Extract and process data while maintaining security, monitoring and standards on the Azure Cloud platform.

Problem Statement: Data engineers usually receive requirements for data processing & transformation from different sources, Azure offers a wide range of services & tools to help you achieve this.

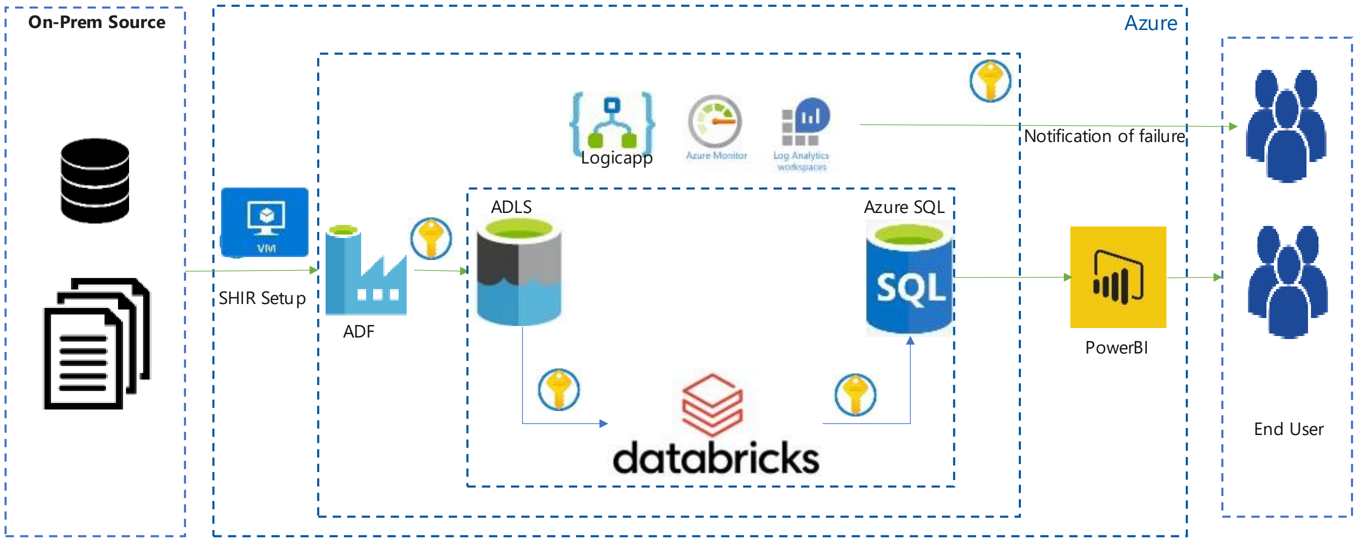
We received a requirement to ingest files from on-premises to azure cloud platform & perform data quality checks while maintaining security, monitoring & data lake transformation layers before loading data into data warehouse.

Services use for Transformation

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:

Logical architecture



Technical requirements:

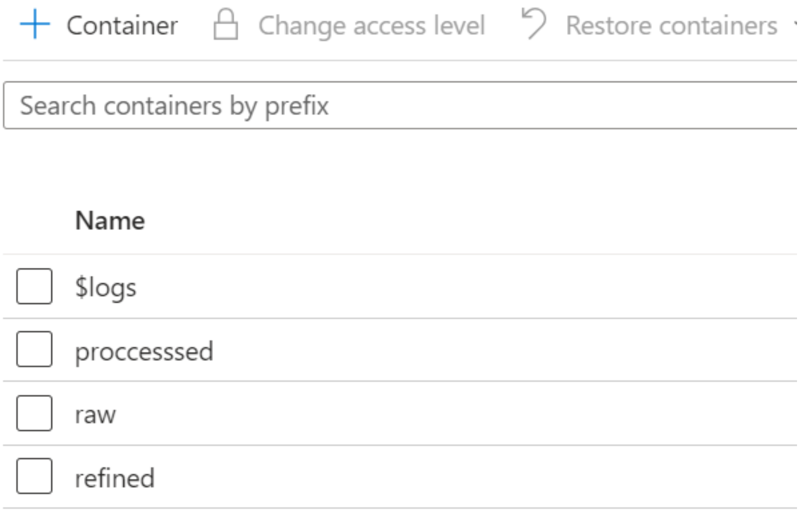
Set up Azure environment with listed services

