Test Case Documentation for Fabric Project

Table of Contents

1. Introduction

- **Project Name**: Fabric Pipeline Project
- **Prepared By**: Kummeri Ajay Kumar
- **Date**: 19-09-2024

This document provides a detailed overview of the test cases for the Fabric project. The project involves data transfer from Azure Data Lake Storage (ADLS) to Microsoft Fabric's One Lake, data transformation using PySpark, table creation or updating (dimension and fact tables), archiving, and updating Power BI reports. The purpose of the test cases is to ensure each pipeline step functions correctly, handles errors, and achieves the desired outcomes.

2. Test Plan

- **Objective**: The objective of the test plan is to validate the functionality, performance, and reliability of the dynamic data pipeline.
- **Scope**: The test cases include unit tests for individual steps, integration tests for the pipeline workflow, and end-to-end tests for the complete execution, covering:
  - Data copying from ADLS to One Lake. ○ Data transformation using PySpark (handling null values, schema validation, and duplicates).
  - Creation and updating of dimension and fact tables (SCD1, SCD2).
  - Archiving raw data.
  - Refreshing the semantic model in Power BI and updating reports.

3. Test Environment

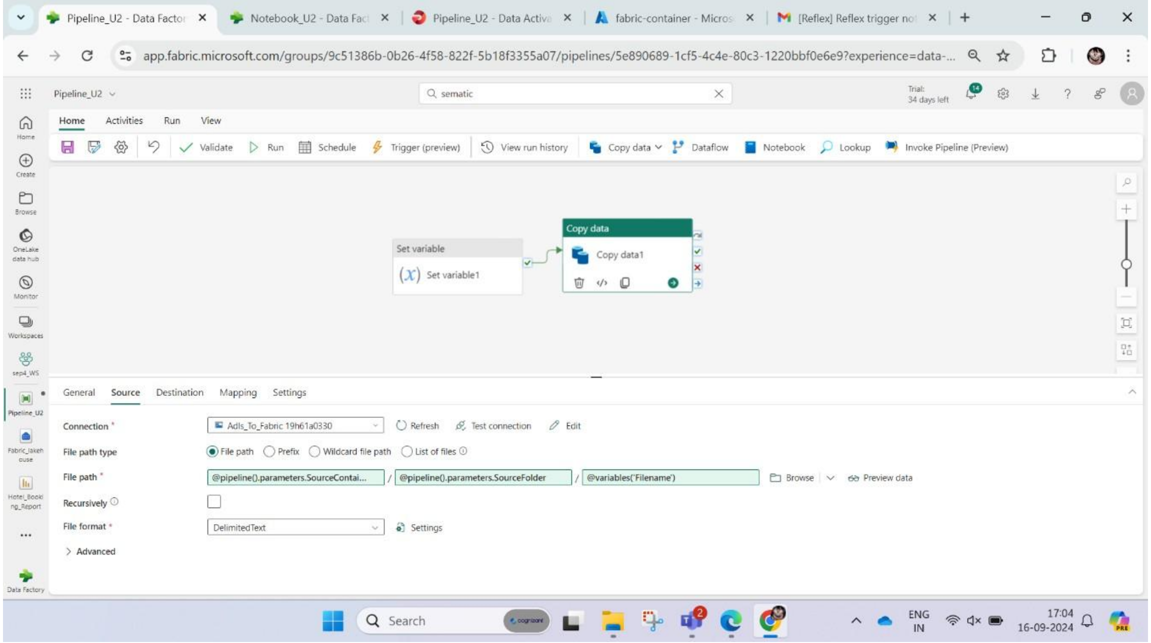
- **Hardware**: Cloud-based environment on Microsoft Fabric.
- **Software**: Microsoft Fabric (with One Lake), PySpark, Power BI.
- **Data**: Mock datasets representing customer, hotel, and reservation data, including scenarios with null values, duplicates, and schema variations.

4. Test Cases

Test Case 1: Copying Data from ADLS to One Lake

- **Description**: Ensure data is copied from Azure Data Lake Storage (ADLS) to One Lake with valid source, destination, and an empty folder.
- **Preconditions**: Source and destination paths are accessible and valid.
- **Steps**:

1. Connect and configure the source parameters and variables.



General | Source | Destination | Mapping | Settings

Connection * Adls_To_Fabric 19h61a0330

File path type ● File path ○ Prefix ○ Wildcard file path ○ List of files

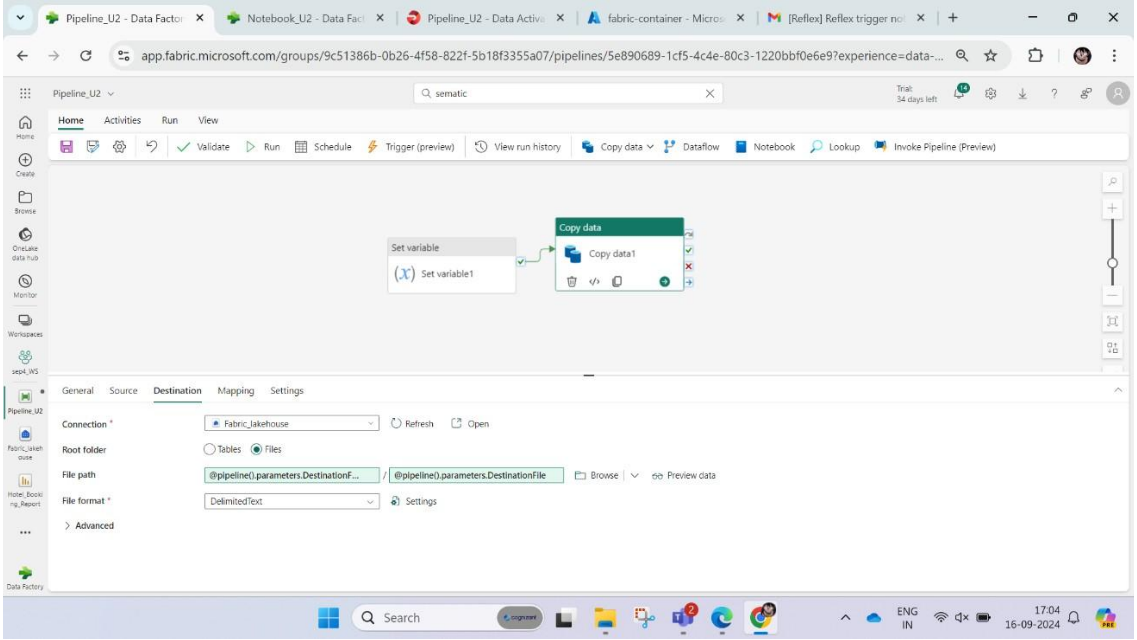
File path * @pipeline().parameters.SourceContai... / @pipeline().parameters.SourceFolder / @variables('Filename')

Recursively □

File format * DelimitedText

Figure 1. Assigning parameters to Source (ADLS) path

2. Configure the destination parameters and variables.



General | Source | Destination | Mapping | Settings

Connection * Fabric_lakehouse

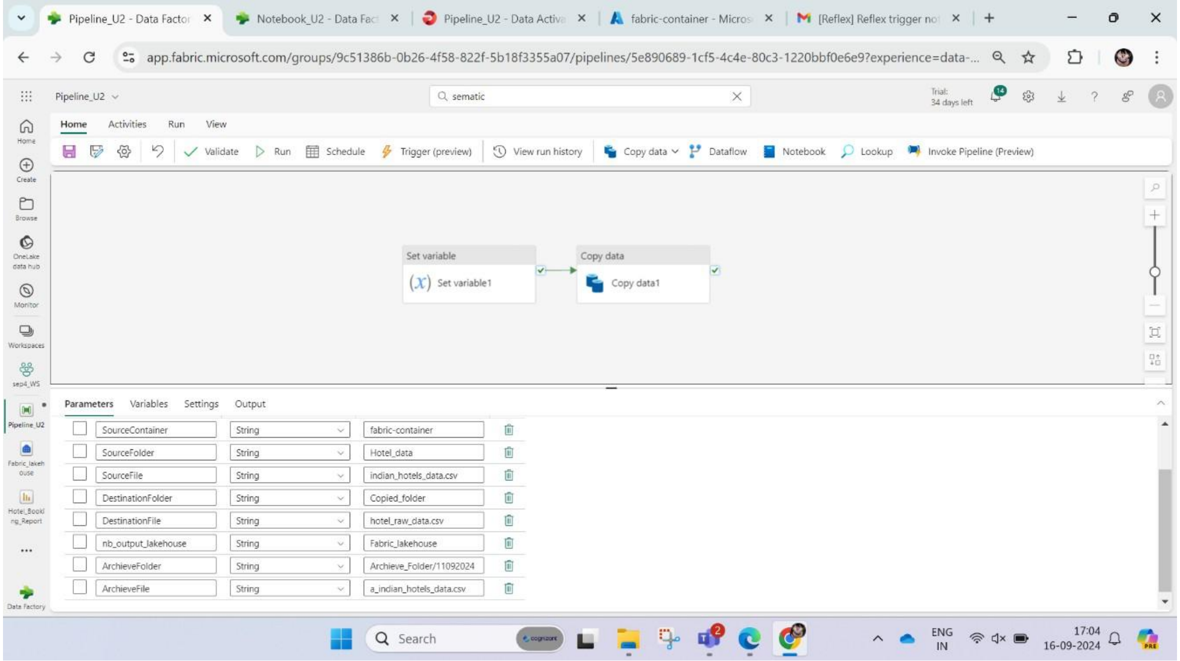
Root folder ○ Tables ● Files

File path @pipeline().parameters.DestinationF... / @pipeline().parameters.DestinationFile

