warehouse numbers, marked as True.

3.1 Metabase Installation

Prerequisite : Validate the EC2 Instance.

Before i start this portion, I make sure that EC2 outbound security rules has CustomTCP Port 3000 open anywhere 0.0.0.0/0.

Then, start the EC2 instance ‘MetabaseINS’ .

3.2Metabase Connection to the Snowflake Database

Connect Snowflake database, schema ANALYTICS with Metabase. A screenshot of a web page

Description automatically generated

3.3 Create a report using Metabase

generate a report in Metabase based on our business requirements.