1.Azure data lake setup & maintain 3 different layers in ADLS

- RAW: Store raw data from source

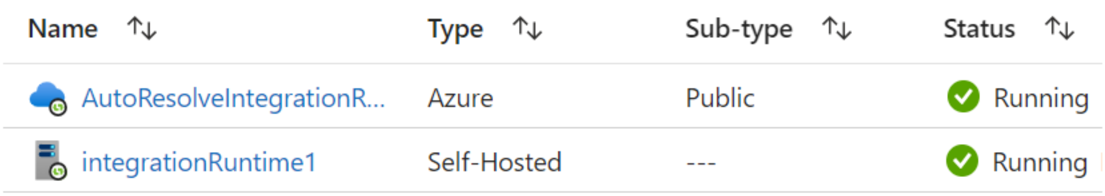
- Refined: store cleansed data after processing

- Processed: store transformed data



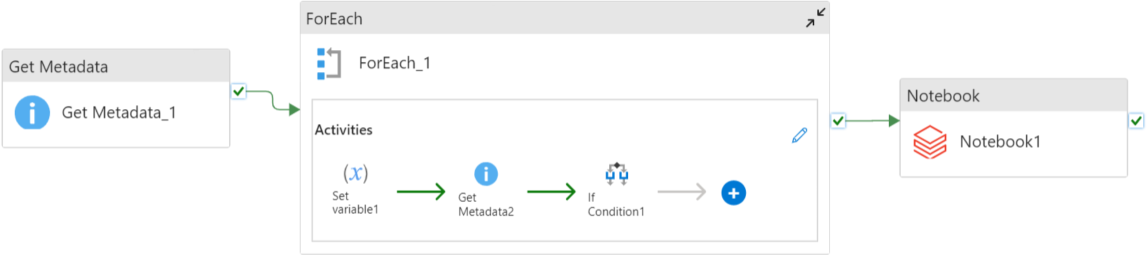
	Name
□	$logs
□	proccesssed
□	raw
□	refined

2. Set up VM for self-hosted IR to connect on prem server to pick up file



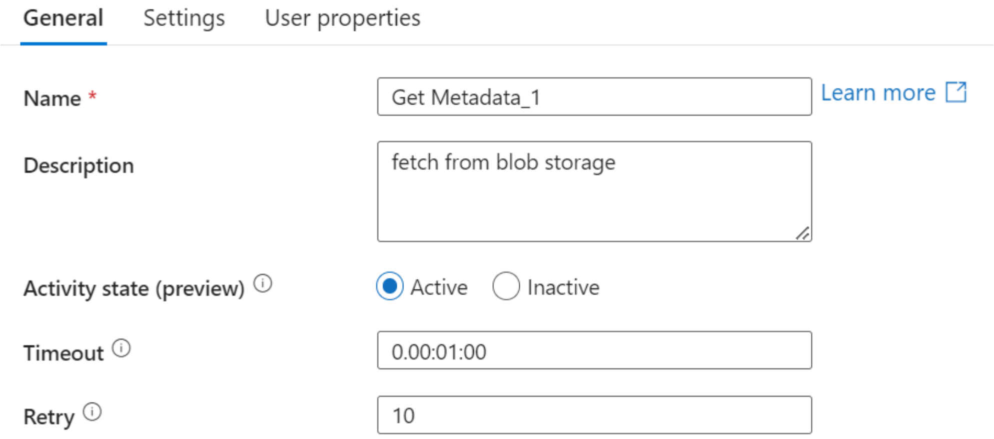
Name ↑↓	Type ↑↓	Sub-type ↑↓	Status ↑↓
AutoResolveIntegrationR...	Azure	Public	✓ Running
integrationRuntime1	Self-Hosted	---	✓ Running

