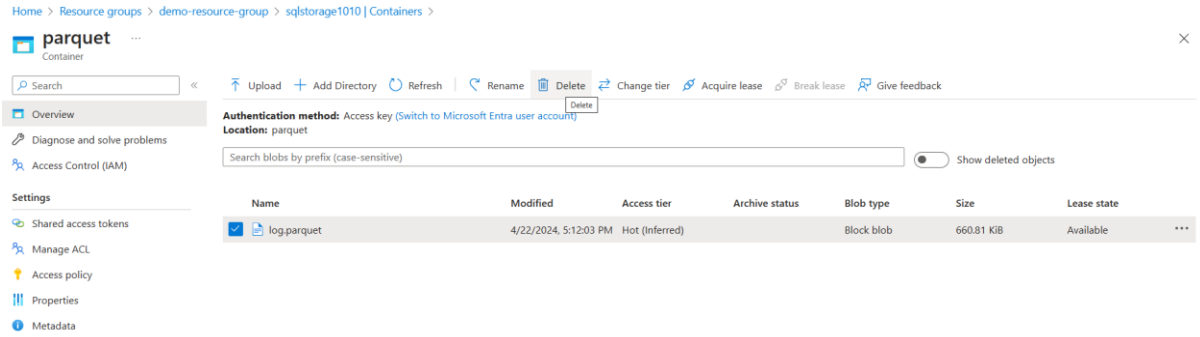
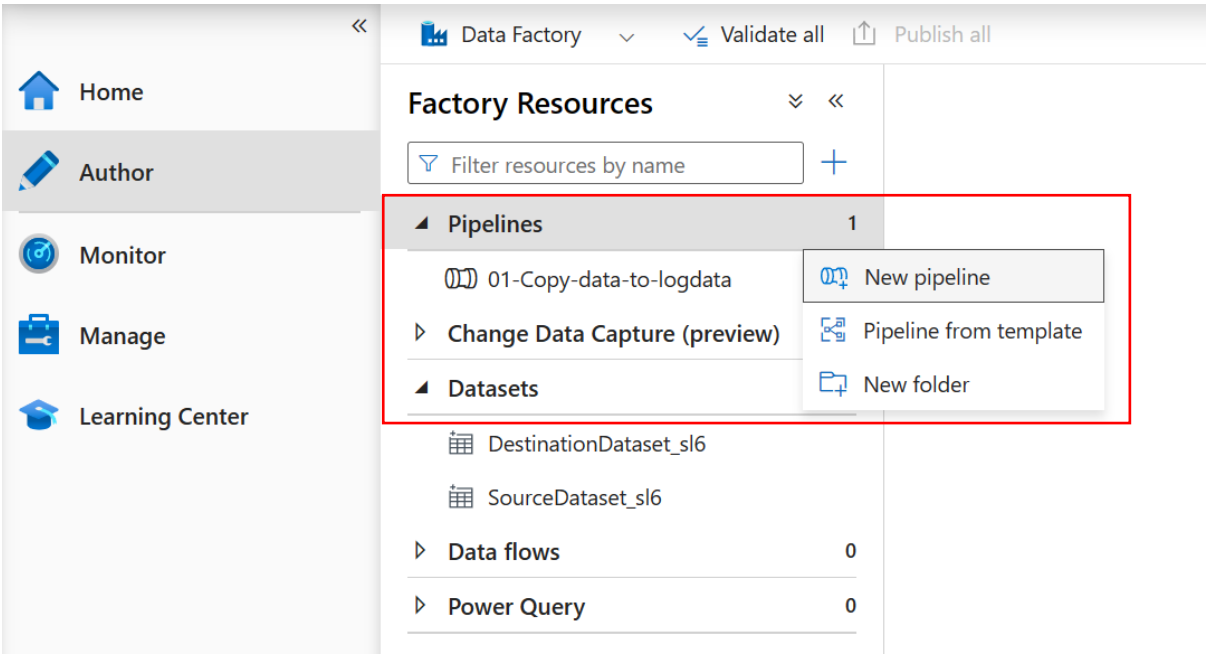
😊 Generating a Parquet File

In this lab, we're using Azure Data Factory to create a Parquet file from a CSV file. The end goal is to automate the process of converting data from one format to another for efficient storage and analysis. This involves configuring pipelines, datasets, and activities in Azure Data Factory to copy data from the CSV file to a Parquet file, ensuring compatibility and optimization for downstream analytics tasks.

