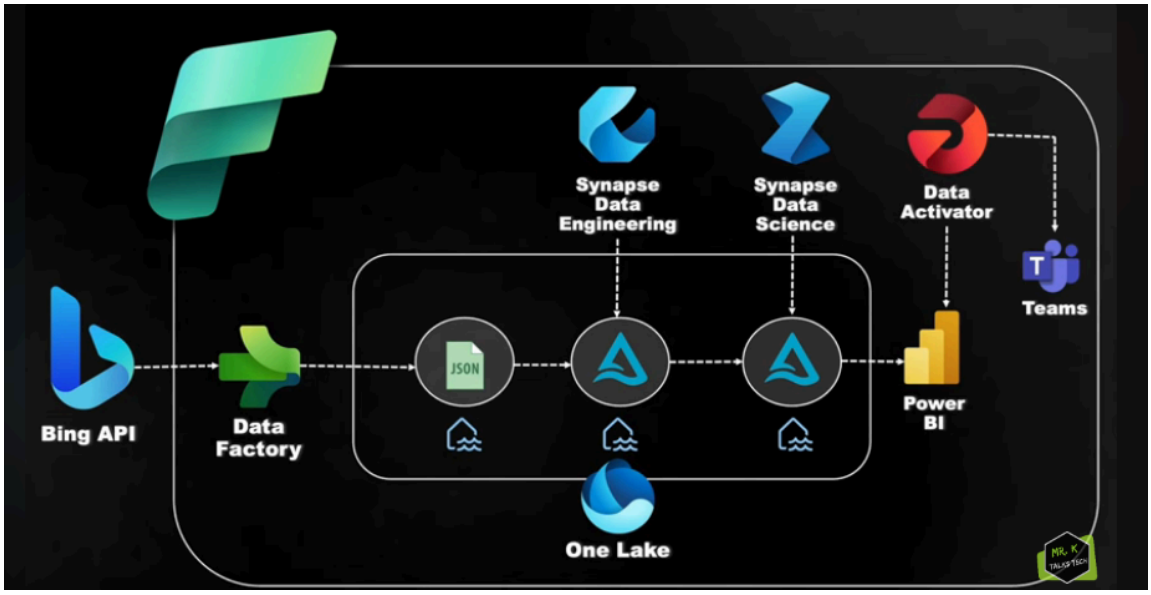
Project Overview / Architecture

1. We'll be using Bing search data to build BI dashboards using Fabric
2. Our data source will be Bing API
3. Here in this proj..we inject data from Bing API via Synapse Data factory and store it in One lake
4. Initially the data would be in Json format and we clean the data using synapse data engineering and store them in lake DB in the delta table format
5. We use this clean data and perform NLP sentiment analysis using Synapse Data Science and store them in Lake DB
6. This data will be the final data to make PowerBI Reports…and we use data activator for analytic purpose ..

   In **Azure Fabric**, the **Data Activator** is a feature used to trigger workflows, automation, and event-driven actions based on data changes or specific conditions in a data store. It allows you to monitor data streams and react to changes, essentially acting as a bridge between your data and the applications that need to react to it.

7. Overall Architecture

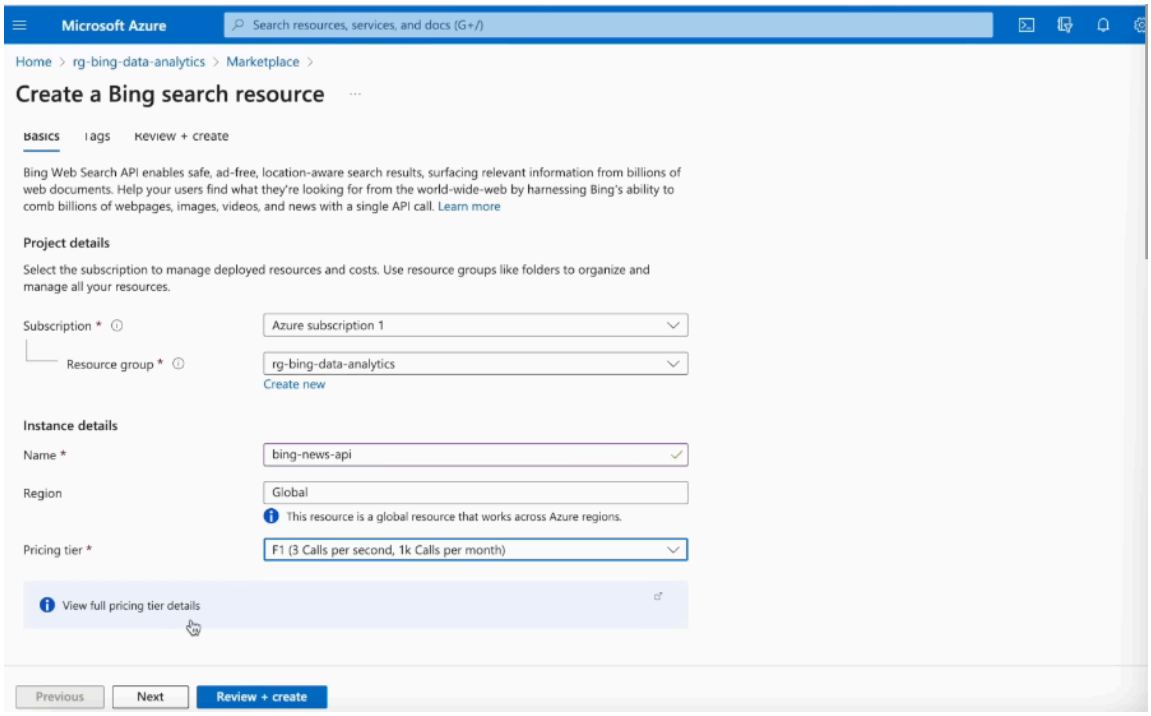


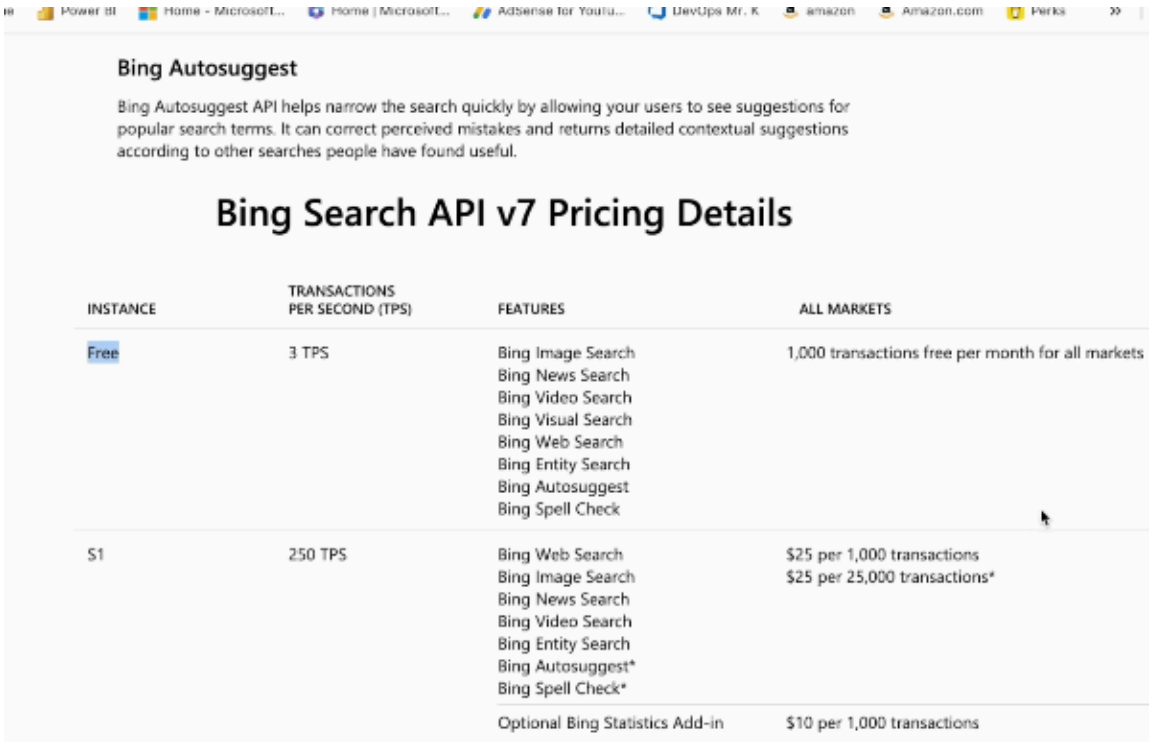
8. 

Environment Setup

1. We create a Resource group and after that we create a data resource for Bing API

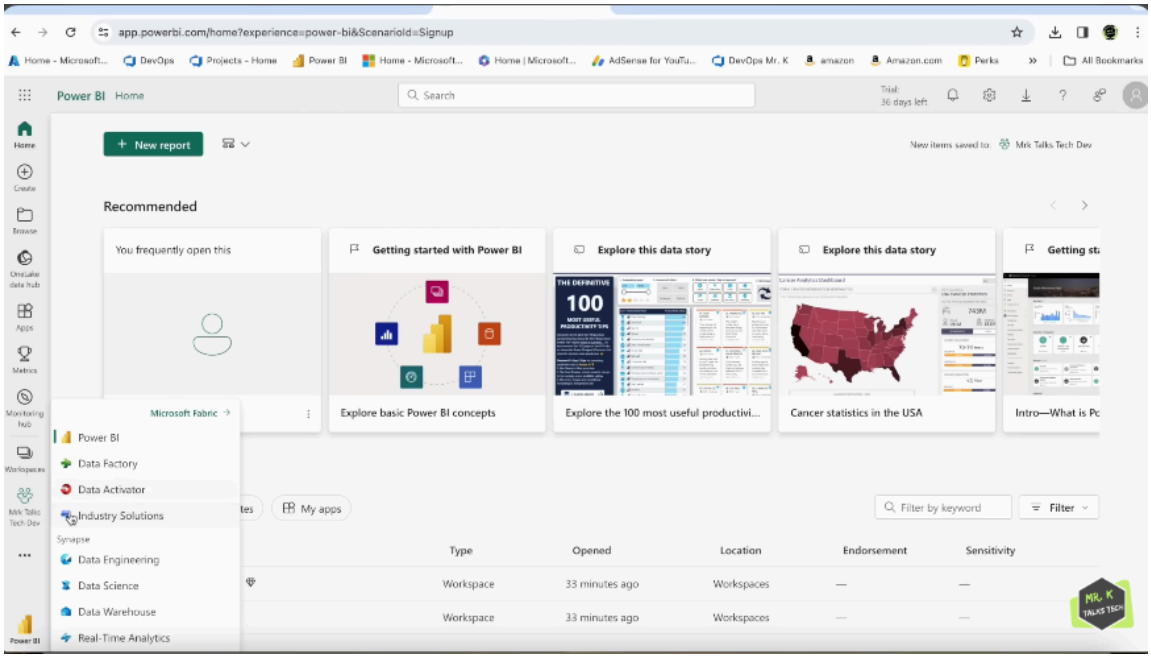
2. In marketplace search for bing and select bing v7



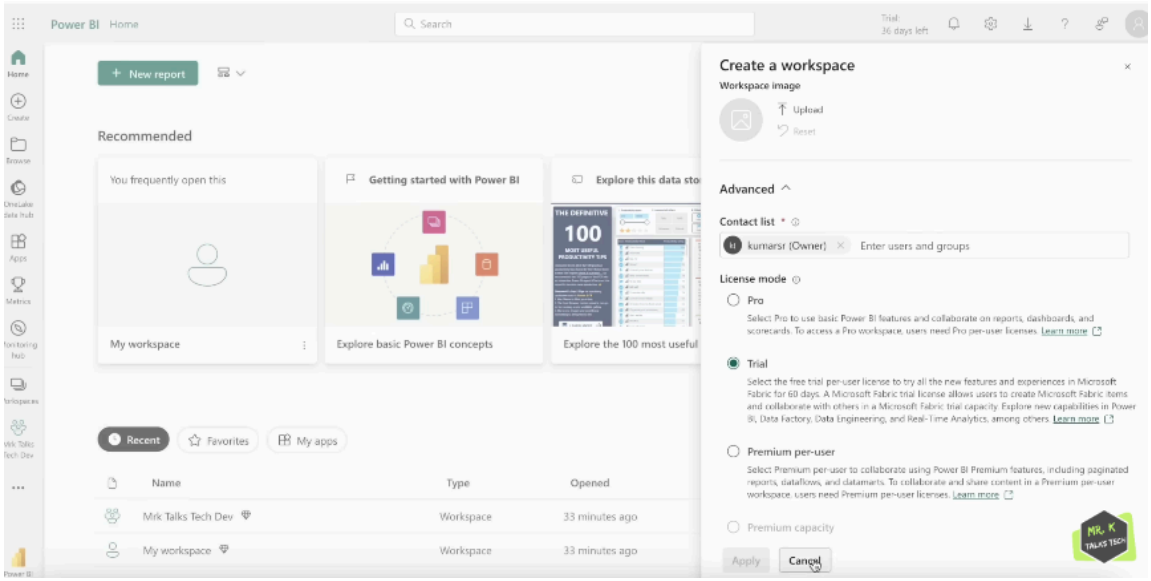


3. Here we have created the bing search API
4. We can connect to this API via endpoints and ingest the news data
5. And we have successfully created a resource and bing search API

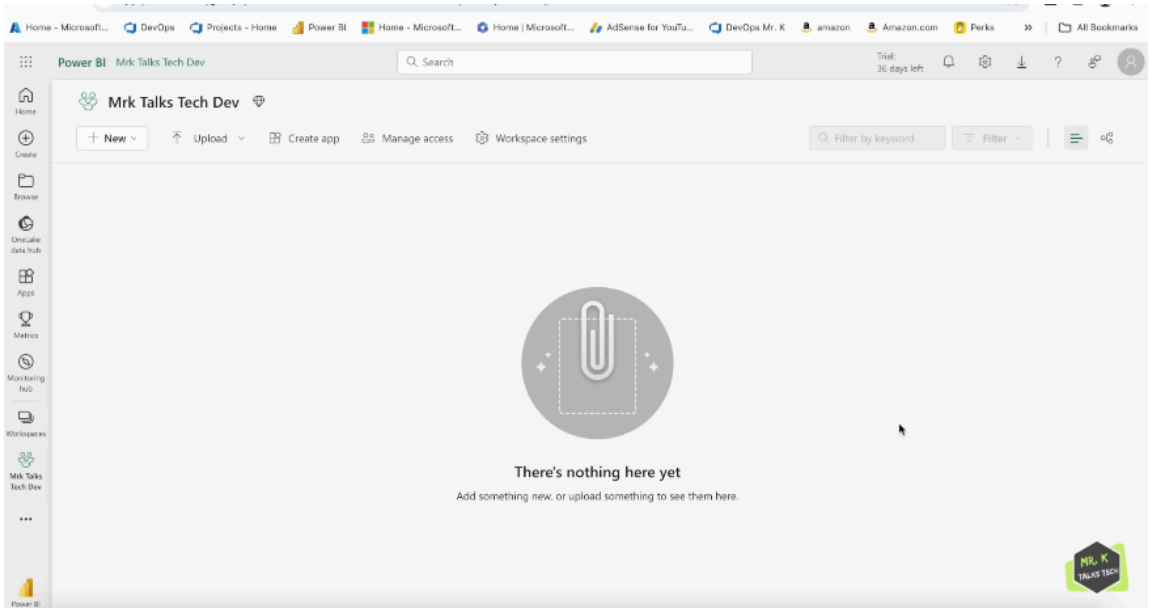
6. Our current fabric workspace



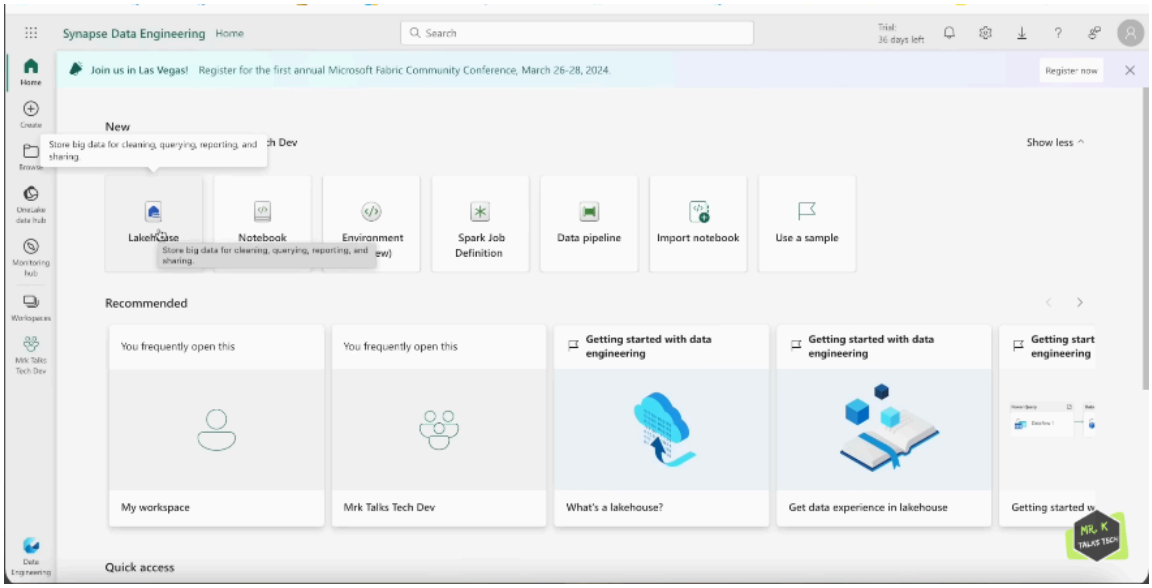
7. Inside the fabric we need to create a new workspace for our project and enable fabric access



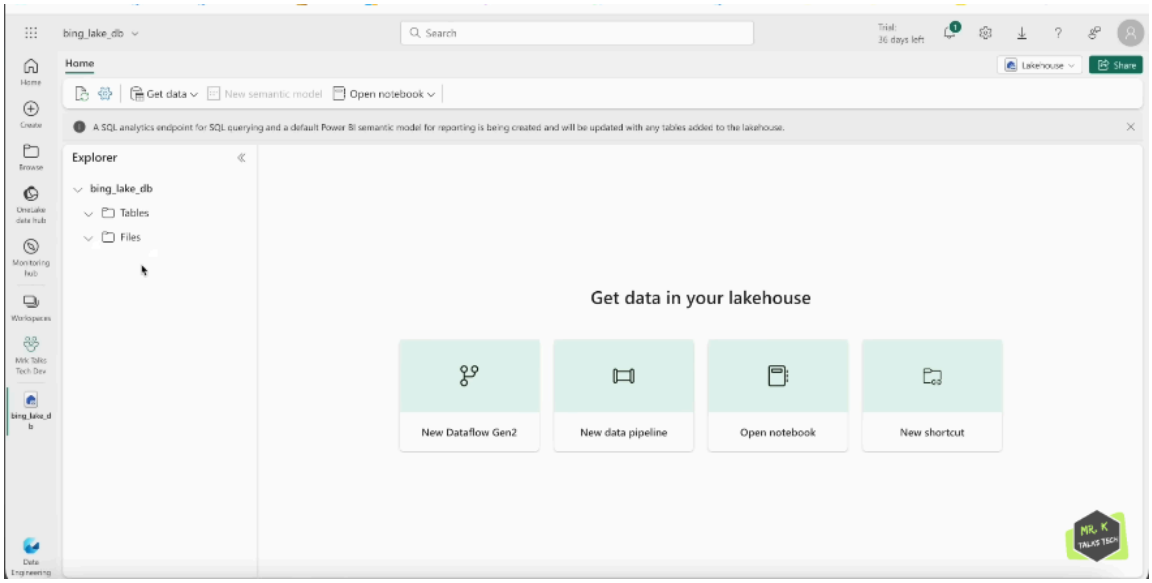
8. This is the workspace that we will be using for fabric proj…wer we can get access to fabric features



