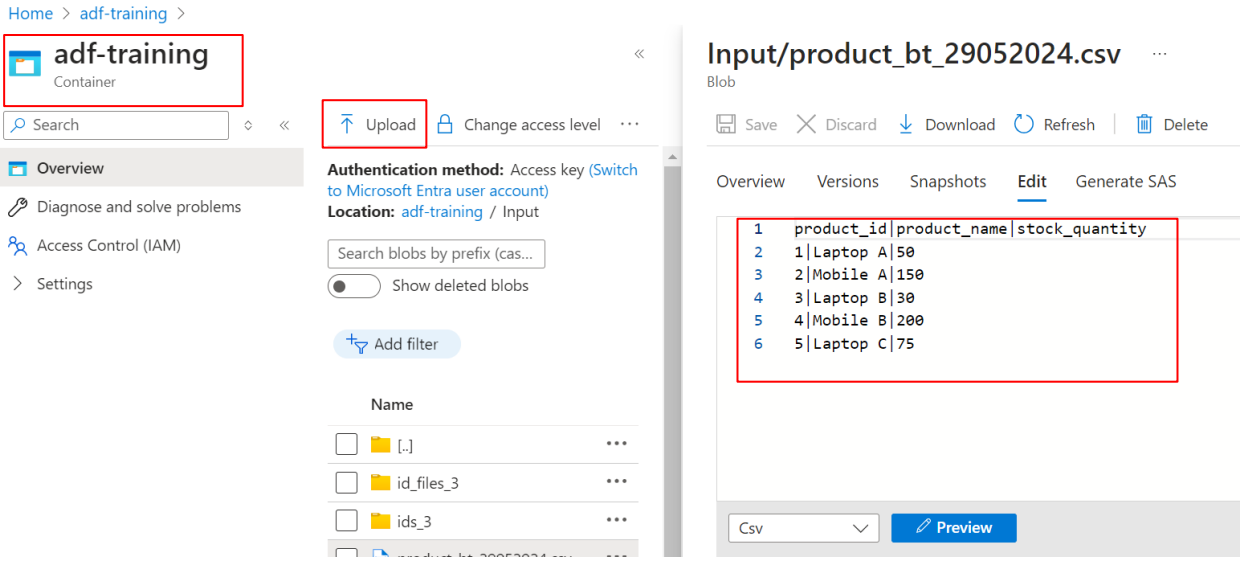
Formal Documentation of Azure Data Factory Pipeline – Training

Use Case: Calculating the sum of product quantity using dataflow.

Steps:

1. Uploading Input File in the Container

- A container named adf-training was previously created in the Azure storage account.
- For this pipeline, the file was uploaded into the container in the input folder
- Once the upload was successful, the file was added to the input folder. We are transforming the data in the input file.
- The task was to calculate the sum of the stock quantity of Laptops and Mobiles (refer image 1)