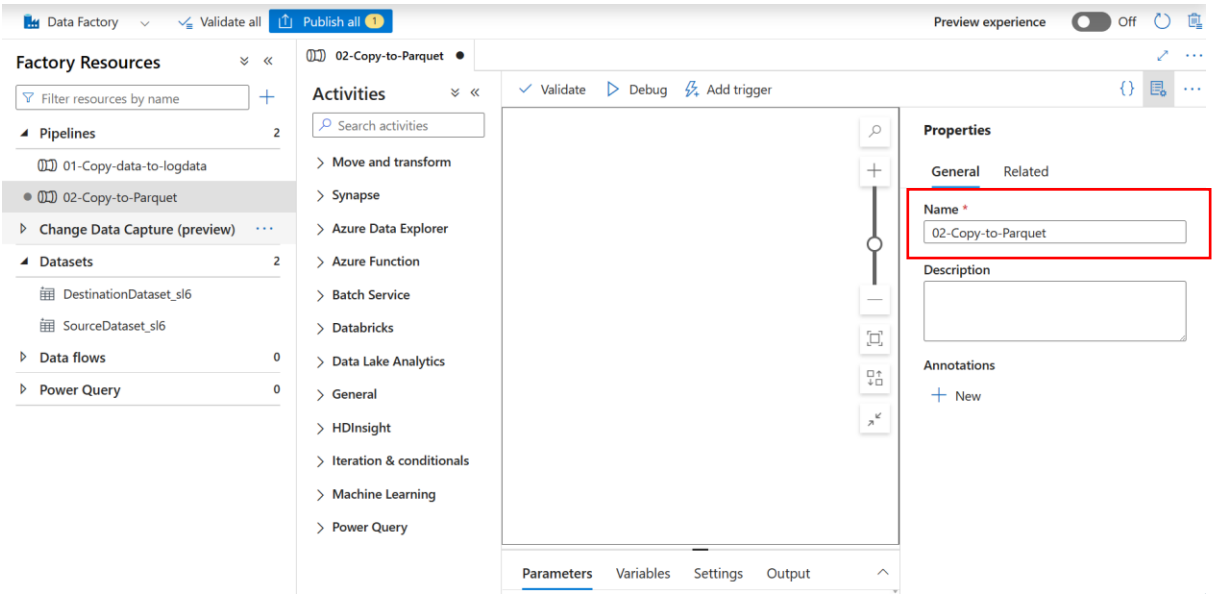
1. Now before starting the lab first, we go to our storage account then go inside our parquet container and we are going to delete our log. parquet file.



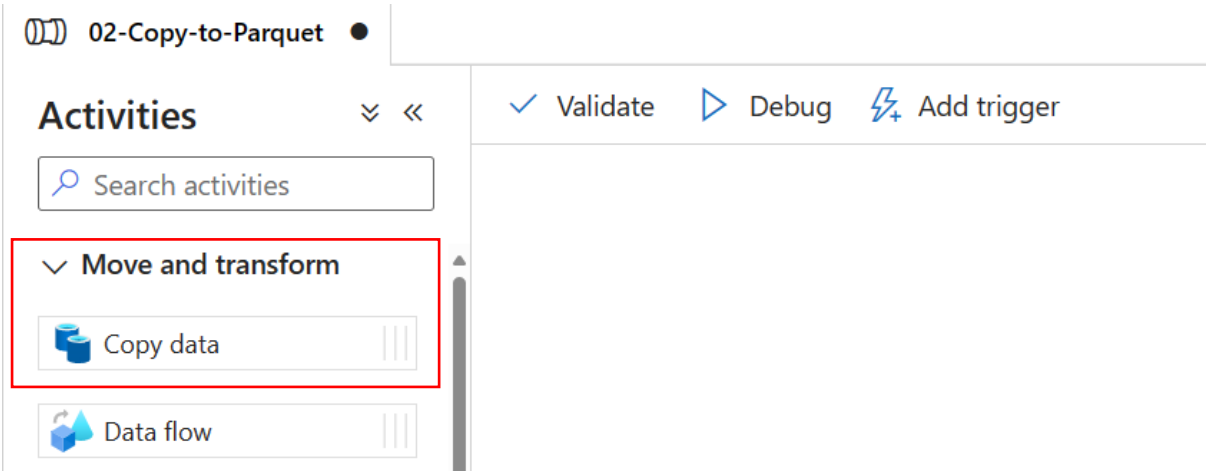
2. In this lab we are going to create a parquet file using our CSV file in the data factory.
3. Now in your data factory wizard first go to Author there you can see Pipelines, now click on the three dots, and choose new Pipeline from here itself.



