Data extraction & processing while maintaining security, monitoring & standard on azure platform.

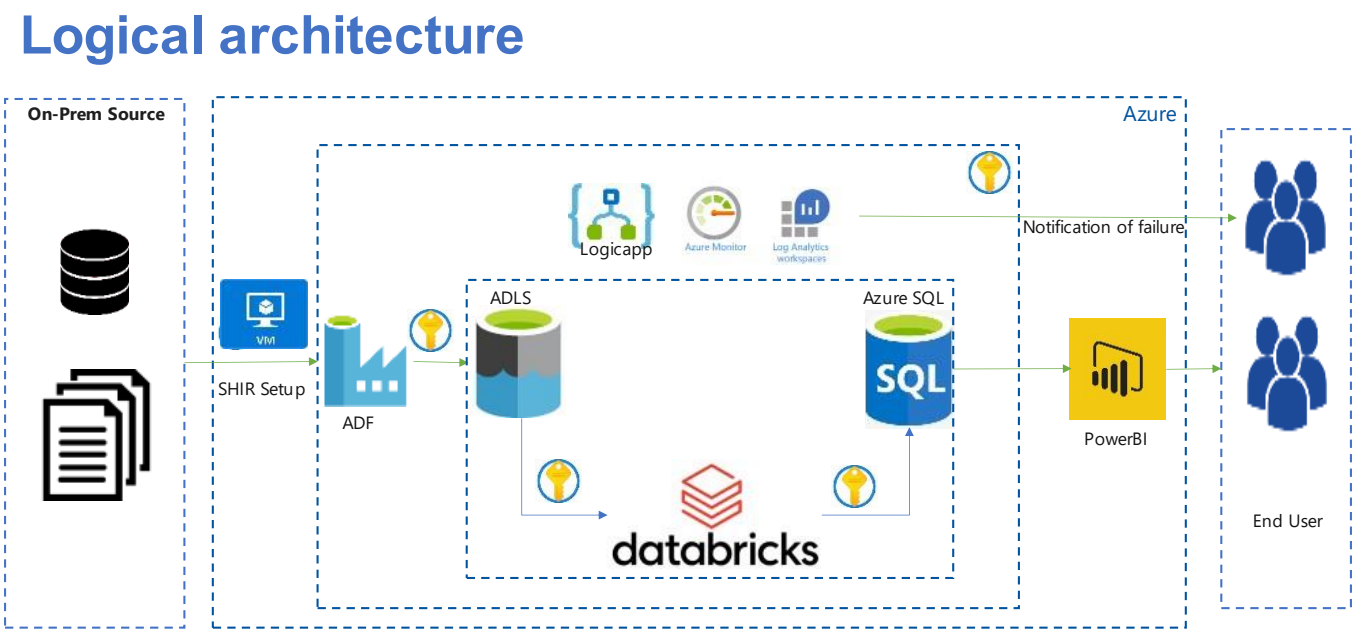
Problem statements: As data engineer usually will get different requirement for data processing & transformation of various sources, Azure offers a wide range of services and tools to help you achieve this.

We received requirement to pick up files from on premises to azure cloud platform & perform checks on data quality while maintaining security, monitoring & data lake transformation layers before loading data in Data warehouse.

Services use:

1. **Azure account:** For using cloud platform
2. **Azure Data lake : -** For storing data & maintaining transformation layer
3. **Azure DataFactory:-** For data ingestion from onprem to azure & data processing
4. **Azure Databricks:** For data transformation & delta tables
5. **Azure Synapse:** For DWH & data modelling purpose
6. **Azure Keyvualt:** For maintaining credentials & secrets at centralize location.
7. **Azure monitor** : For monitor Azure resources
8. **Azure log analytics:** For querying logs
9. **Logic Apps:** For sending email on failure & success
10. **Azure VM :** For setup SHIR

Architecture Diagram:

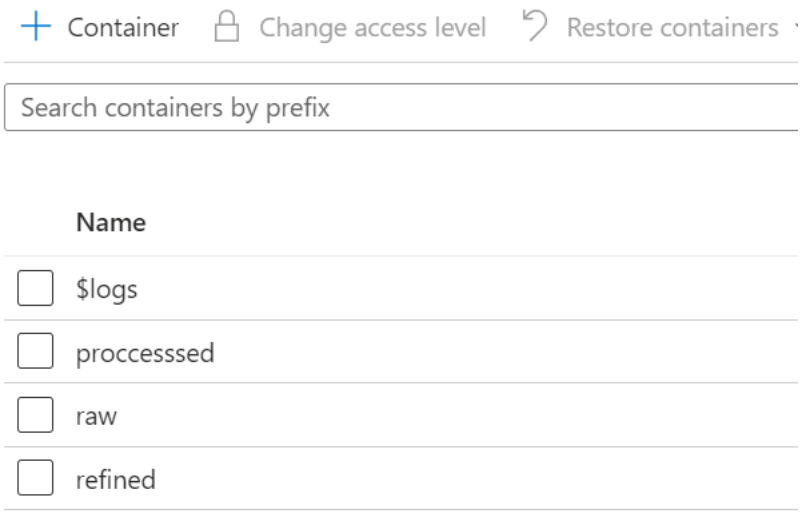


Technical Requirement:

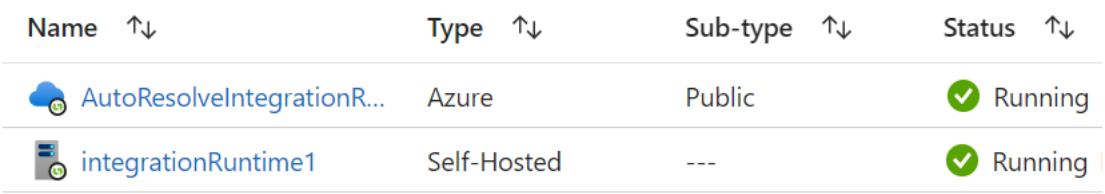
Setup azure environment with listed services

1. **Azure Data lake setup & maintain 3 different layer in ADLS**

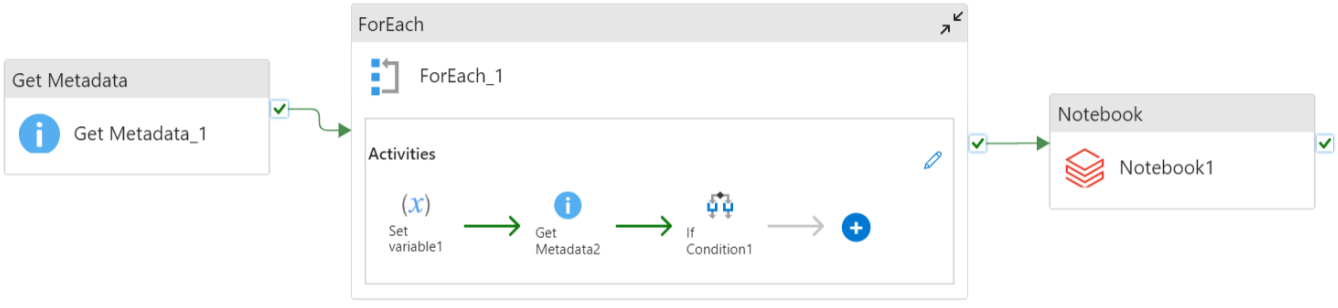
- RAW: Stora raw data from source
- Refined: store cleansed data after processing
- Processed : store transformed data