9. Go to Synapse data engineering comp and create a lakehouse

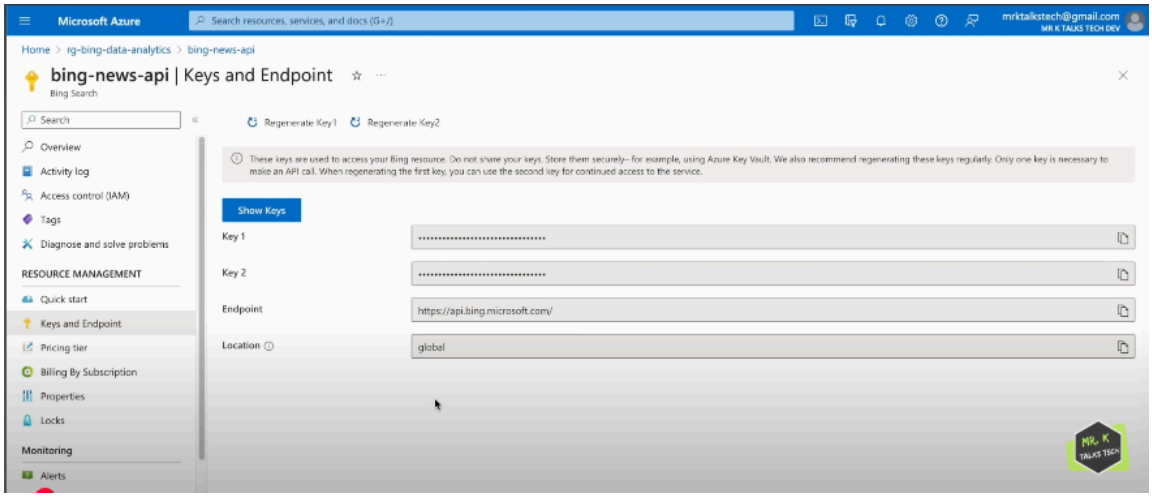


10. After creating the lakehouse

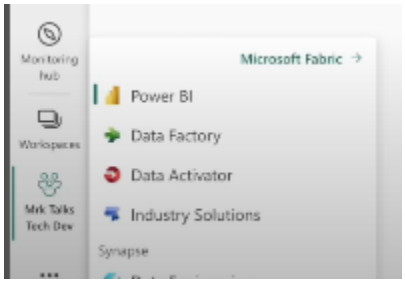


Data Ingestion

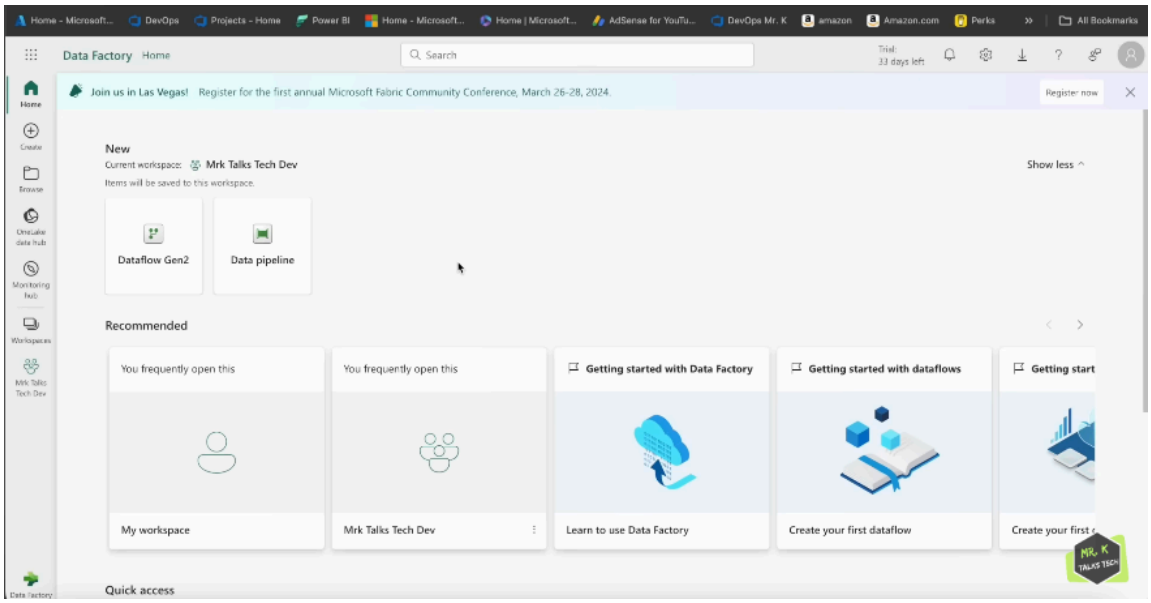
1. Here we will be using ADF for ingesting our data
2. Before that we to get the data we use the endpoints from our API

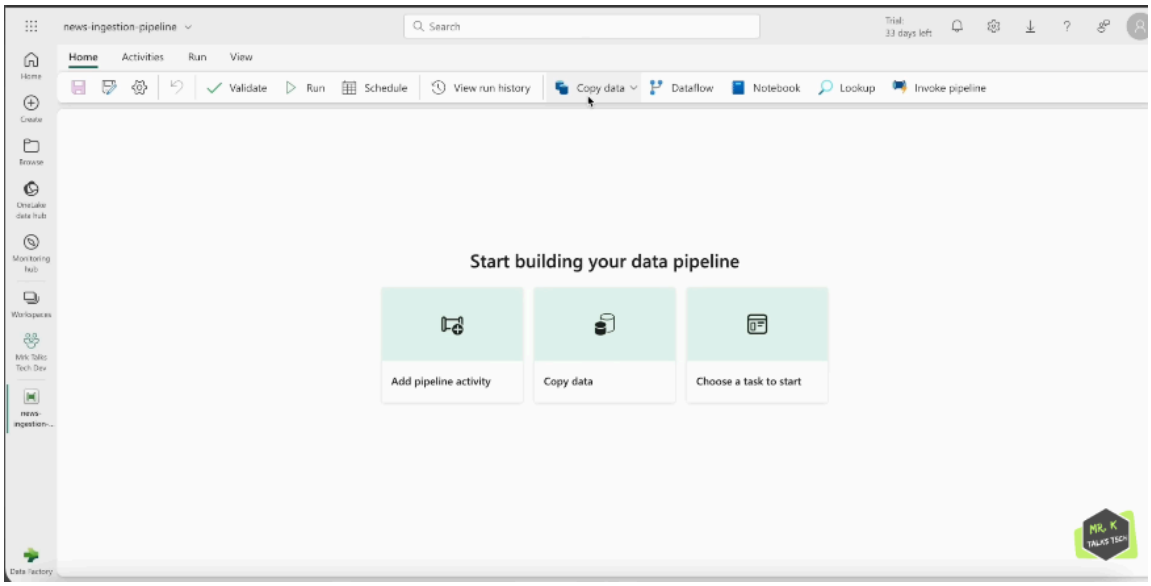


3. Go to the lakehouse DB and from bottom left menu choose data factory

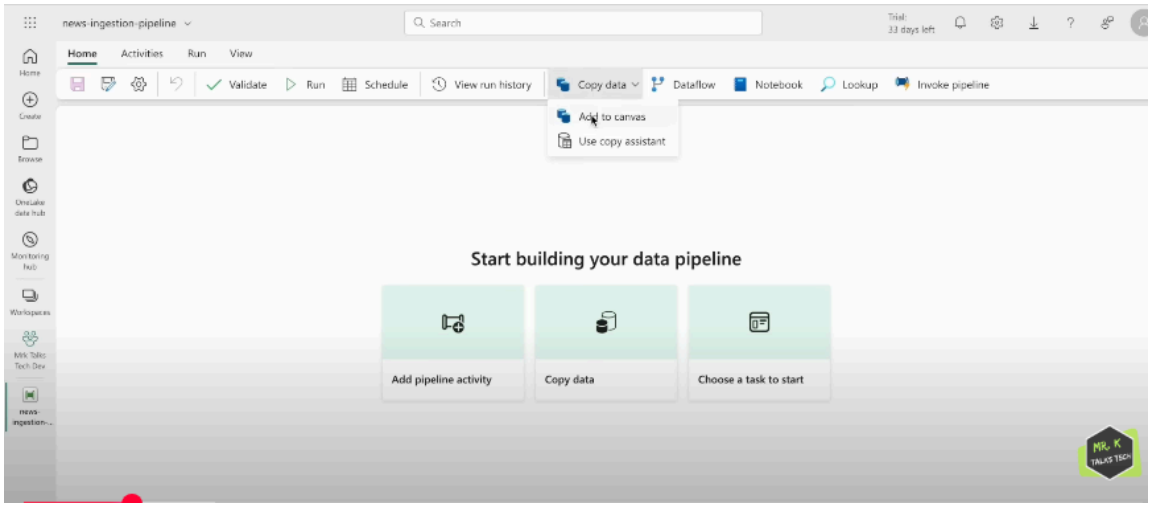


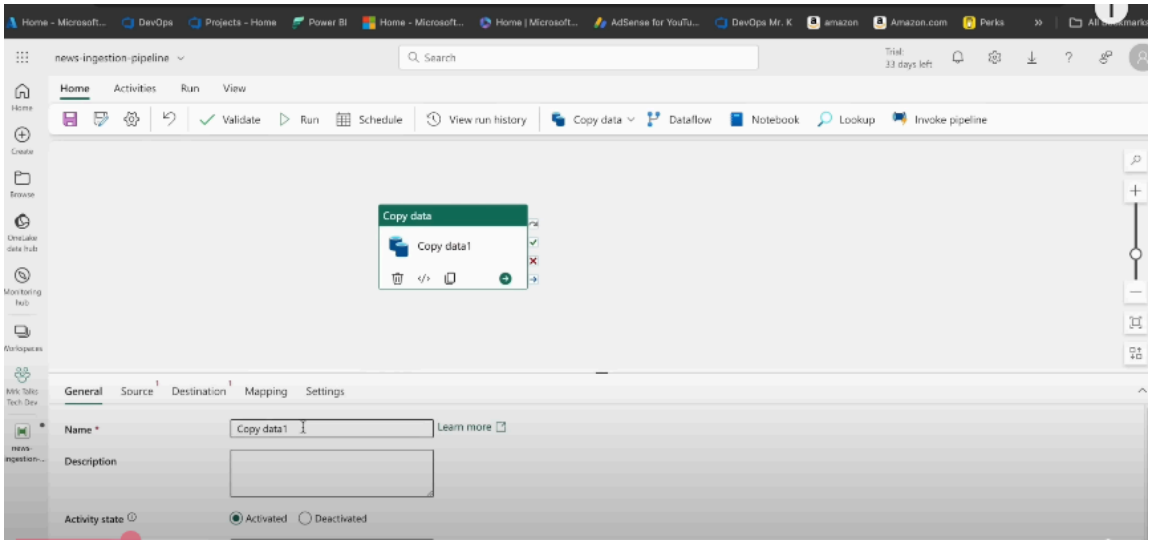
4. Click on data pipeline and create it a name news_ingestion





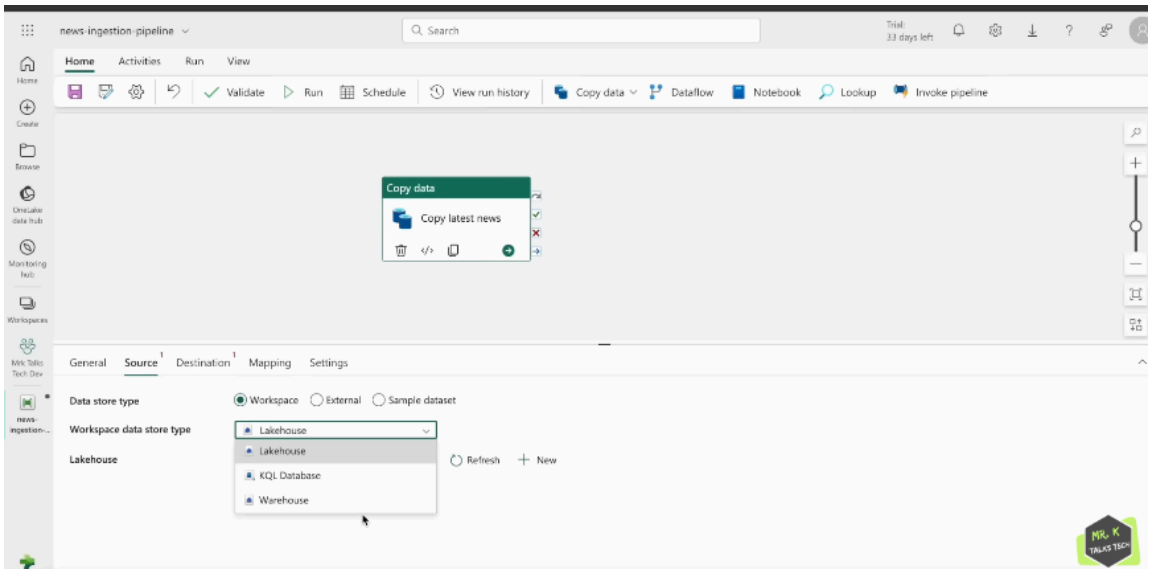
5. Now click on the copy data activity



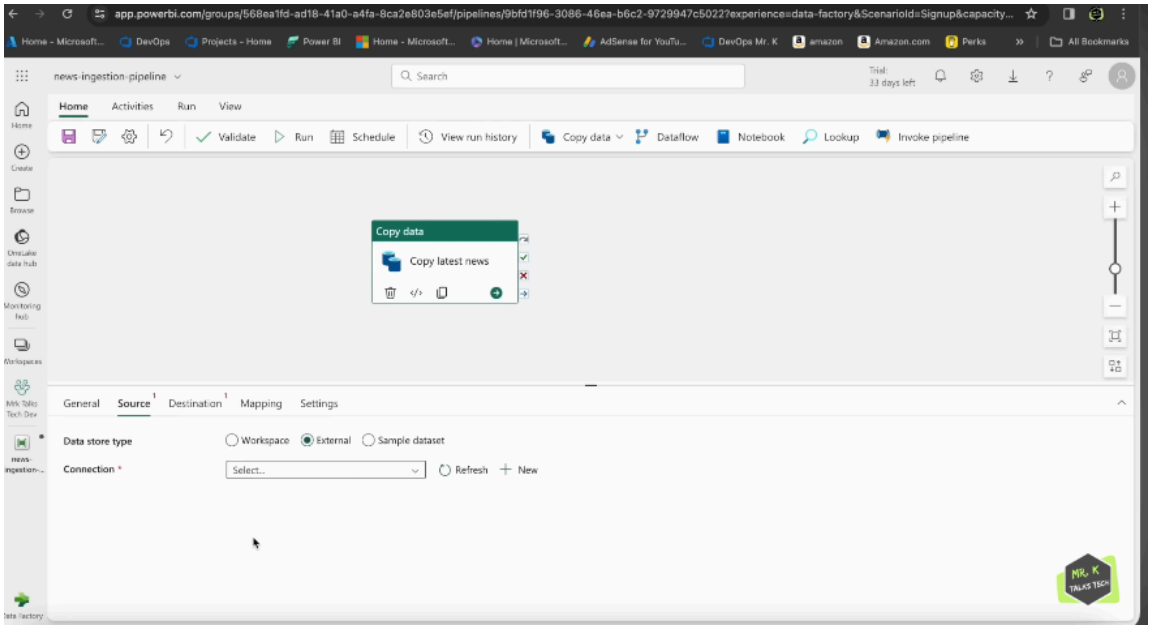


6. Give a name to our copy activity and provide the source

7. Here if the source is “fabric lakehouse or KQL or warehouse “ then we can choose workspace

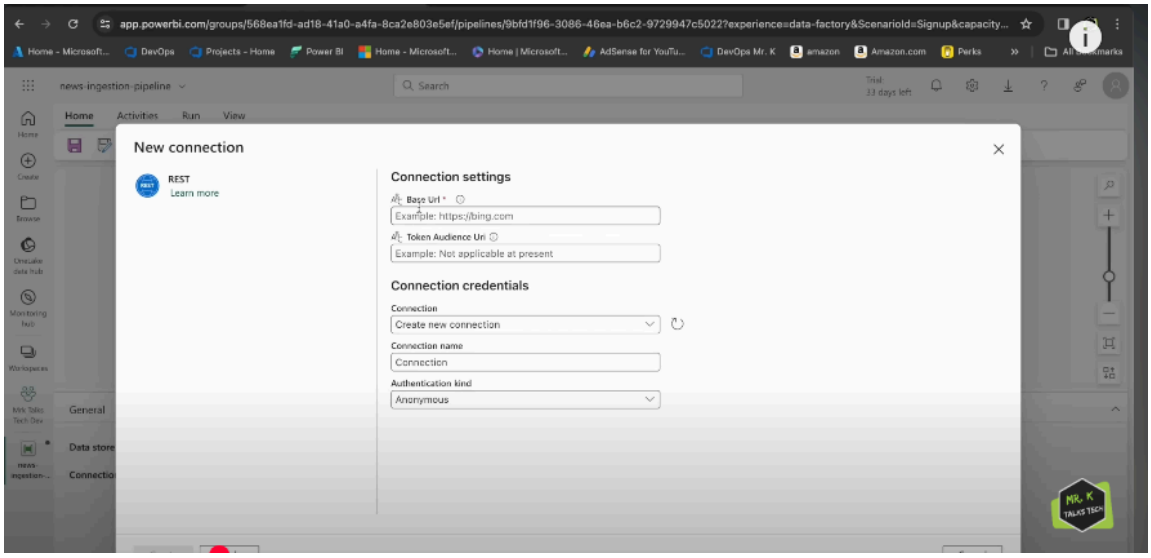


8. Since our data is coming via Azure API we need to choose as external source



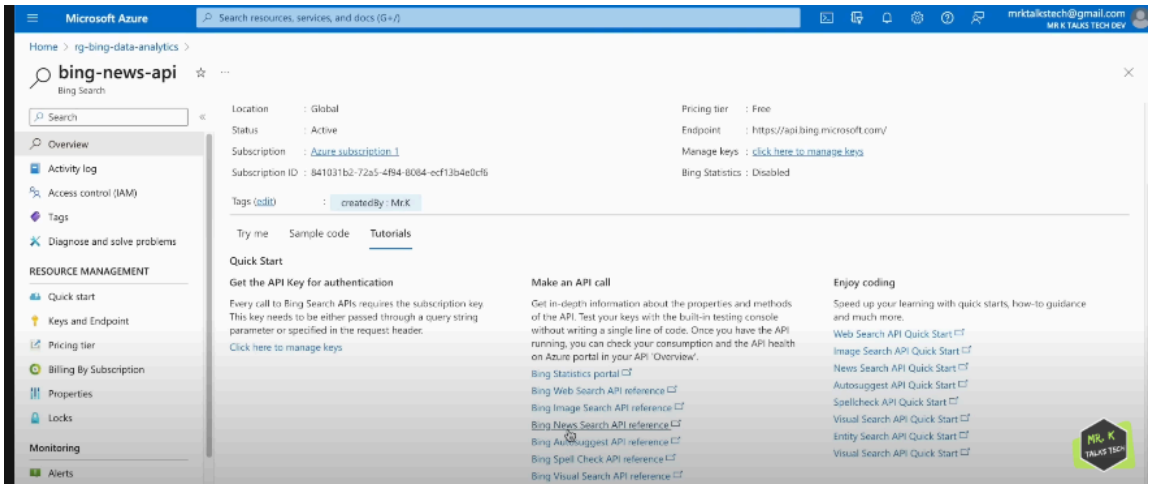
Next we need to configure the connection