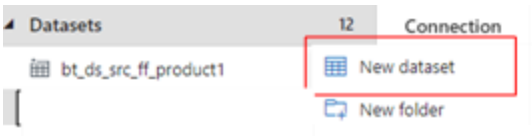


-Image 1

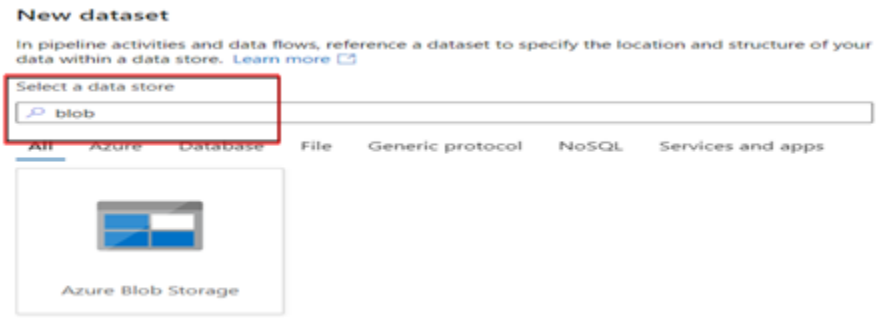
2. Dataset Creation

Source Dataset

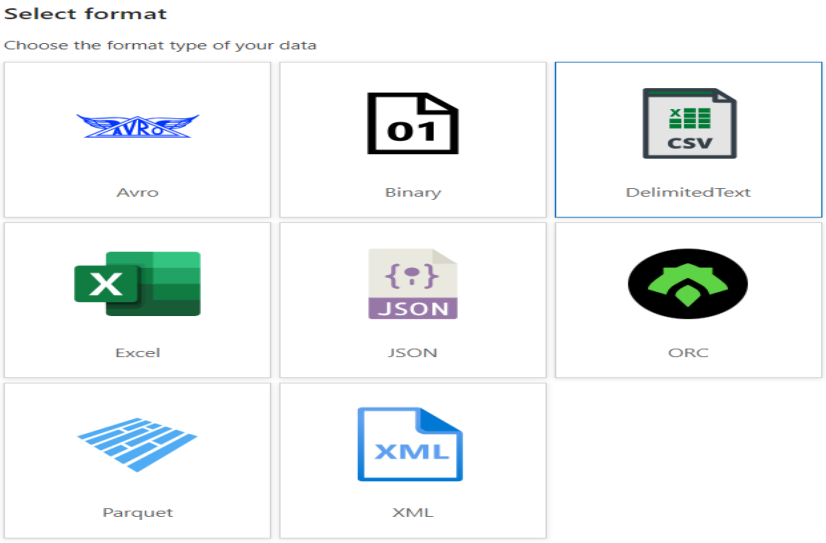
- A new source dataset was created for the source data file.
-
- Source dataset name: bt_ds_src_ff_product_sum
- Inside Azure Data Factory, in the Author tab, I selected the Dataset option and clicked on "New Dataset" (refer to image 1).
- I chose the Azure Blob Storage option (refer to image 2).
- Next, I selected the Delimited Text file format, which brought me to the properties page where I defined the dataset name and path (refer to image 3).
- I specified the dataset name, selected the linked service, and provided the path of my input file. These steps created my source dataset (refer to image 4).
- I opened my source file and updated the connection, changing the column delimiter option to pipe (|) because my CSV file is pipe delimited (refer image 5)



