

Coding Challenge – 5

Azure Synapse ETL Pipeline

Name: Aathirainathan P

Date: 19-12-2024

1) Build an ETL pipeline with azure synapse with dataflow running on it.

Tasks Done With Steps:

Create an Azure Synapse Workspace:

The screenshot shows the Microsoft Azure portal with the URL portal.azure.com/#create/Microsoft.Synapse. The page is titled 'Create Synapse workspace'. The 'Workspace details' section is filled out as follows:

- Workspace name: codingchallengesynapse
- Region: Central India
- Select Data Lake Storage Gen2: From subscription (radio button selected)
- Account name: codingchallengedlacc
- File system name: my-container

A warning message in a callout box states: "Additional configuration is required. After you create your workspace, perform these tasks:

- Assign other users to the **Contributor** role on workspace
- Assign other users the appropriate [Synapse RBAC roles](#) using Synapse Studio

Contact an **Owner** of the storage account, and ask them to perform the following tasks:"

At the bottom, there are buttons for 'Review + create' and 'Next: Security >'. The taskbar at the bottom of the screen shows various application icons and the date/time '19-12-2024'.

Microsoft Azure Synapse Analytics - Overview

Your deployment is complete

Deployment name : Microsoft.Azure.SynapseAnalytics-20241219163133
Subscription : MML Learners
Resource group : rg-azuser2356_mml.local-eylrk

Start time : 12/19/2024, 4:32:38 PM
Correlation ID : dc21ae44-04bc-4883-b542-401872ba21f4

Deployment details

Next steps

Go to resource group

Give feedback

Tell us about your experience with deployment

Cost management

Get notified to stay within your budget and prevent unexpected charges on your bill.

Set up cost alerts >

Microsoft Defender for Cloud

Secure your apps and infrastructure

Go to Microsoft Defender for Cloud >

Free Microsoft tutorials

Start learning today >

Work with an expert

Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.

Find an Azure expert >

32°C Partly sunny

Microsoft Azure | Synapse Analytics > codingchallengesynapse

Synapse Analytics workspace
codingchallengesynapse

New

Ingest

Perform a one-time or scheduled data load.

Explore and analyze

Learn how to get insights from your data.

Visualize

Build interactive reports with Power BI capabilities.

Discover more

Knowledge center

Browse partners

Recent resources

Create data flow activity:

The screenshot shows the Microsoft Azure Synapse Analytics pipeline interface. On the left, the navigation pane lists various services like Synapse, Move and transform, and Databricks. The main workspace displays a pipeline named 'Pipeline 1' containing a single 'Data flow' activity named 'Dataflow1'. The 'Properties' panel on the right shows the general settings for the pipeline, including the name 'Pipeline 1' and a description field.

Create a source and sink and Configure the source:

The screenshot shows the configuration of the 'Dataflow1' activity within the pipeline. The 'Source settings' tab is selected, displaying the configuration for 'source1'. The 'Dataset' dropdown is set to 'DelimitedText3'. Under 'Options', the 'Allow schema drift' checkbox is checked. The 'Sampling' section shows 'Enable' selected. The 'Properties' panel on the right shows the general settings for the data flow, including the name 'Dataflow1' and a description field.

Screenshot of Microsoft Azure Synapse Analytics Dataflow1 pipeline configuration.

The pipeline consists of a source1 (DelimitedText) and a sink1 (Add sink dataset). The source has 24 columns. A validation step is selected.

New integration dataset dialog is open, showing a grid of data store options:

All	Azure	Database	File	Generic protocol
Amazon S3	Azure Blob Storage	Azure Cosmos DB for NoSQL		
Azure Data Explorer (Kusto)	Azure Data Lake Storage Gen2	Azure Database for MySQL		

Buttons: Continue, Cancel.

System tray: 32°C, Mostly sunny, Search, Weather, Task View, File Explorer, Taskbar icons, ENG IN, 16:58, 19-12-2024.

Screenshot of Microsoft Azure Synapse Analytics Dataflow1 pipeline configuration.