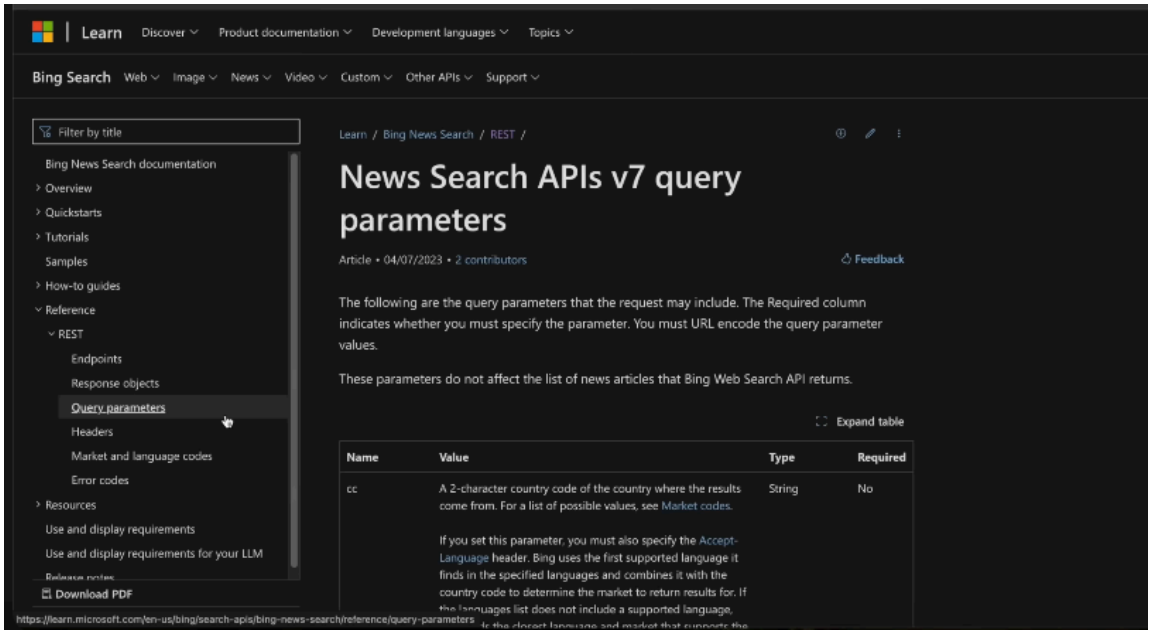
9. Create a new connection and select REST

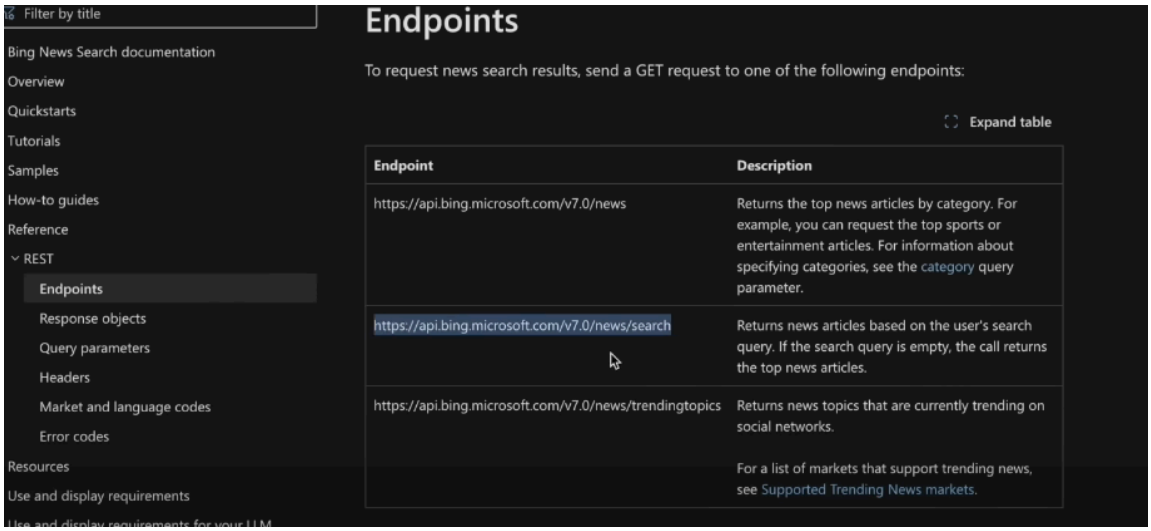


now we need to give url of API

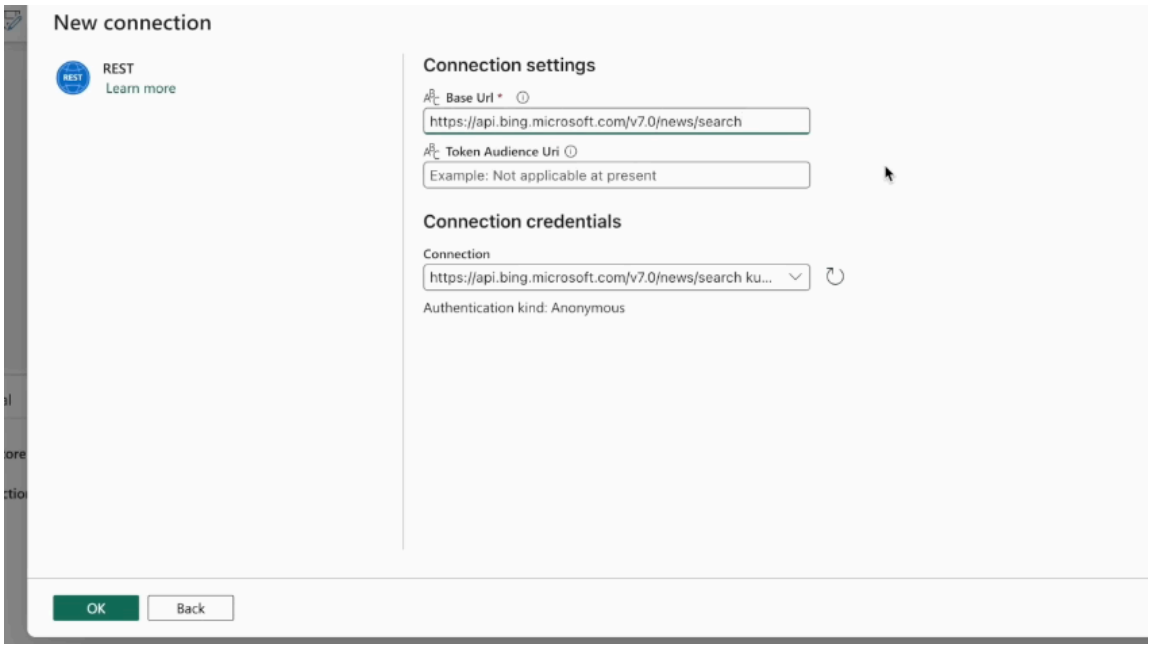
10. To get the URL link…go to our bing API and in the overview→Tutorial go to news seach API



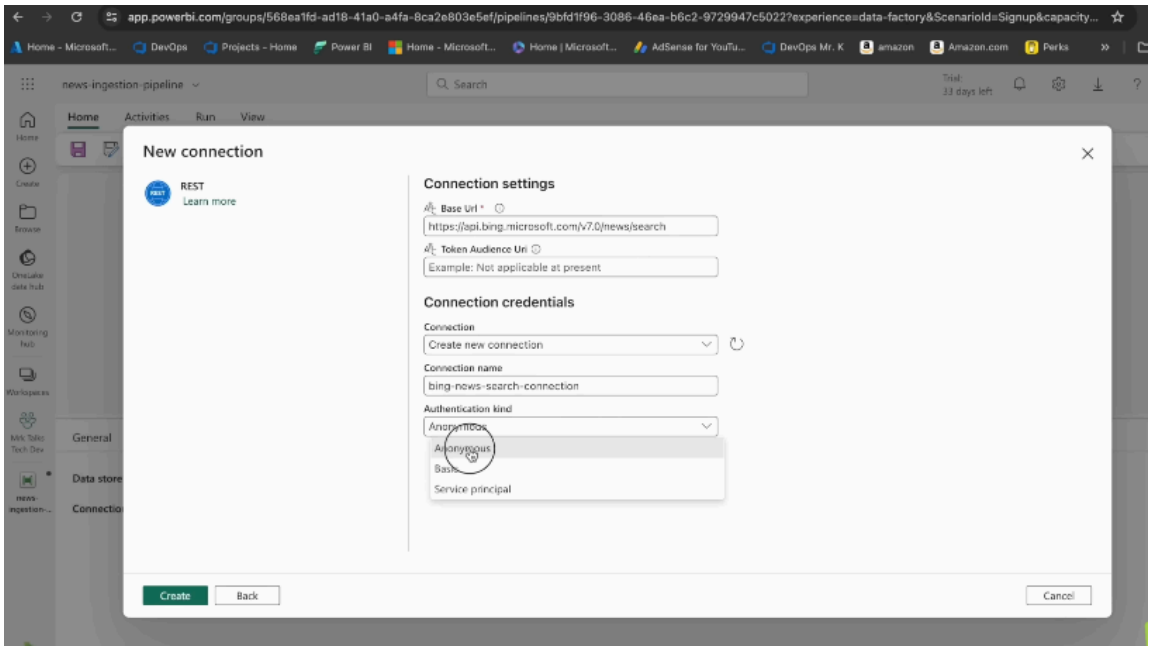




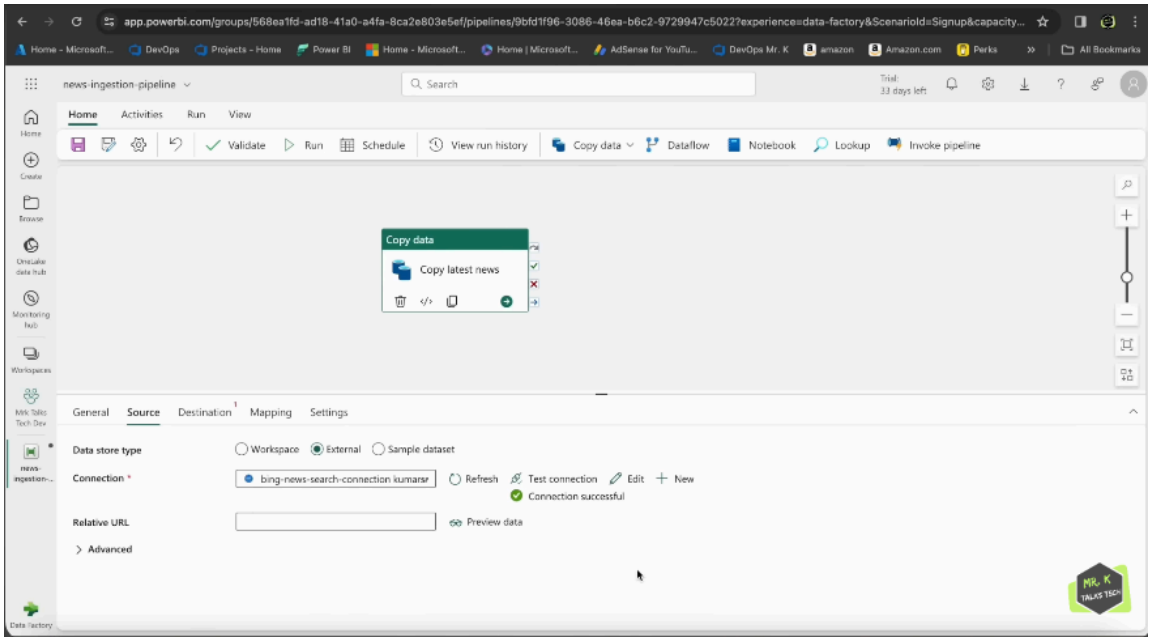
Copy the end point which is our base URL



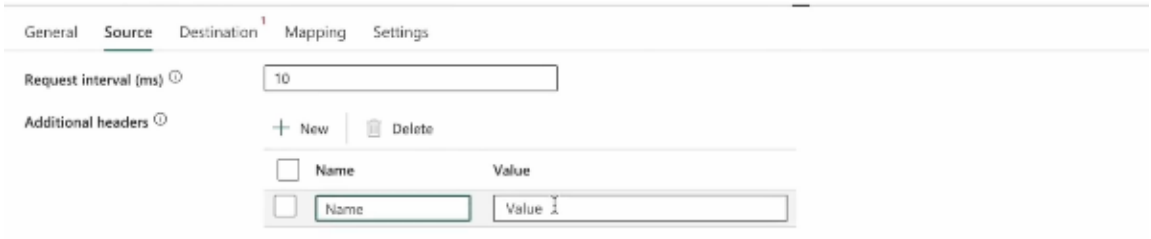
next we have to provide the connection details of this connection



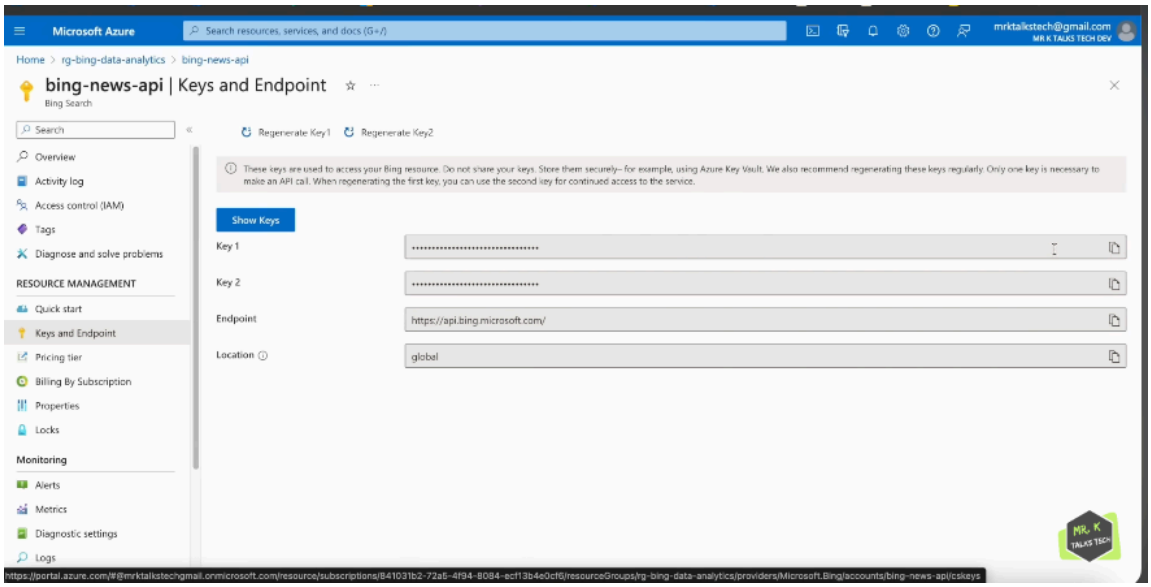
and click on create



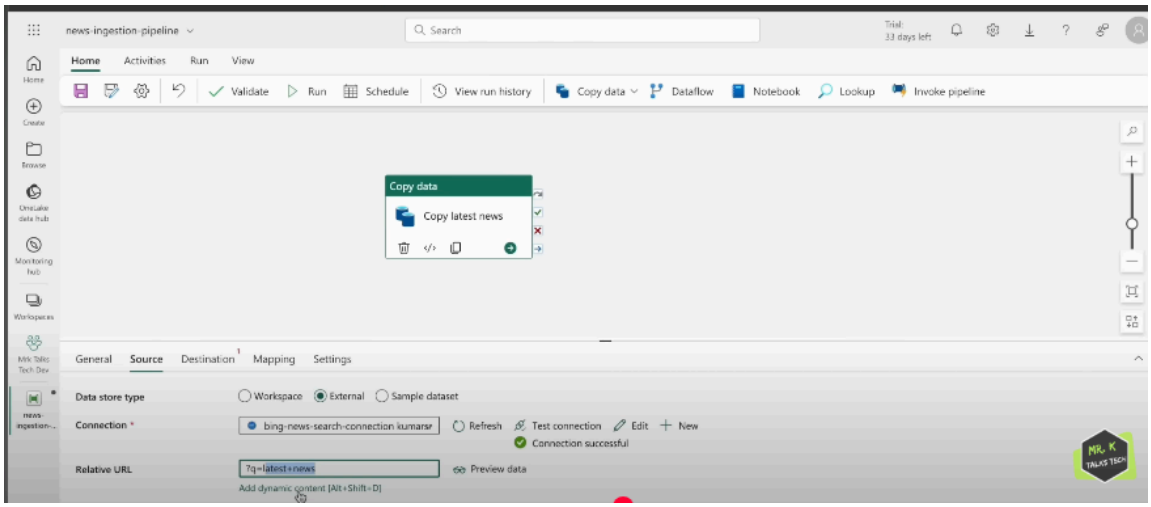
11. Next we need to give to access to the bing Api data for Fabric
12. We will authenticate this API using headers…which contains the key of the API
13. For that in the source tab→Advanced…click on additional headers



we can give a name and we have to give this key1 or key2 of our bing API as the value