File format * DelimitedText

Figure 2. Assigning parameters to Destination

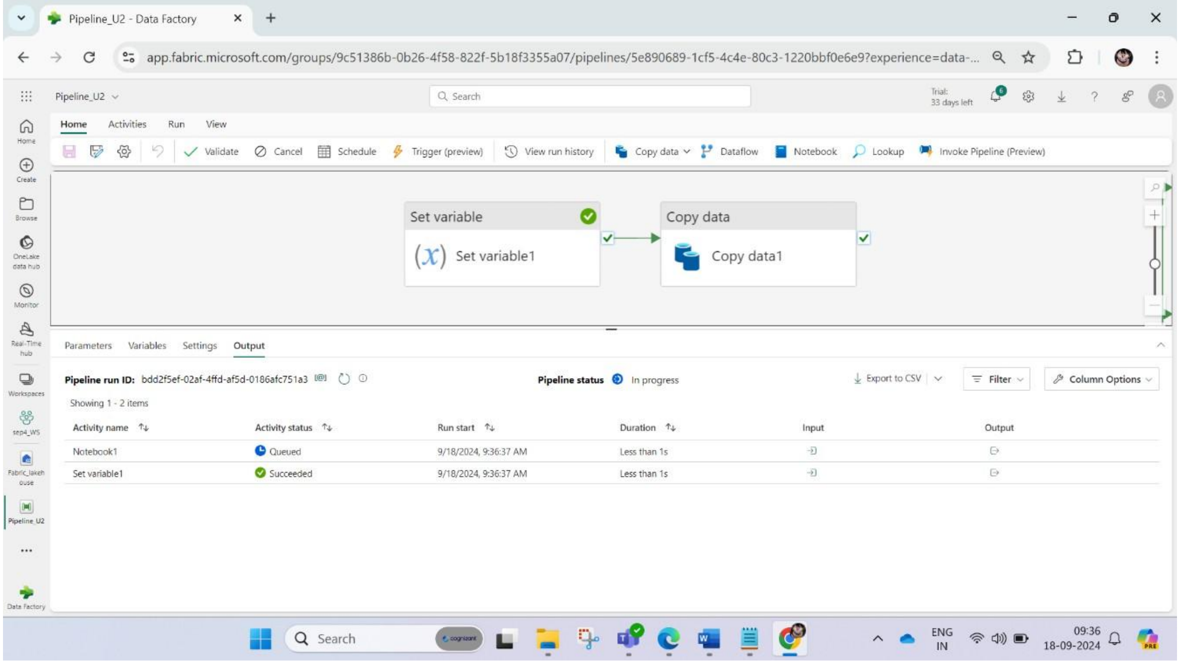
3. Name the custom source (ADLS) and destination (One Lake) paths via Parameters



□	SourceContainer	String	fabric-container
□	SourceFolder	String	Hotel_data
□	SourceFile	String	indian_hotels_data.csv
□	DestinationFolder	String	Copied_folder
□	DestinationFile	String	hotel_raw_data.csv
□	nb_output_lakehouse	String	Fabric_lakehouse
□	ArchieveFolder	String	Archieve_Folder/11092024
□	ArchieveFile	String	a_indian_hotels_data.csv

Figure 3. Assigning values to Parameters

4. Run the pipeline to copy data.



Pipeline run ID: bdd2f5ef-02af-4ffd-af5d-0186afc751a3

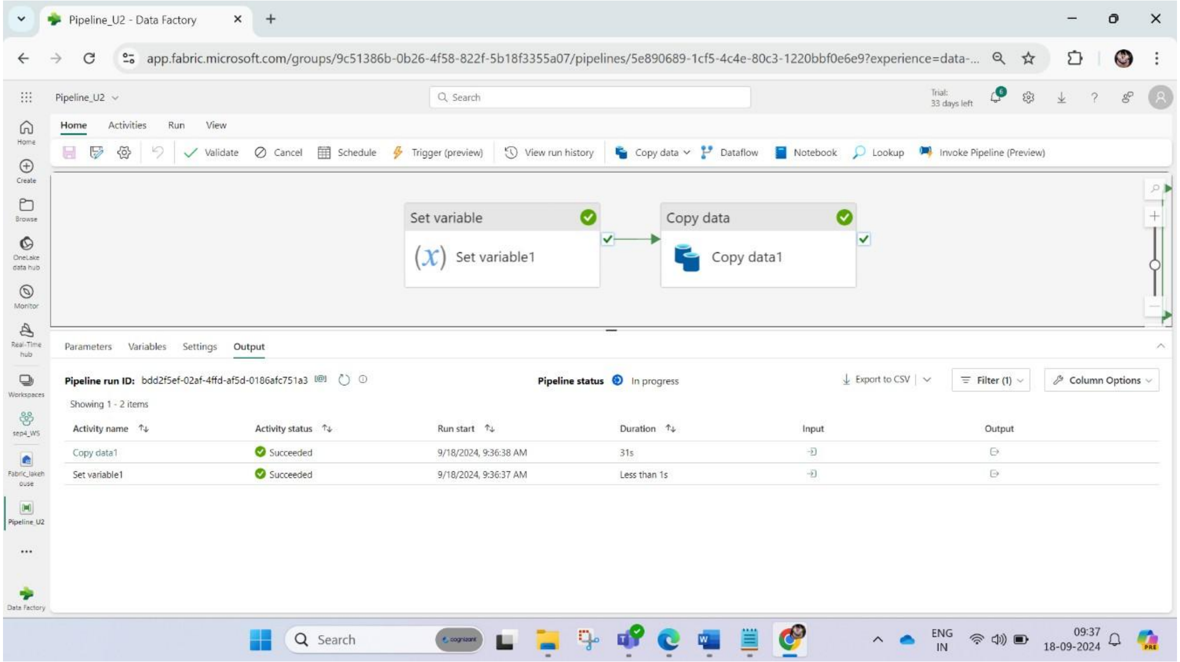
Pipeline status In progress

Showing 1 - 2 items

Activity name	Activity status	Run start	Duration	Input	Output
Notebook1	Queued	9/18/2024, 9:36:37 AM	Less than 1s		
Set variable1	Succeeded	9/18/2024, 9:36:37 AM	Less than 1s		

Figure 4. Running the pipeline

5. Verify that data is copied to the destination folder without errors.



Pipeline run ID: bdd2f5ef-02af-4ffd-af5d-0186afc751a3

Pipeline status In progress

Showing 1 - 2 items

Activity name	Activity status	Run start	Duration	Input	Output
Copy data1	Succeeded	9/18/2024, 9:36:38 AM	31s		
Set variable1	Succeeded	9/18/2024, 9:36:37 AM	Less than 1s		

Figure 5. Pipeline ran Successfully and copied data

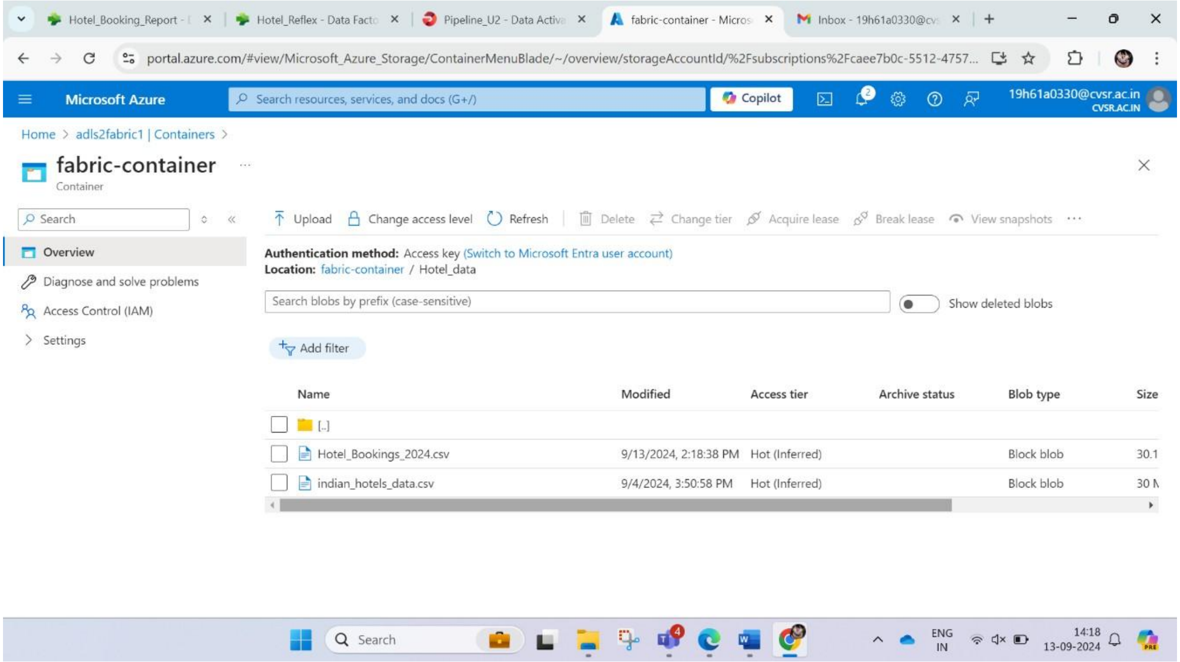
- **Expected Result**: Data should be copied successfully from ADLS to One Lake without any errors or discrepancies.