3. Set Up Azure Data Factory



The following checks have been performed in ADF for data consistency:

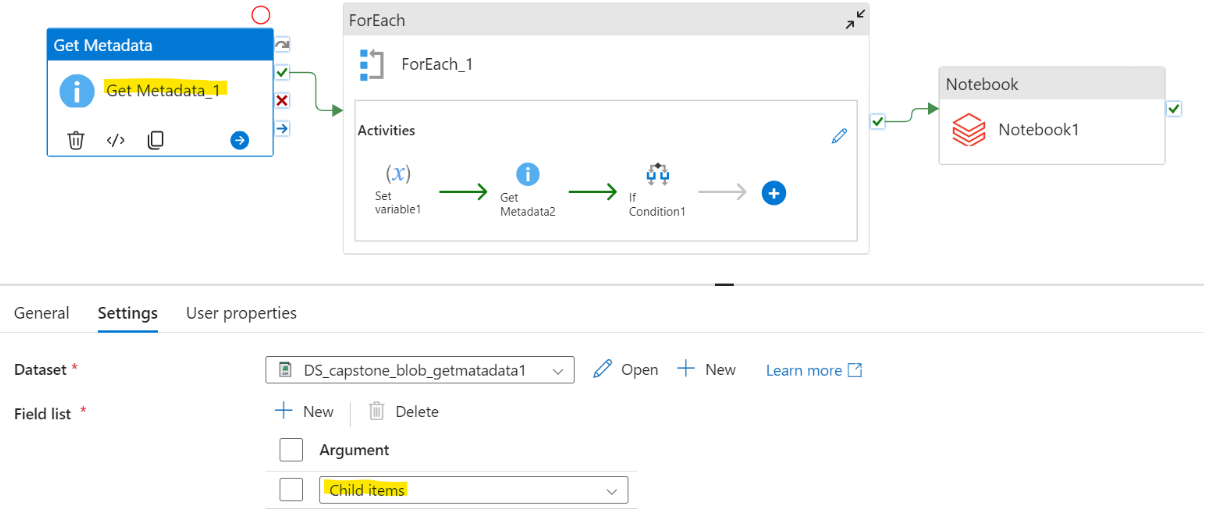
- Check that the file is available in the path. If it's not, there should be a timeout after 1 minute:



Name *	Get Metadata_1
Description	fetch from blob storage
Activity state (preview)	● Active ○ Inactive
Timeout	0.00:01:00
Retry	10

- Check whether the file size is greater than 20b or not. If the file size is greater than 20b then it needs to be processed:

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



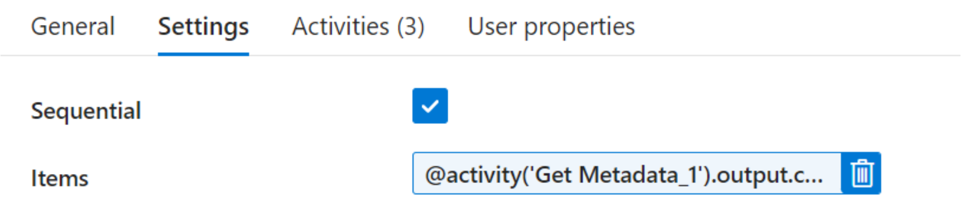
Dataset * DS_capstone_blob_getmatadata1 Open New Learn more