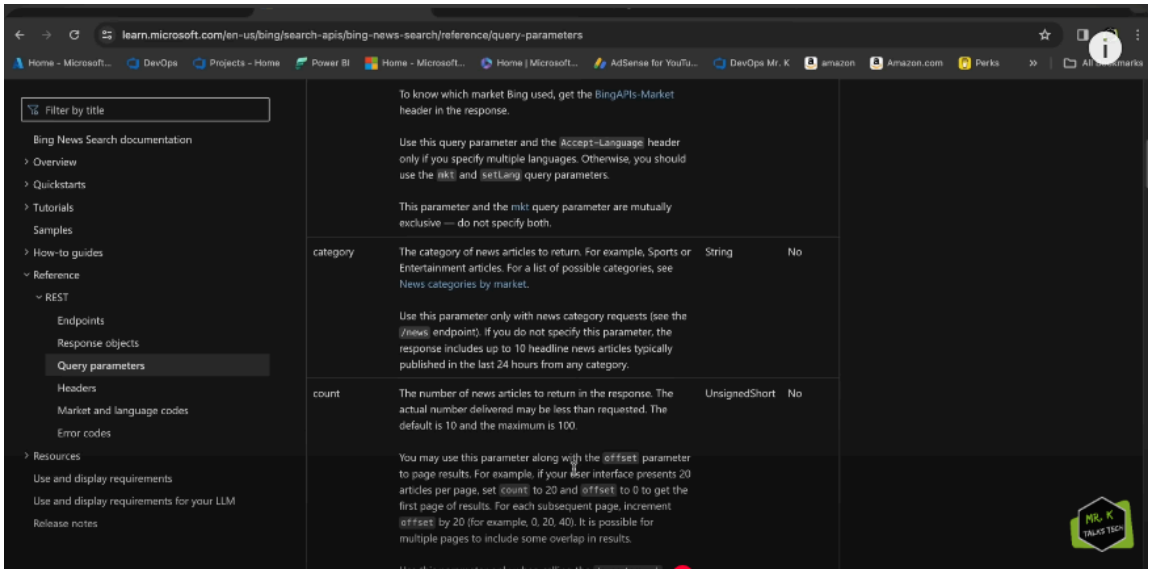


14. Here we need to only get the latest news from our API… for this …we need to pass the url keyword in the relative url

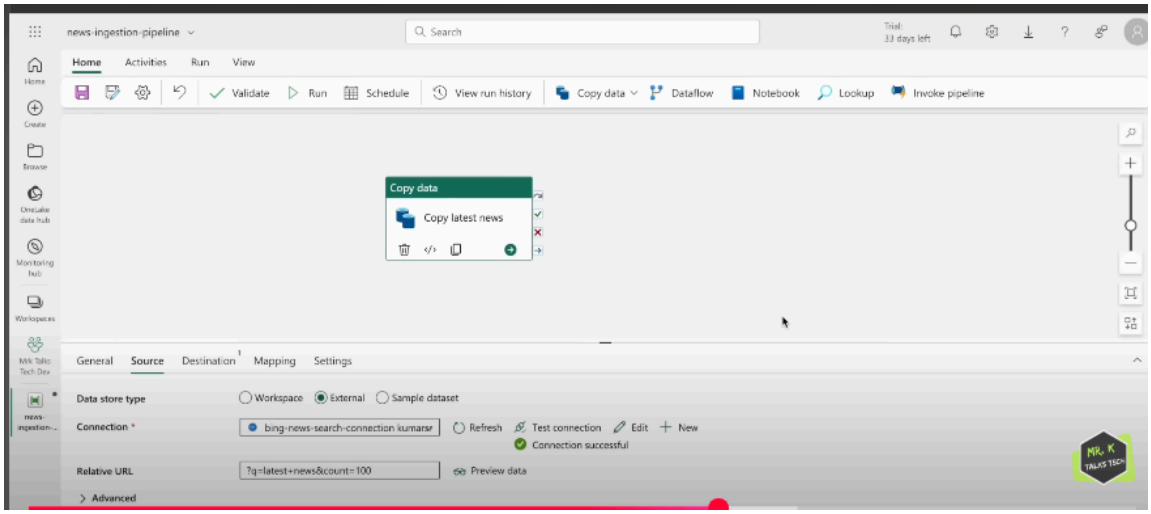


here we gave our search as latest news…so this gives the latest news

15. Here we can see the query parameters for our Bing API

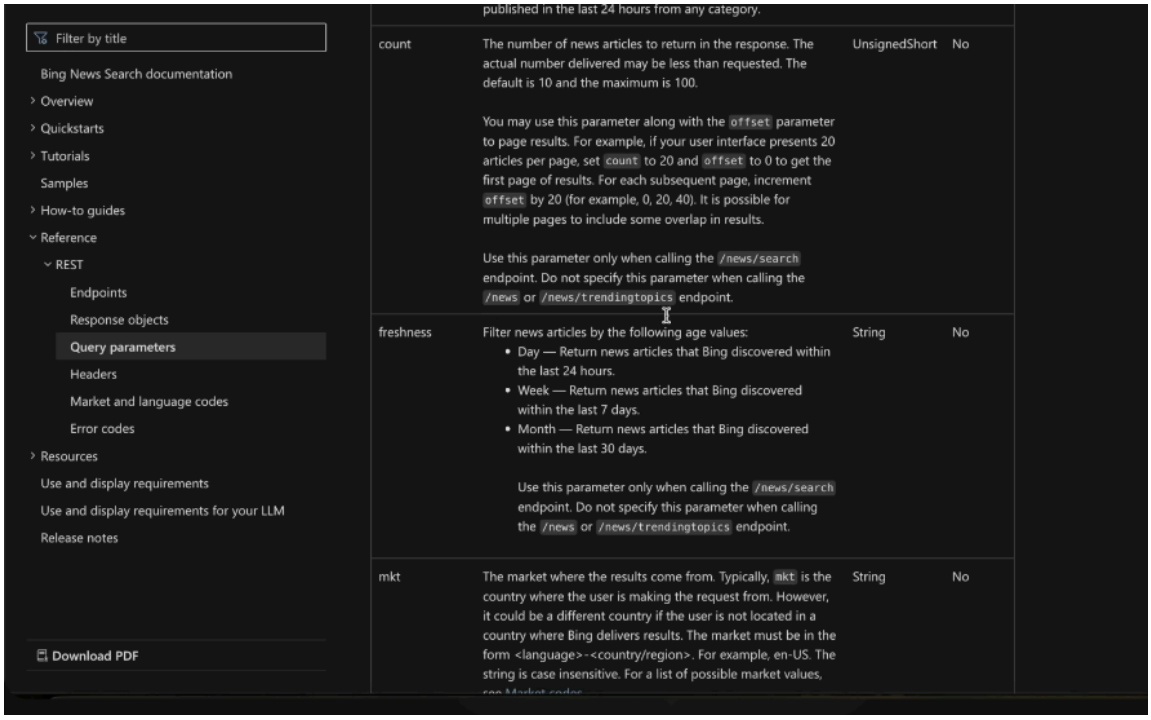


and we use count function to get 100 articles

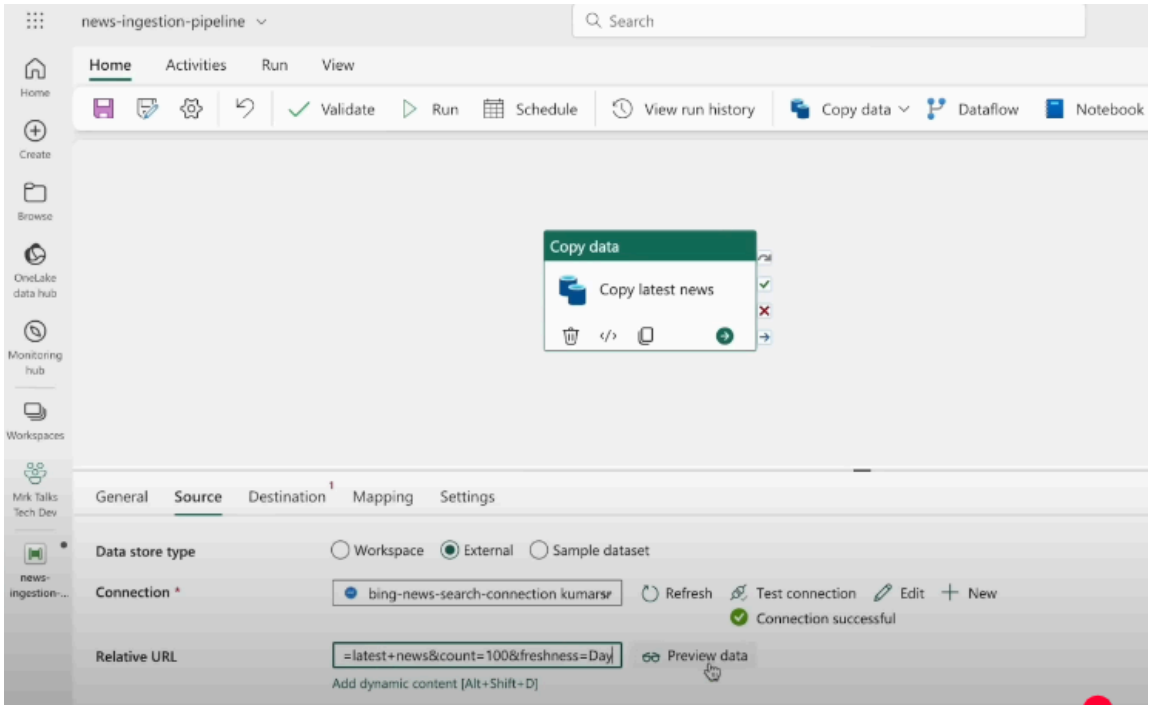


so this is how we use the query parameters

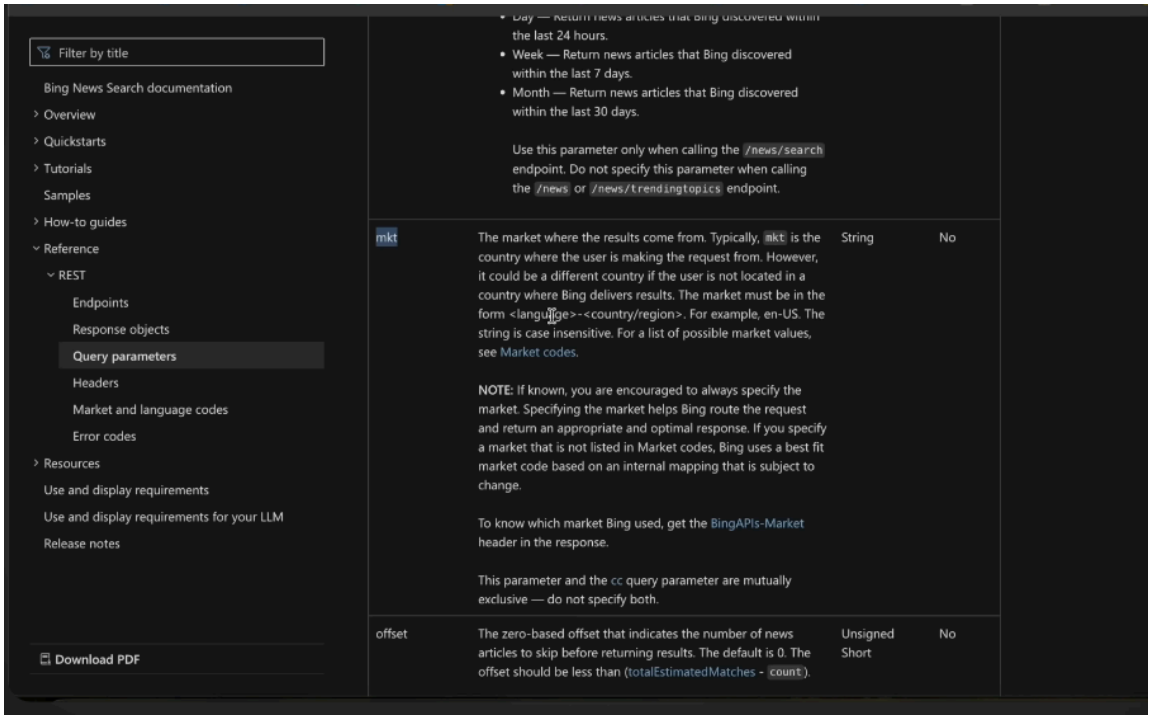
16. Next we will freshness parameter



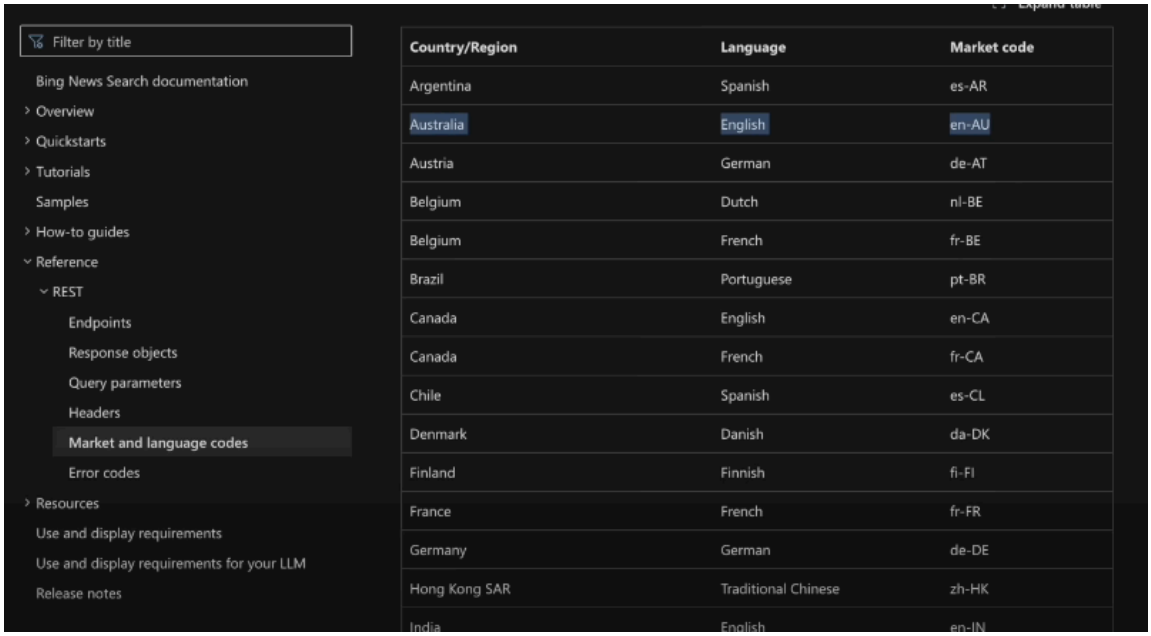
we will use day to get latest news



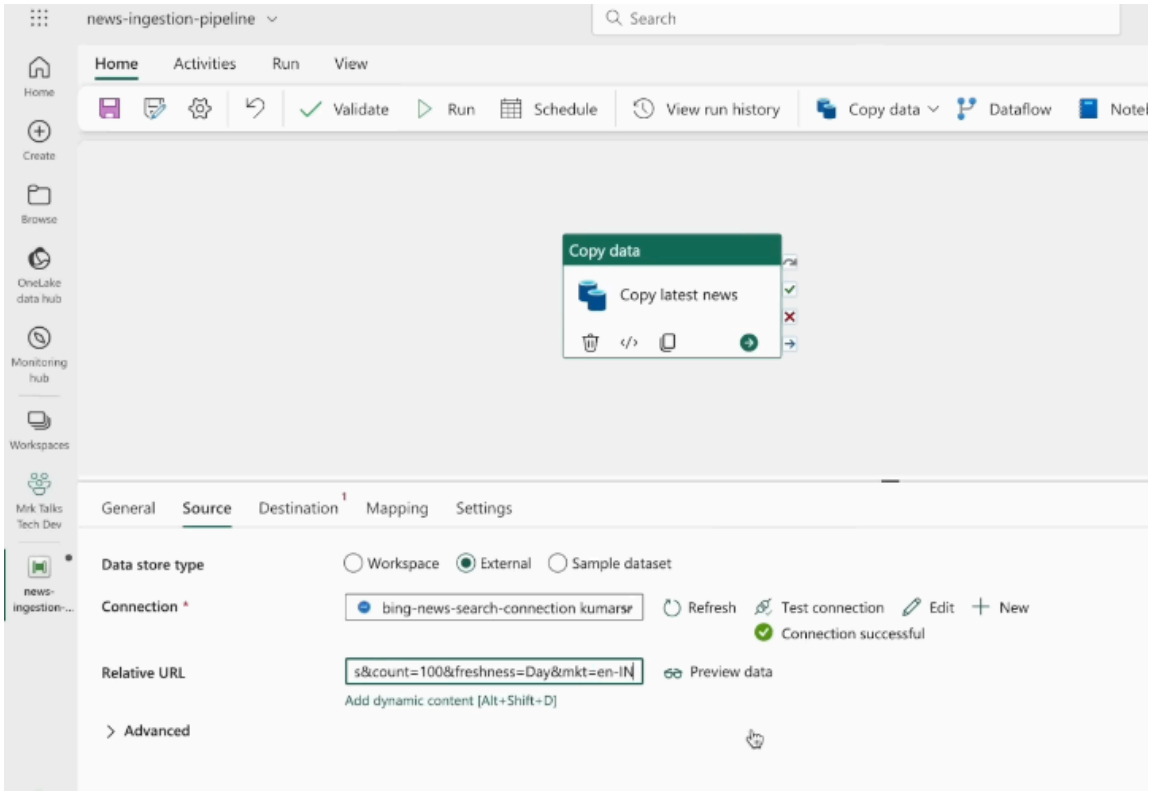
17. Next we have mkt paramter



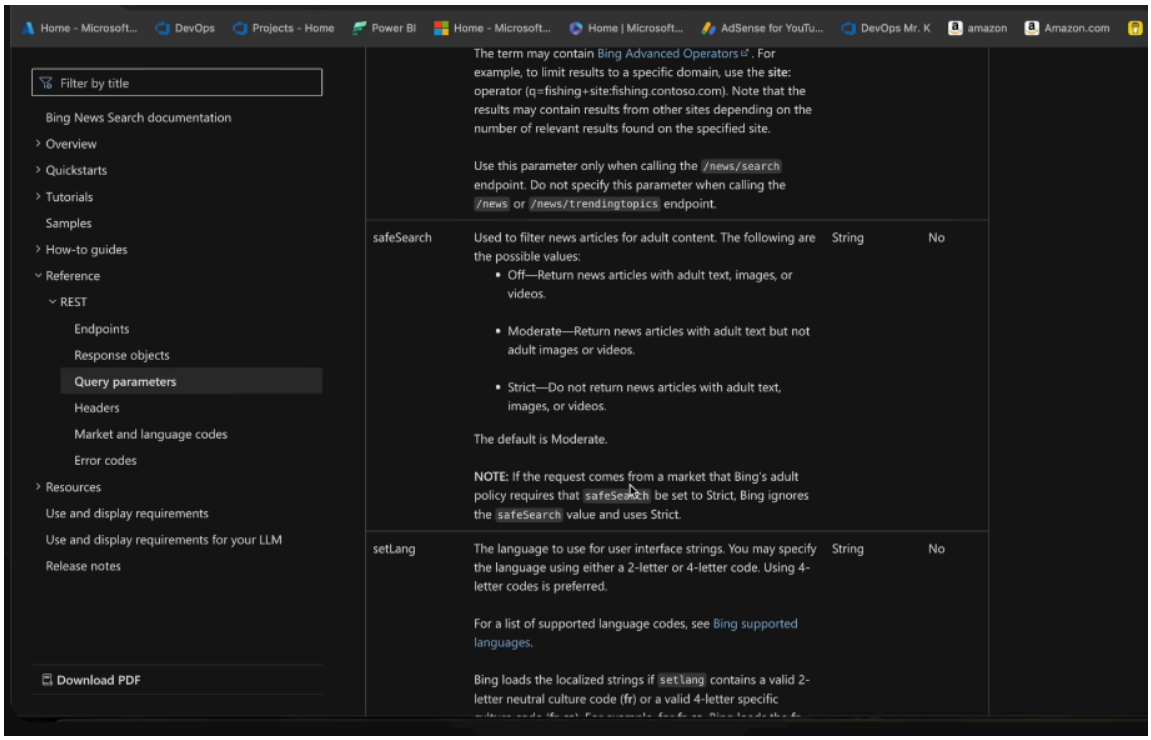
Market codes



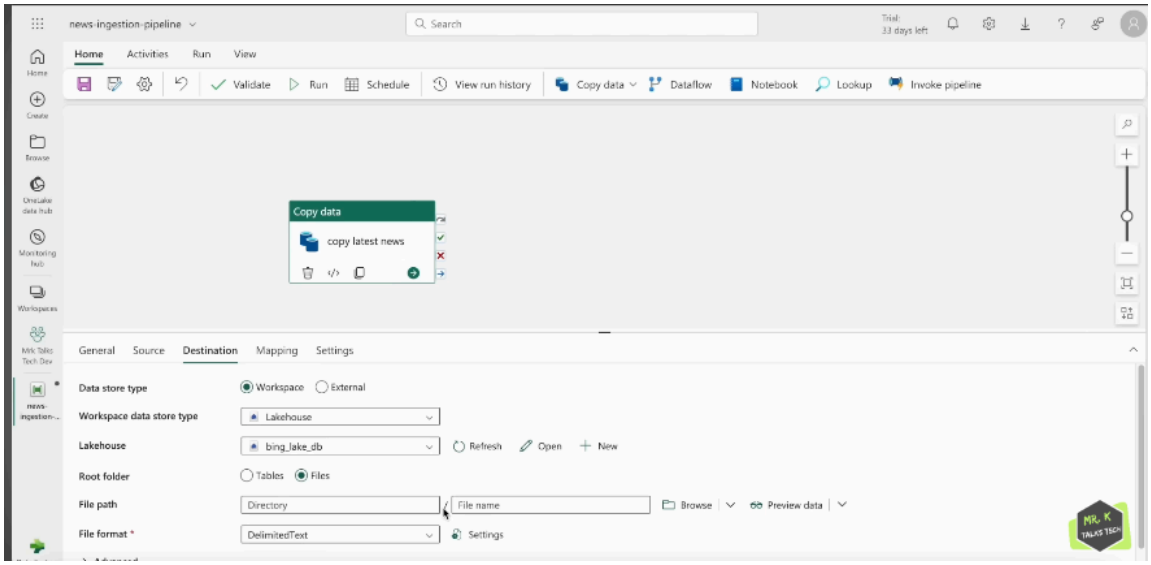
Lets use this param



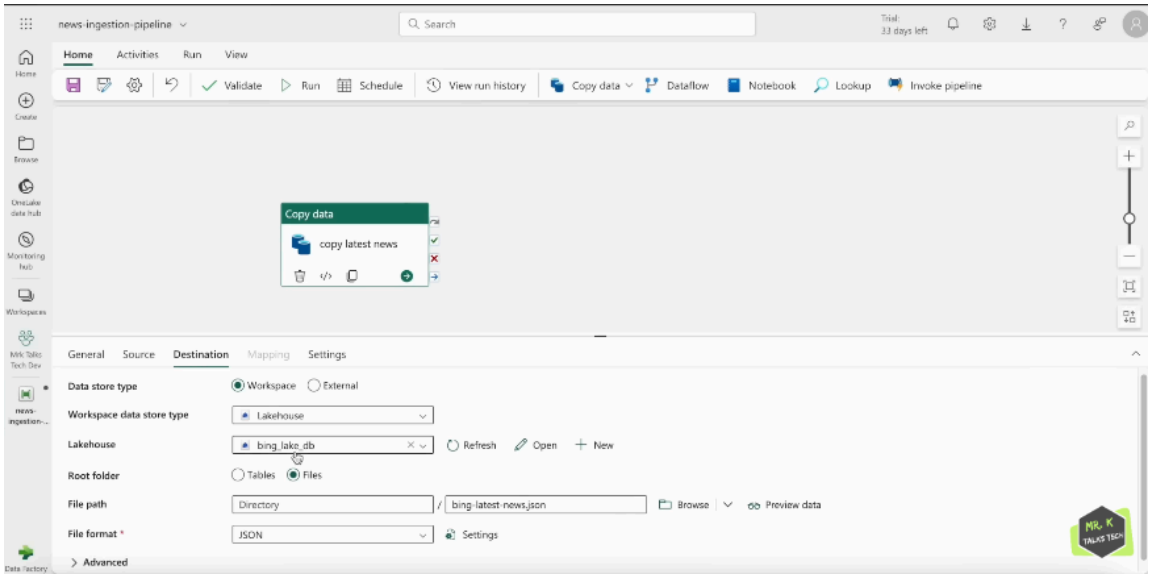
18. To see more parameters you see here



19. Next we need to config a destination..as our dest is fabric lakehoue we choose workspace

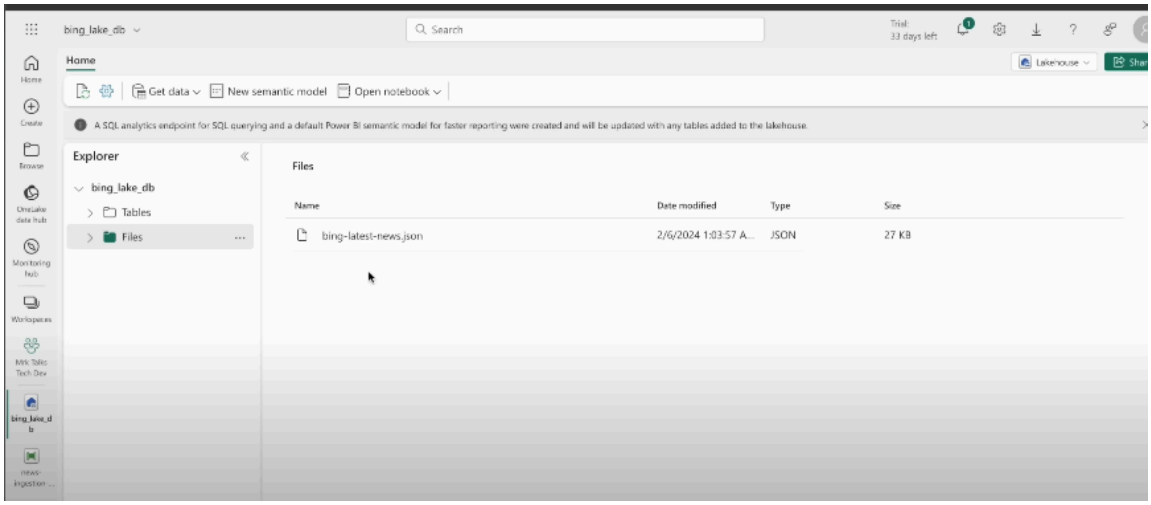


We have given our bing_lake_db as our dest and file format as JSON as our API gives json data



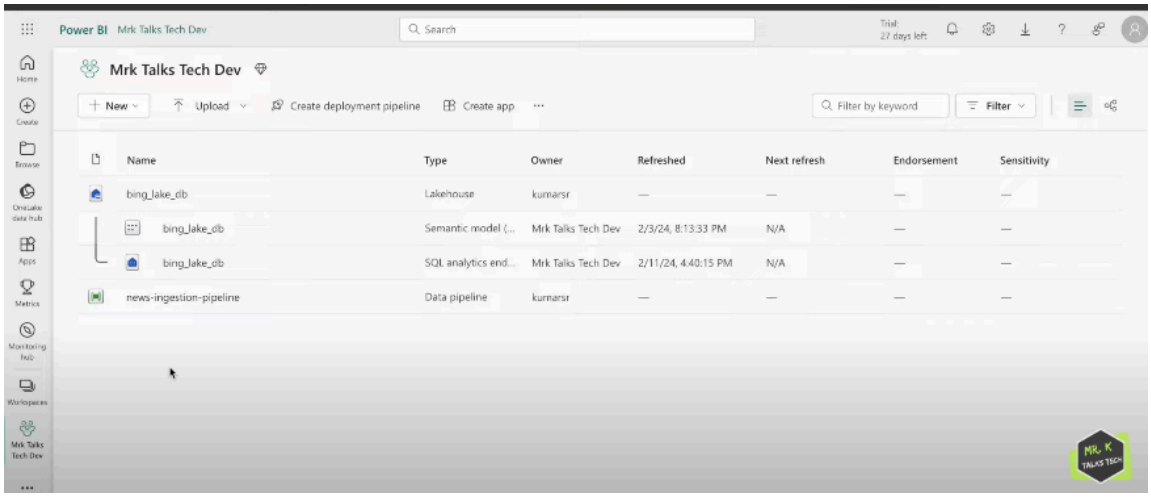
20. Next we run our pipeline…then our copy activity copies the data from API and stores in bing_lake_db..

21. Here we can see ingested file



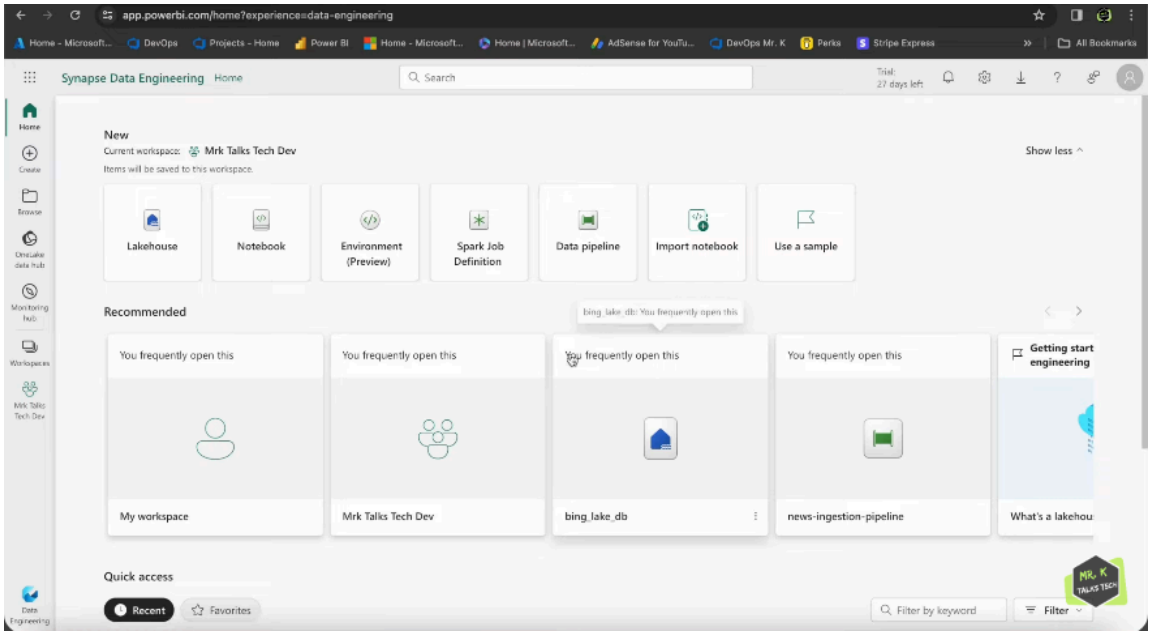
Data Transformation

1. We use Raw json file that we ingested and clean it

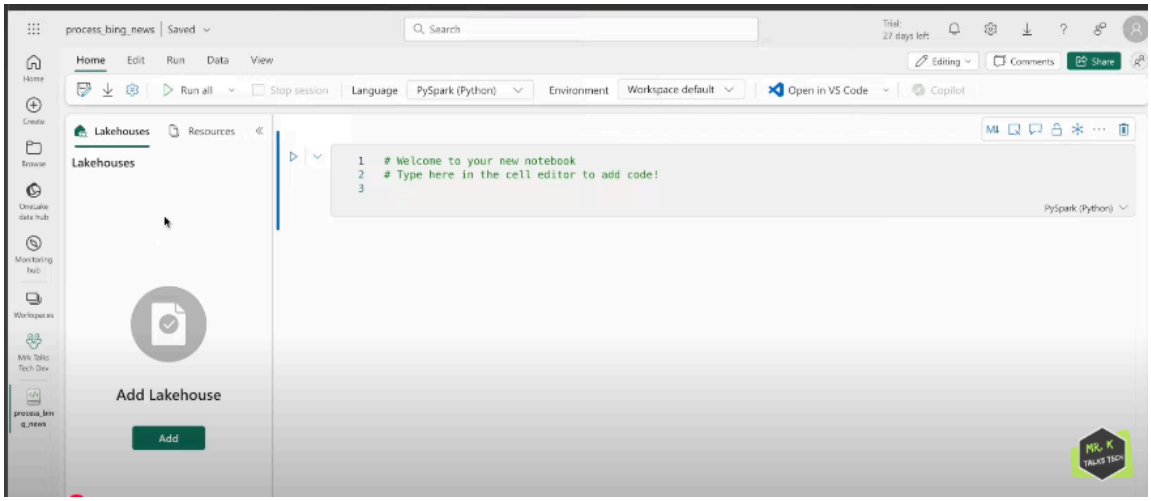


here in our home we have our lake house db and the data pipeline that we have created

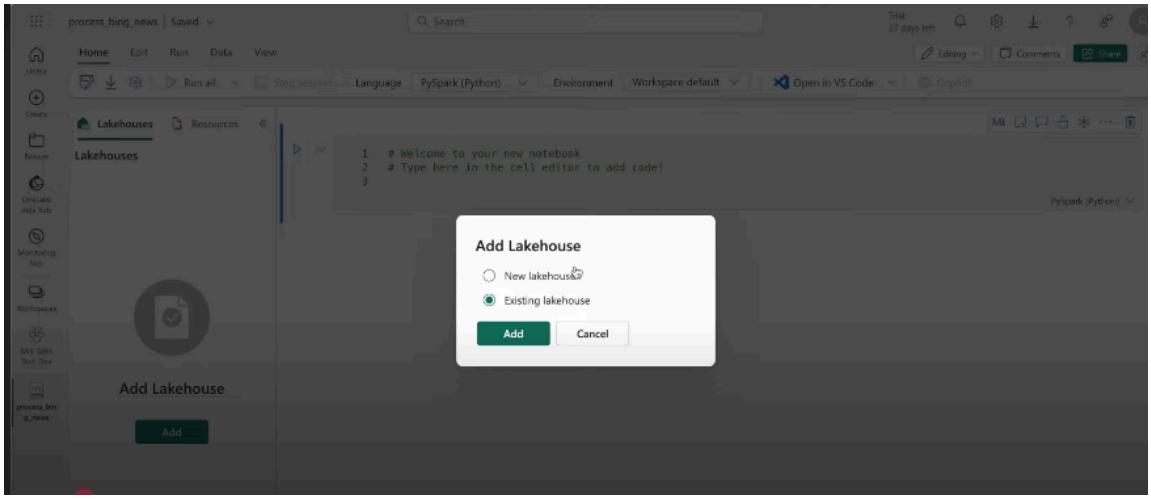
2. For this transformation we use synapse DE

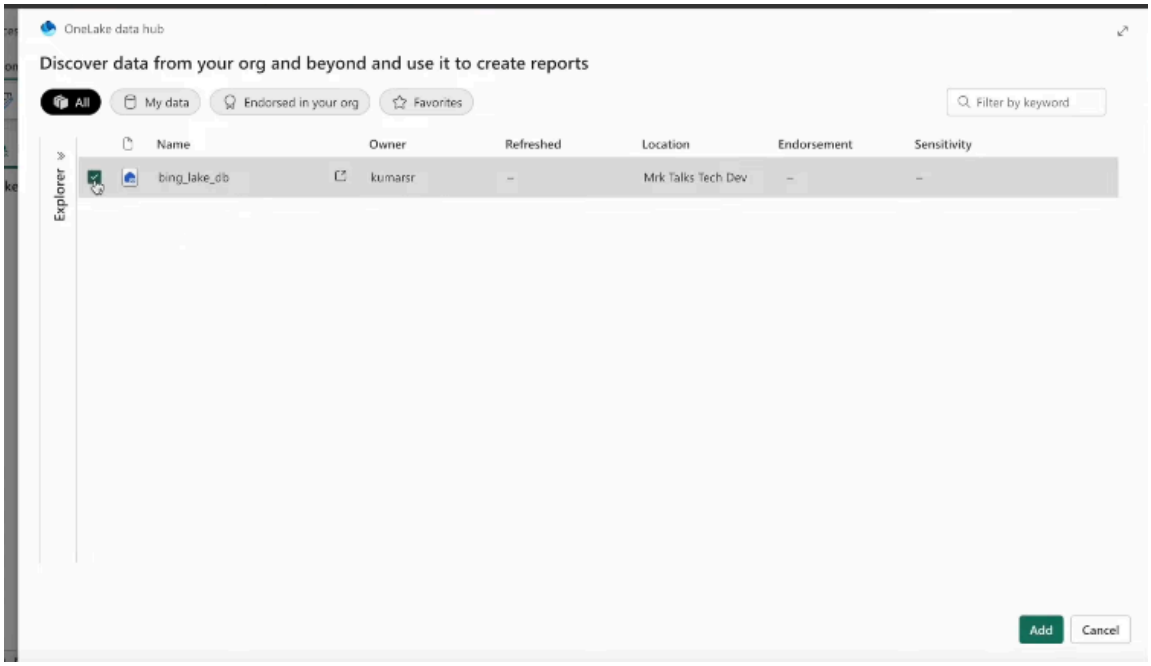


3. Here we create a spark notebook and name it as process_bing_news

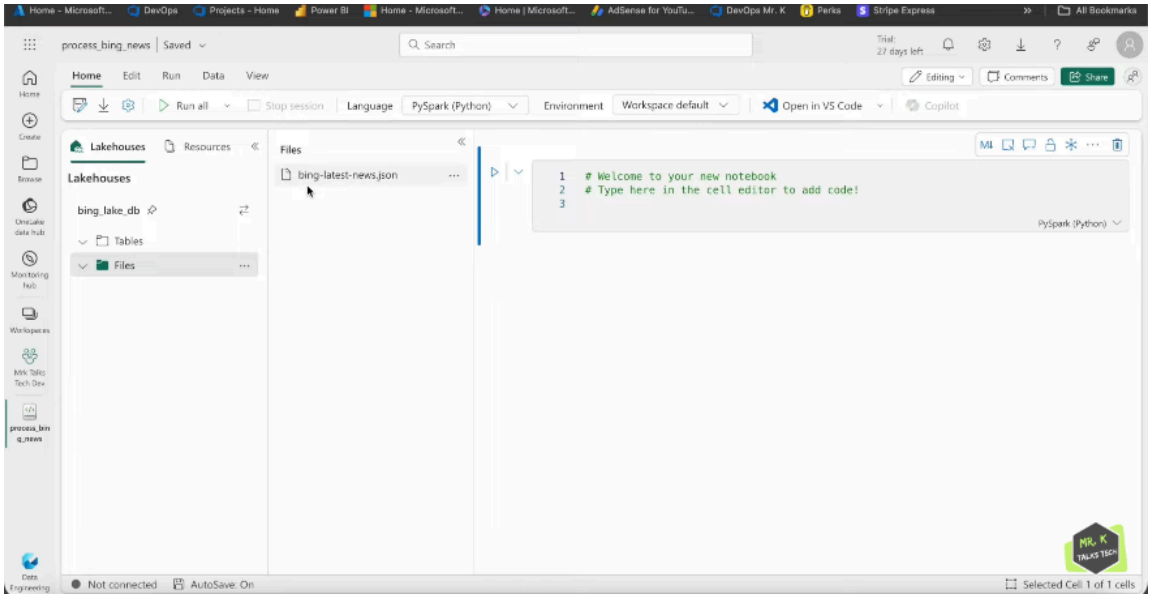


4. Inside the notebook ..we can directly access the lakehouse db