The pipeline consists of a source1 (DelimitedText) and a sink1 (Add sink dataset). The source has 24 columns. A validation step is selected.

Select format dialog is open, showing a grid of file formats:

Avro	CSV	Excel
JSON	ORC	Parquet
XML	DAT	

Buttons: Continue, Back, Cancel.

System tray: 32°C, Mostly sunny, Search, Weather, Task View, File Explorer, Taskbar icons, ENG IN, 16:58, 19-12-2024.

The screenshot shows the Microsoft Azure Synapse Analytics Dataflow pipeline configuration interface. On the left, a sidebar lists 'Pipeline 1' and 'Dataflow1'. The main area displays a data flow diagram with a 'source1' node connected to a 'sink1' node. Below the diagram, the 'Source settings' tab is selected, showing 'Dataset' set to 'DelimitedText2', 'Options' including 'Allow schema drift' checked, and 'Sampling' set to 'Disable'. To the right, a 'Browse' window is open, showing a file tree under 'Root folder > source-container'. A file named 'Global_Superstore2.csv' is selected. At the bottom, a toolbar includes icons for search, refresh, and navigation.

Configure the sink:

The screenshot shows the Microsoft Azure Synapse Analytics Dataflow pipeline configuration interface. The 'Sink' tab is selected in the bottom navigation bar. The 'sink1' node is highlighted in the data flow diagram. On the right, a 'Set properties' dialog is open for the sink. It contains fields for 'Name' (set to 'DelimitedText4'), 'Linked service' (set to 'AzureDataLakeStorage1'), 'File path' (set to 'my-container'), 'First row as header' (checked), and 'Import schema' (set to 'From connection/store'). The 'OK' button is visible at the bottom of the dialog. The status bar at the bottom indicates '16:59 19-12-2024'.

Validate and Debug the pipeline:

The screenshot shows the Microsoft Azure Synapse Analytics Pipeline Editor. A pipeline named "Pipeline 1" is open, containing a single activity named "Data flow1". The "Data flow" tab is selected. The "Properties" pane on the right shows the pipeline is named "Pipeline 1". The "Output" pane displays the pipeline run status as "In progress". A tooltip indicates that the data flow activity will start as soon as the debug session is ready. The pipeline run ID is e1b5266a-e90a-4b0f-b628-638fe5902279.

JPY/INR -1.10% 17:01 19-12-2024

The screenshot shows the Microsoft Azure Synapse Analytics Pipeline Editor after the pipeline run has completed. The pipeline run status is now "Succeeded". The "Properties" pane on the right shows the pipeline is named "Pipeline 1". The "Output" pane displays the pipeline run status as "Succeeded". A tooltip indicates that the pipeline run consumption can be viewed. The pipeline run ID is e1b5266a-e90a-4b0f-b628-638fe5902279.

Watchlist Ideas 17:05 19-12-2024

Extract the data from the delta lake storage and do the transformations and loading:

```
▶ ✓ 04:44 PM (15s) 1 Python ⚙ ⏹ ⋮  
  
%python  
# Fetching the csv file from the blob storage  
storage_account_name = "codingchallengedlacc"  
container_name = "my-container"  
storage_account_key = "hMW9uKwa01lepiUgz0tGQd9P9UKbeVq29mCY5wCz2GIPpsMELvdjGaAcJMmDjsiX8RHCznioSigW+ASTcpQTgQ=="  
  
# Unmount the directory if it is already mounted  
if any(mount.mountPoint == "/mnt/superstore" for mount in dbutils.fs.mounts()):  
    dbutils.fs.unmount("/mnt/superstore")  
  
# Mount dl Storage  
dbutils.fs.mount(  
    source=f"wasbs://{container_name}@{storage_account_name}.blob.core.windows.net",  
    mount_point="/mnt/superstore",  
    extra_configs={  
        f"fs.azure.account.key.{storage_account_name}.blob.core.windows.net": storage_account_key  
    }  
)  
  
True
```