Field list * New Delete

☐ Argument

☐ Child items

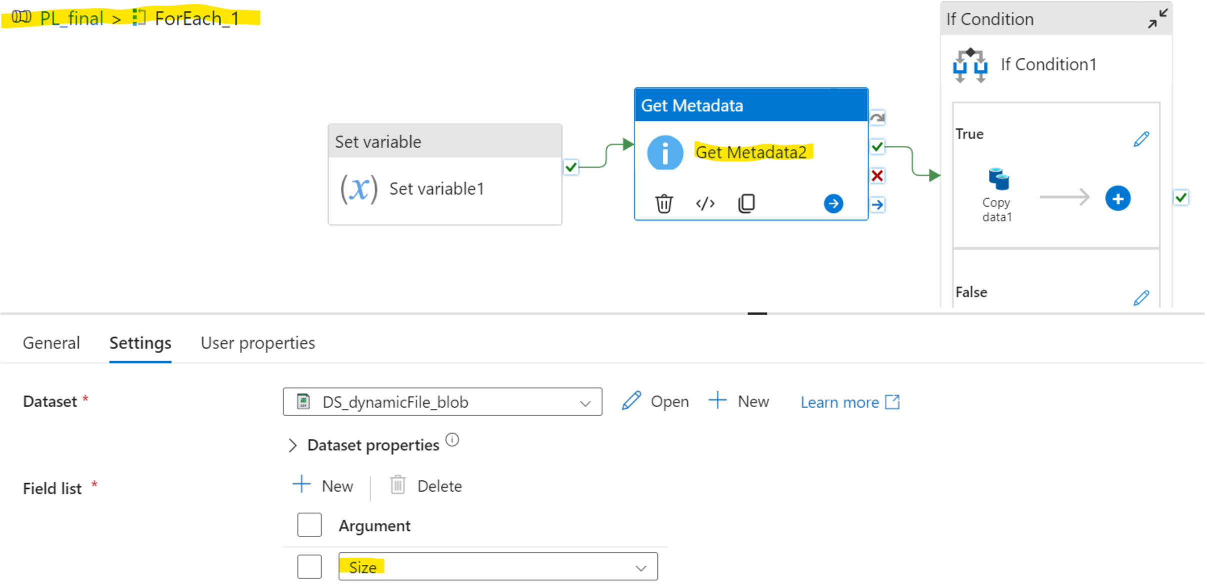
Get metadata output passed to foreach loop activity to run loop for every file



Sequential ☑

Items @activity('Get Metadata_1').output.c...

Inside for each loop used variable to store filename & using 2nd getmetadata get file size.