- **Actual Result**: Data is copied successfully from ADLS to One Lake without any errors at Given destination in One Lake

Test Case 2: Trigger on New File Upload

- **Description**: Ensure the pipeline trigger works when a new file is uploaded to the directory.
- **Preconditions**: Trigger is configured for the directory.
- **Steps**:

1. Upload a new file (Hotel_ Bookings_2024.csv) to the specified directory(fabriccontainer/Hotel_data/).



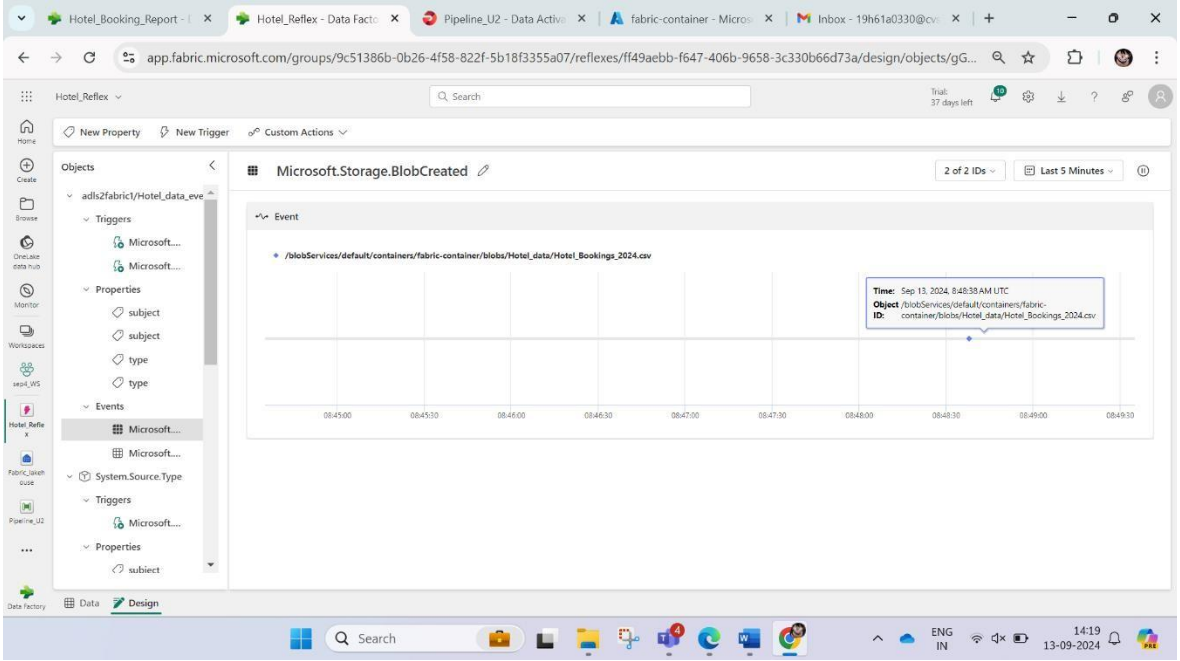
portal.azure.com/#view/Microsoft_Azure_Storage/ContainerMenuBlade/~/overview/storageAccountId/%2Fsubscriptions%2Fcaee7b0c-5512-4757...
Microsoft Azure
Home > adls2fabric1 | Containers >
fabric-container
Container
Overview
Diagnose and solve problems
Access Control (IAM)
Settings
Upload
Change access level
Refresh
Delete
Change tier
Acquire lease
Break lease
View snapshots
Show deleted blobs
Add filter

Authentication method: Access key (Switch to Microsoft Entra user account)
Location: fabric-container / Hotel_data

Name	Modified	Access tier	Archive status	Blob type	Size
[..]					
Hotel_Bookings_2024.csv	9/13/2024, 2:18:38 PM	Hot (Inferred)		Block blob	30.1
indian_hotels_data.csv	9/4/2024, 3:50:58 PM	Hot (Inferred)		Block blob	30 M

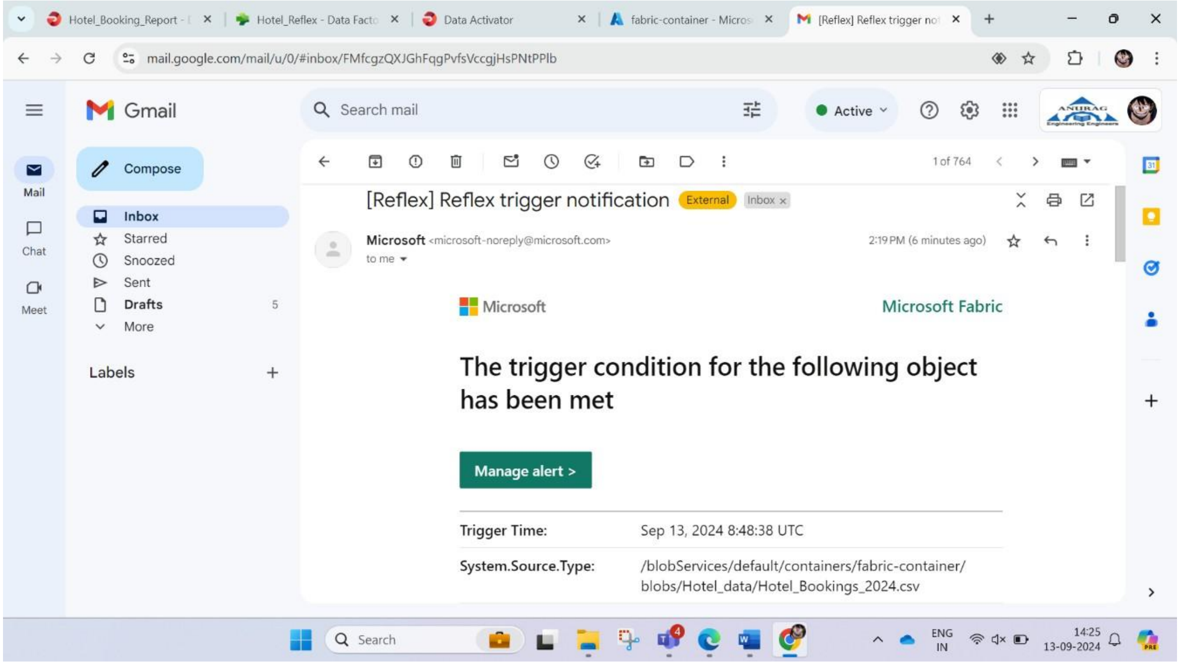
Figure 6. File Uploaded to ADLS path

2. Monitor the pipeline whether triggered or not.



Time: Sep 13, 2024, 8:48:38 AM UTC
Object ID: /blobServices/default/containers/fabric-container/blobs/Hotel_data/Hotel_Bookings_2024.csv

Figure 7. Monitoring Microsoft blob created Trigger



[Reflex] Reflex trigger notification External Inbox

Microsoft <microsoft-noreply@microsoft.com>
to me

2:19 PM (6 minutes ago)

Microsoft Microsoft Fabric

The trigger condition for the following object has been met

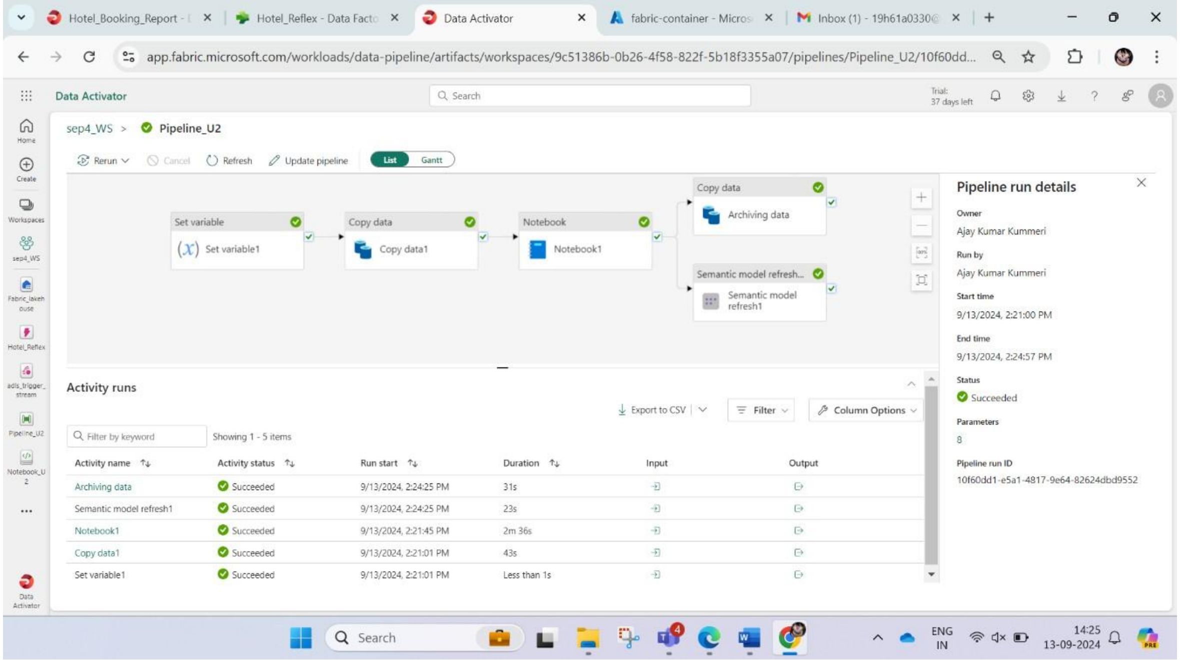
Manage alert >

Trigger Time: Sep 13, 2024 8:48:38 UTC

System.Source.Type: /blobServices/default/containers/fabric-container/blobs/Hotel_data/Hotel_Bookings_2024.csv

Figure 8. Triggered notification

3. Monitor the pipeline run after triggering.



Activity runs

Activity name	Activity status	Run start	Duration	Input	Output
Archiving data	✓ Succeeded	9/13/2024, 2:24:25 PM	31s		
Semantic model refresh1	✓ Succeeded	9/13/2024, 2:24:25 PM	23s		
Notebook1	✓ Succeeded	9/13/2024, 2:21:45 PM	2m 36s		
Copy data1	✓ Succeeded	9/13/2024, 2:21:01 PM	43s		
Set variable1	✓ Succeeded	9/13/2024, 2:21:01 PM	Less than 1s		