Verifying the mount:

```
▶ ✓ 04:44 PM (6s) 2 Python ⚙ ⏹ ⋮  
  
#verifying the mount  
display(dbutils.fs.ls("/mnt/superstore"))  
  
▶ (2) Spark Jobs  
  
Table + Q Y □  
  


|   | path                                         | name                    | size     | modificationTime |
|---|----------------------------------------------|-------------------------|----------|------------------|
| 1 | dbfs:/mnt/superstore/Global_Superstore2.c... | Global_Superstore2.c... | 12089916 | 1734606845000    |

  


↓ 1 row | 5.79 seconds runtime Refreshed 41 minutes ago


```

Load the dataset:

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("SuperstoreETL").getOrCreate()

# Load the dataset
file_path = "/mnt/superstore/Global_Superstore2.csv"
df = spark.read.csv(file_path, header=True, inferSchema=True)
df.show(5)

▶ (3) Spark Jobs
▶ df: pyspark.sql.dataframe.DataFrame = [Row ID: integer, Order ID: string ... 22 more fields]
-----+
| 32298| CA-2012-124891|2012-07-31|2012-07-31| Same Day| RH-19495| Rick Hansen| Consumer|New York City| New York|United States|
10024| US| East| TEC-AC-10003033|Technology| Accessories|Plantronics CS510...| 2309.65| 7| 0|762.1845| 933.57| Critic
call|
| 26341| IN-2013-77878|2013-02-05|2013-02-07|Second Class| JR-16210| Justin Ritter| Corporate| Wollongong|New South Wales| Australia|
NULL| APAC|Oceania| FUR-CH-10003950| Furniture| Chairs|Novimex Executive...|3709.395| 9| 0.1|-288.765| 923.63| Critic
all|
| 25330| IN-2013-71249|2013-10-17|2013-10-18| First Class| CR-12730| Craig Reiter| Consumer| Brisbane| Queensland| Australia|
NULL| APAC|Oceania| TEC-PH-10004664|Technology| Phones|Nokia Smart Phone...|5175.171| 9| 0.1| 919.971| 915.49| Medi
um|
| 13524|ES-2013-1579342|2013-01-28|2013-01-30| First Class| KM-16375|Katherine Murray|Home Office| Berlin| Berlin| Germany|
NULL| EU|Central| TEC-PH-10004583|Technology| Phones|Motorola Smart Ph...| 2892.51| 5| 0.1| -96.54| 910.16| Medi
um|
| 47221| SG-2013-4320|2013-11-05|2013-11-06| Same Day| RH-9495| Rick Hansen| Consumer| Dakar| Dakar| Senegal|
NULL|Africa| Africa|TEC-SHA-10000501|Technology| Copiers|Sharp Wireless Fa...| 2832.96| 8| 0| 311.52| 903.04| Critic
```

Add Total Cost Column:

Convert Sales and Profit to Float:

04:45 PM (1s)

5

Python    

```
#Convert Sales and Profit to Float

df_transformed = df_transformed.withColumn("Sales", col("Sales").cast("float"))
df_transformed = df_transformed.withColumn("Profit", col("Profit").cast("float"))

df_transformed.show(5)
```

▶ (1) Spark Jobs

▶ df_transformed: pyspark.sql.dataframe.DataFrame = [Row ID: integer, Order ID: string ... 23 more fields]

Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	City	State	Country			
Postal Code	Market	Region	Product ID	Category	Sub-Category	Product Name	Sales	Quantity	Discount	Profit	Shipping Cost	Order Priority	TotalCost
32298	CA-2012-124891	2012-07-31	2012-07-31	Same Day	RH-19495	Rick Hansen	Consumer	New York City		New York	United States		
10024	US	East	TEC-AC-10003033	Technology	Accessories	Plantronics CS510...	2309.65	7	0	762.1845	933.57	Critical	1547.465500000002
26341	IN-2013-77878	2013-02-05	2013-02-07	Second Class	JR-16210	Justin Ritter	Corporate	Wollongong	New South Wales	Australia			
NULL	APAC	Oceania	FUR-CH-10003950	Furniture	Chairs	Novimex Executive...	3709.395	9	0.1	-288.765	923.63	Critic	

Filter Rows with Zero or Negative Profit:

Add a Profit Margin Column:

8