Dataset * DS_dynamicFile_blob Open New Learn more

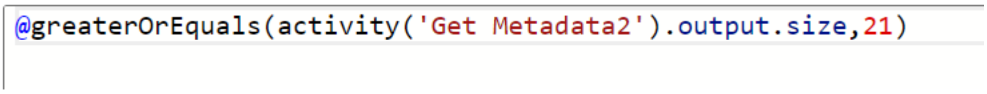
> Dataset properties

Field list * New Delete

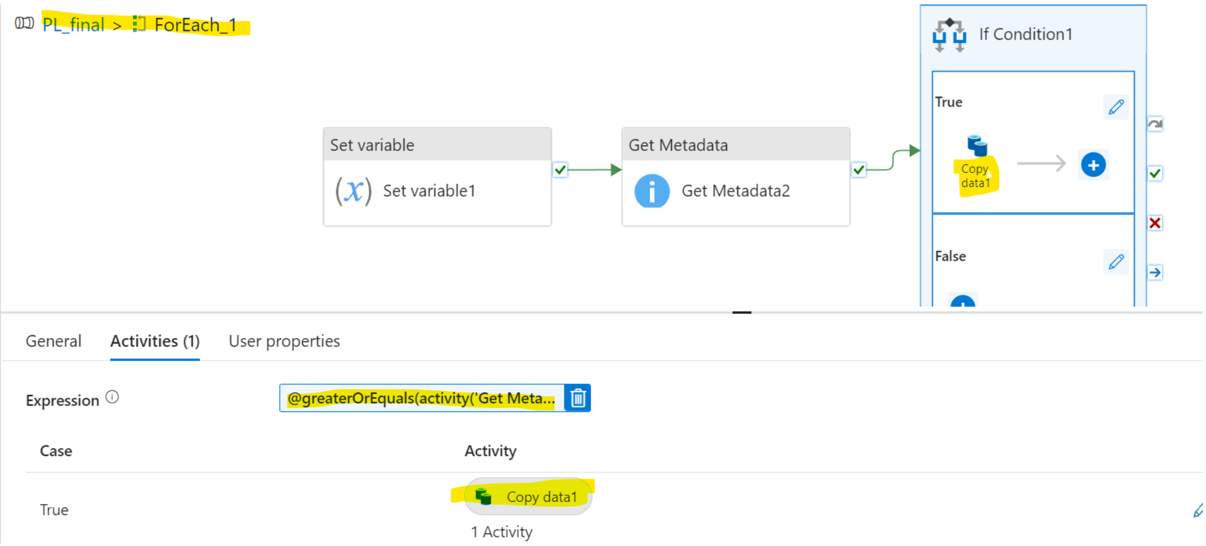
☐ Argument

☐ Size

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



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

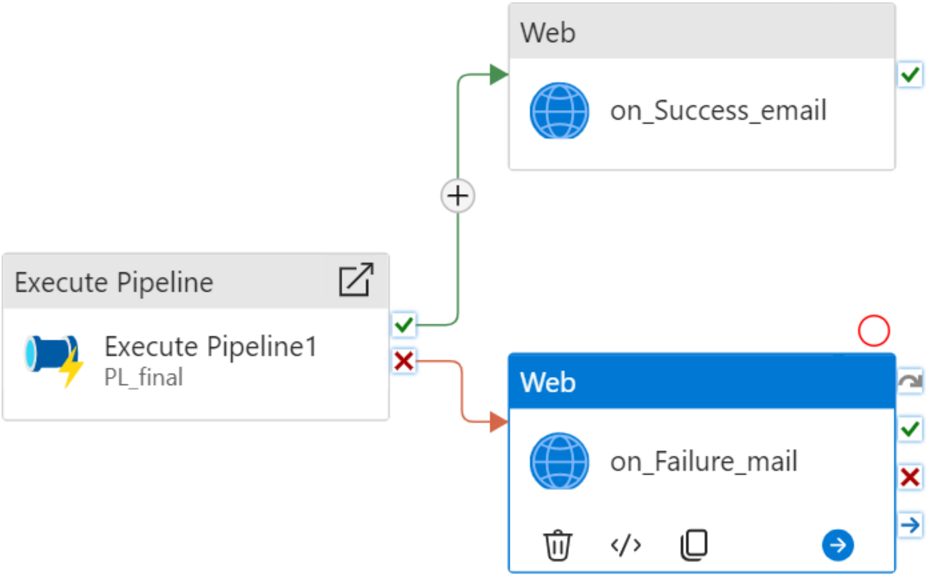


Expression: @greaterOrEquals(activity('Get Meta...

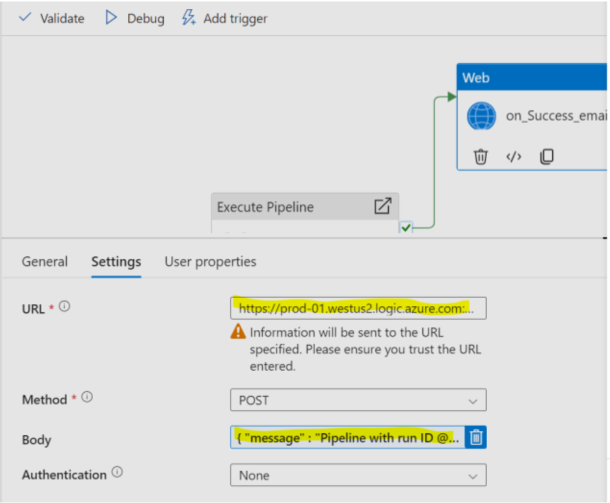
Case	Activity
True	Copy data1 1 Activity