Pipeline run details

Owner
Ajay Kumar Kummeri

Run by
Ajay Kumar Kummeri

Start time
9/13/2024, 2:21:00 PM

End time
9/13/2024, 2:24:57 PM

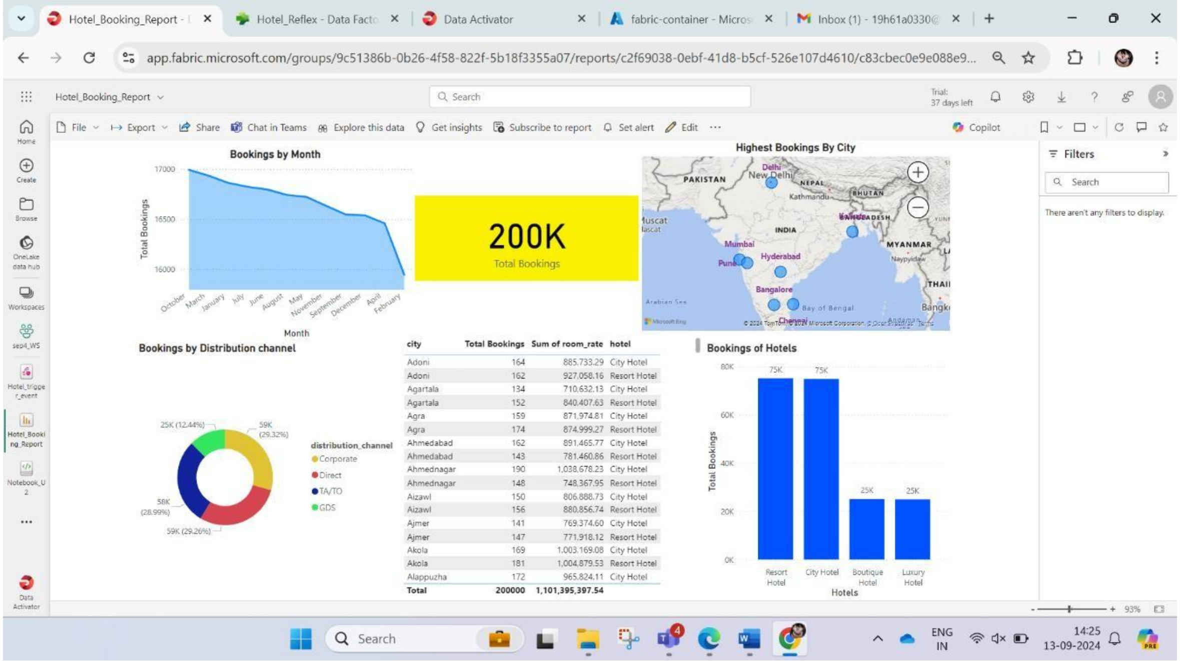
Status
✓ Succeeded

Parameters
8

Pipeline run ID
10f60dd1-e5a1-4817-9e64-82624dbd9552

Figure 9. Successfully Triggered Pipeline

4. verify the data update after triggering.



city	Total Bookings	Sum of room_rate	hotel
Adoni	164	885,733.29	City Hotel
Adoni	162	927,058.16	Resort Hotel
Agartala	134	710,632.13	City Hotel
Agartala	152	840,407.63	Resort Hotel
Agra	159	871,974.81	City Hotel
Agra	174	874,999.27	Resort Hotel
Ahmedabad	162	891,465.77	City Hotel
Ahmedabad	143	781,460.86	Resort Hotel
Ahmednagar	190	1,038,678.23	City Hotel
Ahmednagar	148	748,367.95	Resort Hotel
Aizawl	150	806,888.73	City Hotel
Aizawl	156	880,856.74	Resort Hotel
Ajmer	141	769,374.60	City Hotel
Ajmer	147	771,918.12	Resort Hotel
Akola	169	1,003,169.08	City Hotel
Akola	181	1,004,879.53	Resort Hotel
Alappuzha	172	965,824.11	City Hotel
Total	200000	1,101,395,397.54	

Figure 10. Power Bi Triggered Report -1

- **Expected Result**: Pipeline Should be triggered upon file upload and data should be updated.

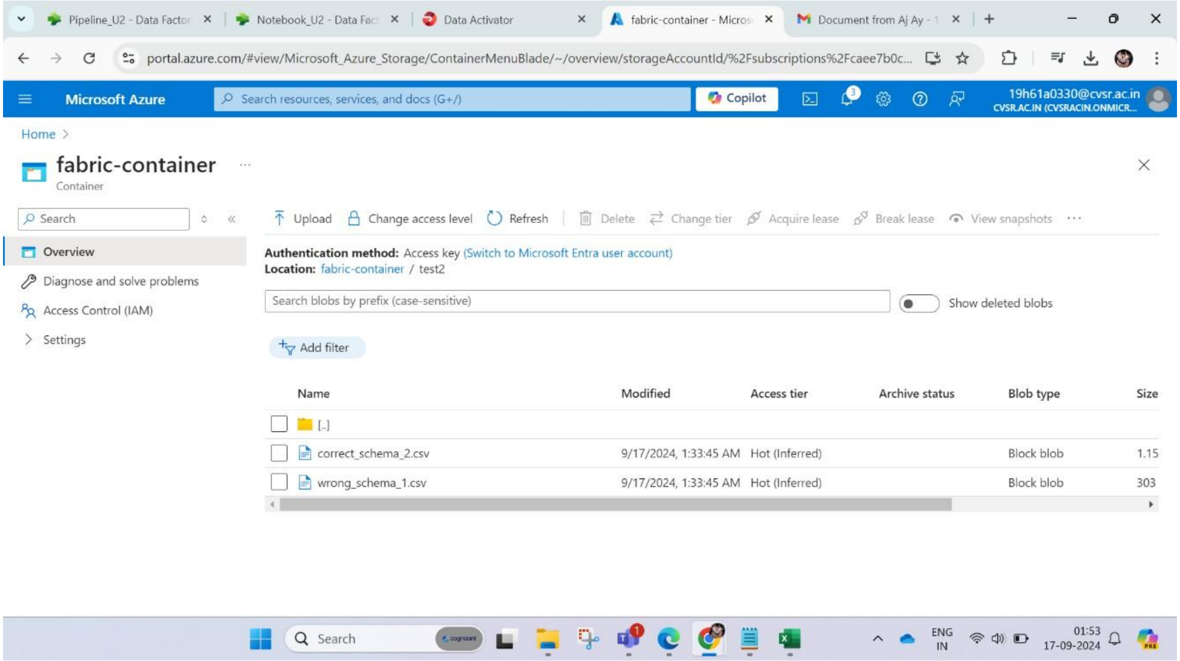
- **Actual Result**: Pipeline is triggered upon file upload and data update reflect in monitor and get updated in report.

Test Case 3: Notebook Schema Mismatch or Match

- **Description**: Ensure the schema of the notebooks matches the expected schema.
- **Preconditions**: Expected schema is defined.
- **Steps**:

1. Upload a wrong schema file to Azure ADLS

   (fabric-container/test2/wrong_schema_1.csv)



fabric-container
Container

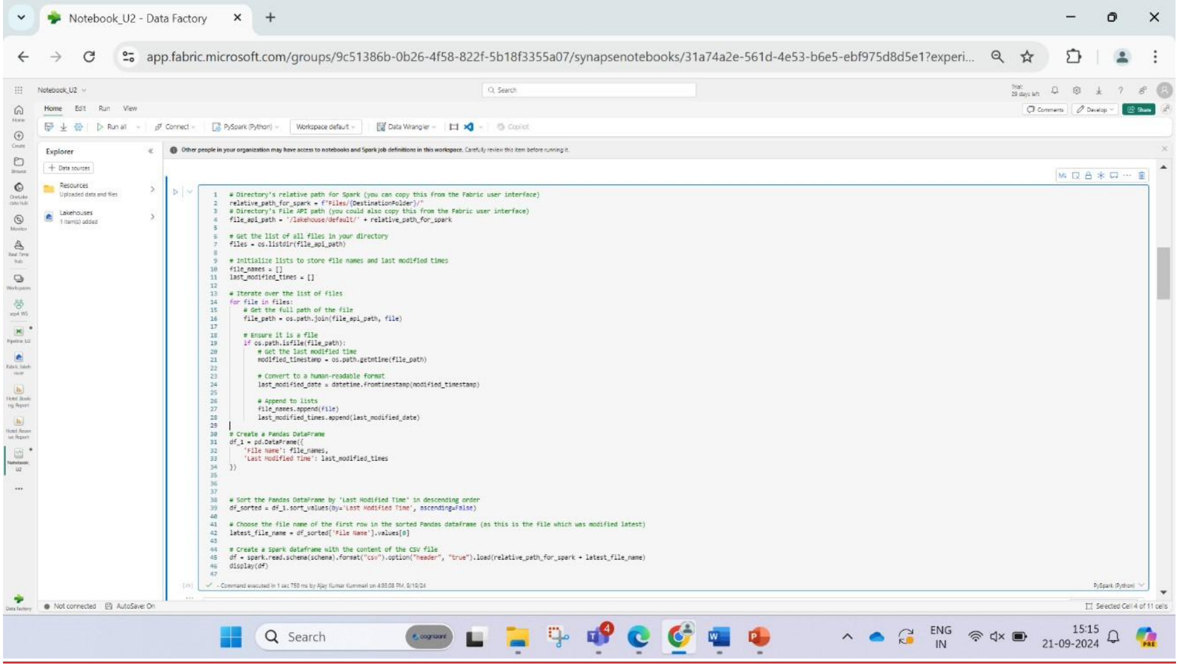
Authentication method: Access key (Switch to Microsoft Entra user account)
Location: fabric-container / test2

Name	Modified	Access tier	Archive status	Blob type	Size
[..]					
correct_schema_2.csv	9/17/2024, 1:33:45 AM	Hot (Inferred)		Block blob	1.15
wrong_schema_1.csv	9/17/2024, 1:33:45 AM	Hot (Inferred)		Block blob	303

portal.azure.com/#view/Microsoft_Azure_Storage/ContainerMenuBlade/~/overview/storageAccountId/%2Fsubscriptions%2Fcaee7b0c...
Microsoft Azure
Home
Overview
Diagnose and solve problems
Access Control (IAM)
Settings
Upload
Change access level
Refresh
Delete
Change tier
Acquire lease
Break lease
View snapshots
Search blobs by prefix (case-sensitive)
Show deleted blobs
Add filter
01:53
17-09-2024

Figure 11. Wrong_schema.csv File Uploading

2. When you run the pipeline the Notebook extract recently Copied file from ADLS to One Fabric whether triggered or manual uploaded file.

