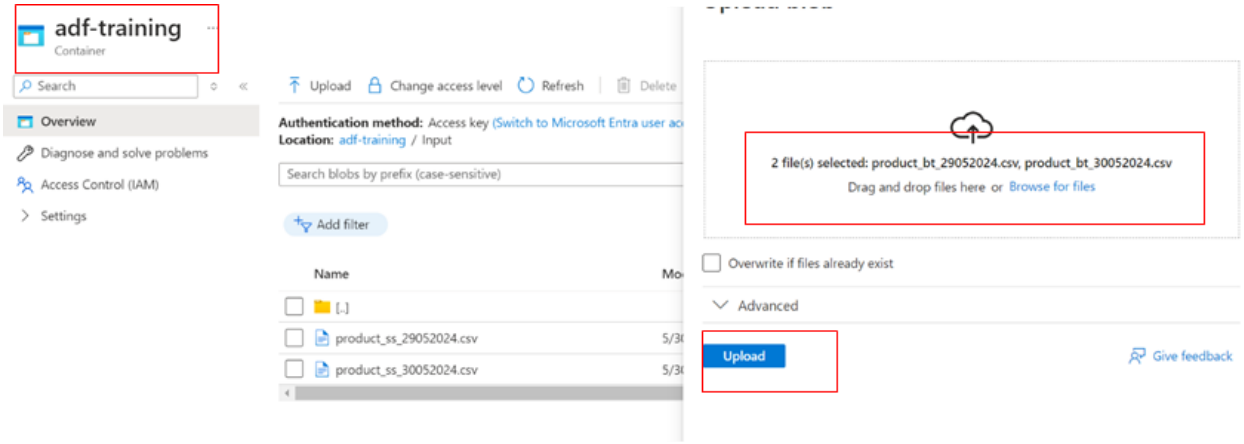
Formal Documentation of Azure Data Factory Pipeline – Training

Use Case: Merging Two CSV Files Using Copy Activity

Steps:

1. Loading the File into the Container.

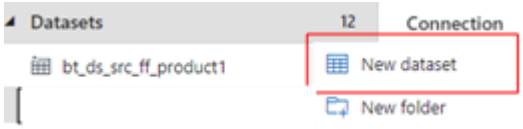
- A container named adf-training was previously created in the Azure storage account.
- For this pipeline, the files were uploaded into the container, in the input folder (refer to image 1).
- File names: product_bt_29052024.csv, product_bt_30052024.csv
- Once the upload was successful, the files were added to the input folder.



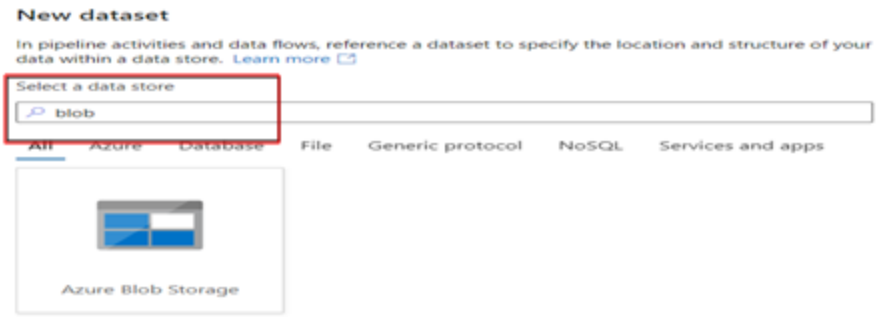
2. Dataset Creation.