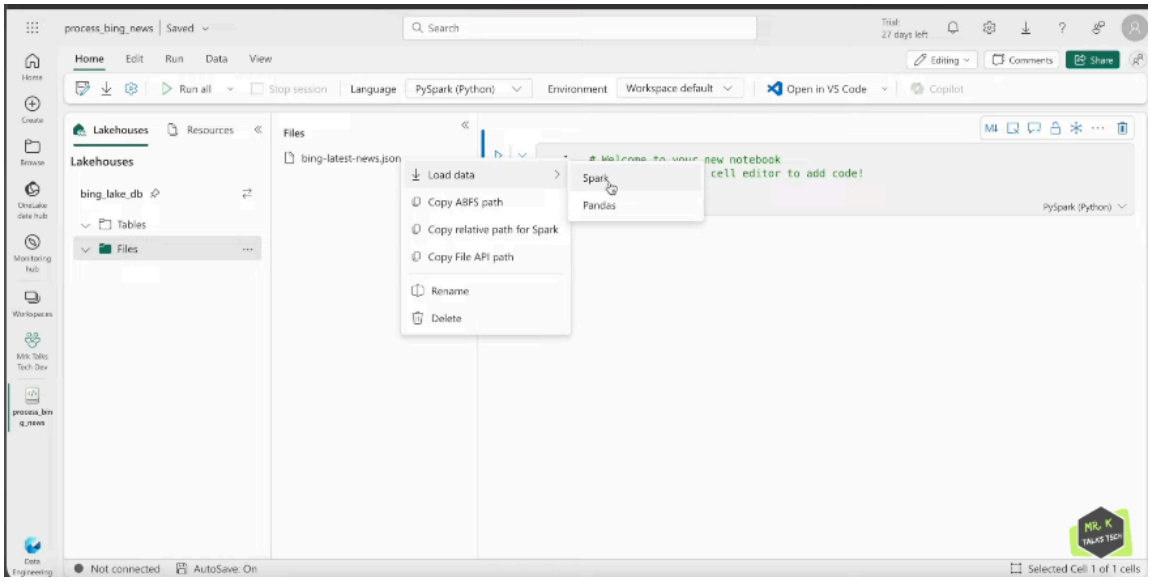




5. If we add our lakehouse to notebook..then we can see our files inside the notebook



6. To read our data choose spark



7. If our json file has multiple lines then need to provide option in spark.read

8. Here we get our output in a single row

but we have our news data in value column…so we only select this value col

9. Next we need to process this value col..to see all the rows of values column…we use explode function

In **Apache Spark**, the `explode` function is used to transform a column containing arrays or maps into multiple rows, where each element in the array or each key-value pair in the map results in a new row. This is especially useful when working with data in a nested format (like JSON) or when dealing with arrays or lists within your dataset, and you want to "flatten" these structures into individual rows for further analysis.

python Copy

```python
from pyspark.sql.functions import explode
df_exploded = df.select(explode(df["column_name"]).alias("new_column"))
```

Suppose we have a DataFrame with a column of arrays, and we want to explode that array into separate rows. Here's an example:

Input DataFrame:

Let's assume we have the following DataFrame:

id	items
1	["apple", "banana", "cherry"]
2	["orange", "grape"]
3	["kiwi"]

Using `explode`:

We can use the `explode` function to turn the `items` column into separate rows:

python Copy

```python
from pyspark.sql.functions import explode

df_exploded = df.select("id", explode(df["items"]).alias("item"))
df_exploded.show()
```

Output:

id	item
1	apple
1	banana
1	cherry
2	orange
2	grape
3	kiwi

In this case:

- For `id = 1`, the `items` array had 3 elements, so 3 new rows were created.
- For `id = 2`, the `items` array had 2 elements, so 2 new rows were created.
- For `id = 3`, the `items` array had 1 element, so 1 new row was created.