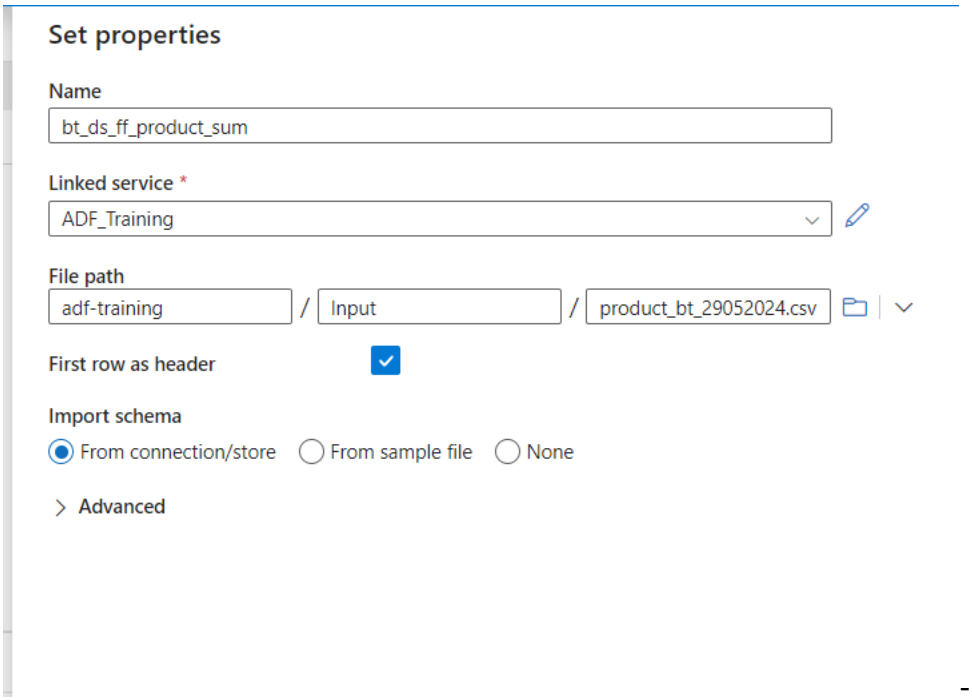
- Image 1



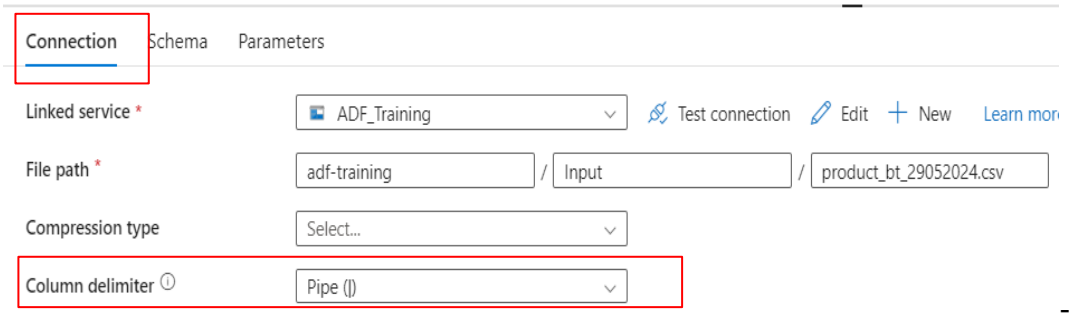
- Image 2



- Image 3



- Image 4



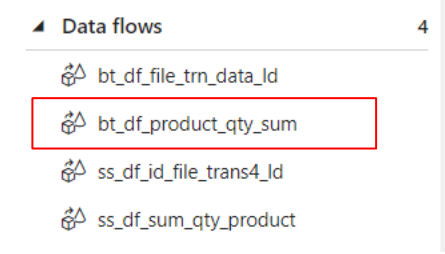
- image 5

Target Dataset

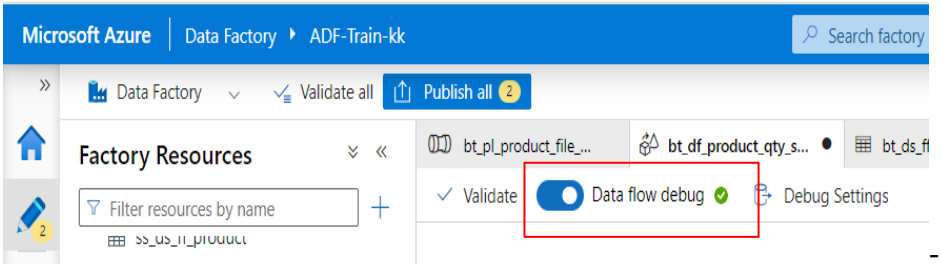
- I followed similar steps for the target dataset.
- In the Author tab, I selected the datasets, clicked on "New Dataset," selected Azure Blob Storage, and then selected the Delimited Text format, which brought me to the properties page where I defined the dataset name and path.
- Assigned path: adf-training/output
- Target dataset name: bt_ds_tgt_product_df_sum
- I opened my source file and updated the connection, changing the column delimiter option to pipe (|) because my CSV file is pipe delimited. All other options remained unchanged.

3. Dataflow Creation

- Dataflow name: bt_df_product_qty_sum (refer image 1)
- The mandatory step is to enable dataflow debug (refer image 2)



- Image 1



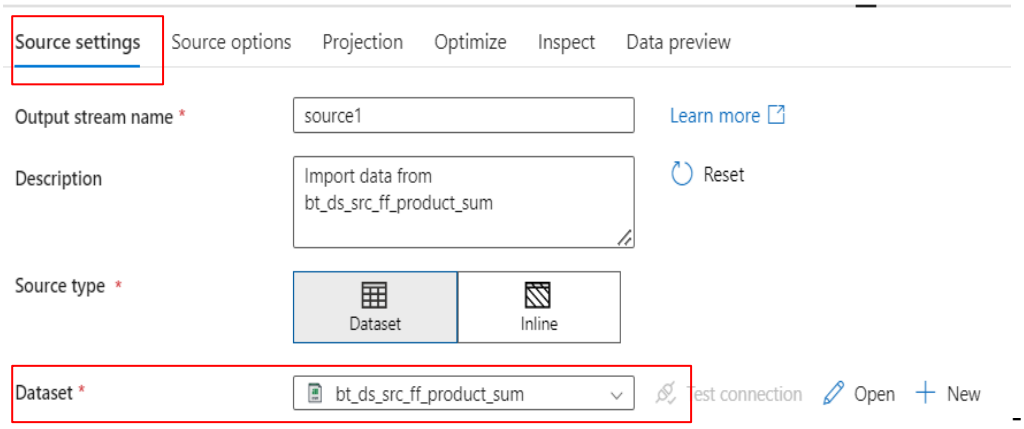
- Image 2

Dataflow Steps:

A. Selecting Source.

- Added source to add my source file (refer image 1)
- In source settings, I added my source dataset (refer image 2)

