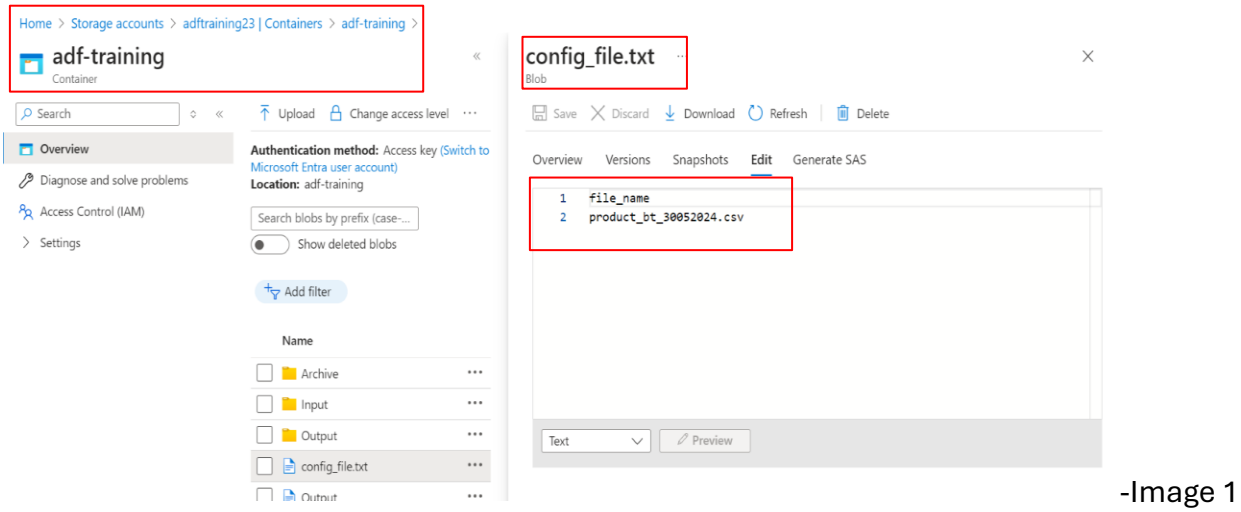
Formal Documentation of Azure Data Factory Pipeline – Training.

Use case: Transfer the data from source to target based on the file name specified in the config file.

Steps:

1. Creating a config file:

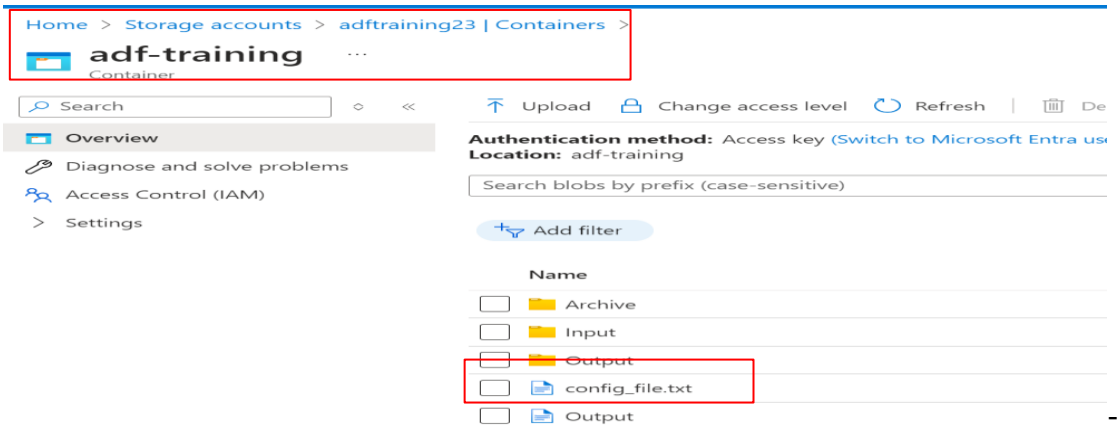
- Created a config file via Notepad, specifying the file name from which I want to copy the data to the target (refer image 1).
- Saved the file as config_file.txt.



-Image 1

2. Loading the file in the container:

- Uploaded a config file named config_file.txt to a container named adf-training which was previously created in the Azure storage account (refer image 1)
- Note: Not in the input or output folder but in the container along with the input and output folder.