Example: Exploding a Map

If the column contains a map (key-value pairs), `explode` can be used to create multiple rows, each representing a key-value pair.

Input DataFrame:

Assume the DataFrame looks like this:

id	attributes
1	{"color": "red", "size": "M"}
2	{"color": "blue", "size": "L"}
3	{"color": "green", "size": "S"}

Using `explode`:

python Copy

```python
from pyspark.sql.functions import explode

df_exploded = df.select("id", explode(df["attributes"]).alias("attribute", "value"))
df_exploded.show()
```

Output:

id	attribute	value
1	color	red
1	size	M
2	color	blue
2	size	L
3	color	green
3	size	S

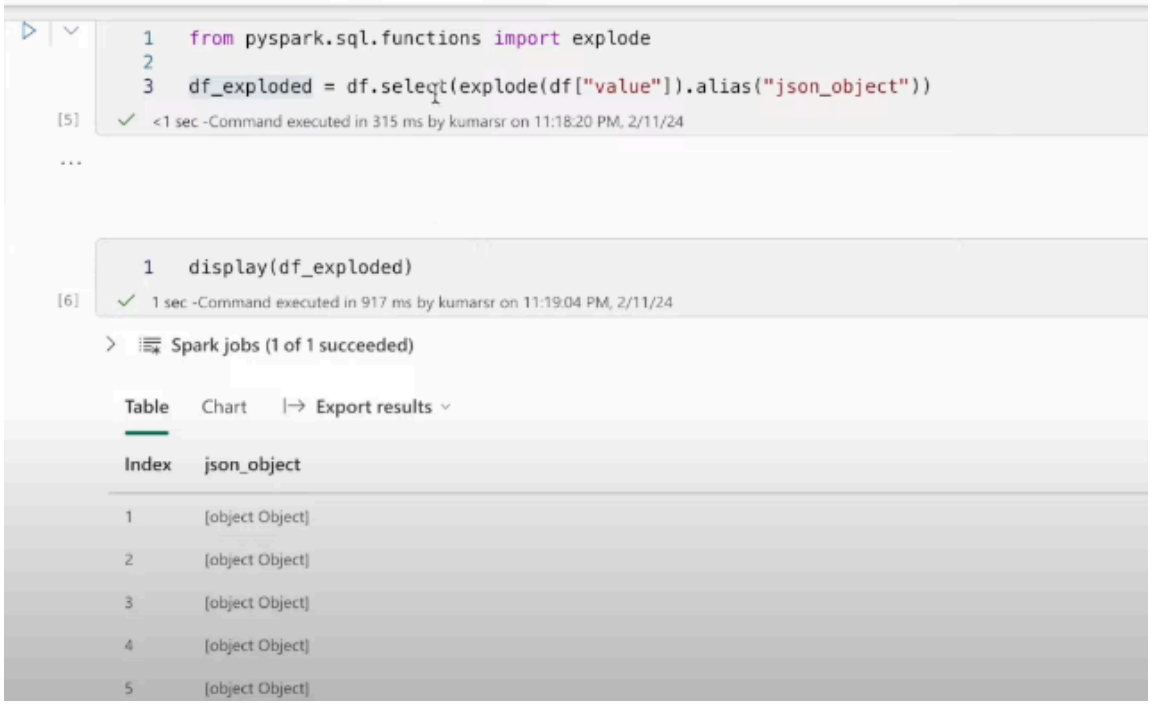
In this case:

- For `id = 1`, the map had two key-value pairs, so two new rows are created.
- For `id = 2`, the map had two key-value pairs as well, so two new rows are created.
- For `id = 3`, the map had two key-value pairs, so two new rows are created.

Summary:

- **explode** is useful for flattening arrays or maps into individual rows.
- It helps in breaking down complex nested data into more manageable, row-based formats, which is particularly useful for analysis and transformations.

10. We use explode for our value col



here all these objects are news articles

11. Next we use toJSON function to see our objects in json string format

The `to_json` function in Apache Spark is used to convert a column containing complex data types (like structs, arrays, or maps) into a **JSON string** representation. This is particularly useful when working with structured data (like **structs** or **maps**) and you need to serialize it into a JSON format for storage or transfer.

python Copy

```python
from pyspark.sql.functions import to_json
df_json = df.select(to_json(df["column_name"]).alias("json_column"))
```

Example: Converting a Struct to JSON

Input DataFrame:

Let's assume you have the following DataFrame with a column that contains **structs**:

id	name	address
1	Alice	{"street": "123 Elm St", "city": "NYC"}
2	Bob	{"street": "456 Oak St", "city": "Boston"}

In this case, the **address** column is a **struct** with two fields: `street` and `city`.

Using `to_json`:

You can use the `to_json` function to convert the **address** column into a JSON string.

python Copy

```python
from pyspark.sql.functions import to_json

df_json = df.select("id", "name", to_json(df["address"]).alias("address_json"))
df_json.show(truncate=False)
```

Output:

id	name	address_json
1	Alice	{"street":"123 Elm St","city":"NYC"}
2	Bob	{"street":"456 Oak St","city":"Boston"}

In this example:

- The **address** struct is converted into a JSON string representation: `{"street":"123 Elm St","city":"NYC"}` for Alice and `{"street":"456 Oak St","city":"Boston"}` for Bob.

12. Our json structure would look like this

```
1  json_list = df_exploded.toJSON().collect()
```

✓ 4 sec -Command executed in 3 sec 670 ms by kumarsr on 11:28:05 PM, 2/11/24

> Spark jobs (1 of 1 succeeded) Log

```
1  print(json_list)
```

✓ <1 sec -Command executed in 322 ms by kumarsr on 11:28:18 PM, 2/11/24


```
lawmakers could lead to a new water resource in the eastern part of the state. On Friday, Fe
Senate introduced Senate Bill 497 which seeks to establish the Pike Reservoir Project Distri
consider creation of new lake, dam in Bourbon County","provider":[{"_type":"Organization","i
{"contentUrl":"https://www.bing.com/th?
id=ODF.UM0amhg3WAObCEGqc4w_Cw&pid=news"}},"name":"Yahoo"}],"url":"https://www.yahoo.com/news
lake-dam-193131924.html"}}', '{"json_object":{"about":[{"name":"Super
Bowl","readLink":"https://api.bing.microsoft.com/api/v7/entities/ce1fece8-34c4-6249-a1aa-
2e779294760e"}],"datePublished":"2024-02-10T19:23:20.0000000Z","description":"Each year, fan
Gatorade will end up dousing the winning head coach at the end of the Super Bowl. See what t
are.","image":{"thumbnail":{"contentUrl":"https://www.bing.com/th?
id=OVFT.aA6lMboTb2QOn0v8ES7V1S&pid=News","height":393,"width":700}},"mentions":[{"name":"Sup
shower"},{"name":"Big Game"}],"name":"What color will the Gatorade bath be at the end of 202
odds","provider":[{"_type":"Organization","image":{"thumbnail":{"contentUrl":"https://www.bi
id=ODF.m1iod5ODNIyyKu23kGIllQ&pid=news"}},"name":"USA Today on MSN.com"}],"url":"https://www
us/sports/nfl/what-color-will-the-gatorade-bath-be-at-the-end-of-2024-super-bowl-see-the-lat
'{"json_object":{"about":[{"name":"United Kingdom","readLink":"https://api.bing.microsoft.co
6bb2-4646-8f7c-3e6b3a53c831"},{"name":"Wales","readLink":"https://api.bing.microsoft.com/api
d525-d360-f2eb5bf3410b"},{"name":"BBC","readLink":"https://api.bing.microsoft.com/api/v7/ent
78a579d79f5b"}],"category":"Sports","datePublished":"2024-02-10T16:44:00.0000000Z","descript
news, BBC coverage, standings and statistics for Saturday\'s Six Nations match between Engla
Twickenham.","image":{"thumbnail":{"contentUrl":"https://www.bing.com/th?
id=OVFT.qCGwjb_G0SzAlZjuKfuiPy&pid=News","height":351,"width":624}},"name":"Six Nations 2024
preview, team news, kick-off time & BBC coverage","provider":[{"_type":"Organization","image
{"contentUrl":"https://www.bing.com/th?
id=ODF.yhngt24TSWuyw3ur0Pt3WQ&pid=news"}},"name":"BBC"}],"url":"https://www.bbc.co.uk/sport/
'{"json_object":{"about":[{"name":"NASDAQ","readLink":"https://api.bing.microsoft.com/api/v7
3bb6-29450420b38e"}],"category":"ScienceAndTechnology","datePublished":"2024-02-
10T12:42:00.0000000Z","description":"Celebrations may be in order for Vanda Pharmaceuticals
shareholders, with the covering analyst delivering a significant upgrade to their statutory
consensus statutory numbers for both revenue and earnings per share (EPS) increased." "image
```


```
1  print(json_list[0])
```

✓ <1 sec -Command executed in 304 ms by kumarsr on 11:28:36 PM, 2/11/24 PySpark (Python)


```
{"json_object":{"about":[{"name":"New Delhi","readLink":"https://api.bing.microsoft.com/api/v7/entities/b474d3c7-
a39a-d5ba-7426-18e00042f03e"},{"name":"Russia","readLink":"https://api.bing.microsoft.com/api/v7/entities/ed4fce79-
8ad4-352b-205b-e4db36c49bbe"},{"name":"Threat","readLink":"https://api.bing.microsoft.com/api/v7/entities/9ccdc704-
0b71-57e0-a4f1-c3036ff79bd4"}],"datePublished":"2024-02-10T23:41:00.0000000Z","description":"The US is trying to
threaten the relationship between New Delhi and Moscow with sanctions, according to the Russian envoy. Read more
about the growing bilateral ties and the call for urgent reforms of the United Nations.","image":{"thumbnail":
{"contentUrl":"https://www.bing.com/th?
id=OVFT.G2ludo3yUnFT2bCx6r5Voi&pid=News","height":379,"width":700}},"mentions":[{"name":"New Delhi"},
{"name":"Russia"},{"name":"Threat"}],"name":"US threatening sanctions to tear New Delhi away from Moscow: Russian
Envoy","provider":[{"_type":"Organization","image":{"thumbnail":{"contentUrl":"https://www.bing.com/th?
id=ODF.VBzHmoY4z9NL3FbRonyM1w&pid=news"}},"name":"Indiatimes on MSN.com"}],"url":"https://www.msn.com/en-
in/news/India/us-threatening-sanctions-to-tear-new-delhi-away-from-moscow-russian-envoy/ar-BB1i5I8t"}}
```


13. Next we convent this Json string to Json dictionary to retrieve the data(like name,description etc) clearly and easily

What are JSON loads () in Python?

The **json.loads()** method can be used to **parse a valid JSON string** and convert it into a Python Dictionary. It is mainly used for deserializing native string, byte, or byte array which consists of JSON data into Python Dictionary.

14. Lets use this on our last article and see the output

```
1  import json
2
3  news_json = json.loads(json_list[25])
```

PySpark

```
1  print(news_json)
```

✓ <1 sec -Command executed in 317 ms by kumarsr on 11:34:26 PM, 2/11/24 — PySpark (Python)


```
{'json_object': {'about': [{'name': 'New Jersey', 'readLink':
'https://api.bing.microsoft.com/api/v7/entities/05277898-b62b-4878-8632-09d29756a2ff'}, {'name': 'WWE', 'readLink':
'https://api.bing.microsoft.com/api/v7/entities/413f922e-0c5f-f9d2-0bd2-f5970294a12e'}], 'category': 'LifeStyle',
'datePublished': '2024-02-11T00:46:35.0000000Z', 'description': 'WWE star Daria Berenato and fitness model Toni
Cassano tied the knot on Feb. 10 in an elegant ceremony at the Legacy Castle in New Jersey in front of 140 guests.',
'image': {'thumbnail': {'contentUrl': 'https://www.bing.com/th?id=OVFT.Uf1PpdBMv2QSYmMztUz_BC&pid=News', 'height':
466, 'width': 700}}, 'mentions': [{'name': 'New Jersey'}, {'name': 'WWE'}, {'name': 'Marriage'}], 'name': 'WWE Star
Daria Berenato Marries Toni Cassano in New Jersey Wedding Officiated by Maria Menounos (Exclusive)', 'provider':
[{'_type': 'Organization', 'image': {'thumbnail': {'contentUrl': 'https://www.bing.com/th?
id=OOF.kridaNmIIz5AdgcUGG5eBQ&pid=news'}}, 'name': 'People on MSN.com'}], 'url': 'https://www.msn.com/en-
us/tv/celebrity/wwe-star-daria-berenato-marries-toni-cassano-in-new-jersey-wedding-officiated-by-maria-menounos-
exclusive/ar-BB1i5MS8'}}
```


now we can use this json dict and easily process the data.
Like if we want to see the description of news article then we use

```
1  print(news_json['json_object']['description'])
```

✓ <1 sec -Command executed in 344 ms by kumarsr on 11:43:07 PM, 2/11/24 — PySpark (Python)

```
WWE star Daria Berenato and fitness model Toni Cassano tied the knot on Feb. 10 in an elegant ceremony at the Legacy
Castle in New Jersey in front of 140 guests.
```

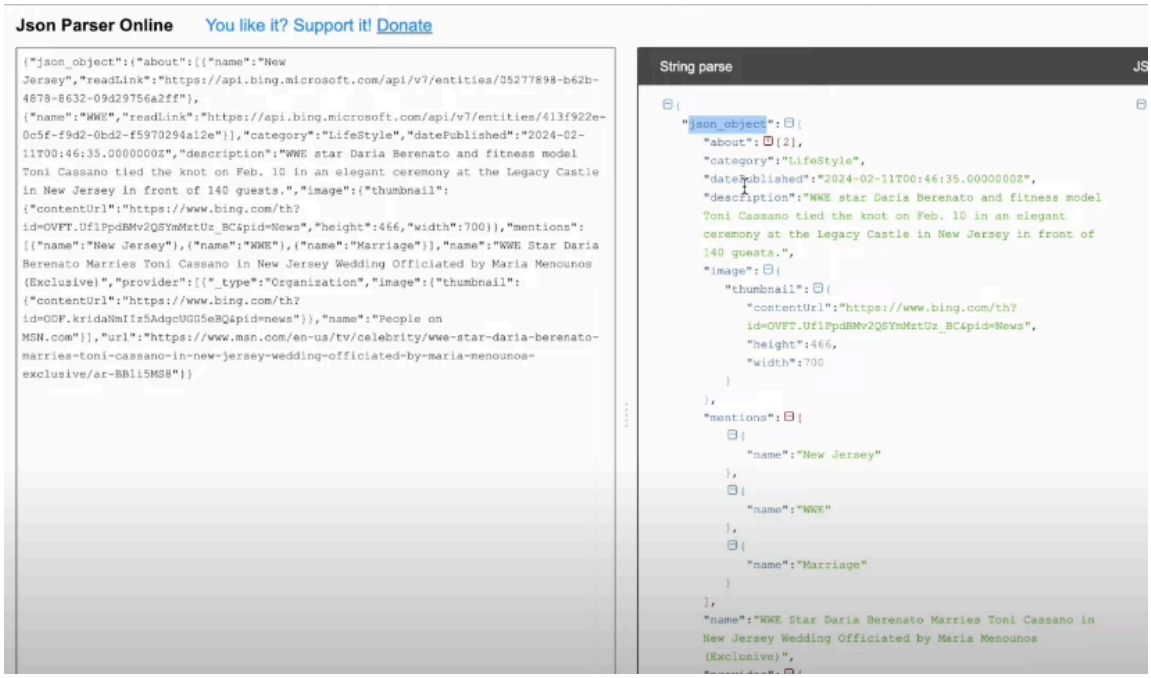
15. In our output we need this columns

```
1  description
2  title
3  category
4  image
5  url
6  provider
7  datePublished
```

PySpark (Python)

and this would be our schema

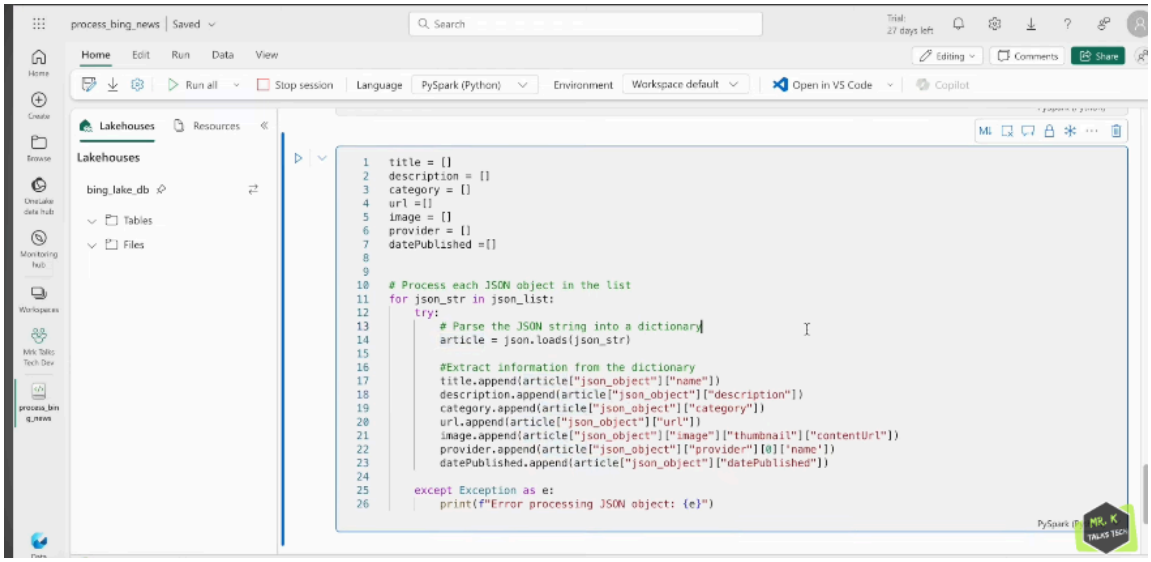
16. To see our json data clearly ..we can use jason parser



17. We can use this code to get the required data from our JSOn file

```
print(news_json["json_object"]["name"])
print(news_json["json_object"]["description"])
print(news_json["json_object"]["category"])
print(news_json["json_object"]["url"])
print(news_json["json_object"]["image"]["thumbnail"]["contentUrl"])
print(news_json["json_object"]["provider"][0]['name'])
print(news_json["json_object"]["datePublished"])
```

18. Now we need to write a code to extract all 26 articles
19. we can use a for loop to iterate all the 26 articles



here in this code …json list has all the articles from our input file

20. Now if we run the above code we get this output

```
Error processing JSON object: 'category'
Error processing JSON object: 'category'
Error processing JSON object: 'category'
Error processing JSON object: 'image'
Error processing JSON object: 'category'
Error processing JSON object: 'category'
Error processing JSON object: 'category'
Error processing JSON object: 'category'
Error processing JSON object: 'image'
Error processing JSON object: 'image'
Error processing JSON object: 'image'
```