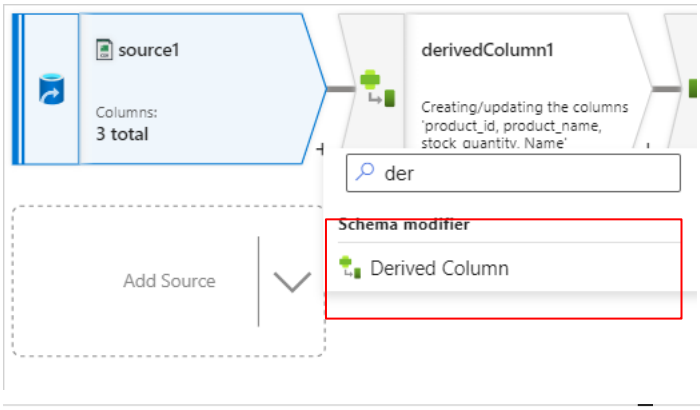
- Image 1



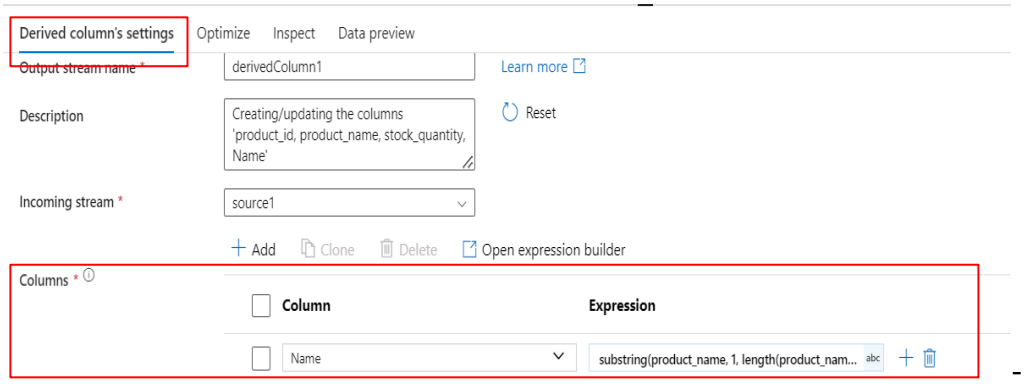
-Image 2

B. Derived Column

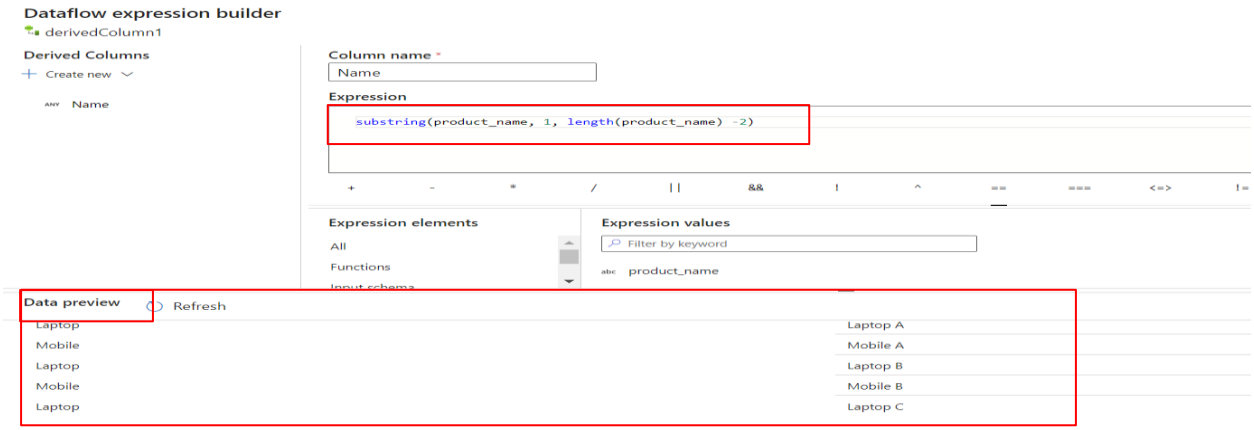
- I then selected the derived column (refer image 1) and added a new column named "Name" to extract the product_name without the last Alphabet (e.g., from Laptop A to Laptop).
- I had 3 columns earlier in my file: product_id, product_name, stock_quantity.
- In derived column's settings, I clicked on add column, assigned the column name as "Name" and clicked on expression (refer image 2).
- I wrote an expression which correctly extracted the product_name and created a new column as "Name" (Eg: Laptop A to Laptop) (refer image 3).