```
from pyspark.sql.functions import col

# Add a Profit Margin Column
df_transformed = df_transformed.withColumn("ProfitMargin", (col("TotalProfit") / col("TotalSales")) * 100)

# Show the result
df_transformed.show(5)
```

▶ (2) Spark Jobs

▶ df_transformed: pyspark.sql.dataframe.DataFrame = [Category: string, Region: string ... 5 more fields]

Category Region	TotalSales	TotalProfit	TotalQuantity	TotalDiscount	ProfitMargin
Furniture Canada	10595.279964447021	2613.24	78	0.0	24.664190174953884
Technology EMEA	300854.583026886	17494.44300000036	2259	189.1000056862831	5.814916569988444
Furniture East	205540.3473367691	2501.8162	2151	90.6000018119812	1.2171898278934408
Technology Africa	322367.0430994034	44129.493000001	2031	143.1999975964427	13.689207363046899
Technology East	264872.0816922188	47439.55759999996	1927	76.30000080168247	17.910365372189244

only showing top 5 rows

Remove Duplicate Rows based on 'Category' and 'Region':

9

```
# Remove Duplicate Rows based on 'Category' and 'Region'
df_transformed = df_transformed.dropDuplicates(["Category", "Region"])

df_transformed.show(5)
```

▶ (2) Spark Jobs

▶ df_transformed: pyspark.sql.dataframe.DataFrame = [Category: string, Region: string ... 5 more fields]

Category Region	TotalSales	TotalProfit	TotalQuantity	TotalDiscount	ProfitMargin
Furniture Canada	10595.279964447021	2613.24	78	0.0	24.664190174953884
Technology EMEA	300854.583026886	17494.44300000036	2259	189.1000056862831	5.814916569988444
Furniture East	205540.3473367691	2501.8162	2151	90.6000018119812	1.2171898278934408
Technology Africa	322367.0430994034	44129.493000001	2031	143.1999975964427	13.689207363046899
Technology East	264872.0816922188	47439.55759999996	1927	76.30000080168247	17.910365372189244

only showing top 5 rows

Rename Columns for Clarity:

04:45 PM (1s) 10 Python

```
#Rename Columns for Clarity
df_transformed = df_transformed.withColumnRenamed("Sales", "TotalSales") \
| | | | | | | .withColumnRenamed("Profit", "TotalProfit")

df_transformed.show(5)
```

▶ (2) Spark Jobs

df_transformed: pyspark.sql.dataframe.DataFrame = [Category: string, Region: string ... 5 more fields]

Category	Region	TotalSales	TotalProfit	TotalQuantity	TotalDiscount	ProfitMargin
Furniture	Canada	10595.279964447021	2613.24	78	0.0	24.664190174953884
Technology	EMEA	300854.583026886	17494.44300000036	2259	189.1000056862831	5.814916569988444
Furniture	East	205540.3473367691	2501.8162	2151	90.6000018119812	1.2171898278934408
Technology	Africa	322367.0430994034	44129.493000001	2031	143.1999975964427	13.689207363046899
Technology	East	264872.0816922188	47439.55759999996	1927	76.30000080168247	17.910365372189244

only showing top 5 rows

Add a Year Column:

04:45 PM (1s) 11 Python

```
from pyspark.sql.functions import col, year, to_date

# Add a Year Column
df = df.withColumn("Year", year(to_date(col("Order Date"), "MM/dd/yyyy")))

# Show the result
df_transformed.show(5)
```

▶ (2) Spark Jobs

df: pyspark.sql.dataframe.DataFrame = [Row ID: integer, Order ID: string ... 23 more fields]

Category	Region	TotalSales	TotalProfit	TotalQuantity	TotalDiscount	ProfitMargin
Furniture	Canada	10595.279964447021	2613.24	78	0.0	24.664190174953884
Technology	EMEA	300854.583026886	17494.44300000036	2259	189.1000056862831	5.814916569988444
Furniture	East	205540.3473367691	2501.8162	2151	90.6000018119812	1.2171898278934408
Technology	Africa	322367.0430994034	44129.493000001	2031	143.1999975964427	13.689207363046899
Technology	East	264872.0816922188	47439.55759999996	1927	76.30000080168247	17.910365372189244

only showing top 5 rows

Filter the Data for a Specific Year:

```
▶ 04:45 PM (2s) 12 Python ⚡ ⏺ ⏹
```

```
from pyspark.sql.functions import col, year, to_date, sum
#Filter the Data for a Specific Year

# Step 1: Add the Year Column to the Original DataFrame
df = df.withColumn("Year", year(to_date(col("Order Date"), "MM/dd/yyyy")))

# Step 2: Perform the aggregation including 'Year'
df_transformed = df.groupBy("Category", "Region", "Year") \
    .agg(
        sum("Sales").alias("TotalSales"),
        sum("Profit").alias("TotalProfit"),
        sum("Quantity").alias("TotalQuantity"),
        sum("Discount").alias("TotalDiscount")
    )

# Step 3: Filter the Data for a Specific Year
df_transformed = df_transformed.filter(col("Year") == 2012)

# Show the result
df_transformed.show(5)
```

▶ (2) Spark Jobs

```
▶ df: pyspark.sql.dataframe.DataFrame = [Row ID: integer, Order ID: string ... 23 more fields]
▶ df_transformed: pyspark.sql.dataframe.DataFrame = [Category: string, Region: string ... 5 more fields]
```

Category	Region	Year	TotalSales	TotalProfit	TotalQuantity	TotalDiscount
Furniture	Oceania	2012	100519.0080000002	8623.818	607.0	26.1999999999998
Technology	North	2012	126353.5630000004	25098.862999999998	809.0	7.378
Technology	Africa	2012	64734.58200000024	6320.742000000001	404.0	36.6999999999999
Furniture	Canada	2012	1600.68	290.19	16.0	0.0
Technology	Oceania	2012	89761.7670000005	14203.827	688.0	23.49999999999986

only showing top 5 rows

Sort Data by Total Sales:

04:46 PM (1s) 13 Python

```
#Sort Data by Total Sales
df_transformed = df_transformed.orderBy(col("TotalSales").desc())
df_transformed.show()
```

▶ (2) Spark Jobs

df_transformed: pyspark.sql.dataframe.DataFrame = [Category: string, Region: string ... 5 more fields]

Category	Region	Year	TotalSales	TotalProfit	TotalQuantity	TotalDiscount
Technology	Central	2012	237291.8116200002	31373.85431999996	1791.0	61.1419999999993
Furniture	Central	2012	183778.9994999998	5231.084800000005	1876.94	85.14000000000004
Office Supplies	Central	2012	180068.2920000002	26988.422600000013	6003.084	198.4000000000001
Technology	North	2012	126353.5630000004	25098.86299999998	809.0	7.378
Office Supplies	South	2012	116962.6720000002	10905.97419999997	3831.186	151.7000000000016
Furniture	South	2012	106962.4015	8656.69339999998	1108.0	38.9499999999999
Technology	South	2012	103154.04796	11673.36195999993	1034.824	38.896
Furniture	Oceania	2012	100519.0080000002	8623.818	607.0	26.1999999999998
Office Supplies	North	2012	91426.151	17629.37099999996	2648.0	36.89999999999984
Technology	Oceania	2012	89761.7670000005	14203.827	688.0	23.49999999999986
Furniture	North	2012	82451.2580000002	8177.70800000001	839.0	50.4999999999996
Technology	Southeast Asia	2012	77886.9887999996	6068.45879999999	534.0	32.0900000000001
Technology	Central Asia	2012	69458.7600000001	13756.85999999999	394.0	5.5
Technology	North Asia	2012	64934.6250000001	13026.85500000001	359.0	4.0
Technology	Africa	2012	64734.58200000024	6320.74200000001	404.0	36.6999999999999

Save the transformed data in DBFS:

04:46 PM (14s) 14 Python

