A diagram of a company

Description automatically generated

**Environment Setup:**

Created a new resource group in Microsoft azure specific to the project.

Bing Api resource (**Bing Search v7)** is then created inside the resource group which is used as data source.

A screenshot of a computer

Description automatically generated

After that as part of environment set up, we created a Lakehouse database in the data engineering component in the fabric workspace.

A screenshot of a computer

Description automatically generated

**Data Ingestion using Data Factory:**

For Data Ingestion, I connected to Bing API(rest API) and using data factory pipelines in the fabric successfully ingested into one lake data storage in fabric as Json file format.

Since here rest APIs are used, headers are used to authenticate.

Key word latest news is passed to request latest news through API.

And freshness parameter is used to get news for the last 24 hours.

Next parameter used is count, to request 100 latest news articles.

Next, we used mkt parameter to select latest news related to India region only.

News was collected for last 24 hours and India country news were collected successfully in file format onto one lake data storage in Microsoft fabric.

Relative URL used is “**?q=latest+news&count=100&freshness=Day&mkt=en-IN**”

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Data Transformation (Incremental Load):**

The data transformation process involved reading a JSON file containing news articles into a Spark DataFrame, selecting only the 'value' column with the JSON objects, and exploding/flattening the nested JSON data into separate rows. The exploded DataFrame was then converted to a list of JSON strings. Each JSON string was parsed, and relevant fields like title, description, category, URL, image, provider, and datePublished were extracted into separate Python lists. These lists were combined and used to create a final DataFrame with a defined schema, where the 'datePublished' column was formatted to 'dd-MMM-yyyy' format. Finally, this cleaned and structured DataFrame was written as a Delta table to the 'bing\_lake\_db.tbl\_latest\_news' table in the lakehouse database for further analysis and use cases.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Incremental Load:**

Type 1 merge concept in Data Warehousing is used for this project.

So, we need only the latest news and new records to write the old records.

A screenshot of a computer

Description automatically generated

**Sentiment Analysis using Synapse Data Science (Incremental Load):**

Pre-Trained Machine Learning Model is used to do sentiment analysis in Synapse Data Science.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**PowerBI Reporting:**

A new semantic model is created on our sentiment analysis delta table that is there in our lakehouse.

A new filter is added for date published column to get only the latest news.

A measure is created on sentiment column for positive, negative, and neutral sentiments using DAX code to get percentage of each sentiment news from the latest news articles.

And Cards are used to display each sentiment percentage for the latest news i.e. last 24 hours news.

A screenshot of a computer

Description automatically generated

**Building Pipelines:**

Now we are integrating all the activities that we have done before and building a pipeline.

After extracting news using copy activity, it is connected to data transformation notebook activity.

After success of notebook activity, it is connected to sentiment analysis notebook activity.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

To automate this flow even better, in copy data activity in source data:

We are creating a new parameter called search\_term.

A screenshot of a computer

Description automatically generated

Next in copy data activity in source we are passing that parameter.

“**?q=@{pipeline().parameters.search\_term}&count=100&freshness=Day&mkt=en-IN**”

A screenshot of a computer

Description automatically generated

When we preview data, we get all the latest news.

Next, we are scheduling the pipeline, to run at exactly 6:00am in the morning every day.

By doing this we can see everyday in the morning we can view powerbi report and see associated sentiment.

Next configurations are made to schedule the pipeline.

A screenshot of a computer

Description automatically generated

To test this end-to-end pipeline, pipeline was run manually by clicking on the run button.

A screenshot of a computer

Description automatically generated

Everyday as per scheduled trigger pipeline runs automatically and data is populated and PowerBI report is updated since this is end to end process.

**Setting up alerts using Data Activator:**

I set up alerts using Data Activator for a Power BI dashboard that monitors negative sentiment news. I wrote a condition to automatically send alerts via Teams messages if the percentage of negative sentiment news exceeds 20% for news published in the last 24 hours.