-Image 1



-Image 2

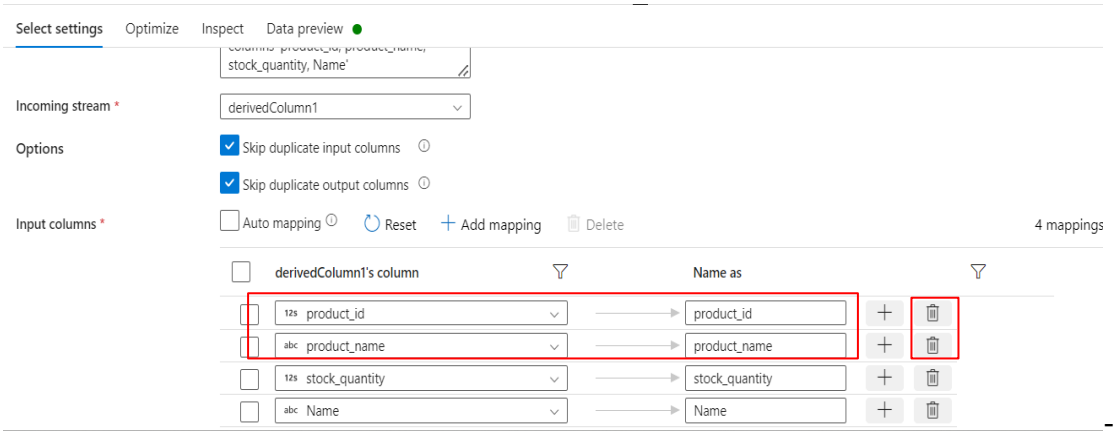


-Image 3

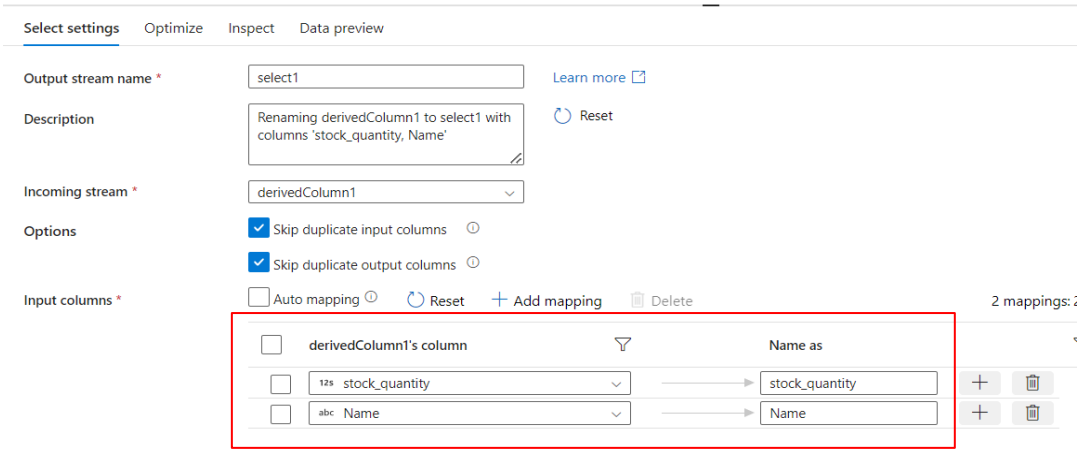
C. Select

- I then used the select function to get only the required columns (refer image 1)
- After the derived function, I had 4 columns however to calculate the sum and clean the data, I only needed 2 columns "Name" and "stock_quantity".
- In the select settings, I deleted the other 2 columns, product_id and product_name, and was able to get the data in the desired format.
- I checked the same from the data preview tab (refer image 4).