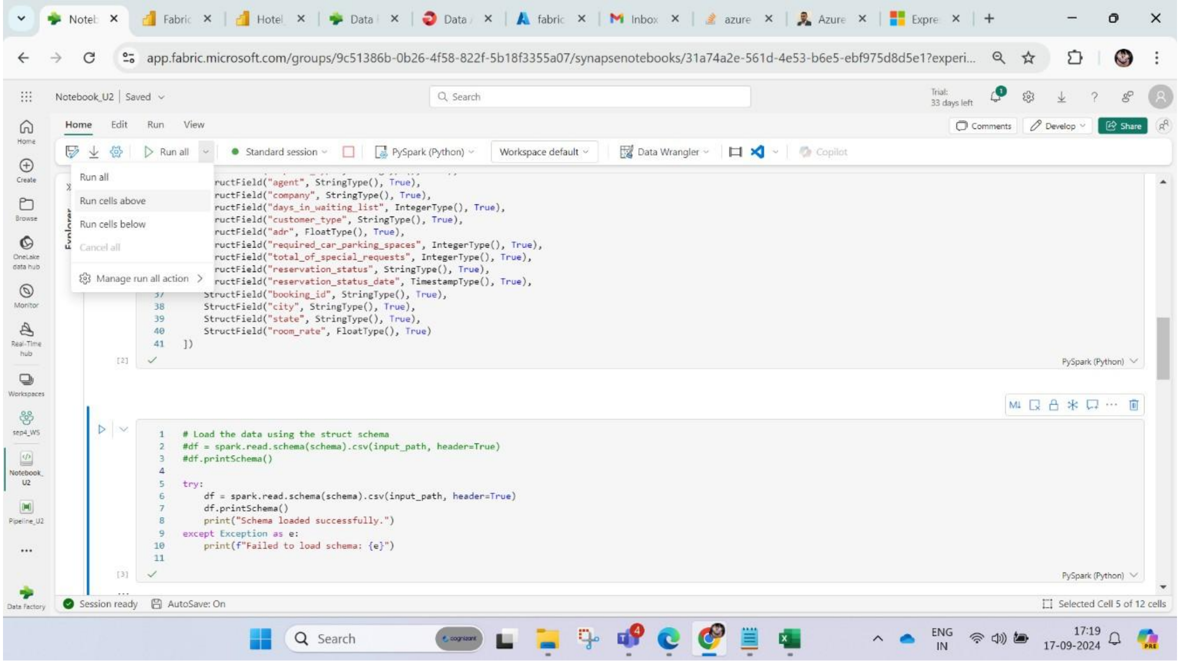


```
# Directory's relative path for Spark (you can copy this from the Fabric user interface)
relative_path_for_spark = f"Files/{DestinationFolder}/"
# Directory's File API path (you could also copy this from the Fabric user interface)
file_api_path = '/lakehouse/default/' + relative_path_for_spark

# Get the list of all files in your directory
files = os.listdir(file_api_path)

# Initialize lists to store file names and last modified times
file_names = []
last_modified_times = []

# Iterate over the list of files
for file in files:
    # Get the full path of the file
    file_path = os.path.join(file_api_path, file)

    # Ensure it is a file
    if os.path.isfile(file_path):
        # Get the last modified time
        modified_timestamp = os.path.getmtime(file_path)

        # Convert to a human-readable format
        last_modified_date = datetime.fromtimestamp(modified_timestamp)

        # Append to lists
        file_names.append(file)
        last_modified_times.append(last_modified_date)

# Create a Pandas DataFrame
df_1 = pd.DataFrame({
    'File Name': file_names,
    'Last Modified Time': last_modified_times
})



# Sort the Pandas DataFrame by 'Last Modified Time' in descending order
df_sorted = df_1.sort_values(by='Last Modified Time', ascending=False)

# Choose the file name of the first row in the sorted Pandas dataframe (as this is the file which was modified latest)
latest_file_name = df_sorted['File Name'].values[0]

# Create a Spark dataframe with the content of the csv file
df = spark.read.schema(schema).format("csv").option("header", "true").load(relative_path_for_spark + latest_file_name)
display(df)
```

- Command executed in 1 sec 750 ms by Ajay Kumar Kummeri on 4:03:08 PM, 9/19/24

Notebook_U2 - Data Factory
app.fabric.microsoft.com/groups/9c51386b-0b26-4f58-822f-5b18f3355a07/synapsenotebooks/31a74a2e-561d-4e53-b6e5-ebf975d8d5e1?experi...
Home
Edit
Run
View
Run all
Connect
PySpark (Python)
Workspace default
Data Wrangler
Copilot
Other people in your organization may have access to notebooks and Spark job definitions in this workspace. Carefully review this item before running it.
Explorer
Data sources
Resources
Lakehouses
Not connected
AutoSave: On
15:15
21-09-2024

Figure 12. Notebook - 1

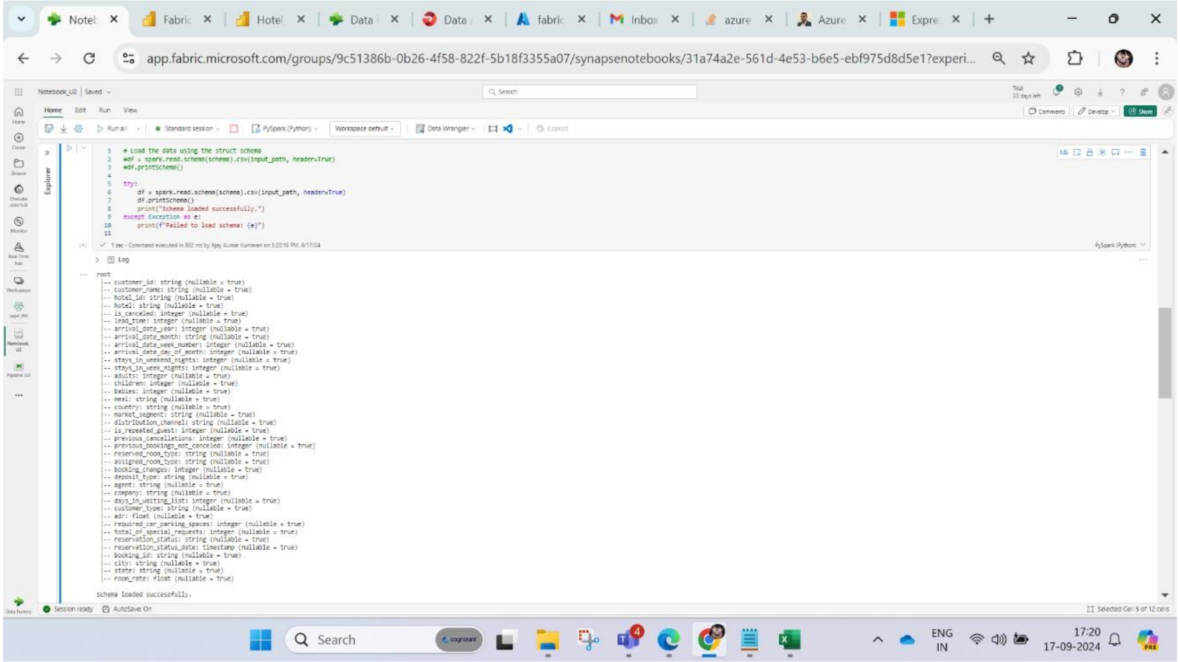


```
ructField("agent", StringType(), True),
ructField("company", StringType(), True),
ructField("days_in_waiting_list", IntegerType(), True),
ructField("customer_type", StringType(), True),
ructField("adr", FloatType(), True),
ructField("required_car_parking_spaces", IntegerType(), True),
ructField("total_of_special_requests", IntegerType(), True),
ructField("reservation_status", StringType(), True),
ructField("reservation_status_date", TimestampType(), True),
    StructField("booking_id", StringType(), True),
    StructField("city", StringType(), True),
    StructField("state", StringType(), True),
    StructField("room_rate", FloatType(), True)
])
```

```
# Load the data using the struct schema
#df = spark.read.schema(schema).csv(input_path, header=True)
#df.printSchema()

try:
    df = spark.read.schema(schema).csv(input_path, header=True)
    df.printSchema()
    print("Schema loaded successfully.")
except Exception as e:
    print(f"Failed to load schema: {e}")
```

Figure 13. Notebook Run - 2

3. Compare the schema of the output with the Correct schema and verify whether the wrong_schema1.csv is read in notebook.