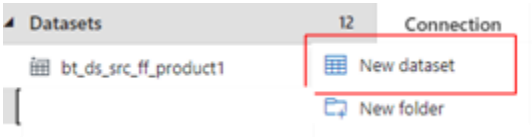


-Image 1

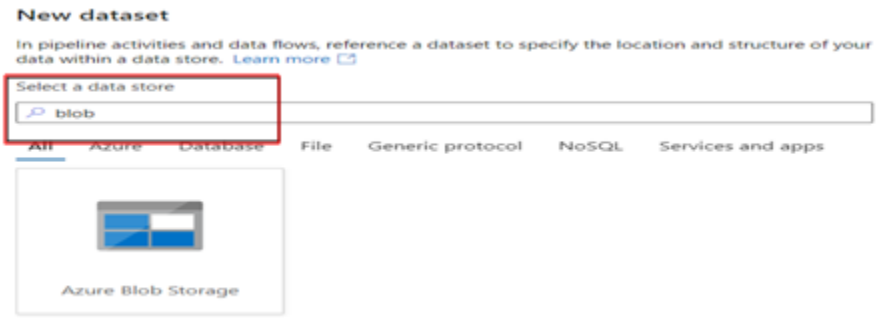
3. Dataset Creation:

Source dataset:

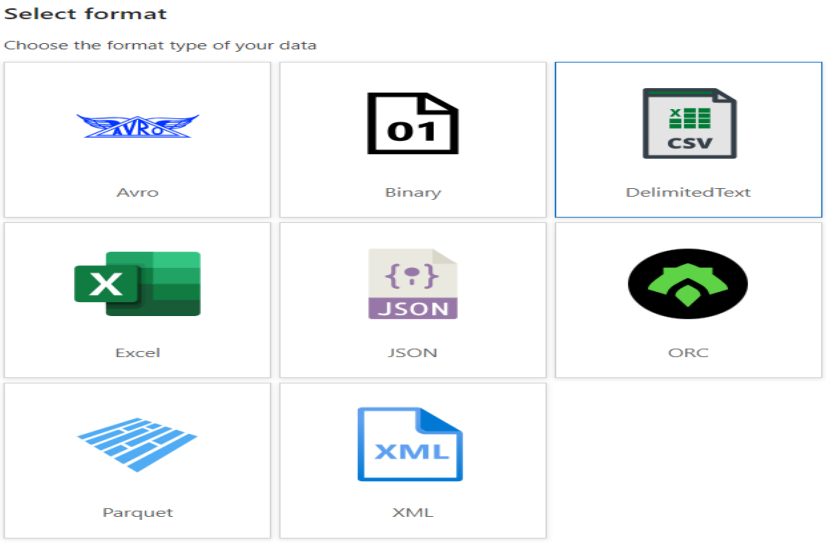
- Created a new source dataset for the source data files with the name bt_ds_src_look_up.
- Inside Azure Data Factory, in the Author tab, I selected the Dataset option and clicked on "New Dataset" (refer to image 1).
- Chose the Azure Blob Storage option.
- Next, I selected the Delimited Text file format, which brought me to the properties page where I defined the dataset name and path (refer image 2,3)