21. We are getting this error bcuz …there are some null values in our json file…so we have used try except to find any errors ..and we got this error
22. So to solve this…we only process the news article which has all the values(no nulls)

```
# Process each JSON object in the list
for json_str in json_list:
    try:
        # Parse the JSON string into a dictionary
        article = json.loads(json_str)

        if article["json_object"].get("category") and article["json_object"].get("image", {}).get("thumbnail", {}).

            #Extract information from the dictionary
            title.append(article["json_object"]["name"])
            description.append(article["json_object"]["description"])
            category.append(article["json_object"]["category"])
            url.append(article["json_object"]["url"])
            image.append(article["json_object"]["image"]["thumbnail"]["contentUrl"])
            provider.append(article["json_object"]["provider"][0]['name'])
            datePublished.append(article["json_object"]["datePublished"])

    except Exception as e:
        print(f"Error processing JSON object: {e}")
```

here we used a if condition to check whether the all values are present…if yes then we extract the data from the JSON file…if false then we ignore it

23. Another way to solve this is by using a default value

24. Next we will combine all this data and create a df called df_cleaned

```
from pyspark.sql.types import StructType, StructField, StringType


# Combine the lists
data = list(zip(title,description,category,url,image,provider,datePublished))

# Define schema
schema = StructType([
    StructField("title", StringType(), True),
    StructField("description", StringType(), True),
    StructField("category", StringType(), True),
    StructField("url", StringType(), True),
    StructField("image", StringType(), True),
    StructField("provider", StringType(), True),
    StructField("datePublished", StringType(), True)
])

# Create DataFrame
df_cleaned = spark.createDataFrame(data, schema=schema)
```

```
display(df_cleaned)
```

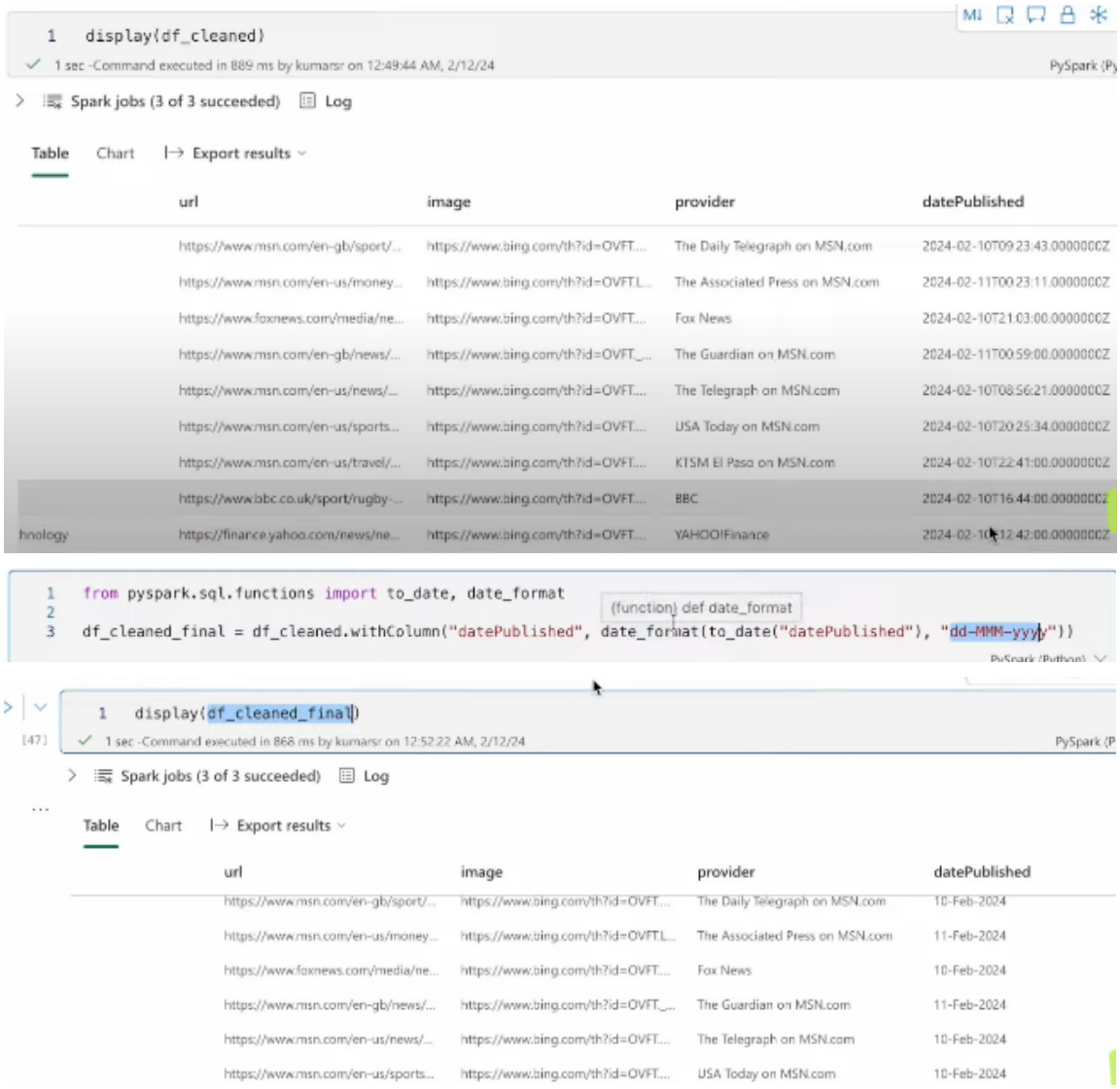
1 sec - Command executed in 889 ms by kumarsr on 12:49:44 AM, 2/12/24 — PySpark (Python)

Spark jobs (3 of 3 succeeded) Log

Table Chart Export results

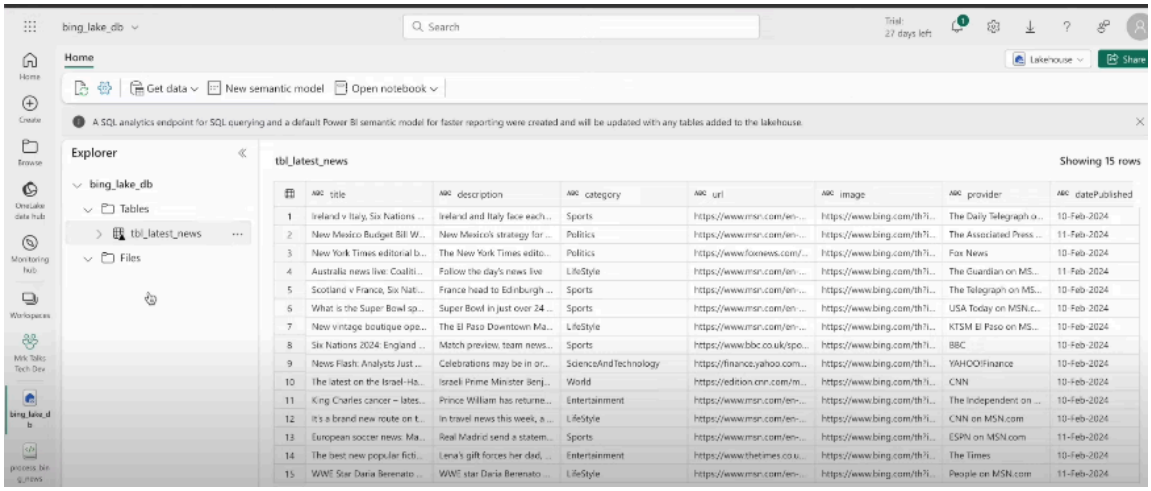
Index	title	description	category	url	image
1	Ireland v Italy, Six Nations 2024: Kick...	Ireland and Italy face each other on ...	Sports	https://www.msn.com/en-gb/sport/...	https://ww
2	New Mexico Budget Bill Would Fou...	New Mexico's strategy for spending...	Politics	https://www.msn.com/en-us/money...	https://ww
3	New York Times editorial board de...	The New York Times editorial board ...	Politics	https://www.foxnews.com/media/ne...	https://ww
4	Australia news live: Coalition will op...	Follow the day's news live	LifeStyle	https://www.msn.com/en-gb/news/...	https://ww
5	Scotland v France, Six Nations 2024:...	France head to Edinburgh on the ba...	Sports	https://www.msn.com/en-us/news/...	https://ww
6	What is the Super Bowl spread? Lat...	Super Bowl in just over 24 hours, be...	Sports	https://www.msn.com/en-us/sports...	https://ww
7	New vintage boutique opens in Do...	The El Paso Downtown Managemen...	LifeStyle	https://www.msn.com/en-us/travel/...	https://ww
8	Six Nations 2024: England vs Wales ...	Match preview, team news, BBC cov...	Sports	https://www.bbc.co.uk/sport/rugby-...	https://w
9	News Flash: Analysts Just Made A Si...	Celebrations may be in order for Va...	ScienceAndTechnology	https://finance.yahoo.com/news/ne...	https
10	The latest on the Israel-Hamas war	Israeli Prime Minister Benjamin Net...	World	https://edition.cnn.com/middleeast/...	https://ww

25. Next we format the datepublished col



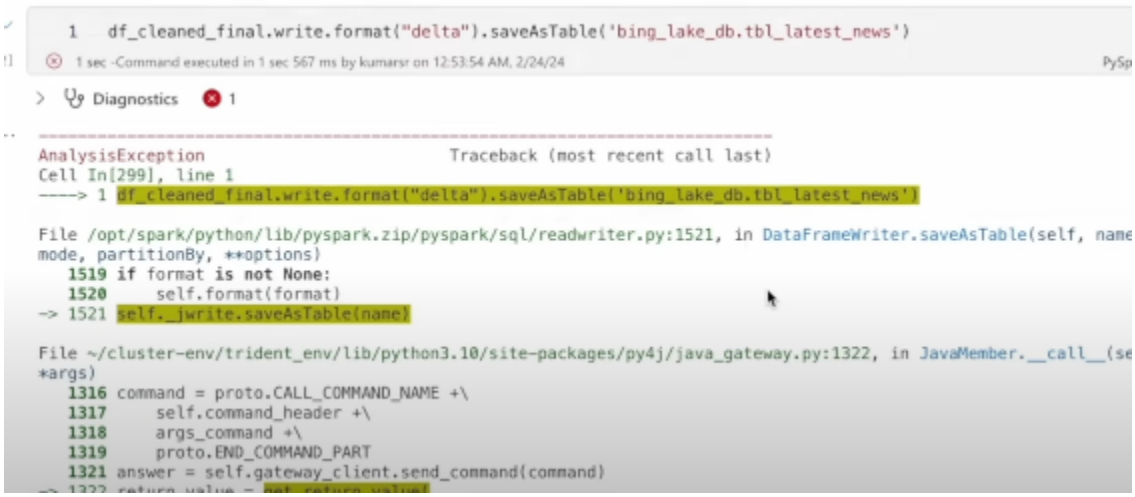
26. Finally we save our cleaned data in delta format

27. Now we can see our data in tables section



Incremental Load - Type1

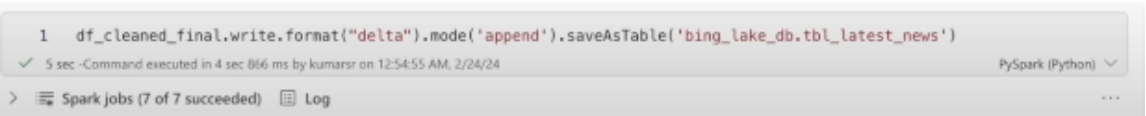
1. Previously we have saved our table in our one lake…and if we again try to save it..we get an error ‘already exists’



2. So to overcome this we can use mode

ANd this is not the best way to do it…if we overwrite the same data ..it costs us additional time and compute power…and the other reason is we can lose the data

3. Next way to solve is by using append mode



if we use this mode..the same table data will be appended to the old table…and this also is not a best way to deal

4. Third way is to use incremental load.and this will be best way
Here if the data already exists in table..then we don't add the new data…

This can be done with the help of SQL Merge Data Warehousing

In the context of data warehousing, the SQL `MERGE` statement is a powerful tool that allows you to synchronize data between two tables (typically a target table and a source table). It combines `INSERT`, `UPDATE`, and `DELETE` operations in a single query. This operation is often used for tasks like data consolidation, updating dimensions, and managing slowly changing dimensions (SCDs).

What does the `MERGE` statement do?

- **Insert**: Adds new records to the target table if they don't already exist.
- **Update**: Modifies existing records in the target table if they have changed.
- **Delete**: Removes records from the target table if they no longer exist in the source.

Let's consider a data warehouse where we have a `sales_fact` table (target) and a `new_sales` table (source) with the same structure. We want to update the `sales_fact` table with new sales data from `new_sales`, add any new records, and delete records that no longer exist in the source.

Tables:

- `sales_fact`:
  - `sale_id`
  - `product_id`
  - `sale_date`
  - `quantity_sold`
  - `total_amount`

- `new_sales`:
  - `sale_id`
  - `product_id`
  - `sale_date`
  - `quantity_sold`
  - `total_amount`

Scenario:

1. If a sale exists in both `sales_fact` and `new_sales` with the same `sale_id`, update the record in `sales_fact`.
2. If a sale exists in `new_sales` but not in `sales_fact`, insert it into `sales_fact`.
3. If a sale exists in `sales_fact` but not in `new_sales`, delete it from `sales_fact`.

SQL `MERGE` Query:

sql Copy

```sql
MERGE INTO sales_fact AS SF
USING new_sales AS NS
ON SF.sale_id = NS.sale_id
WHEN MATCHED THEN
    UPDATE SET SF.quantity_sold = NS.quantity_sold, SF.total_amount = NS.total_amount
WHEN NOT MATCHED THEN
    INSERT (sale_id, product_id, sale_date, quantity_sold, total_amount)
    VALUES (NS.sale_id, NS.product_id, NS.sale_date, NS.quantity_sold, NS.total_amount
WHEN NOT MATCHED BY SOURCE THEN
    DELETE;
```

Explanation:

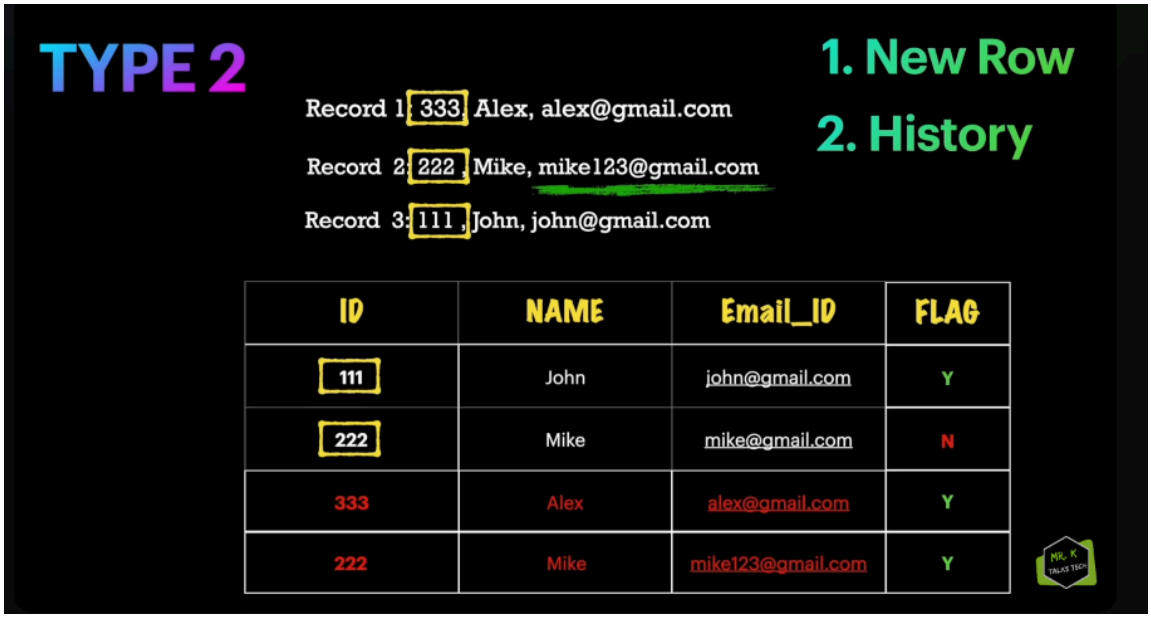
- **Matching rows** (when `sale_id` is the same in both `sales_fact` and `new_sales`) will be updated with the latest `quantity_sold` and `total_amount`.
- **Non-matching rows** in `new_sales` will be inserted into `sales_fact` to reflect new sales.
- **Rows in** `sales_fact` that do not exist in `new_sales` will be deleted (which may represent outdated data, like canceled sales).

5. We have two types

6. IN type 1…if we have new unique primary key ..then it will add a new row

If the key is already present and if there any changes it will update it…
If it will ignore the duplicates

7. It is similar to SCD types



8. We will use type 1 in our proj

```
from pyspark.sql.utils import AnalysisException

try:

    table_name = 'bing_lake_db.tbl_latest_news'

    df_cleaned_final.write.format("delta").saveAsTable(table_name)

except AnalysisException:

    print("Table Already Exists")

    df_cleaned_final.createOrReplaceTempView("vw_df_cleaned_final")

    spark.sql(f"""  MERGE INTO {table_name} target_table
                    USING vw_df_cleaned_final source_view

                    ON source_view.url = target_table.url

                    WHEN MATCHED AND
                    source_view.title <> target_table.title OR
                    source_view.description <> target_table.description OR
                    source_view.category <> target_table.category OR
                    source_view.image <> target_table.image OR
                    source_view.provider <> target_table.provider OR
                    source_view.datePublished <> target_table.datePublished

                    THEN UPDATE SET *

                """)
```

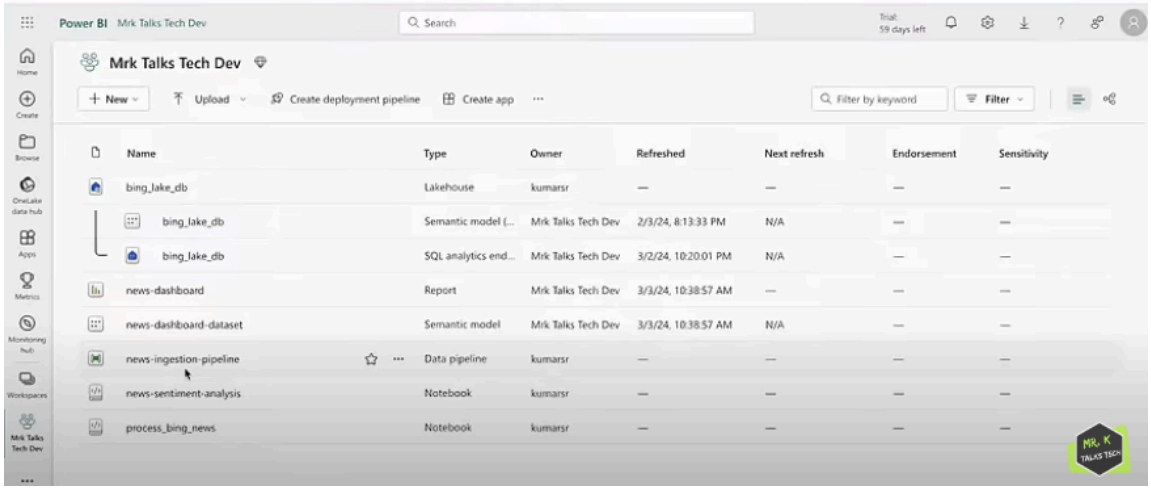
9. Here in our code we use URL as join col as each article has unique URL…if when they are matched.. then we check if any cols has updated or not….if not updated then we leave as it is…cuz this is a duplicate…if there any changes …then we update the entire row again