Source Dataset

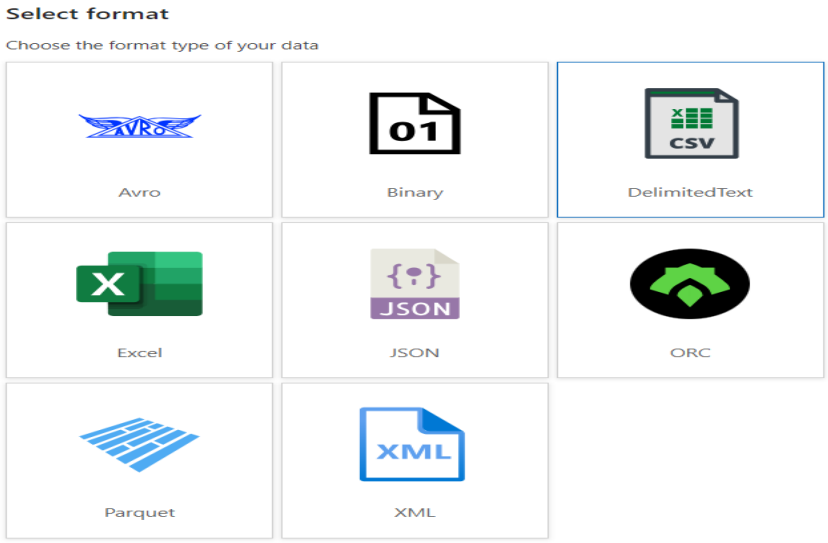
- A new source dataset was created for the source data files.
- Source dataset name: bt_ds_src_ff_multi_product
- Inside Azure Data Factory, in the Author tab, I selected the Dataset option and clicked on "New Dataset" (refer to image). I chose the Azure Blob Storage option (refer to image 1,2).
- Next, 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: adf-training/Input (refer to image 4).
- These steps created my source dataset (refer to image 5). I opened my source file and updated the connection, changing the column delimiter option to pipe (|) because my CSV file is pipe delimited.



