Contents

[Project Overview 1](#_Toc175414602)

[Architecture Diagram 3](#_Toc175414603)

[Flow of data 3](#_Toc175414604)

[Services/ Tools used 4](#_Toc175414605)

[Storage Access / mounting 4](#_Toc175414606)

[Folder Structure/notebook organization 6](#_Toc175414607)

[Data Loading 8](#_Toc175414608)

[Full Load 8](#_Toc175414609)

[Incremental Load 8](#_Toc175414610)

[Scheduling 8](#_Toc175414611)

## Project Overview

The project is about a sample shipment tracking. I have sampled some of the possible entities of a shipment tracking project like orders, shipment details, invoices and transactions. In the landing zone/src we have csv files containing data of each entity.

A screenshot of a computer

Description automatically generated

Data from src layer is loaded into bronze layer as it is in Delta Tables.

A screenshot of a computer

Description automatically generated

As I have been following the medallion architecture, cleaned data from bronze layer is read, cleaned and loaded to silver layer in Delta tables.

Here I have created few more tables like temp\_customer, temp\_carriers which work as intermediary store for populating dimensions and facts in gold layer.

A screenshot of a computer

Description automatically generated

And in the last stage curated, calculated data is loaded to gold layer in form of facts and dimensions. For the demonstration purpose I have created only a single fact here.

A screenshot of a computer

Description automatically generated

At every later I have used a structure, watermark to support incremental or bulk load of data. This structure hold the timestamp of last run of data populate.

## Architecture Diagram

A diagram of a diagram

Description automatically generated

Src is the landing zone, I have data in csv format in this layer.

Bronze is the raw layer, data from src/landing zone is copied to bronze layer as is without any modification.

Silver layer hold cleansed data with additional entities and details like ingestion date, understandable column names etc.

Gold layer is the datawarehouse layer where data is stored in facts and dimensions.

## Flow of data

A screenshot of a computer

Description automatically generated

## Services/ Tools used

Azure Data Bricks 🡪 Cleaning & loading data

Azure Data Lake 🡪 Data Storage

Azure Key Vault 🡪 Secured storage for access token

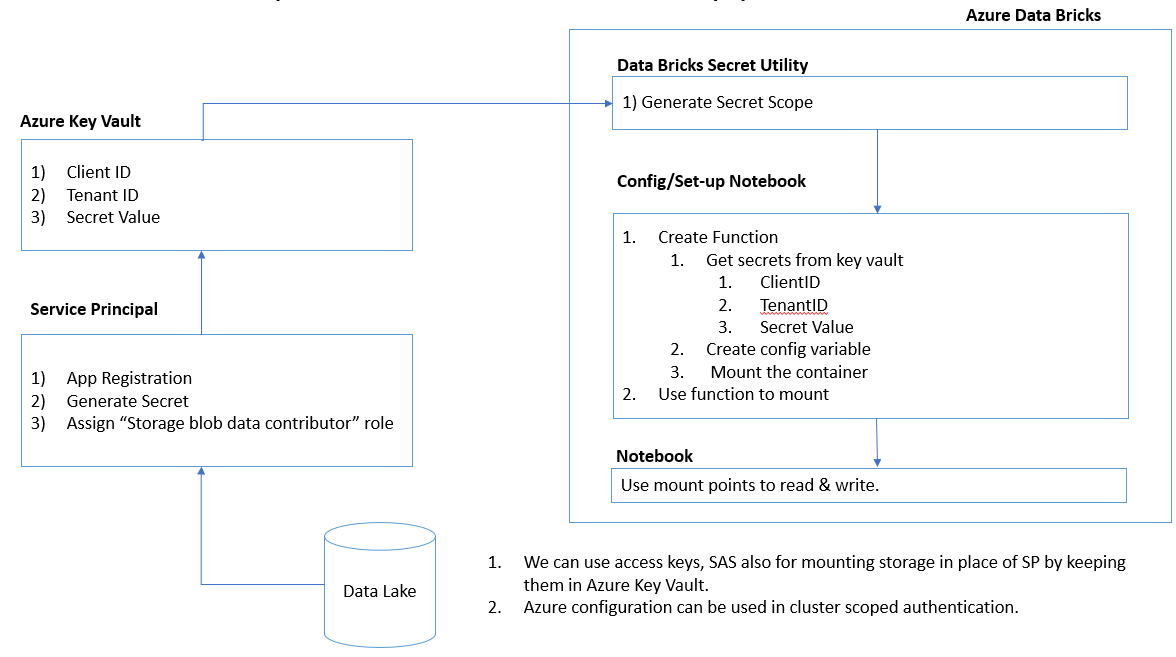
Azure Active Directory 🡪 User Authentication

Azure Data Factory 🡪 Scheduling and Execution

Azure Delta Tables 🡪 Silver & Gold layer storage format

Spark SQL 🡪 Data querying and cleaning

## Storage Access / mounting



For authenticating to Data Lake, I have used session scoped credentials using Service Principal. Client Id and secret value of Data lake is stored in Azure key vault which is read by Databricks Secret Utility. For mounting data storage I have created a notebook and mounted required containers.

A screenshot of a computer

Description automatically generated

A computer screen shot of text

Description automatically generated

here in above example

1. Reading client\_id, tenant\_id and client\_secret from databricks secret utility which is further reading data from Azure Key Vault.
2. Created a function “mount\_adls” which is used to mount the data lake containers.
3. Once container “dev” is mounted, the mounted container can be directly used in any notebook instead of using configuration again and again.

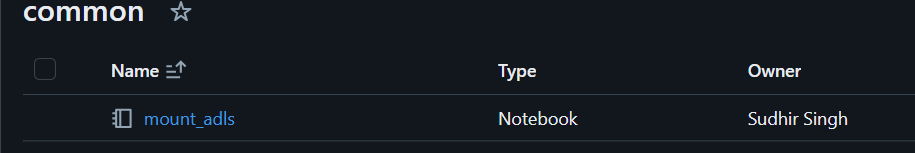
## Folder Structure/notebook organization

A screenshot of a computer

Description automatically generated

At the top level I have “notebooks” folder which has 2 main folders.

1. Common : Contain common code for mounting storage location.



1. ShipmentTracking
   1. Executor
   2. Load
   3. Set up

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

In set-up folder I have created notebooks for common configuration like creating variables to read storage layer, defining database schema for bronze, silver and gold.

A screenshot of a computer

Description automatically generated

In load folder there are notebooks to cleanse and move data from one layer to another.

A screenshot of a computer

Description automatically generated

Finally executor folder holds notebooks to call, data loading notebooks in order for Full Load or Incremental load.

## Data Loading

A diagram of data flow

Description automatically generated with medium confidence

### Full Load

1. Create notebooks \_setfullload from load folder to truncate all the tables and reset watermarkvalue.
2. Call notebooks to load data from scratch using dbutils.notebook.run.

### Incremental Load

1. Call notebooks to load data, the notebooks will read last populated timestamp from watermark table and process only incremental data.

## Scheduling

Here I have used Azure Data Factory “data bricks” task to schedule the entire process.

A screenshot of a computer

Description automatically generated