**1st End-to-End DE Project on Azure**

**Follow Mr. K Talks YouTube video:** [Watch Video](https://www.youtube.com/watch?v=mECDWTYiKp4&list=PLrG_BXEk3kXx6KE4nBmhf6QwSHMbznP2W)

**Environment Setup**

1. **Create a resource group and the following resources:**
   * Azure Data Factory (ADF)
   * Databricks
   * Key Vault
   * Storage Account (Data Lake)
   * Synapse (Access may be restricted due to trial version)
2. **In SQL Server Management Studio (SSMS):**
   * Restore database: AdventureWorksLT2022 (lightweight version)
   * Create login and user, then update security to db\_datareader

**Data Ingestion**

1. **In ADF:**
   * Install the self-hosted integration runtime.
   * Create a pipeline.

**Sample for one table:**

* + Create a copy address activity.
  + Set up source and sink.
  + Publish.
  + Check Blob Storage to confirm updates.

**Another pipeline for copying all tables:**

* + Execute the following query to get table names:

sql

Copy code

SELECT s.name AS SchemaName,

t.name AS TableName

FROM sys.tables t

INNER JOIN sys.schemas s

ON t.schema\_id = s.schema\_id

WHERE s.name = 'SalesLT'

* + Use a Lookup activity to paste the query.
  + Use a ForEach activity to copy all tables.

**Data Transformation**

1. **In Databricks:**
   * Create a cluster (select the lowest compute node, DVS3\_2).
   * Enable credential passthrough as Databricks is not yet connected to Data Lake.

**Create the first notebook:**

* + Mount Data Lake storage to Databricks using credential passthrough.
  + Mount each folder (do this for all containers: bronze, silver, gold).

python

Copy code

configs = {

"fs.azure.account.auth.type": "CustomAccessToken",

"fs.azure.account.custom.token.provider.class": spark.conf.get("spark.databricks.passthrough.adls.gen2.tokenProviderClassName")

}

dbutils.fs.mount(

source = "abfss://bronze@datalakegen2shai.dfs.core.windows.net/",

mount\_point = "/mnt/bronze",

extra\_configs = configs

)

**Create two additional notebooks for transformation:**

* + Bronze to Silver: Date transformation.
  + Silver to Gold: Column name rename using regex.

1. **Create a linked service in ADF for Databricks:**
   * Connect Databricks to ADF.

**In ADF:**

* + Add a Databricks notebook activity.
  + Configure the connection for each activity.

**Data Loading**

1. **Use Synapse:**
   * Synapse can be serverless (compute only, storage in Data Lake) or dedicated (compute and storage).
   * Create a SQL pool (e.g., gold\_db).
   * Automatically linked to Azure Data Lake (setup occurs upon workspace creation).
2. **Create a linked service for Azure SQL DB (serverless) for stored procedures:**
   * Create a pipeline to create views.
     + Use Get Metadata activity to add dataset from Data Lake (Binary, since it will get the table name).
     + Create a ForEach activity to generate dynamic content (Get Tablenames).output.childItems.
     + Inside ForEach, add a Stored Procedure activity.
     + Configure the procedure with parameter @ViewName set to @item().name (for each loop).

**Alternative Approach:**

* + Perform Data Loading in ADF directly to the Azure SQL database.

**Data Reporting**

In Power BI, create visuals and graphs to draw insights from the Sales data.

**Security and Governance (AAD):**

* Set up security groups (e.g., DE group, DA group) for easier role assignment management.

**Testing**

1. **Create a schedule in ADF:**
   * Add a trigger > Type: Schedule > Timezone > Recurrence: Every day > Time