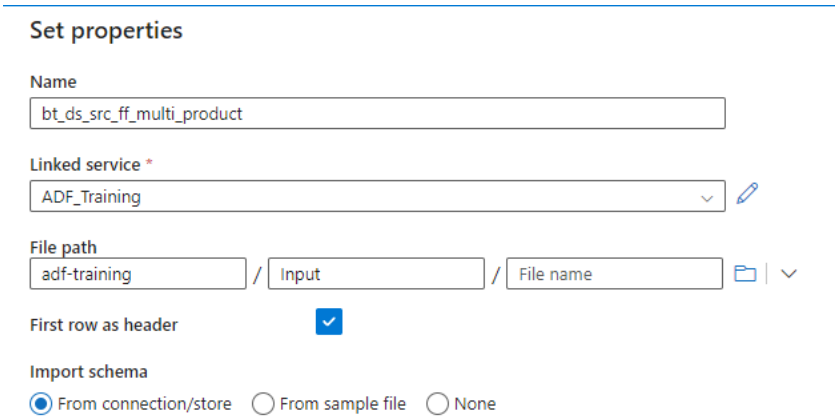
- Image 1



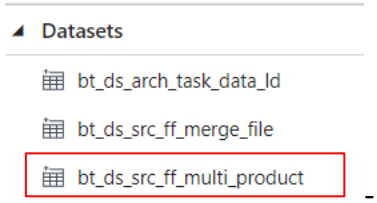
- Image 2



- Image 3



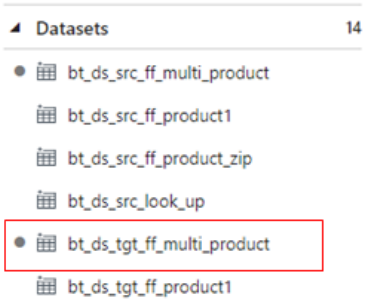
- Image 4



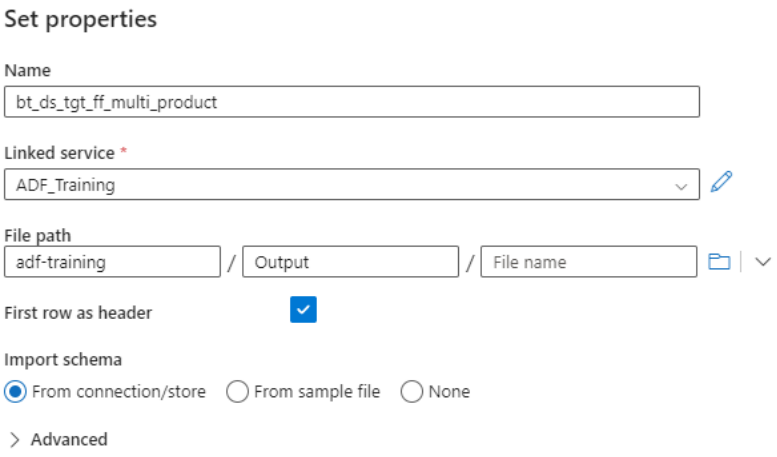
- Image 5

Target Dataset

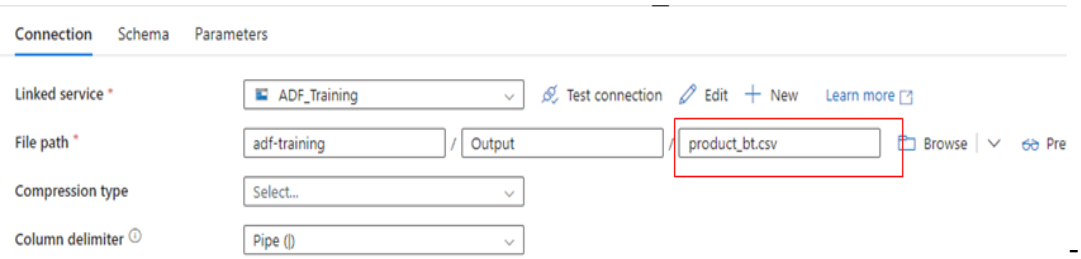
- I followed similar steps for the target dataset. In the Author tab, I selected the Datasets option, clicked on "New Dataset," selected Azure Blob Storage, and then selected the Delimited Text format, which brought me to the properties page where I defined the dataset name and path.
- Assigned path: adf-training/output
- Target dataset name: bt_ds_tgt_ff_multi_product(refer to image 1,2)
- I opened my sink file and updated the connection, changing the column delimiter option to pipe (|) because my CSV file is pipe delimited. All other options remained unchanged.
- Additionally, the file name must be specified in the file path to ensure the merged file is created using the designated name (refer image 3).



- Image 1



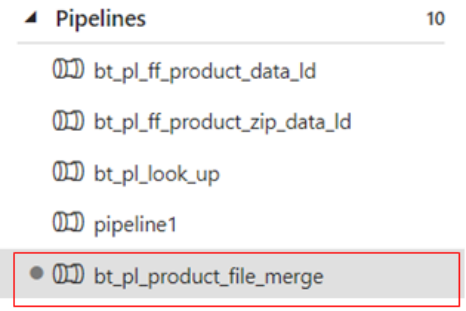
- Image 2



-Image 3

3. Pipeline Creation

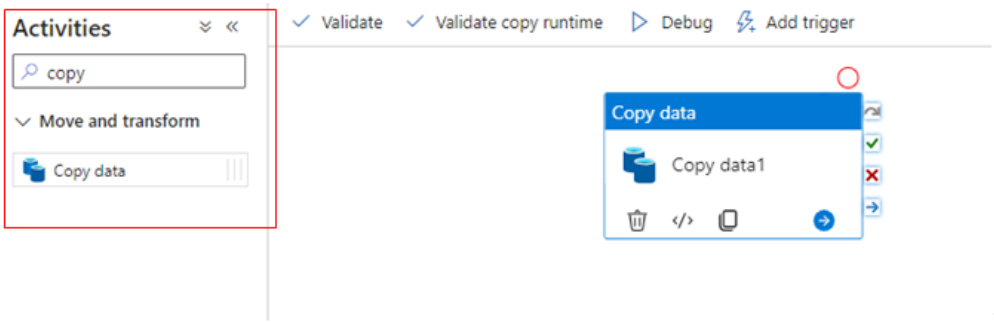
- A new pipeline named “bt_pl_product_file_merge” was created in Azure Data Factory (ADF) to load the merged file in the output folder. (refer to image 1).