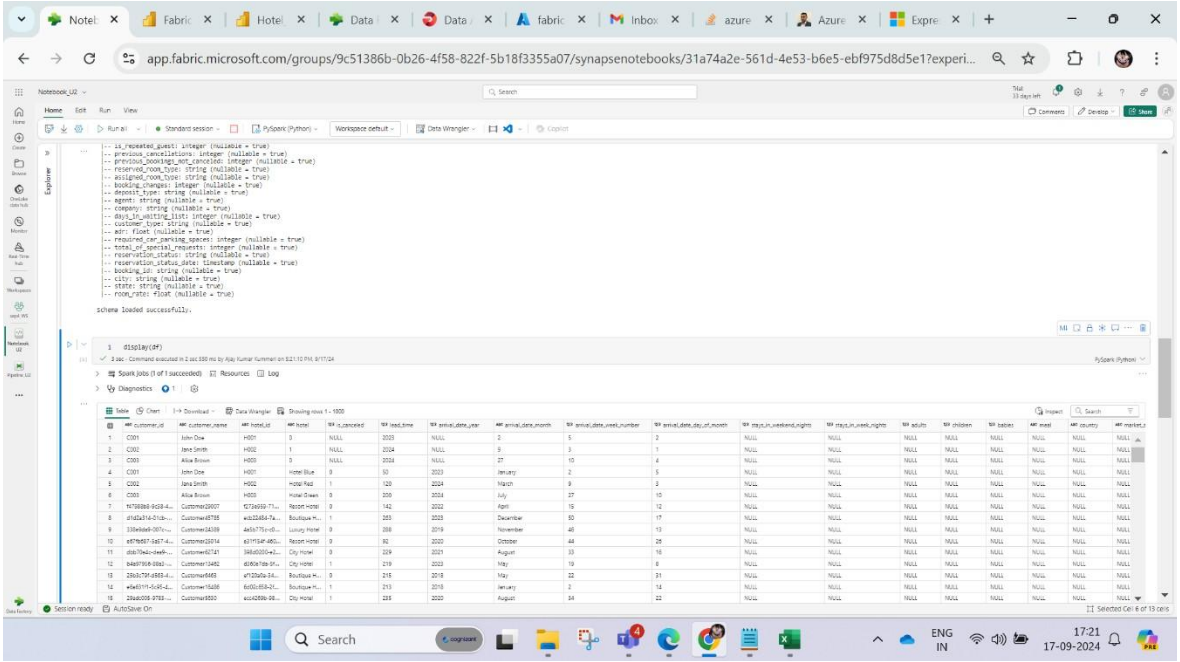


```
# Load the data using the struct schema
#df = spark.read.schema(schema).csv(input_path, header=True)
#df.printSchema()

try:
    df = spark.read.schema(schema).csv(input_path, header=True)
    df.printSchema()
    print("Schema loaded successfully.")
except Exception as e:
    print(f"Failed to load schema: {e}")
```

1 sec - Command executed in 802 ms by Ajay Kumar Kummeri on 5:20:18 PM, 9/17/24

```
root
 |-- customer_id: string (nullable = true)
 |-- customer_name: string (nullable = true)
 |-- hotel_id: string (nullable = true)
 |-- hotel: string (nullable = true)
 |-- is_canceled: integer (nullable = true)
 |-- lead_time: integer (nullable = true)
 |-- arrival_date_year: integer (nullable = true)
 |-- arrival_date_month: string (nullable = true)
 |-- arrival_date_week_number: integer (nullable = true)
 |-- arrival_date_day_of_month: integer (nullable = true)
 |-- stays_in_weekend_nights: integer (nullable = true)
 |-- stays_in_week_nights: integer (nullable = true)
 |-- adults: integer (nullable = true)
 |-- children: integer (nullable = true)
 |-- babies: integer (nullable = true)
 |-- meal: string (nullable = true)
 |-- country: string (nullable = true)
 |-- market_segment: string (nullable = true)
 |-- distribution_channel: string (nullable = true)
 |-- is_repeated_guest: integer (nullable = true)
 |-- previous_cancellations: integer (nullable = true)
 |-- previous_bookings_not_canceled: integer (nullable = true)
 |-- reserved_room_type: string (nullable = true)
 |-- assigned_room_type: string (nullable = true)
 |-- booking_changes: integer (nullable = true)
 |-- deposit_type: string (nullable = true)
 |-- agent: string (nullable = true)
 |-- company: string (nullable = true)
 |-- days_in_waiting_list: integer (nullable = true)
 |-- customer_type: string (nullable = true)
 |-- adr: float (nullable = true)
 |-- required_car_parking_spaces: integer (nullable = true)
 |-- total_of_special_requests: integer (nullable = true)
 |-- reservation_status: string (nullable = true)
 |-- reservation_status_date: timestamp (nullable = true)
 |-- booking_id: string (nullable = true)
 |-- city: string (nullable = true)
 |-- state: string (nullable = true)
 |-- room_rate: float (nullable = true)

Schema loaded successfully.
```

Figure 14. Notebook Schema Result-1



```
 |-- is_repeated_guest: integer (nullable = true)
 |-- previous_cancellations: integer (nullable = true)
 |-- previous_bookings_not_canceled: integer (nullable = true)
 |-- reserved_room_type: string (nullable = true)
 |-- assigned_room_type: string (nullable = true)
 |-- booking_changes: integer (nullable = true)
 |-- deposit_type: string (nullable = true)
 |-- agent: string (nullable = true)
 |-- company: string (nullable = true)
 |-- days_in_waiting_list: integer (nullable = true)
 |-- customer_type: string (nullable = true)
 |-- adr: float (nullable = true)
 |-- required_car_parking_spaces: integer (nullable = true)
 |-- total_of_special_requests: integer (nullable = true)
 |-- reservation_status: string (nullable = true)
 |-- reservation_status_date: timestamp (nullable = true)
 |-- booking_id: string (nullable = true)
 |-- city: string (nullable = true)
 |-- state: string (nullable = true)
 |-- room_rate: float (nullable = true)

schema loaded successfully.
```

```
display(df)
```

3 sec - Command executed in 2 sec 550 ms by Ajay Kumar Kummeri on 5:21:10 PM, 9/17/24

Spark jobs (1 of 1 succeeded) Resources Log

Diagnostics 1

Showing rows 1 - 1000

	customer_id	customer_name	hotel_id	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights	adults	children	babies	meal	country	market_s
1	C001	John Doe	H001	0	NULL	2023	NULL	2	5	2	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
2	C002	Jane Smith	H002	1	NULL	2024	NULL	9	3	1	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
3	C003	Alice Brown	H003	0	NULL	2024	NULL	27	10	4	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
4	C001	John Doe	H001	Hotel Blue	0	50	2023	January	2	5	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
5	C002	Jana Smith	H002	Hotel Red	1	120	2024	March	9	3	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
6	C003	Alice Brown	H003	Hotel Green	0	200	2024	July	27	10	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
7	f47988b8-9c38-4...	Customer29007	f273e959-71...	Resort Hotel	0	142	2022	April	15	12	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
8	d1d2a314-01cb-...	Customer45785	ecb22484-7a...	Boutique H...	1	263	2023	December	50	17	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
9	338e9da9-007c-...	Customer24389	4a5b775c-c0...	Luxury Hotel	0	288	2019	November	46	13	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
10	e67fb687-5a57-4...	Customer25014	e31f154f-460...	Resort Hotel	0	92	2020	October	44	26	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
11	dbb70e4c-dea9-...	Customer62741	398d0200-e2...	City Hotel	0	229	2021	August	33	16	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
12	b4a97956-88a3-...	Customer13462	d360e7da-5f...	City Hotel	1	219	2023	May	19	8	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
13	25b3c79f-d563-4...	Customer6463	ef120a0a-34...	Boutique H...	0	215	2018	May	22	31	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
14	e6e631f1-5c95-4...	Customer15486	6d02c658-2f...	Boutique H...	1	213	2018	January	2	14	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
15	29adc006-9783-...	Customer5630	ecc4269b-98...	City Hotel	1	235	2020	August	34	22	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Selected Cell 6 of 13 cells

Figure 15. Notebook Schema Result-2

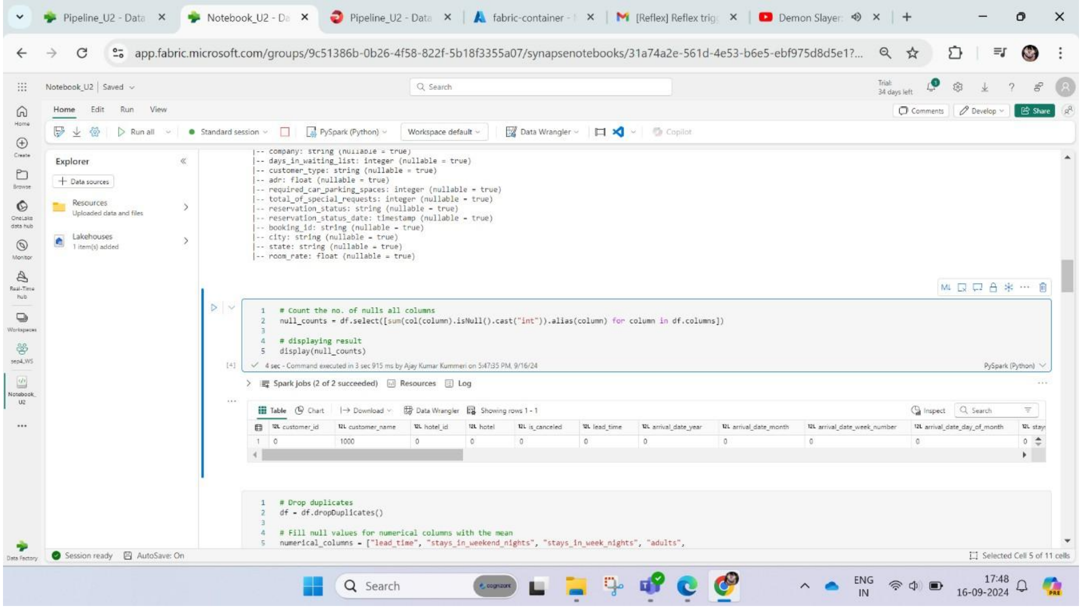
- **Expected Result**: Schema should match with the default schema.

- **Actual Result**: Schema match with the default schema.