2. **Setup vm for selfhosted IR** to connect on prem server to pickup file

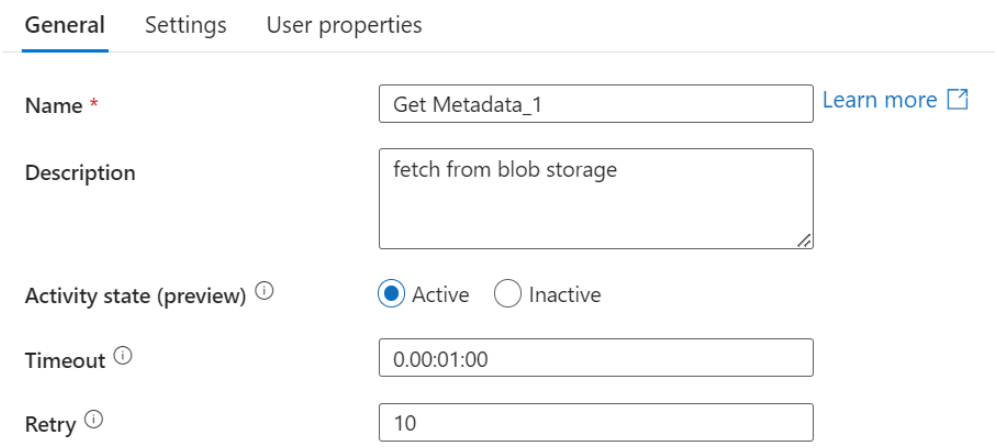


3. **Setup Azure data factory** for data ingestion & proccesessing :



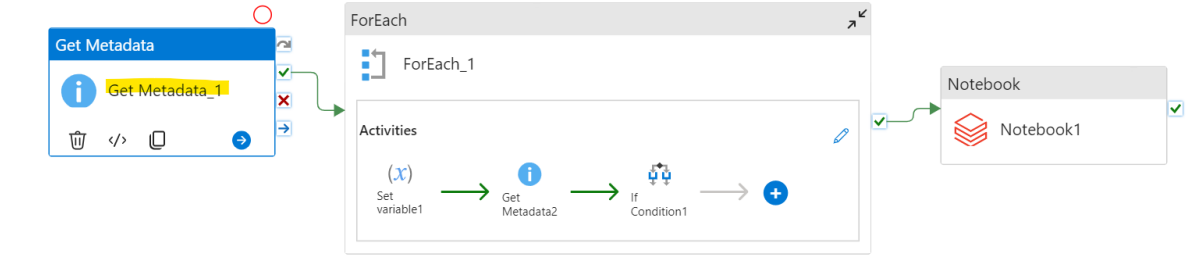
Below checks has been performed in ADF for data consistency:

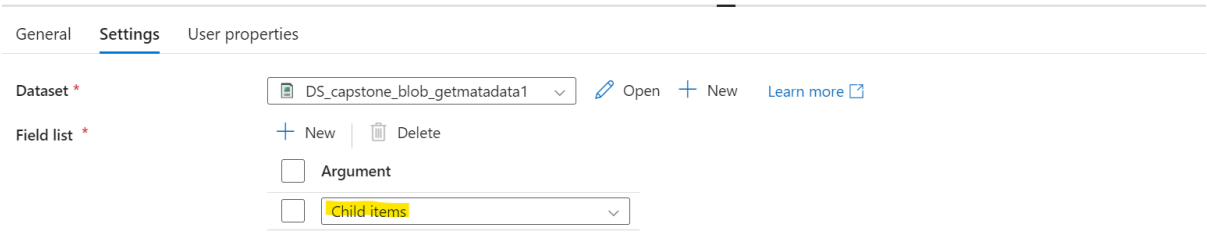
- Check if the file is available in the path. If it's not available, there should be timeout after 1 minute:



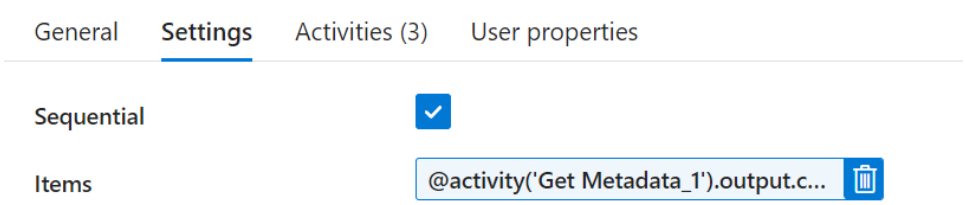
- Check if the file size is greater than 20b or not. If the file size is greater than 20b then needs to process:

We have used 2 get metadata to check first list of files & then stored file name in variable to check size of each file

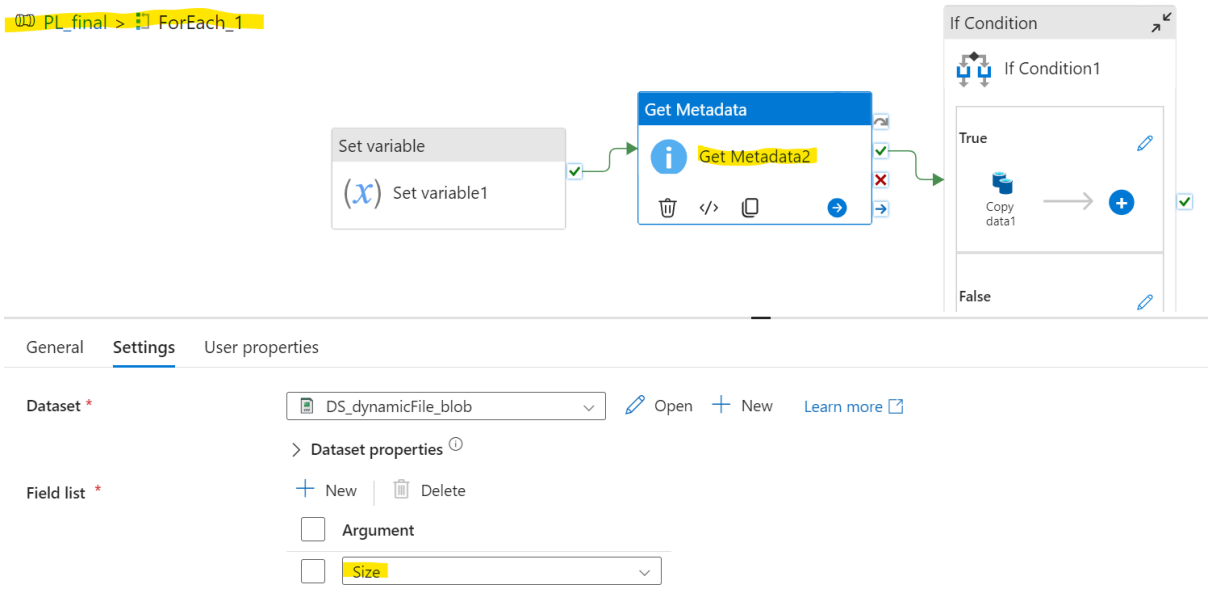




Get metadata output passed to foreach loop activity to run loop for every file

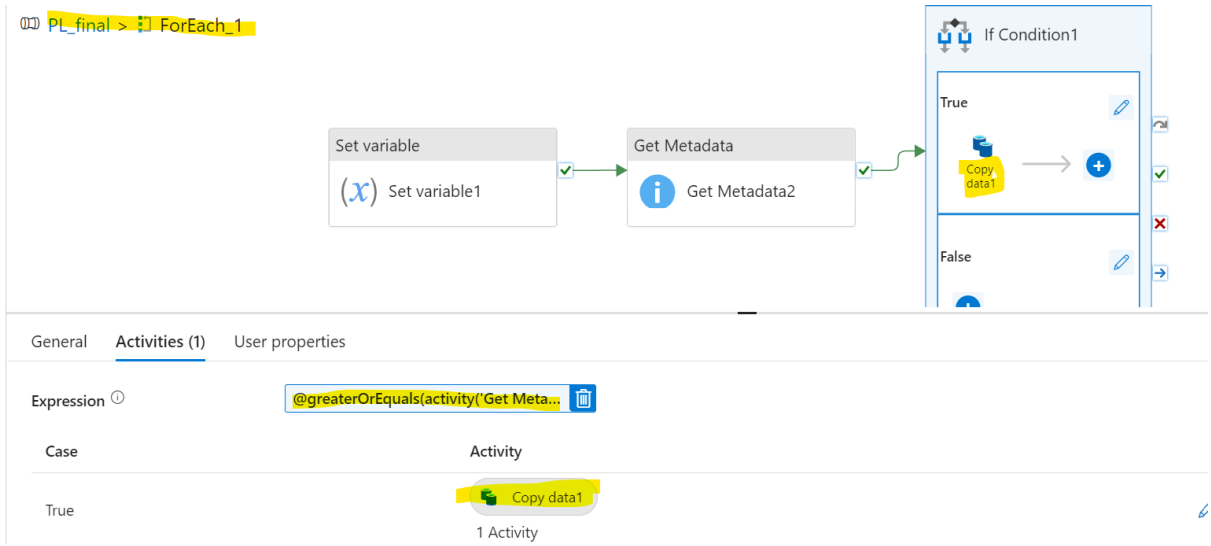