- 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:



General | **Settings** | User properties

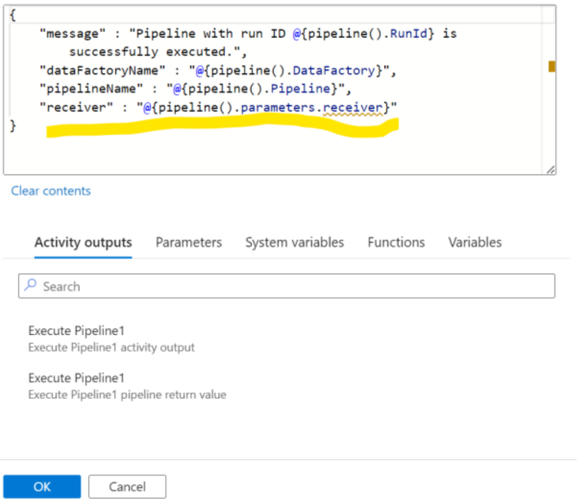
URL * https://prod-01.westus2.logic.azure.com:...

⚠ Information will be sent to the URL specified. Please ensure you trust the URL entered.

Method * POST

Body { "message" : "Pipeline with run ID @...

Authentication None

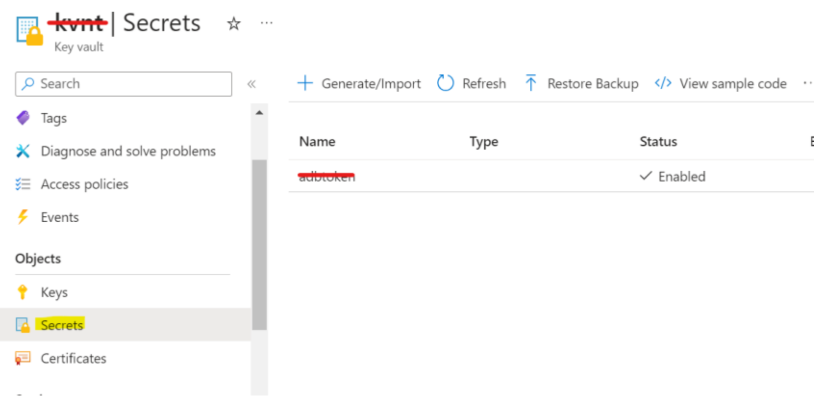


```
{
    "message" : "Pipeline with run ID @{pipeline().RunId} is
        successfully executed.",
    "dataFactoryName" : "@{pipeline().DataFactory}",
    "pipelineName" : "@{pipeline().Pipeline}",
    "receiver" : "@{pipeline().parameters.receiver}"
}
```

Execute Pipeline1
Execute Pipeline1 activity output

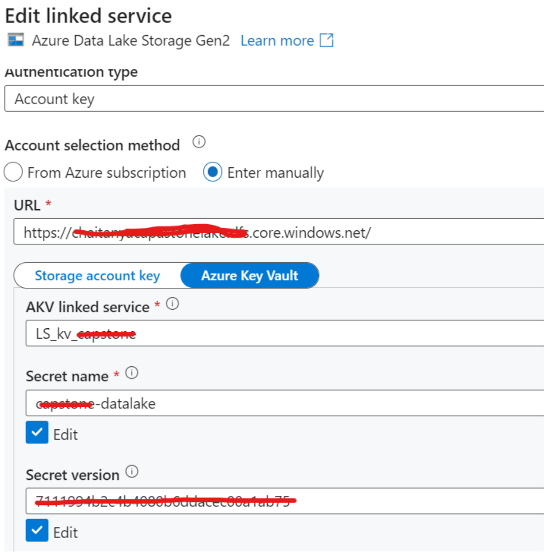
Execute Pipeline1
Execute Pipeline1 pipeline return value