Test Case 4: Handling Null Values in Notebooks

- **Description**: Ensure the notebook handles null values appropriately.
- **Preconditions**: Data with null values is available.
- **Steps**:

1. Run the notebook with data containing null values.



```
 |-- company: string (nullable = true)
 |-- days_in_waiting_list: integer (nullable = true)
 |-- customer_type: string (nullable = true)
 |-- adr: float (nullable = true)
 |-- required_car_parking_spaces: integer (nullable = true)
 |-- total_of_special_requests: integer (nullable = true)
 |-- reservation_status: string (nullable = true)
 |-- reservation_status_date: timestamp (nullable = true)
 |-- booking_id: string (nullable = true)
 |-- city: string (nullable = true)
 |-- state: string (nullable = true)
 |-- room_rate: float (nullable = true)
```

```
# Count the no. of nulls all columns
null_counts = df.select([sum(col(column).isNull().cast("int")).alias(column) for column in df.columns])

# displaying result
display(null_counts)
```

[4] ✓ 4 sec - Command executed in 3 sec 915 ms by Ajay Kumar Kummeri on 5:47:35 PM, 9/16/24

Spark jobs (2 of 2 succeeded)

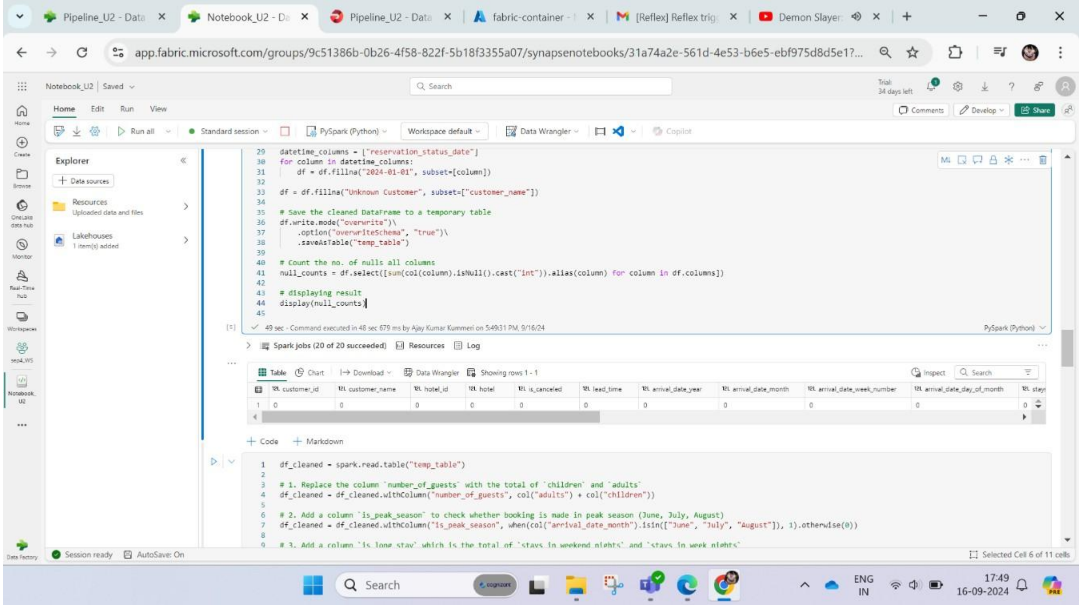
Showing rows 1 - 1

	customer_id	customer_name	hotel_id	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stay
1	0	1000	0	0	0	0	0	0	0	0	0

```
# Drop duplicates
df = df.dropDuplicates()

# Fill null values for numerical columns with the mean
numerical_columns = ["lead_time", "stays_in_weekend_nights", "stays_in_week_nights", "adults",
```

Figure 16. Null Count – 1

2. Verify that null values are handled as per the defined rules.



```
datetime_columns = ["reservation_status_date"]
for column in datetime_columns:
    df = df.fillna("2024-01-01", subset=[column])

df = df.fillna("Unknown Customer", subset=["customer_name"])

# Save the cleaned DataFrame to a temporary table
df.write.mode("overwrite")\
    .option("overwriteSchema", "true")\
    .saveAsTable("temp_table")

# Count the no. of nulls all columns
null_counts = df.select([sum(col(column).isNull().cast("int")).alias(column) for column in df.columns])

# displaying result
display(null_counts)
```

[5] ✓ 49 sec - Command executed in 48 sec 679 ms by Ajay Kumar Kummeri on 5:49:31 PM, 9/16/24

Spark jobs (20 of 20 succeeded)

Showing rows 1 - 1

	customer_id	customer_name	hotel_id	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stay
1	0	0	0	0	0	0	0	0	0	0	0

```
df_cleaned = spark.read.table("temp_table")

# 1. Replace the column `number_of_guests` with the total of `children` and `adults`
df_cleaned = df_cleaned.withColumn("number_of_guests", col("adults") + col("children"))

# 2. Add a column `is_peak_season` to check whether booking is made in peak season (June, July, August)
df_cleaned = df_cleaned.withColumn("is_peak_season", when(col("arrival_date_month").isin(["June", "July", "August"]), 1).otherwise(0))

# 3. Add a column `is_long_stay` which is the total of `stays_in_weekend_nights` and `stays_in_week_nights`
```

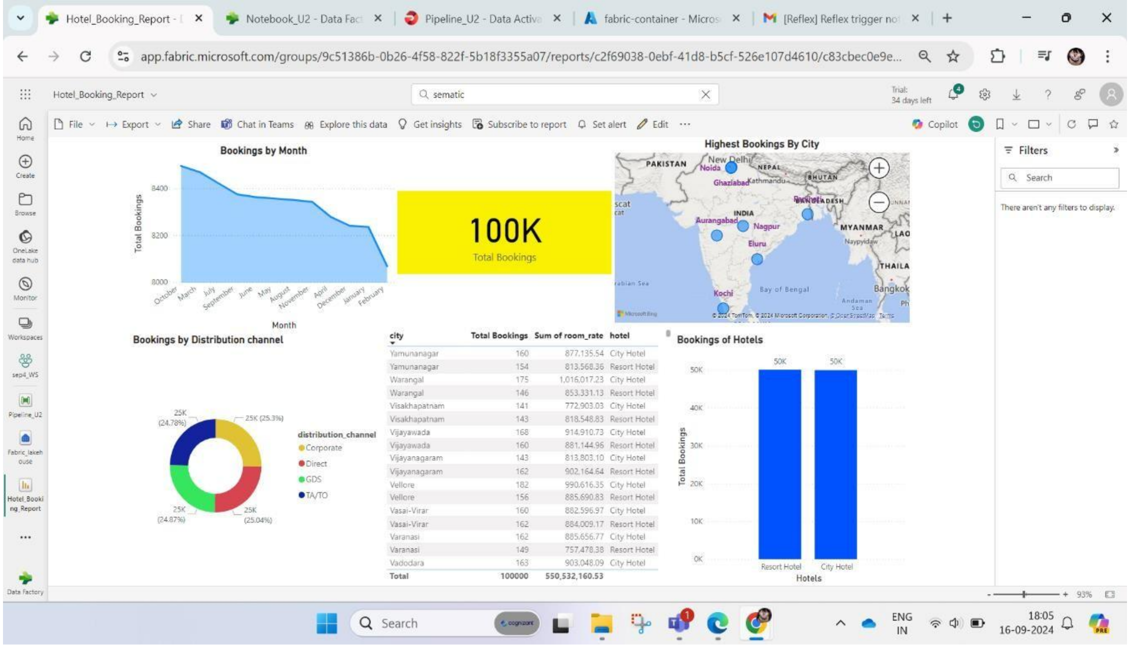
Figure 17. Null Count - 2

- **Expected Result**: Null values should be handled correctly.
- **Actual Result**: Null values are handled correctly.

Test Case 5: Semantic Model Update

- **Description**: Ensure the Power BI semantic model is refreshed and reports are updated based on the new data.
- **Preconditions**: Power BI dataset is configured for the semantic model. Data has been updated in the pipeline.
- **Steps**:

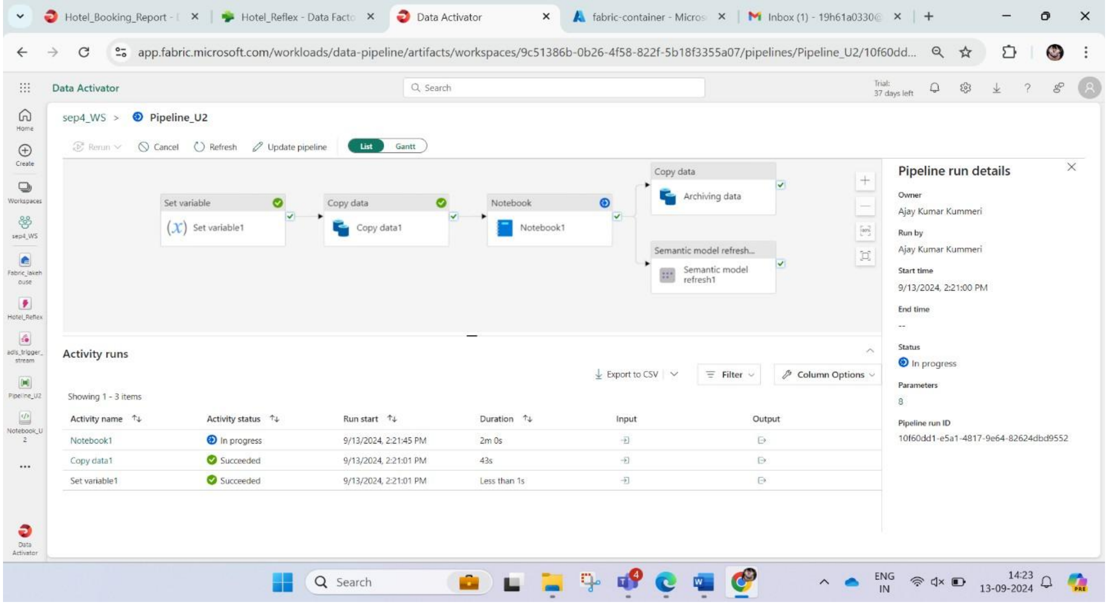
1. Check the report visuals before running a pipeline with new data.