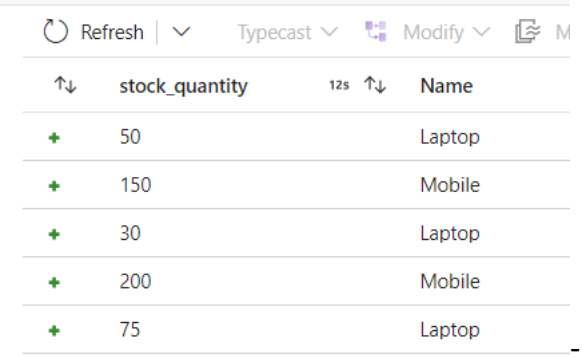
-Image 1



-Image 2



-Image 3

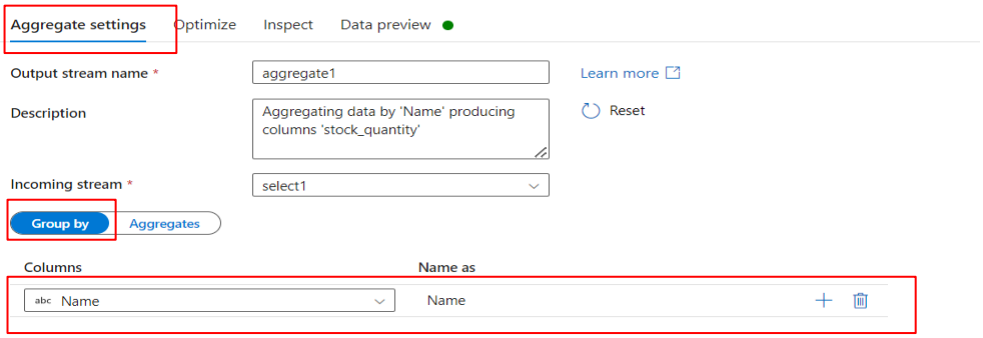


- Image 4

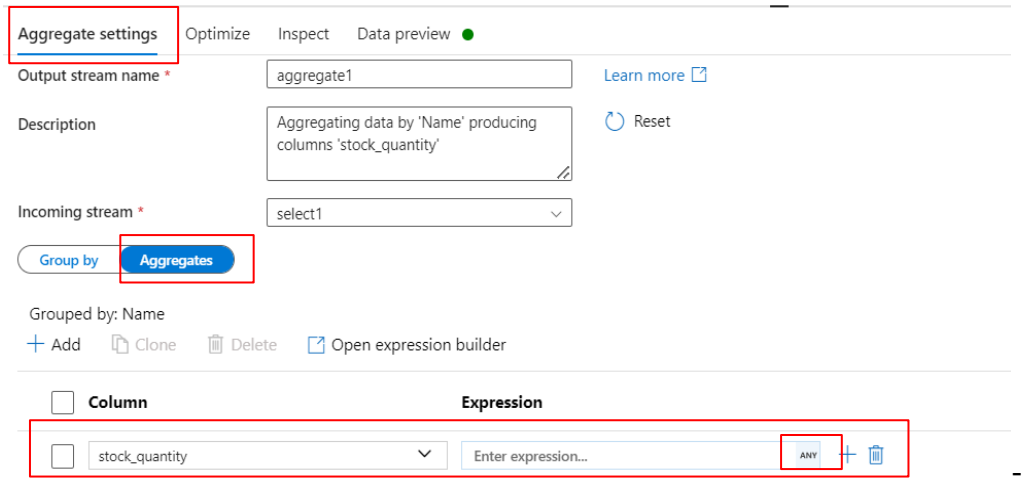
D. Aggregate

- To perform a sum of the stock quantity, I used the aggregate function (refer image 1).
- In the aggregate settings, there are two options, group by and aggregates.
- In group by, I selected the name column as I wanted to group the commodity names and then find the sum (refer image 2)
- In aggregates, I selected stock_quantity and in expression, wrote a sum expression (sum(stock_quantity)) and was able to get the sum of these 2 commodities (refer image 3,4)
- Check the data preview after aggregate (refer image 5)