5. Setup keyvault to store credentials



[redacted] | Secrets ☆ ···
Key vault

Name	Type	Status	E
[redacted]		✓ Enabled	



Edit linked service
Azure Data Lake Storage Gen2 Learn more

Authentication type
Account key

Account selection method ⓘ
○ From Azure subscription ● Enter manually

URL *
https://[redacted].core.windows.net/

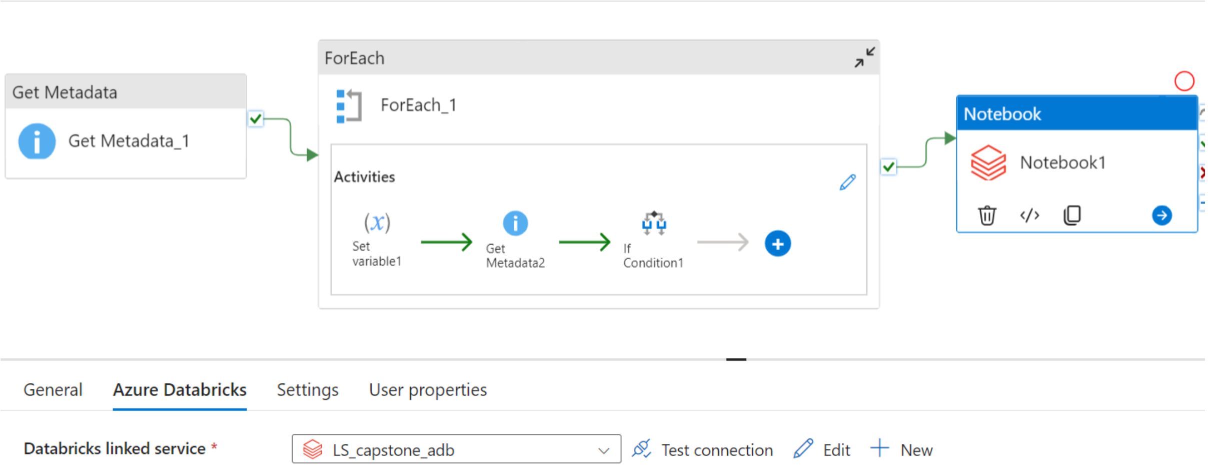
Storage account key | **Azure Key Vault**

AKV linked service * ⓘ
LS_kv_[redacted]

Secret name * ⓘ
[redacted]-datalake
☑ Edit

Secret version ⓘ
[redacted]
☑ Edit

Once all the checks have been performed, the DataFactory pipeline is run to maintain the transformation layer in ADLS.