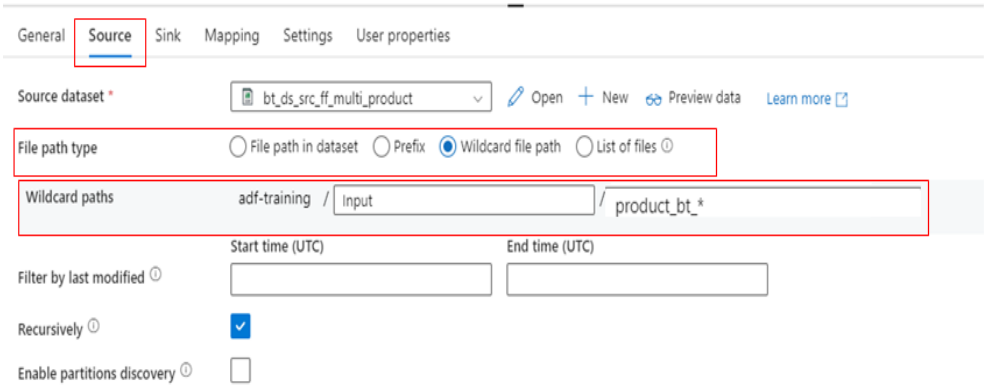
- Image 1

4. Copy Activity

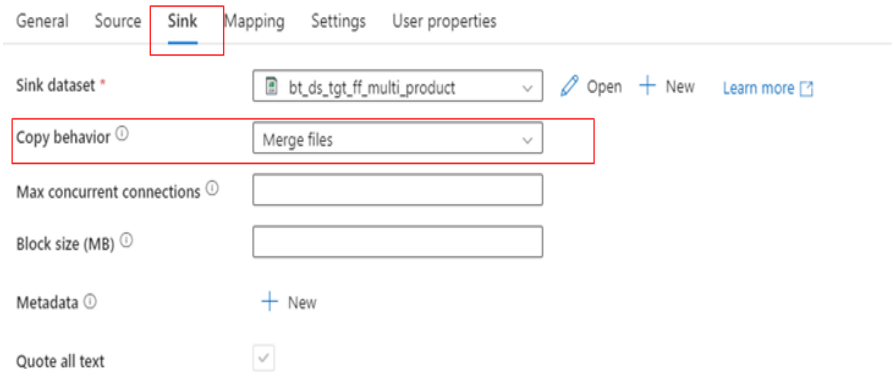
- The Copy Data activity was used to transfer the data from both files to the output file.
- Source: bt_ds_src_ff_multi_product
- Sink: bt-ds_tgt_ff_multi_product
- I selected the Copy activity from the Activities tab (refer to image 1).
- In the source option of the Copy activity, I added the source dataset and selected the Wildcard file path option as this option takes a wildcard pattern to match files to copy from the source location. In this case, the common factor in both CSV files is product_bt_* (putting an asterisk at the end is mandatory) (refer image 2).
- In the sink option, I added the target dataset and changed the file extension from .txt to .csv. Additionally, since our motive is to merge the two files, I selected "Merge files" from the drop-down of copy behavior (refer image 3).