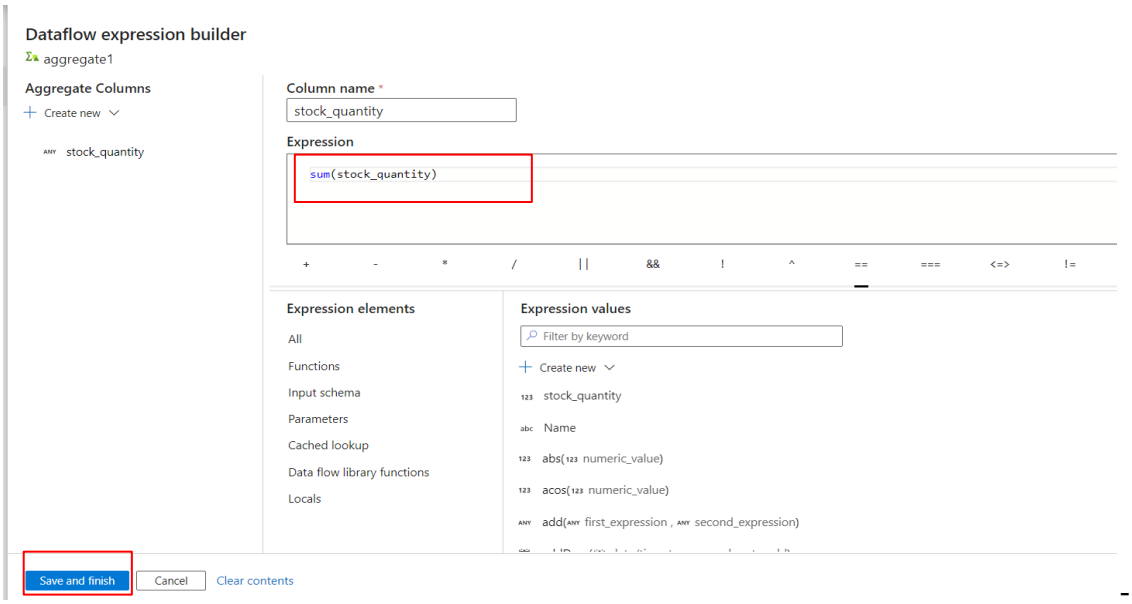
-Image 1



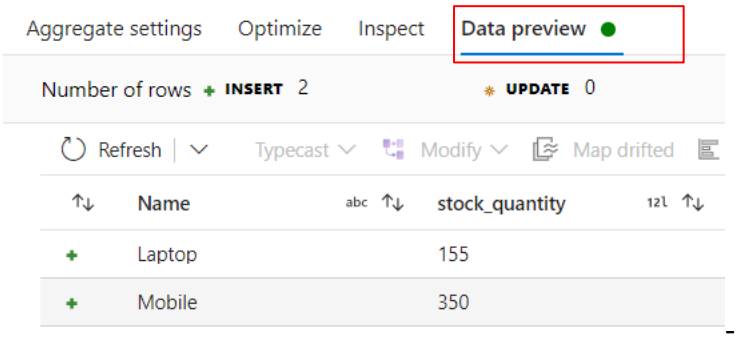
- Image 2



-Image 3



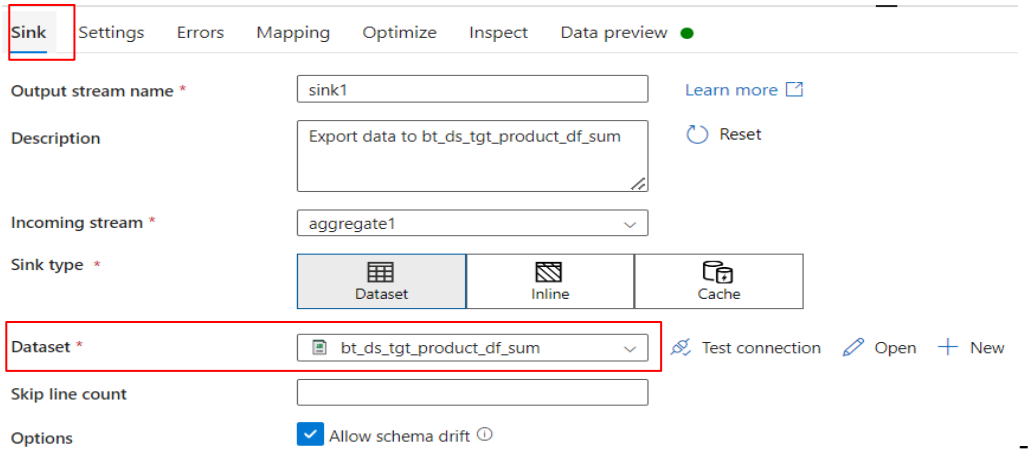
-Image 4



-Image 5

E. Sink

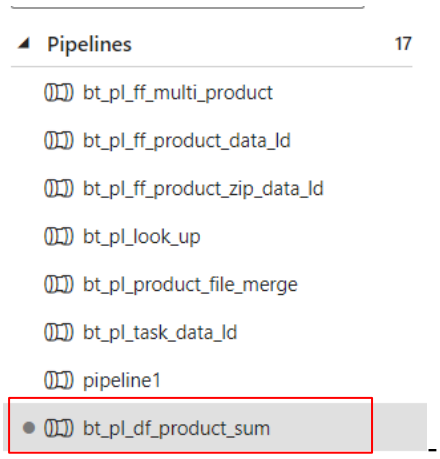
- In sink, I selected my target dataset to copy the sum file to the output folder (refer image 1)



-Image 1

4. Pipeline Creation

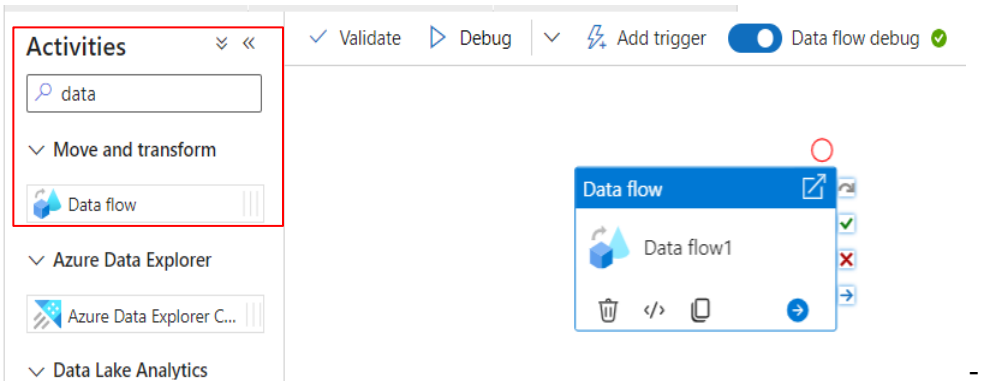
- I then created a pipeline named bt_pl_df_product_sum to execute the dataflow activities (refer image 1)



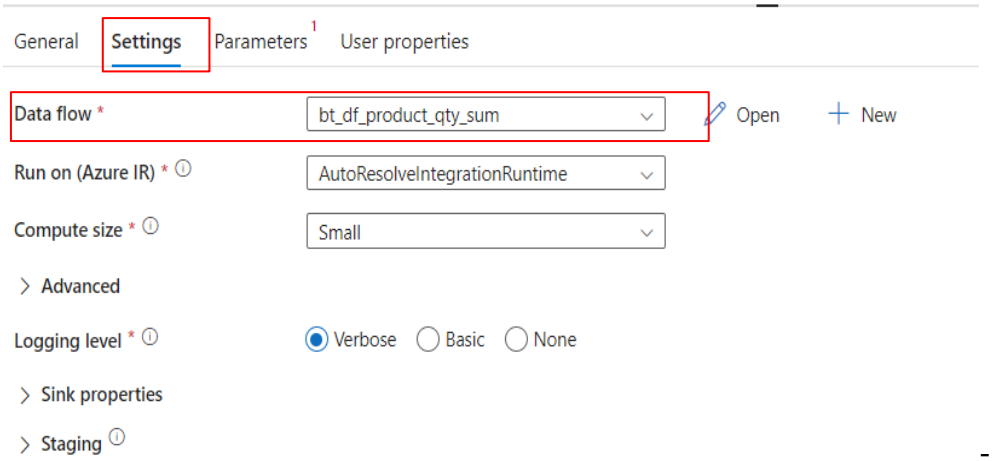
- Image 1

5. Dataflow Activity

- From the activities, I dragged and dropped the dataflow activity (refer image 1).
- Then, in the settings, I added the source file (refer image 2).



