**Project : Customer 360 View for Banking Industry**

**Project Over-view:** In this Project we will see *how to develop an* end-to-end data pipeline using Delta Lake which is an open-source storage layer. Also you will learn how data moves from bronze to gold container, how to make an incremental load, create external tables for data analysis and orchestrate your pipeline.

**Prerequisite:**

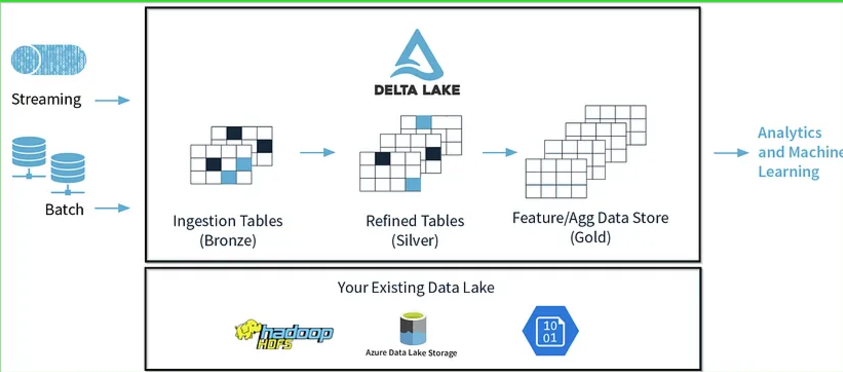
* Azure Account (Free Trial /Pay-as-you-go Subscription)
* Basic knowledge of Azure
* Basic knowledge of Databricks
* Basic knowledge of Python and PySpark

**Business Requirements:**

1.The Marketing area needs to have updated customer data to be able to contact them and make offers.

2.The Finance area requires to have daily loan transactions complemented with customer drivers to be able to analyze them and improve the revenue.

**Architecture:**



**Technologies Used:**

• Data Source: CSV files

• Orchestration: Azure Data Factory

•Alerts: Azure monitor

• Transformation: Azure Databricks

• Security: App Registrations, Azure key-vault

• Storage: Azure Data Lake Gen2

• Analytics: Azure Synapse Analytics

**Dataset Used:**

1.**Customer:** This data needs to be treated with an UPSERT technique because it only send new and modified records daily.

* customerId: unique identifier
* firstName: name
* lastName: last name
* phone: home phone number
* email: email
* gender: Male or Female
* address: place where the customer lives
* is\_active: flag that indicates if the client is with us

2.**Customer Drivers**: This data is generated daily from the RiskModeling area. The data that is sent is an snapshot of the day. This data will be loaded incrementally.

* date: date that the data was generated by RiskModeling area
* customerId: unique identifier of the customer
* monthly\_salary: monthly salary in USD
* health\_score: score - how important is the customer for the bank
* current\_debt: current debt that the customer has with our bank
* category: segment of the customer

3.**Loan Transactions:** Data correspond to the transactions performed in a specific date. This data will be loaded incrementally.

* date: date of the transaction
* customerId: unique identifier of the customer
* paymentPeriod: term of the loan
* loanAmount: amount requested by the customer
* currencyType: currency of the loan (USD, EUR)
* evaluationChannel: channel by which the loan was sold
* interest\_rate: rate of the loan

**Working:**

Step 1 — Create Resource Group

Step 2— Create Azure Databricks Cluster

A screenshot of a computer

Description automatically generated

You can also edit your cluster configuration and change some values as the termination time after inactivity.

**Step 3 —** Create Azure Storage

Make sure to enable hierarchical namespace

create a storage account with container having bronze, silver, gold

A screenshot of a computer

Description automatically generated

**Step 4 —** Create App Registrations

save the application Id, object Id and Directory Id on a notepad

A screenshot of a computer

Description automatically generated

Create Secrets:

copy the secret value and keep it in a notepad, because once you come out of the screen, you won’t be able to get it again.

A screenshot of a computer

Description automatically generated

After creating app registrations need to give access in storage account

IAM - Add Role Assignment - Storage Blob Data Contributor-select app registrations as member

**Step 5 —** Create Azure key vault

Objects – Secrets - Generate/Import.

copy the Vault URI and Resource ID values in a notepad, we will be using it to link our databricks secret scope with azure key vault.