- Image 1



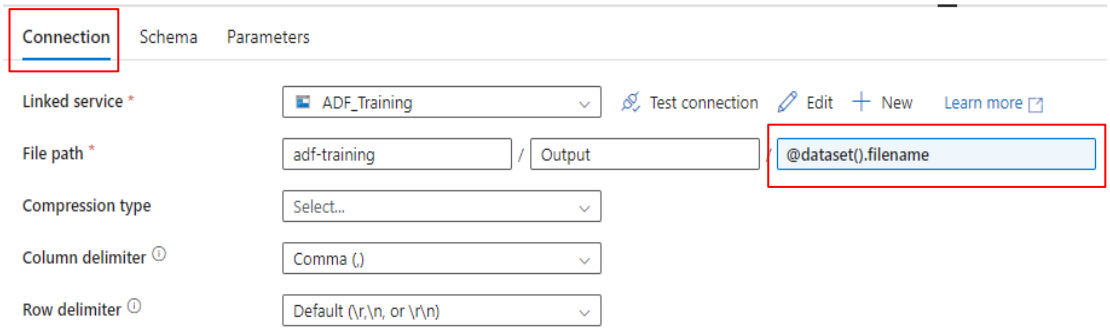
- Image 2



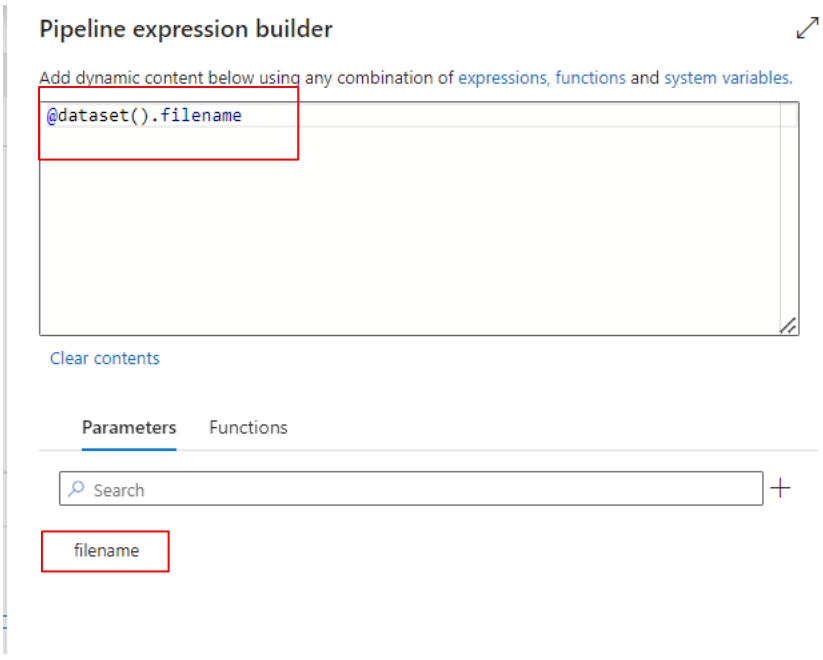
-Image 3

Target dataset:

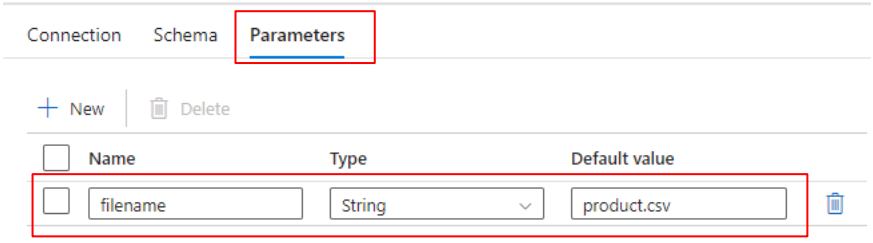
- Followed similar steps for the target dataset.
- Selected the existing target dataset named ss_ds_tg_product_adf_train.
- In the file path, dynamically added the file name. Clicked on add dynamic content and selected the filename option. It gave me an expression @dataset().filename. Selected it and clicked OK.
- In Azure Data Factory (ADF), the expression @dataset().filename is used to dynamically reference the filename of the dataset being processed. (refer image 1)
- By using @dataset().filename, we ensure that the filename used in our operations is always consistent with what is defined in the dataset's configuration file (refer image 2)
- In parameters, clicked on add parameter and assigned name as filename and gave a default value product.csv (refer image 3).