Inside for each loop used variable to store file name & with help of 2nd getmetadata pick size of file.



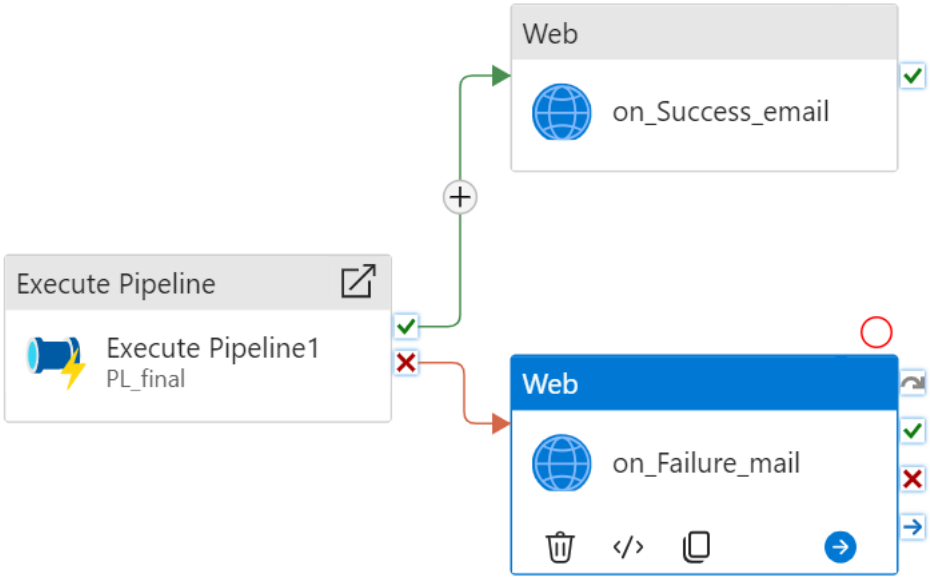
Passed size condition in ifcondition activity to pick files which are greater then 20b

```
@greaterOrEquals(activity('Get Metadata2').output.size,21)
```