-Image 1



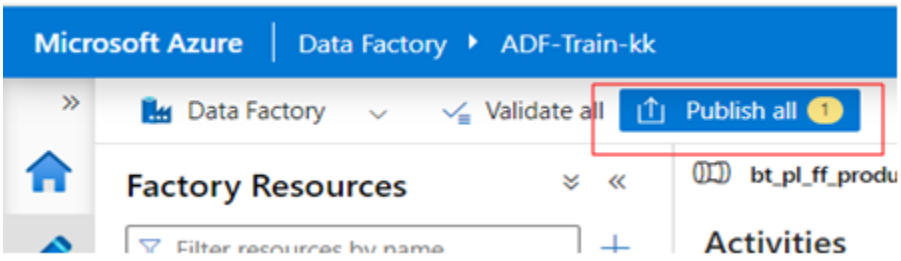
- Image 2



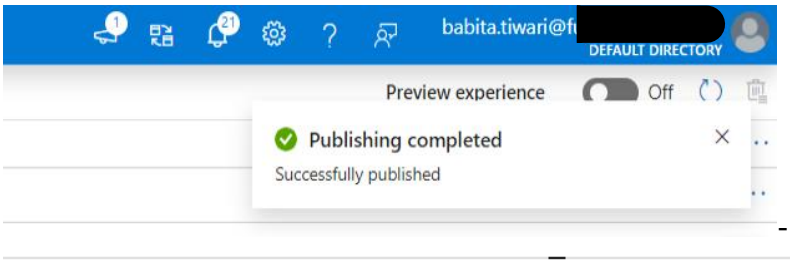
- Image 3

5. Publishing and Executing the Pipeline

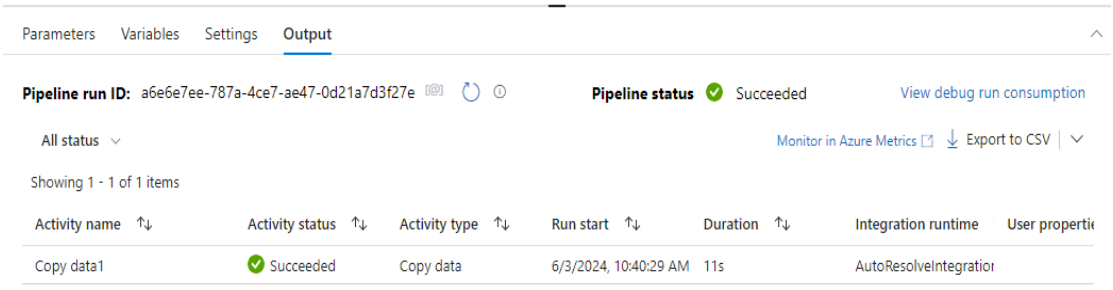
The activities were saved/published, and all were successfully published and executed (refer image 1,2,3).



- Image 1



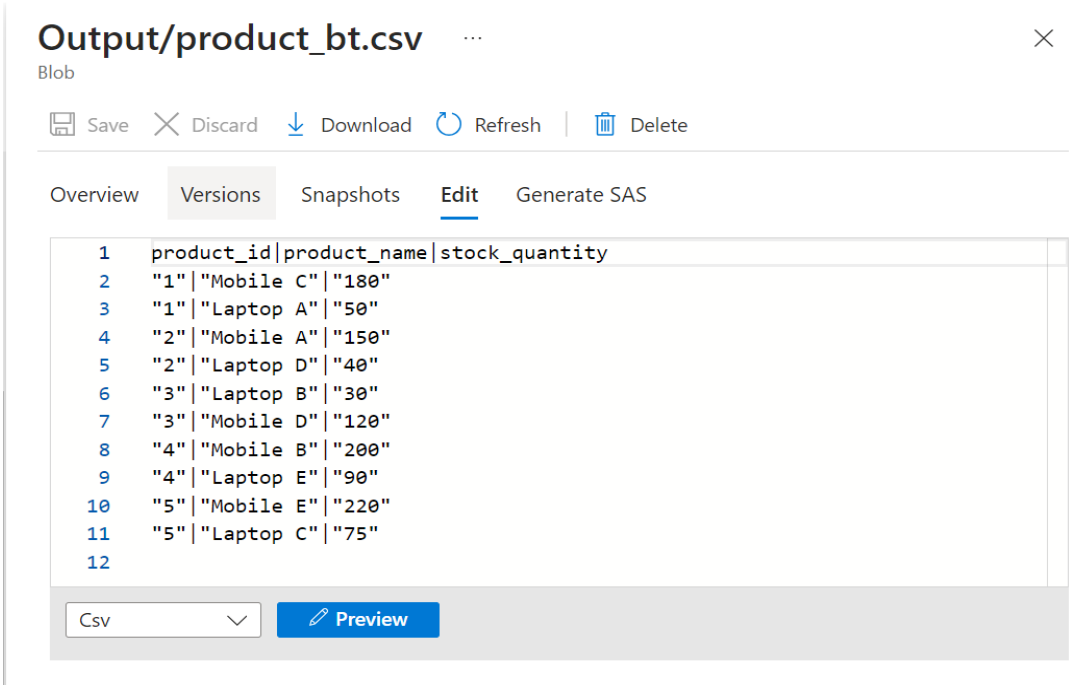
-Image 2



-Image 3

6. Checking the Output Files

- After the execution of the pipeline, I went to the adf-training container in the Azure storage account to check the output.
- A new file named product_bt.csv was created and the data from both files were merged (refer image 1).



- Image 1

Summary:
This documentation outlines the process of merging two CSV files using Azure Data Factory's copy activity. The steps include loading files into the container, creating source and target datasets, pipeline creation, configuring the copy activity, and verifying the merged output file.