city	Total Bookings	Sum of room_rate	hotel
Yamunanagar	160	877,135.54	City Hotel
Yamunanagar	154	813,568.36	Resort Hotel
Warangal	175	1,016,017.23	City Hotel
Warangal	146	853,331.13	Resort Hotel
Visakhapatnam	141	772,903.03	City Hotel
Visakhapatnam	143	818,548.83	Resort Hotel
Vijayawada	168	914,910.73	City Hotel
Vijayawada	160	881,144.96	Resort Hotel
Vijayanagaram	143	813,803.10	City Hotel
Vijayanagaram	162	902,164.64	Resort Hotel
Vellore	182	990,616.35	City Hotel
Vellore	156	885,690.83	Resort Hotel
Vasai-Virar	160	882,596.97	City Hotel
Vasai-Virar	162	884,009.17	Resort Hotel
Varanasi	162	885,656.77	City Hotel
Varanasi	149	757,478.38	Resort Hotel
Vadodara	163	903,048.09	City Hotel
Total	100000	550,532,160.53	

Figure 18. Power BI Report – 2

2. Run the Sematic Model refresh step in the pipeline with new Data.



Activity runs

Export to CSV | Filter | Column Options

Showing 1 - 3 items

Activity name	Activity status	Run start	Duration	Input	Output
Notebook1	In progress	9/13/2024, 2:21:45 PM	2m 0s		
Copy data1	Succeeded	9/13/2024, 2:21:01 PM	43s		
Set variable1	Succeeded	9/13/2024, 2:21:01 PM	Less than 1s		

Pipeline run details

Owner
Ajay Kumar Kummeri

Run by
Ajay Kumar Kummeri

Start time
9/13/2024, 2:21:00 PM

End time
--

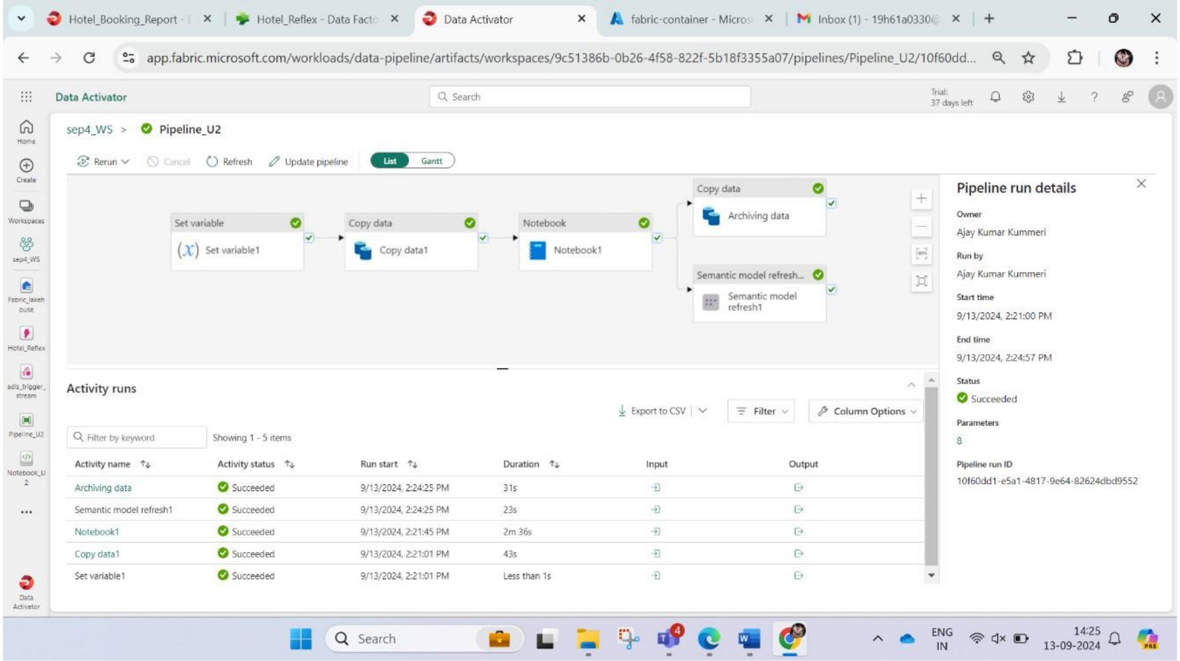
Status
In progress

Parameters
8

Pipeline run ID
10f60dd1-e5a1-4817-9e64-82624dbd9552

Figure 19. Running Pipeline

3. Verify that the semantic model updates successfully.



Pipeline run details

Owner
Ajay Kumar Kummeri

Run by
Ajay Kumar Kummeri

Start time
9/13/2024, 2:21:00 PM

End time
9/13/2024, 2:24:57 PM

Status
Succeeded

Parameters
8

Pipeline run ID
10f60dd1-e5a1-4817-9e64-82624dbd9552

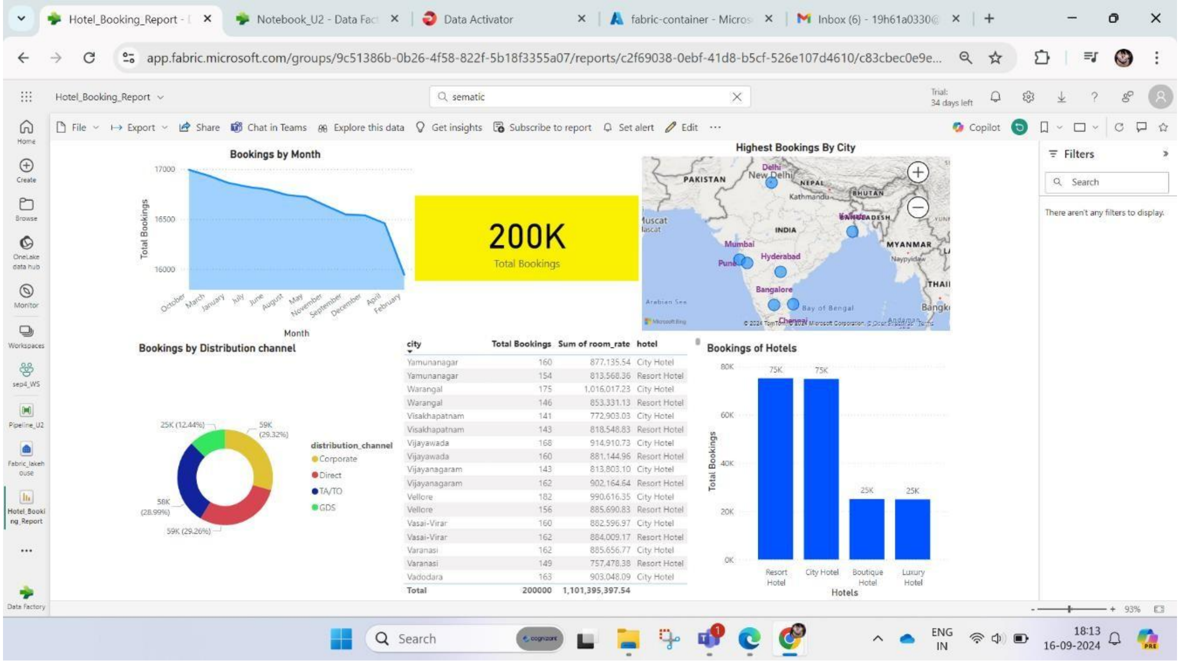
Activity runs

Showing 1 - 5 items

Activity name	Activity status	Run start	Duration	Input	Output
Archiving data	Succeeded	9/13/2024, 2:24:25 PM	31s		
Semantic model refresh1	Succeeded	9/13/2024, 2:24:25 PM	23s		
Notebook1	Succeeded	9/13/2024, 2:21:45 PM	2m 36s		
Copy data1	Succeeded	9/13/2024, 2:21:01 PM	43s		
Set variable1	Succeeded	9/13/2024, 2:21:01 PM	Less than 1s		

Figure 20. Successful pipeline -2

4. After pipeline run Check that the reports are updated with the new data.



city	Total Bookings	Sum of room_rate	hotel
Yamunanagar	160	877,135.54	City Hotel
Yamunanagar	154	813,568.36	Resort Hotel
Warangal	175	1,016,017.23	City Hotel
Warangal	146	853,331.13	Resort Hotel
Visakhapatnam	141	772,903.03	City Hotel
Visakhapatnam	143	818,548.83	Resort Hotel
Vijayawada	168	914,910.73	City Hotel
Vijayawada	160	881,144.96	Resort Hotel
Vijayanagaram	143	813,803.10	City Hotel
Vijayanagaram	162	902,164.64	Resort Hotel
Vellore	182	990,616.35	City Hotel
Vellore	156	885,690.83	Resort Hotel
Vasai-Virar	160	882,596.97	City Hotel
Vasai-Virar	162	884,009.17	Resort Hotel
Varanasi	162	885,656.77	City Hotel
Varanasi	149	757,478.38	Resort Hotel
Vadodara	163	903,048.09	City Hotel
Total	200000	1,101,395,397.54	

Figure 21. Power BI Report After Refresh

Expected Result: Power BI dataset refresh should be completed successfully, and reports should reflect the updated data without any errors.

Actual Result: Power BI dataset refresh completes successfully, and reports reflect the updated data without any errors.

5. Test Execution

- **Testers**: Kummeri Ajay Kumar
- **Schedule**: 06-09-2024 to 19-09-2024

Test cases - Fabric project.xlsx

6. Conclusion

- **Summary**: The test cases ensure the dynamic data pipeline performs as expected, covering data transfer, transformation, table creation, archiving, and reporting. All critical scenarios (valid/invalid inputs, schema mismatches, null handling) were tested, and the pipeline was validated for smooth execution.

Recommendations: Ensure proper monitoring and logging. Periodically review pipeline triggers and dataset refresh settings to maintain performance and accuracy.