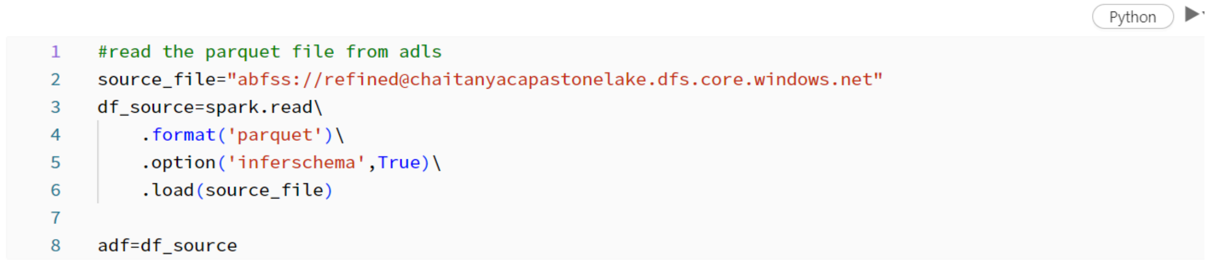


Databricks linked service * LS_capstone_adb

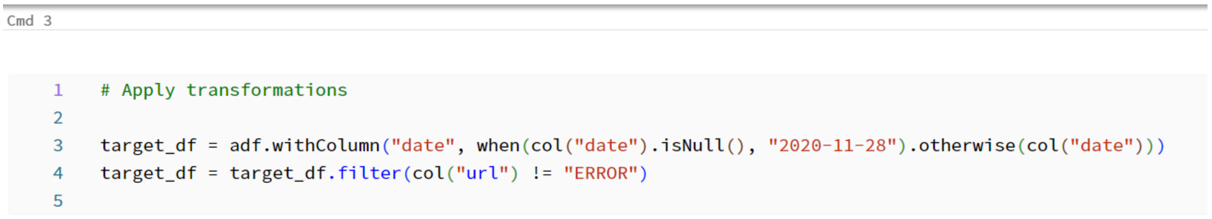
Setup Databricks :

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

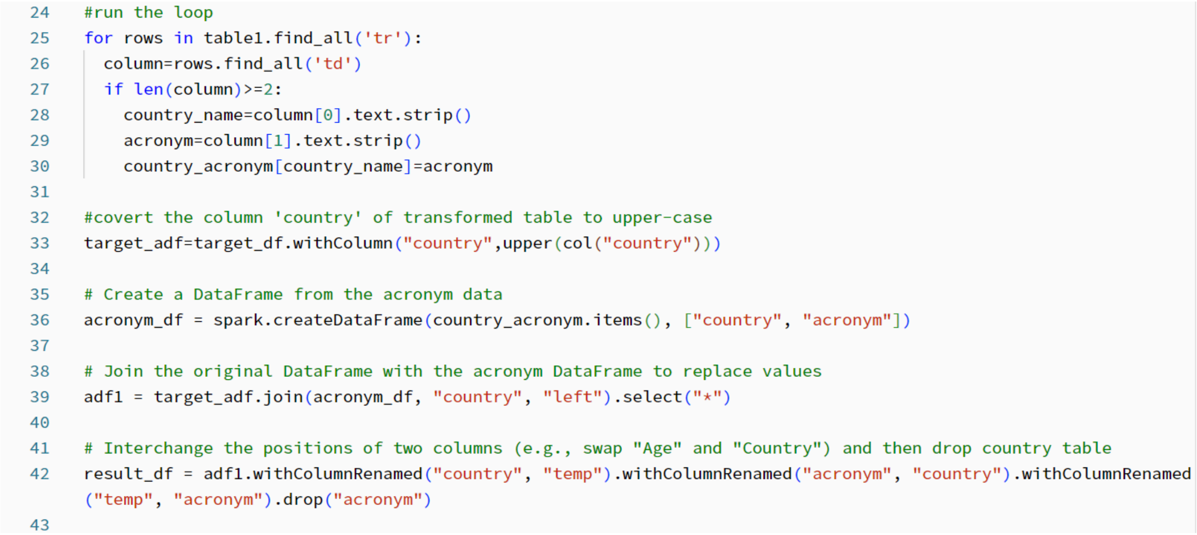
- If date is NULL or blank, give default date as ‘2020-11-28’. Format of date column should be YYYY-MM-DD.
- Remove the entries which has URL field value as ‘ERROR’.
- Transform the values of column country with their acronyms. For eg: Austria would be replaced by ‘AUST’, Belgium by ‘BELG’ etc.



```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
```



```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")

```



```
#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

Integrate all Azure services together in a secure way with the help of VNET & Keyvault, so data can be processed from on prem to Azure & loaded successfully into database for consumption.

During the process, the criticality of data is the most important thing to protect so that the whole flow works smoothly as per the architecture diagram.

Benefits

In all projects 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.