-Image 1



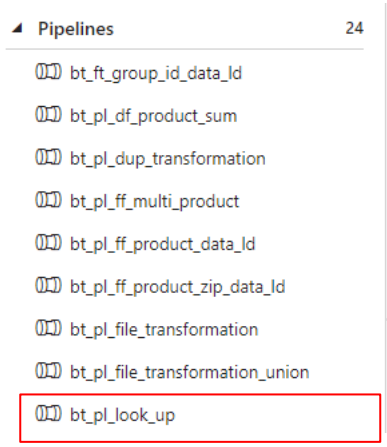
- Image 2



- Image 3

4. Pipeline creation:

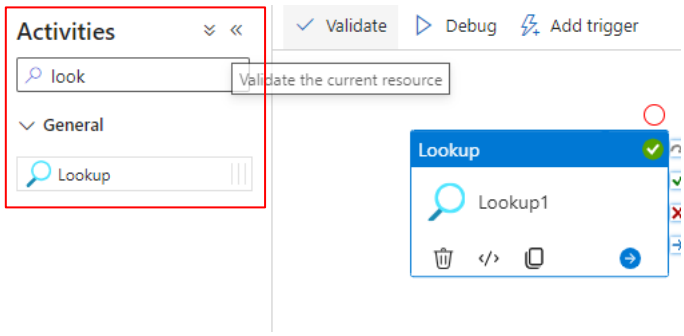
- Created a new pipeline named bt_pl_look_up in Azure Data Factory (ADF) to load the data into the output folder (refer image 1).



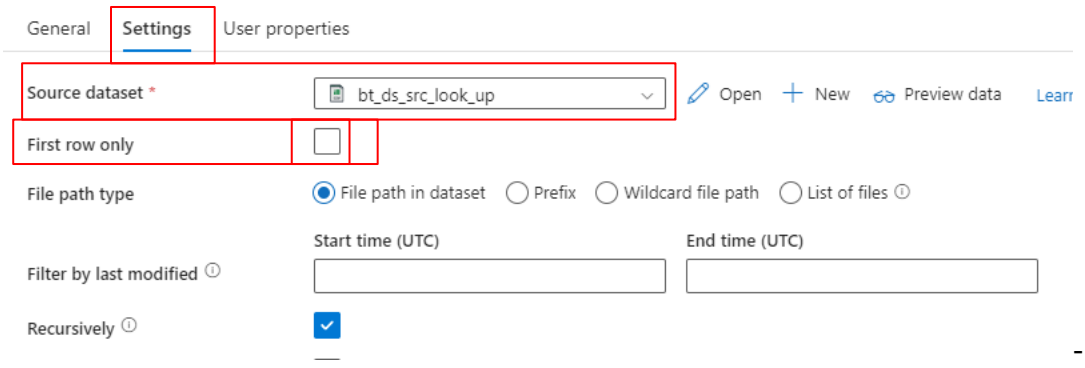
- Image 1

5. Lookup activity:

- Used lookup activity Lookup activity for the purpose of retrieving specific data or configuration settings that are required for the pipeline to process the file correctly (refer image 1).
- This ensures that the correct file is referenced and processed based on the dynamic content specified in the dataset's configuration.
- In the settings, I specified the source data and deselected the first row only option (refer image 2).