10. Similarly if there are no matching URL then it is a new item and we insert entire row

```
except AnalysisException:

    print("Table Already Exists")

    df_cleaned_final.createOrReplaceTempView("vw_df_cleaned_final")

    spark.sql(f"""  MERGE INTO {table_name} target_table
                    USING vw_df_cleaned_final source_view

                    ON source_view.url = target_table.url

                    WHEN MATCHED AND
                    source_view.title <> target_table.title OR
                    source_view.description <> target_table.description OR
                    source_view.category <> target_table.category OR
                    source_view.image <> target_table.image OR
                    source_view.provider <> target_table.provider OR
                    source_view.datePublished <> target_table.datePublished

                    THEN UPDATE SET *

                    WHEN NOT MATCHED THEN INSERT *
                """)
```

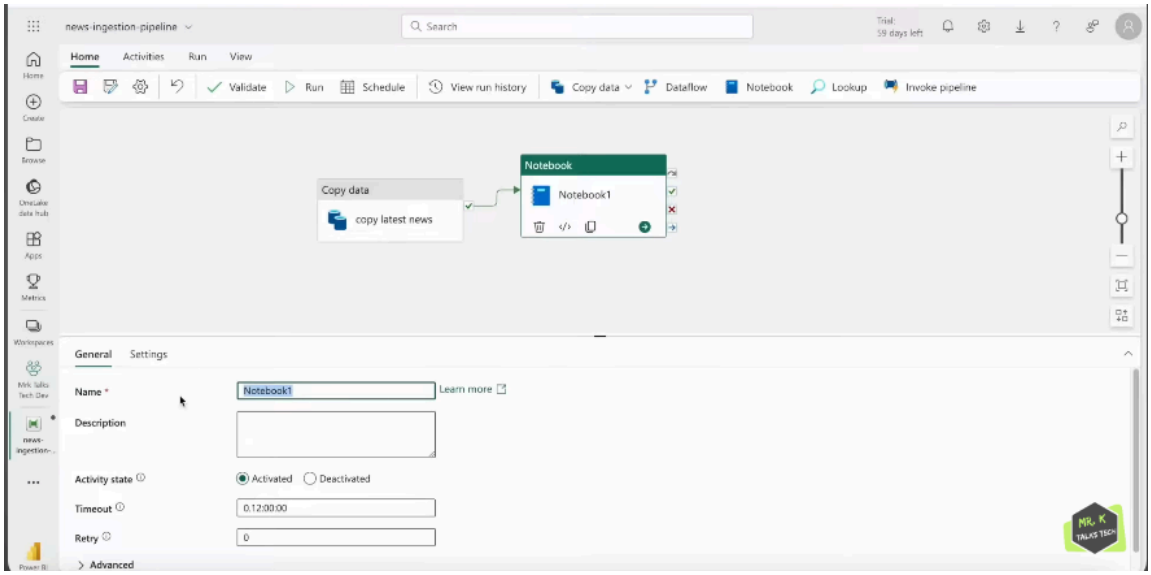
Orchestrating the entire project

1. Now we will build this entire end to end pipeline

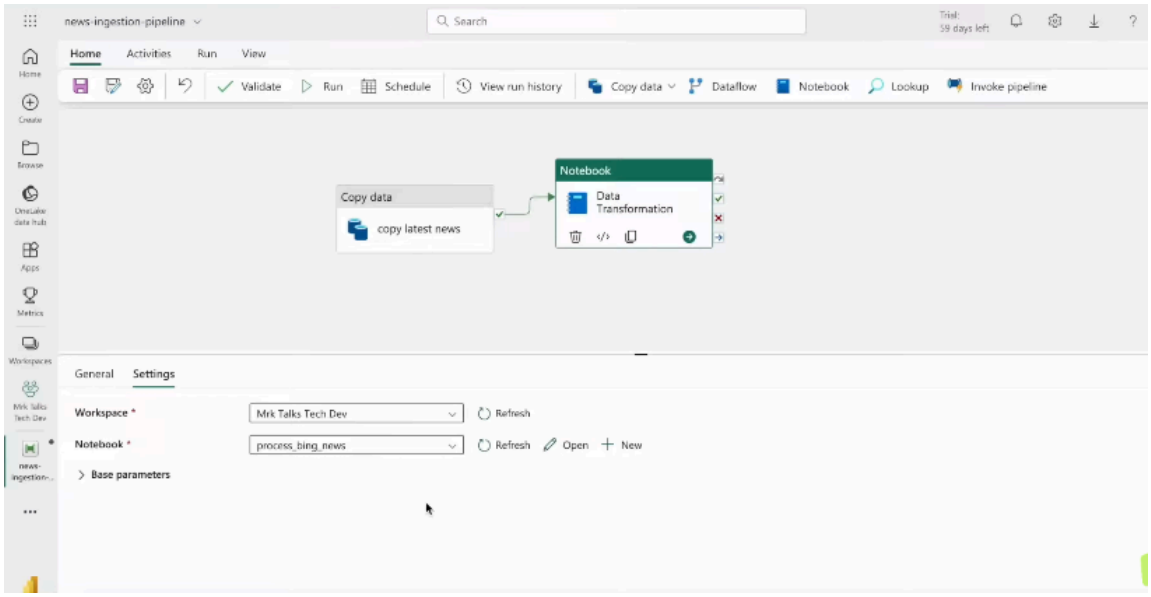


2. First we go to our ingestion pipeline..where we ingest data from Bing API

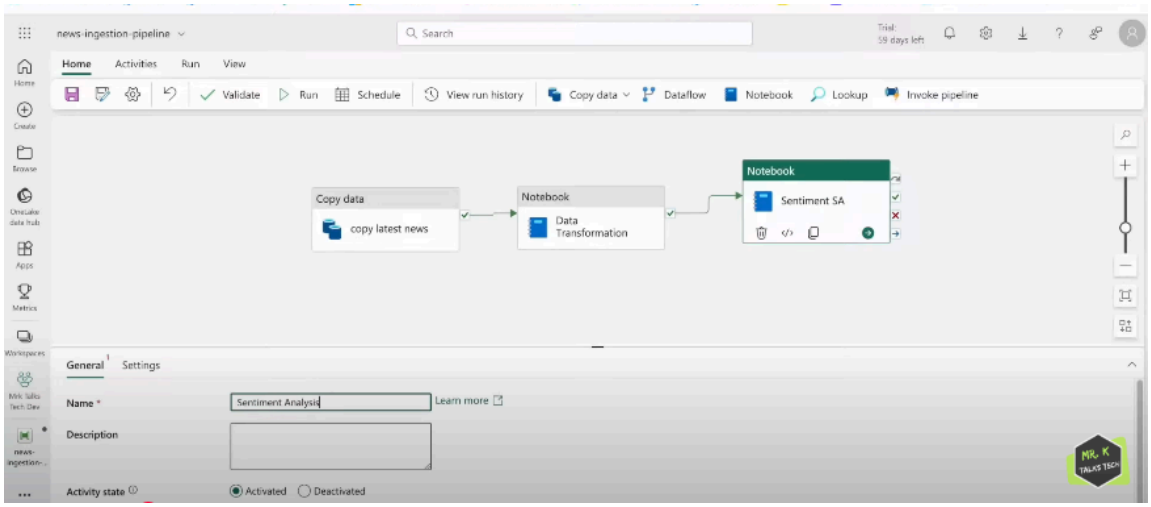
3. Now we will add transformation notebook to this activity

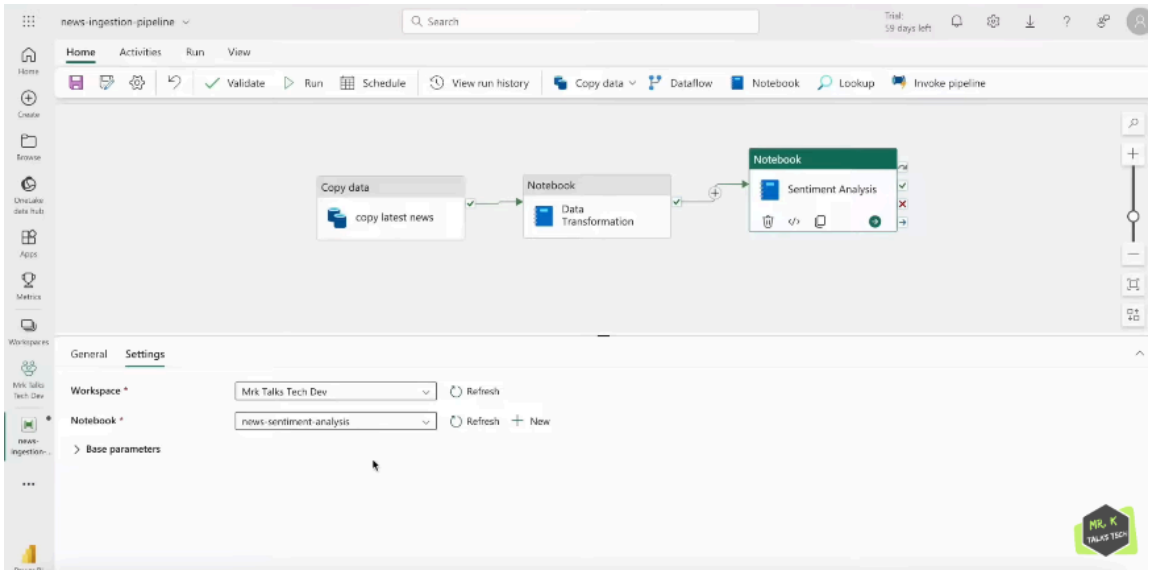


When the copy data is successful then it goes to notebook activity and then we provide the notebook that we have created

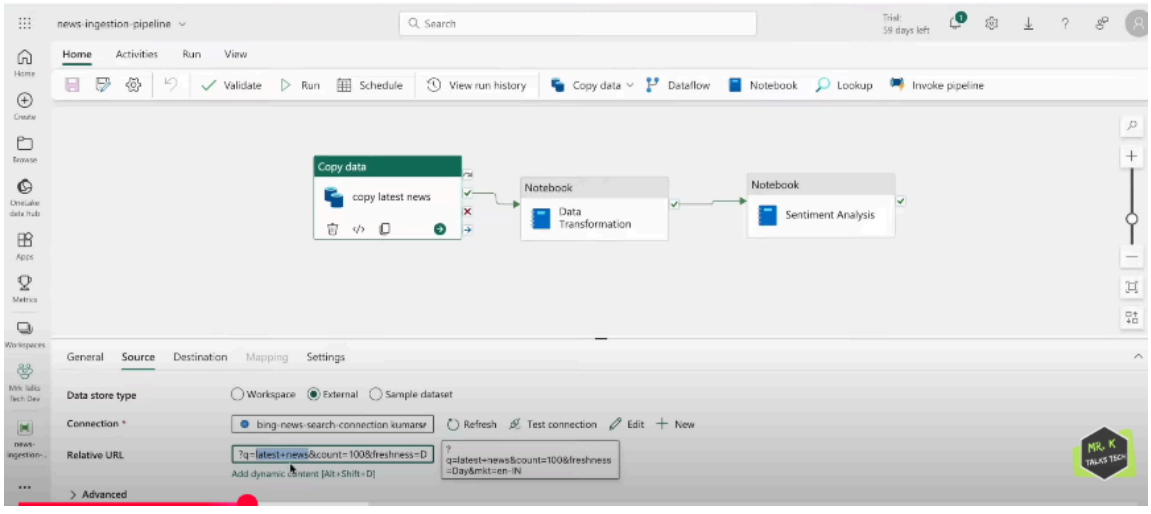


4. Next we will add a sentiment analysis notebook activity



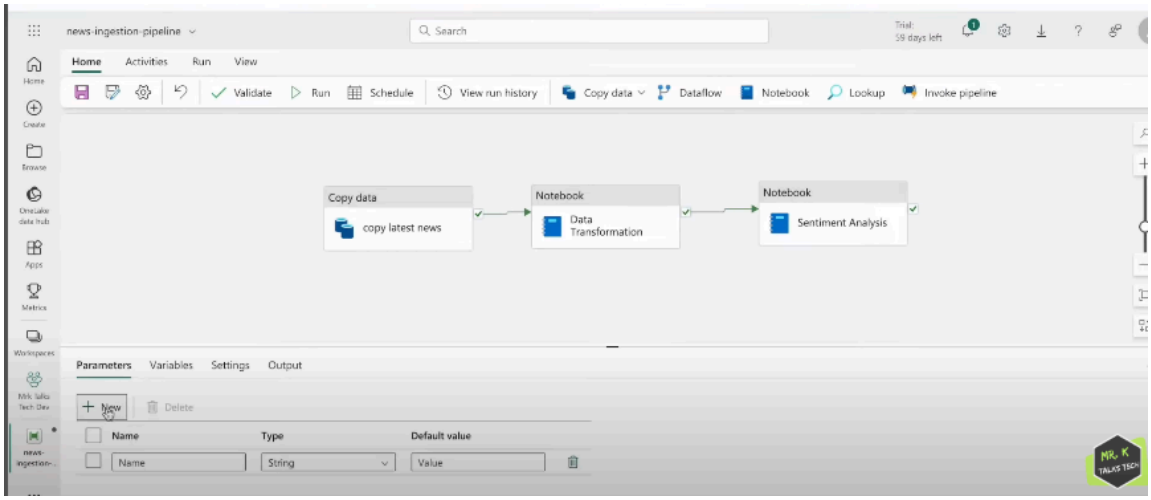


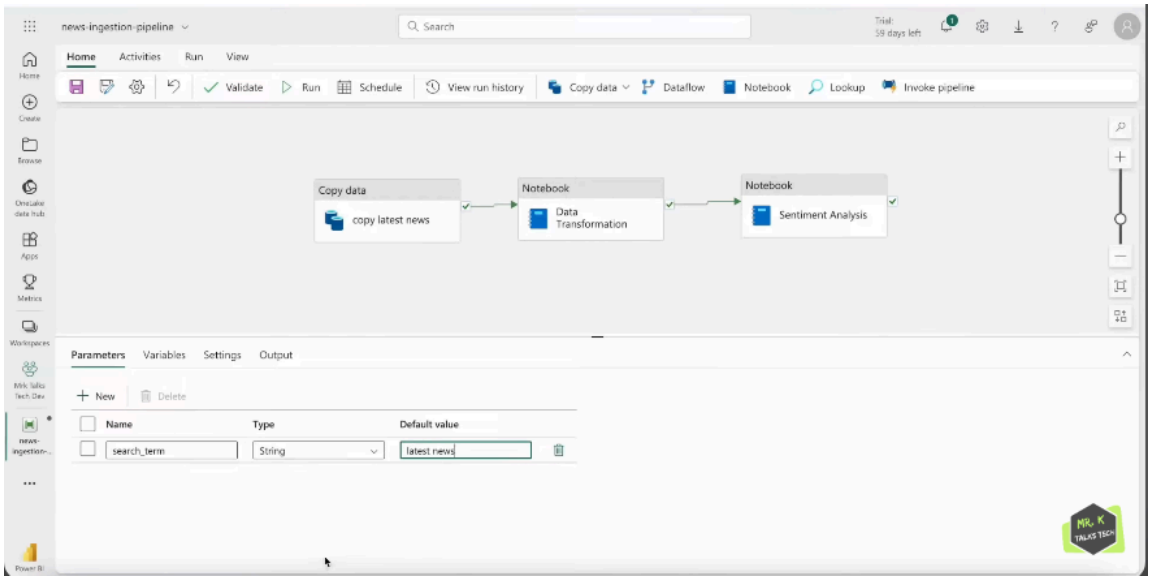
5. This is our entire pipeline
6. Now in the copy activity we have manually encoded the latest_news keyword for bingAPI



ANd we'd like it to be dynamic..by giving a parameter and our freshness argument will gives us the latest news

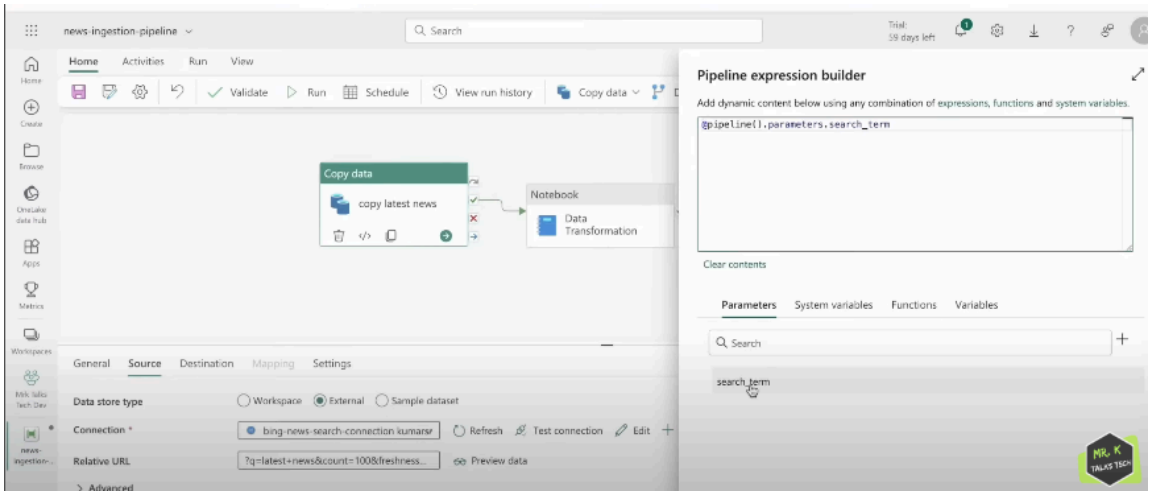
For that …click on white space and click parameters





Next we use this parameter in our relative URL…

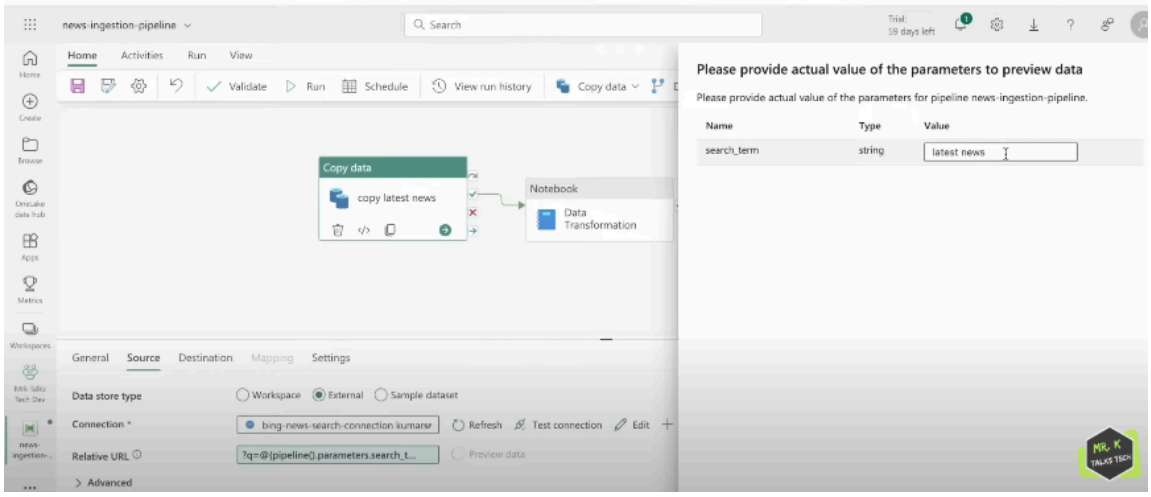
Now go to Add dynamic content and select the param we created…then we'll get the expression..which we can use in url




```
?
q=@pipeline().parameters.search_term&count=100&freshness=Day&mkt=e
n-IN
```

Relative URL :

Now if we preview the data…we can type what we need as value



7. Now we will schedule this pipeline to run every mrng at 6am