```
# Save the transformed data in DBFS
df_transformed.write.format("delta").mode("overwrite").saveAsTable("transformed_data")
```

▶ (9) Spark Jobs

Configure Databricks and the Databricks notebook to Azure Synapse Pipeline:

The screenshot shows the Microsoft Azure Synapse Analytics pipeline configuration interface. On the left, the 'Integrate' sidebar lists 'Pipelines' (Pipeline 1, Pipeline 2). In the center, 'Pipeline 2' is selected, showing its activities: 'Synapse', 'Move and transform', 'Azure Data Explorer', 'Azure Function', 'Batch Service', and 'Databricks'. Under 'Databricks', a 'Notebook' activity is highlighted. The 'Properties' panel on the right shows the pipeline is named 'Pipeline 2'. The 'Azure Databricks' tab is selected, showing a dropdown for 'Databricks linked service' which is currently empty ('No results found'). The status bar at the bottom indicates it's 17:06 on 19-12-2024.

Generate a new token in Databricks to connect to Azure Synapse:

The screenshot shows the Microsoft Azure Databricks access tokens generation interface. The 'Access tokens' page is open, showing a 'Generate new token' dialog. The dialog fields include 'Comment' (set to 'Synapse Connection Token') and 'Lifetime (days)' (set to 1). The 'Generate' button is visible at the bottom right. The background shows the Databricks workspace settings. The status bar at the bottom indicates it's 17:09 on 19-12-2024.

The screenshot shows the Microsoft Azure Synapse Analytics interface. On the left, the 'Integrate' tab is selected, displaying a list of pipelines: Pipeline 1, Dataflow1, Pipeline 2, and Pipeline 3. The 'Activities' section lists various options like Synapse, Move and transform, Azure Data Explorer, etc. A 'Notebook' activity is currently selected. On the right, a 'New linked service' dialog is open for 'Azure Databricks'. It includes fields for 'Authentication type' (Access Token), 'Access token' (containing a redacted value), 'Cluster version' (15.4 LTS), 'Cluster node type' (Standard_D4ds_v5), 'Python version' (2), and 'Worker options' (Fixed). A 'Create' button is at the bottom.

Validate and Debug the pipeline:

The screenshot shows the Microsoft Azure Synapse Analytics interface with the 'Transformation_and_Loading_Pipeline' selected. The 'Properties' pane on the right shows the pipeline's name is 'Transformation_and_Loading_Pipeline'. The 'Output' section displays a table of pipeline run details. One row is shown: 'Transformation...' with 'Activity status' as 'Succeeded' and 'Run start' as '12/19/2024, 5:42:28 PM'. The 'Validate' and 'Debug' buttons are visible at the top of the pipeline editor.

Activity name	Activity status	Activity type	Run start
Transformation...	Succeeded	Notebook	12/19/2024, 5:42:28 PM

Create the final pipeline to connect the ingestion pipeline and Transformation pipeline and then Validate and Debug it:

The screenshot shows the Microsoft Azure Synapse Analytics studio interface. On the left, the 'Integrate' sidebar lists three pipelines: Ingestion_Pipeline, Transformation_and_Loading_Pipeline, and Final_Pipeline. The Final_Pipeline is selected. The main workspace displays two 'Execute Pipeline' activities. The first activity, 'Ingestion_Pipeline', has a green checkmark and is connected by a blue arrow to the second activity, 'Transformation_and_Loading_Pipeline', which also has a green checkmark. Both activities have a yellow lightning bolt icon. To the right, the 'Properties' panel shows the pipeline's name as 'Pipeline 3'. Below the activities, a table provides details about the pipeline run, including the run ID, status (Succeeded), and start time (12/19/2024, 5:18:04). The table also lists the two activities and their statuses.

Activity name	Activity status	Activity type	Run start
Transformation_and_Loading...	Succeeded	Execute Pipeline	12/19/2024, 5:18:04
Ingestion_Pipeline	Succeeded	Execute Pipeline	12/19/2024, 5:15:27

Submitted By:
Aathirainathan P