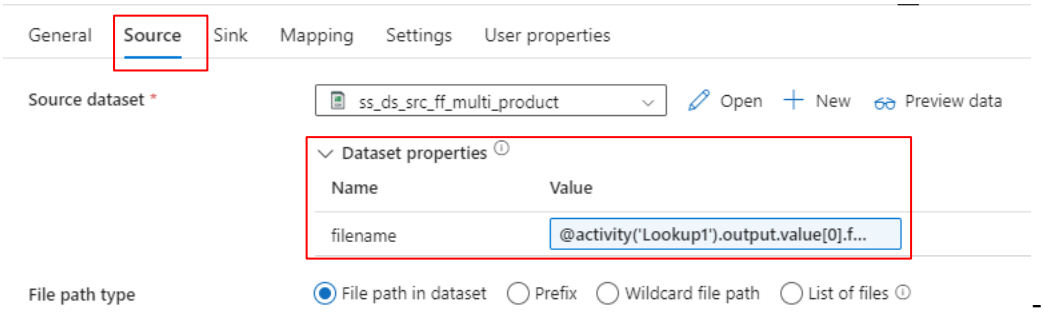
-Image 1



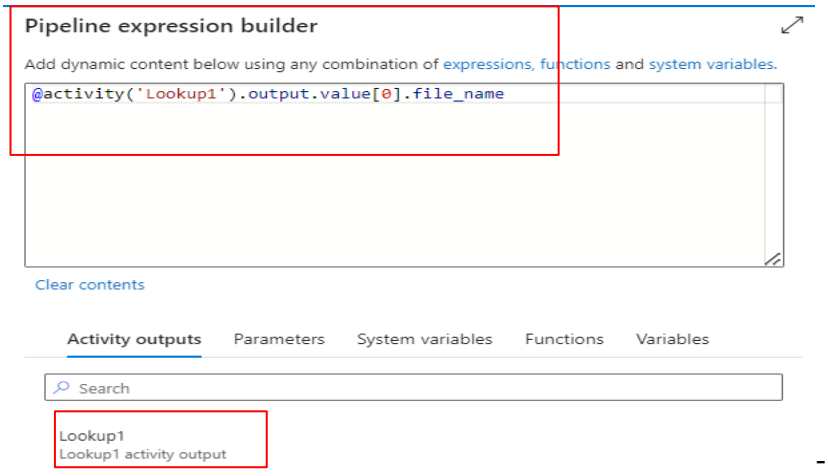
- Image 2

6. Copy activity

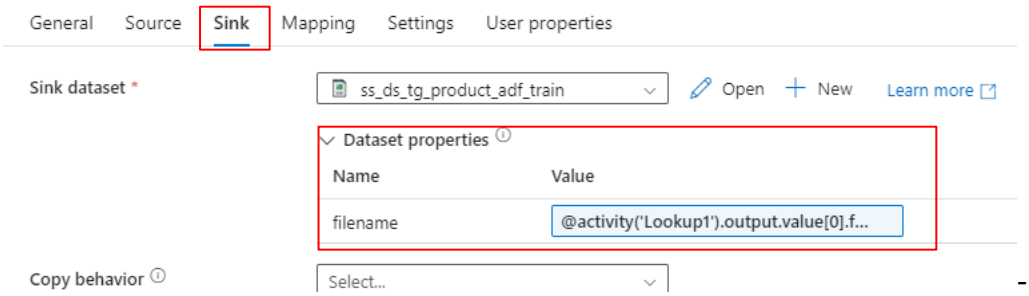
- Selected the copy activity to copy data from a source to a destination.
- In the settings, source option, I added the source file and in dataset properties, I added an expression by clicking on add dynamic content (refer image 1).
- Expression: @activity('Lookup1').output.value[0].file_name (refer image 2).
- The expression @activity('Lookup1').output.value[0].file_name is used to dynamically reference the file_name field from the output of a Lookup activity named Lookup1.
- I did the same in the sink option, selected the target dataset and wrote an expression (refer image 3)
- Also, connected the copy activity with the lookup activity (refer image 4).