A screenshot of a computer

Description automatically generated

Creating secrets in Azure key-vault-need to give “Key Vault Administrator" ,”key vault secret officer “ role

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Step 6 —** Secrets Scope

*Secrets scopes help to store the credentials securely and reference them in notebooks and jobs when required.*

A diagram of a key

Description automatically generated

**Step 7—** Link Databricks Secret Scopes with Azure Key-vault

login to GitHub account – Settings - Developer Settings - Personal access tokens - Tokens (classic) - Generate new token - Generate new token (classic)-give name, expiration ,tick Repo – access token generated – Copy it

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Databricks workspace – settings -  Linked accounts**

A screenshot of a computer

Description automatically generated

Databricks Workspace-Repos- Create – copy the github url here - create

A screenshot of a computer

Description automatically generated

Copy the selected one and paste it in place of databricks – instance i.e, https://<databricks-instance>#secrets/createScope in google chrome..

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Step 7—** Mount azure containers into databricks

Inorder to read data from key-vault using databricks give “Key vault secret user” access.

**A screenshot of a computer

Description automatically generated**

**Data Ingestion using Data Factory**

1.Create a linked service for github(http)

2.Create a linked service for key vault ,and inorder to give access for secrets go to key vault-secret permissions-select all

3.Create an Access policy in Key vault ,so that ADF can access the Key-vault.

A screenshot of a computer

Description automatically generated

4.Create a linked service for adls

5.create a source dataset for csv file using parametrization for relative\_url column

6. create a sink dataset for csv file using parametrization for folder, file columns

7.create a json including all parameters and upload in Adls container.

A screenshot of a computer

Description automatically generated

8.do lookup activity-source dataset-adls with json

A screenshot of a computer

Description automatically generated

9.in for each -settings - items

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

10.To create alerts ,used azure monitor service or directly in ADF ,if there is pipeline failure an email will be send.

A screenshot of a computer

Description automatically generated

**Mount azure containers into databricks:**

**1.Run mount-adls-storage notebook**

**A screen shot of a computer

Description automatically generated**

**Data Ingestion on Databricks Cluster:**

1. **Run the common\_functions, configurations notebooks.**
2. **Run the customer, customerdriver, loantransactions notebooks**
3. **Test the file by running test\_ingestion notebook**
4. **Run database, customer, loantrx notebooks**
5. **Test the file by running test\_ingestion notebook**

**Generate Access Token for Databricks**

Settings – user – Developer – Generate token

A screenshot of a computer

Description automatically generated

Data factory

1.create a pipeline with date as parameter

2. create a **databricks linked service and copy the acees token from databricks**

**3.in settings browse the path of our notebook in databricks**

**4.** **Base parameters section and add one parameter named “p\_file\_date” Click on Add dynamic content i.e, @formatDateTime(date, str\_format)**

Used: *@formatDateTime(****pipeline().parameters.p\_processing\_date****, ‘yyyy-MM-dd’)*

*5.create a* Tumbling window trigger and set the parameter value in trigger i.e, ***@trigger().outputs.windowEndTime End of the window associated with the trigger run.***

A screenshot of a computer

Description automatically generated

**Power BI Visualizations**

1. **Customer Demographics:**
   * **Pie Chart**: Distribution of customers by gender.
   * **Bar Chart**: Number of active vs. inactive customers.
2. **Financial Analysis:**
   * **Bar Chart**: Average monthly salary by customer category.
   * **Line Chart**: Trends in average monthly salary over time.
3. **Loan Analysis:**
   * **Stacked Bar Chart**: Total loan amount by currency type.
   * **Line Chart**: Interest rates over time.
4. **Customer Health Score:**
   * **Histogram**: Distribution of health scores.
   * **Scatter Plot**: Health score vs. current debt.

Challenges:

1.Creating secrets in Azure key-vault -need to give “Key Vault Administrator" ,”key vault secret officer “ role

2.While creating a linked service in ADF for key vault need to give secret name-need to give access in key-vault

Settings - access configuration – vault access policy

3.connecting key-vault to Databricks using secret scope credentials

4.connecting github to Databricks to work on notebooks.

5.tried to create alerts using logic apps- Not possible for free subscription.