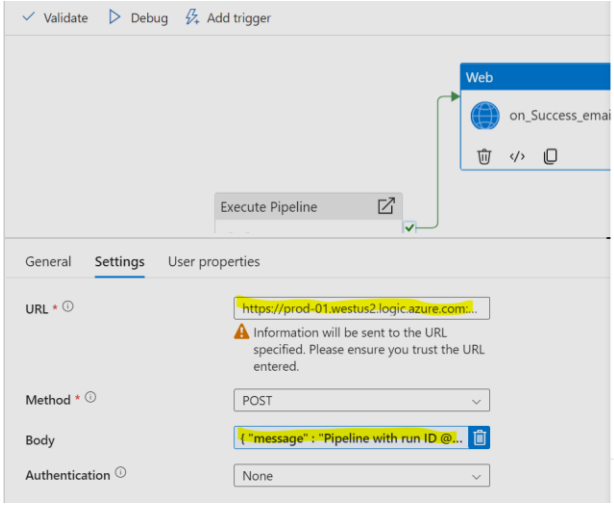


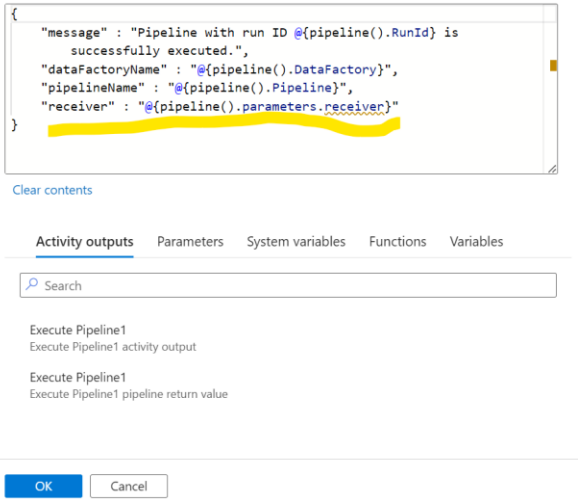
- Setup email configuration on failure & success of pipeline to get notify

Create master pipeline to call logic app to get notified on pipeline failure

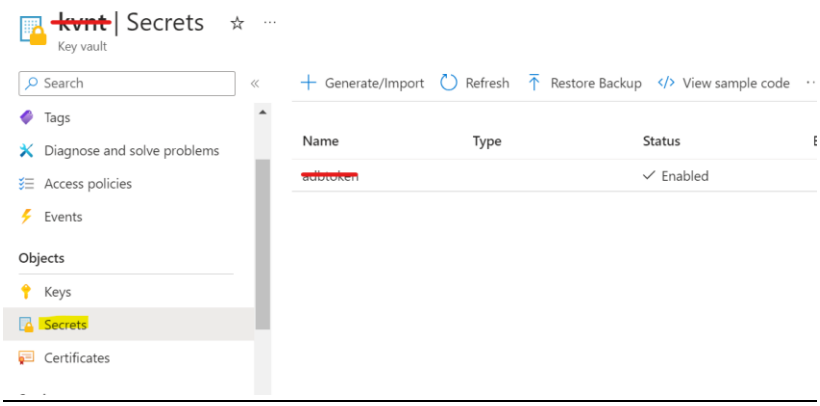


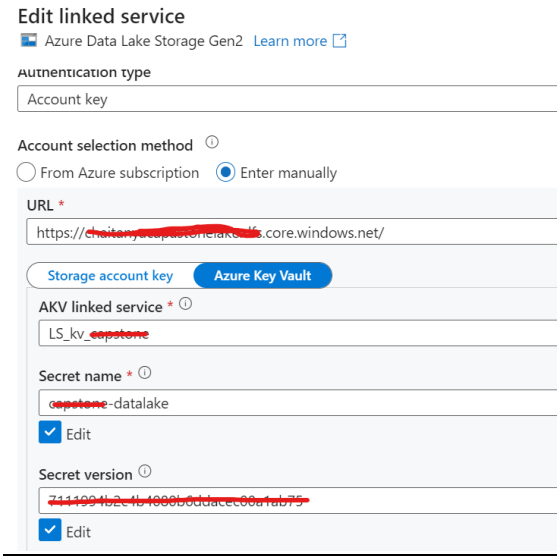
4. **Configured Logicapp to trigger email** & passed logicapp url in be activity:



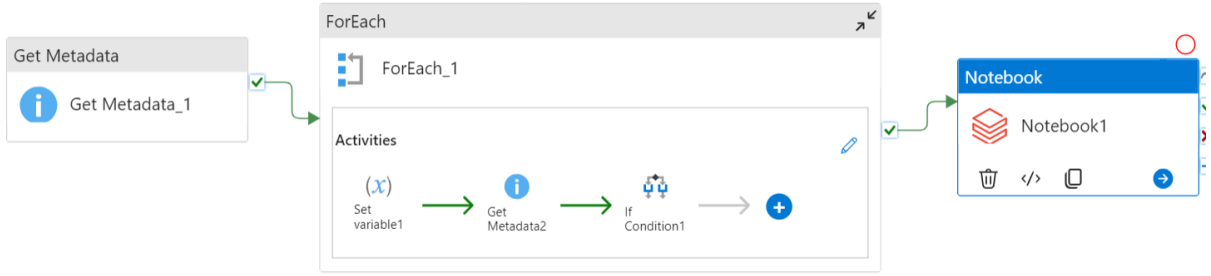


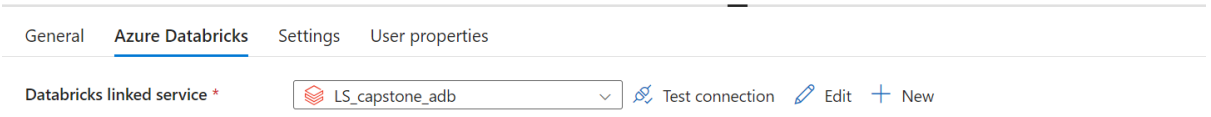
5. Setup keyvault to store credentials





Once all checks has performed databricks notebook will be executed from datafactory pipeline to maintain transformation layer in adls





Setup Azure databricks :

in Databricks couple of transformation applied , I am attaching some of them only as per confidentiality,

- If date is NULL or blank, give default date as '2020-11-28'. Format of date column should be YYYY-MM-DD.
- Remove the entries which has URL field value as 'ERROR'.
- Transform the values of column country with their acronyms. For eg: Austria would be replaced by 'AUST', Belgium by 'BELG' etc. Y

Python

```python
#read the parquet file from adls
source_file="abfss://refined@chaitanyacapastonelake.dfs.core.windows.net"
df_source=spark.read\
    .format('parquet')\
    .option('inferschema',True)\
    .load(source_file)

adf=df_source
```

Cmd 3

```python
# Apply transformations

target_df = adf.withColumn("date", when(col("date").isNull(), "2020-11-28").otherwise(col("date")))
target_df = target_df.filter(col("url") != "ERROR")

```

```python
#run the loop
for rows in table1.find_all('tr'):
  column=rows.find_all('td')
  if len(column)>=2:
    country_name=column[0].text.strip()
    acronym=column[1].text.strip()
    country_acronym[country_name]=acronym

#covert the column 'country' of transformed table to upper-case
target_adf=target_df.withColumn("country",upper(col("country")))

# Create a DataFrame from the acronym data
acronym_df = spark.createDataFrame(country_acronym.items(), ["country", "acronym"])

# Join the original DataFrame with the acronym DataFrame to replace values
adf1 = target_adf.join(acronym_df, "country", "left").select("*")

# Interchange the positions of two columns (e.g., swap "Age" and "Country") and then drop country table
result_df = adf1.withColumnRenamed("country", "temp").withColumnRenamed("acronym", "country").withColumnRenamed("temp", "acronym").drop("acronym")

```

Complete code & ARM template added in github.

Once data transformation done loading data in sql to build pbi dashboard on so can be consumed be end user.

Challenges in implementing solution:

integrating all azure services together in secure way with help of vnet & keyvault so data can be processed from on prem to azure & load successfully in database to consume.

In while process criticality of data is key things to protect so overall flow will work smoothly as per architecture diagram.

Business Benefits:

In all project as data engineering will play a key role to perfoem various data operation & maintain security so its help of this high level flow can implement security best practices can be followed.

Github link:

https://github.com/deeksharm/azuredataengineer