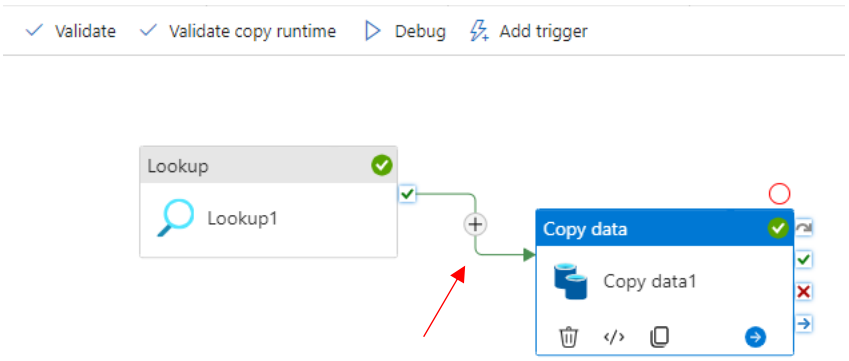
- Image 1



- Image 2



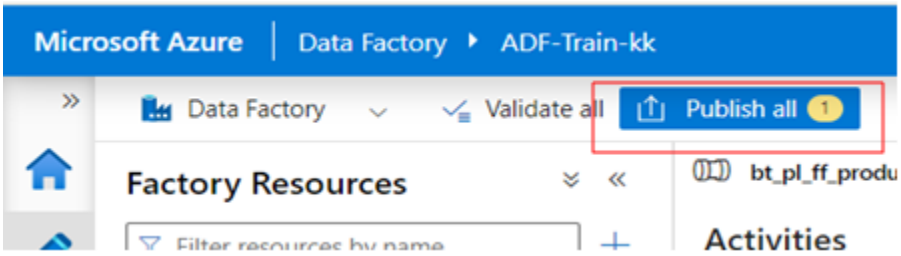
-Image 3



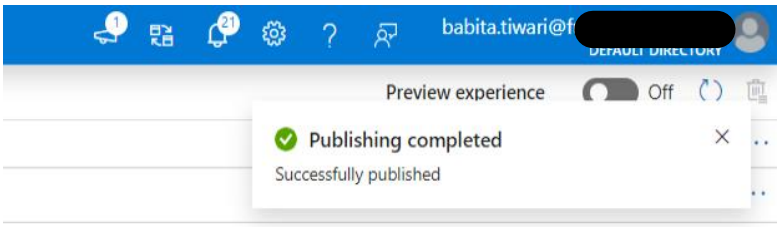
- Image 4

7. Publishing and Executing the Pipeline:

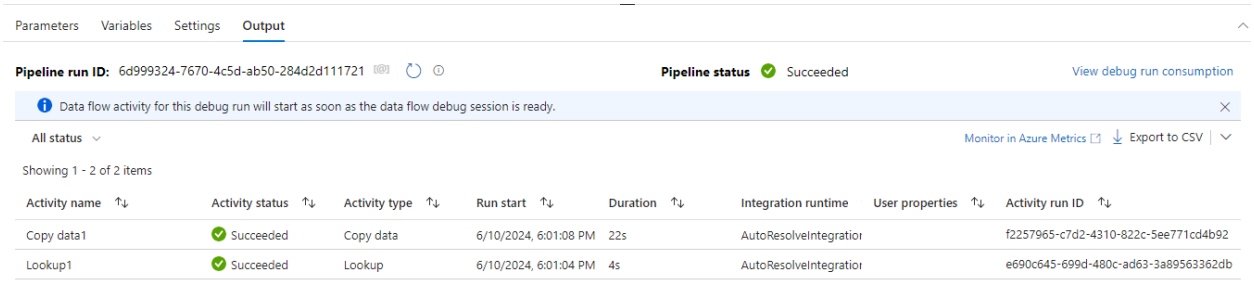
- The activities were saved/published, and all were successfully published and executed (Image 1, 2, 3).



- Image 1



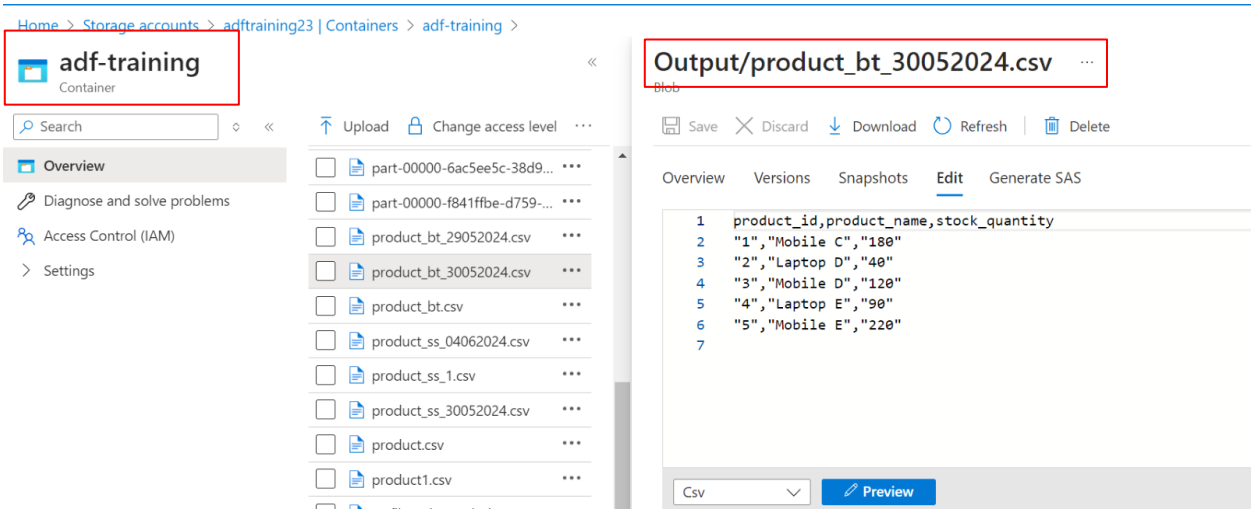
- Image 2



-Image 3

8. Verifying the output:

- I then visited the output folder in the adf-training container and could see that the file was successfully loaded into the output folder (refer image 1).



-Image 1

Summary:

The use case involves transferring data from a source to a target based on the file name specified in a config file.