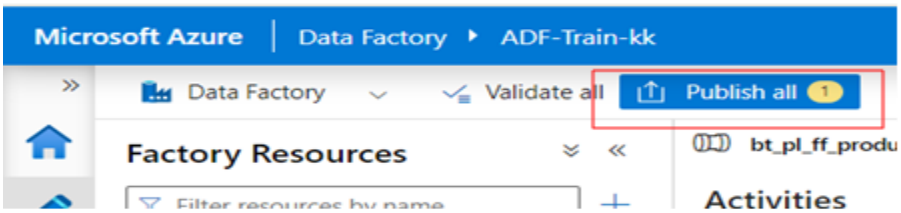
-Image 1



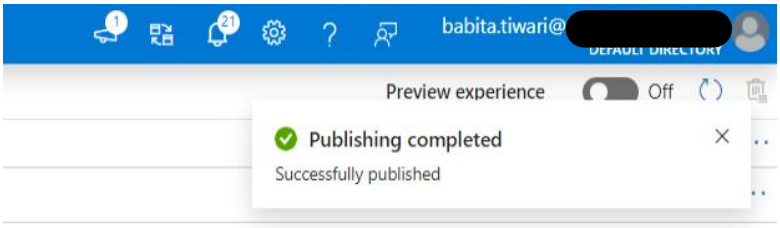
- Image 2

6. Publishing and Executing the Pipeline

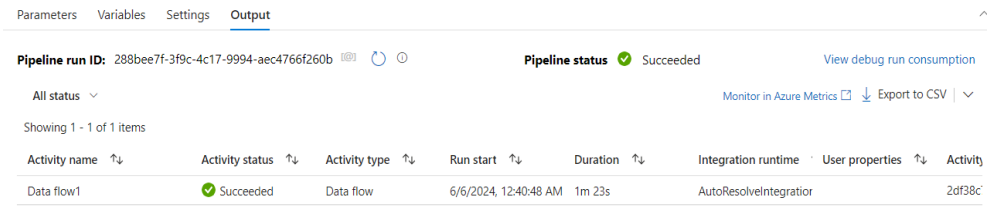
- The activities were saved/published, and all were successfully published and executed (refer to images 1, 2, 3).



- Image 1



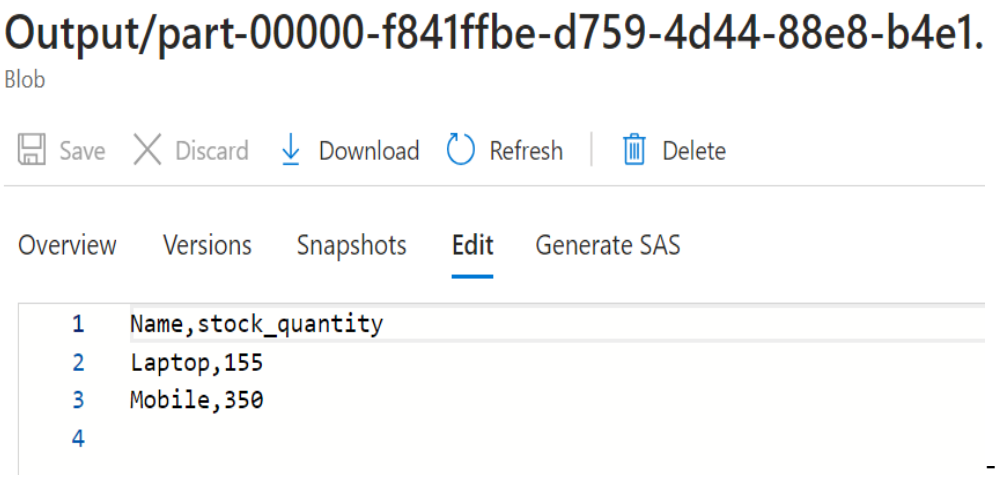
-Image 2



- Image 3

7. Verifying the Output

- I then visited the output folder in the adf-training container and checked the output.
- The file was generated correctly, and I got the sum of both items (refer image 1).



-Image 1

Summary:

This document outlines the steps taken to create an Azure Data Factory pipeline to calculate the sum of product quantities. The process involved uploading the input file to a container, creating source and target datasets, setting up dataflow with source, derived column, select, and aggregate functions, creating and executing a pipeline, and verifying the output. The pipeline successfully calculated and outputted the sum of the stock quantities of Laptops and Mobiles.