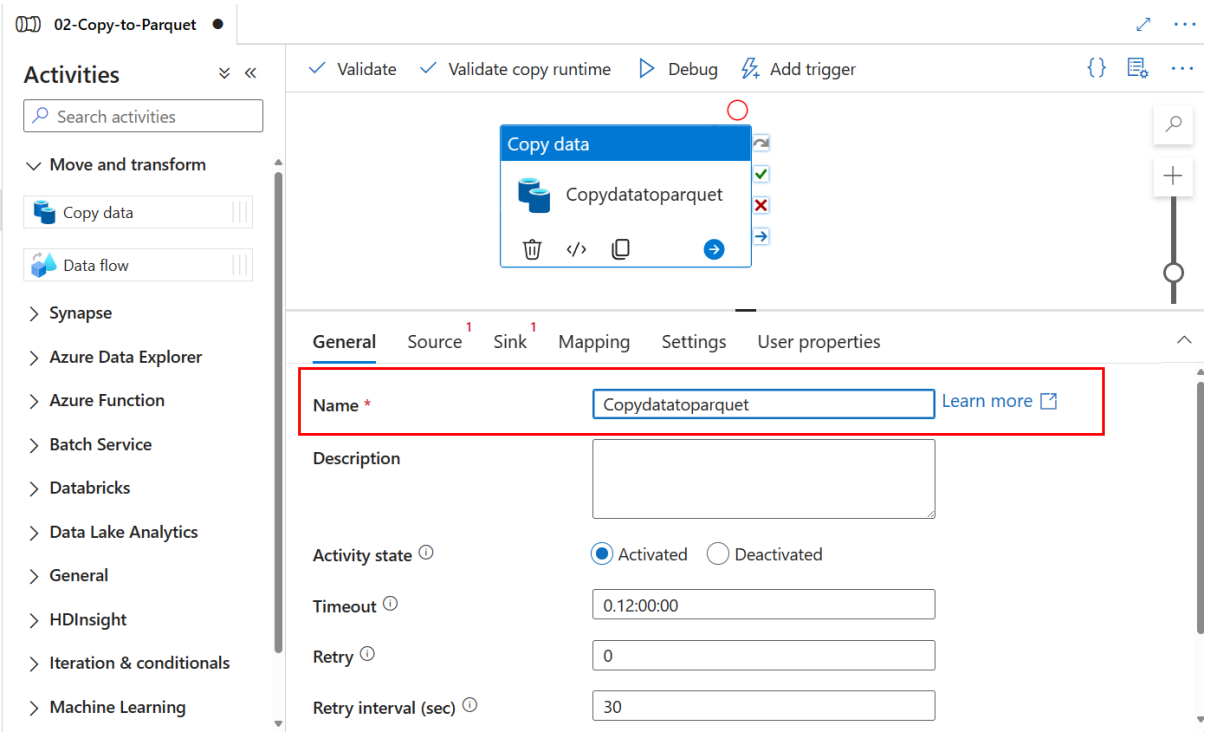
4. Then you will see that this type of interface layout has been generated by the wizard. First, we are going to give a name to this pipeline.



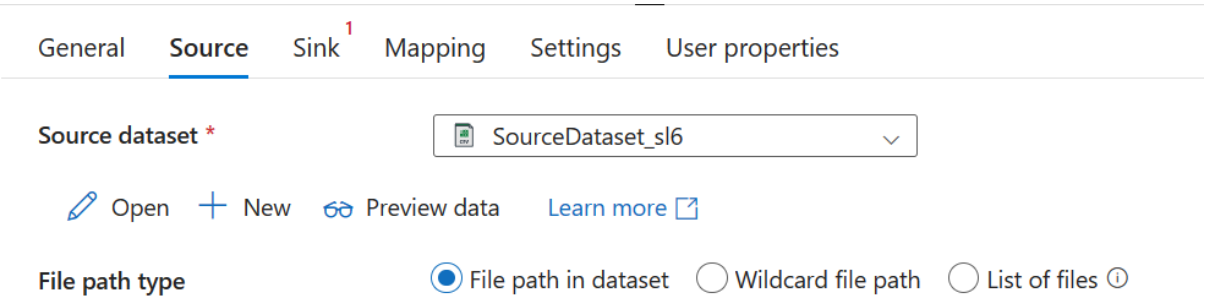
5. Now form the activities if you expand the move and transform section then you will see two options one for copying the data and the other for data flow.
6. Here we must choose to copy data and drag it to the canvas. So, this will allow us to copy data from our log.csv file to the parquet file which we will create moving further.



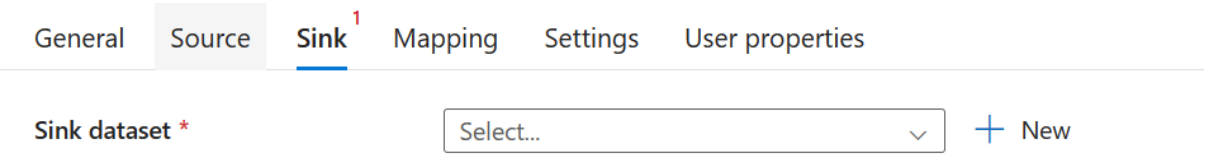
7. First, we are going to give a name to our activity. Then we will move to source.



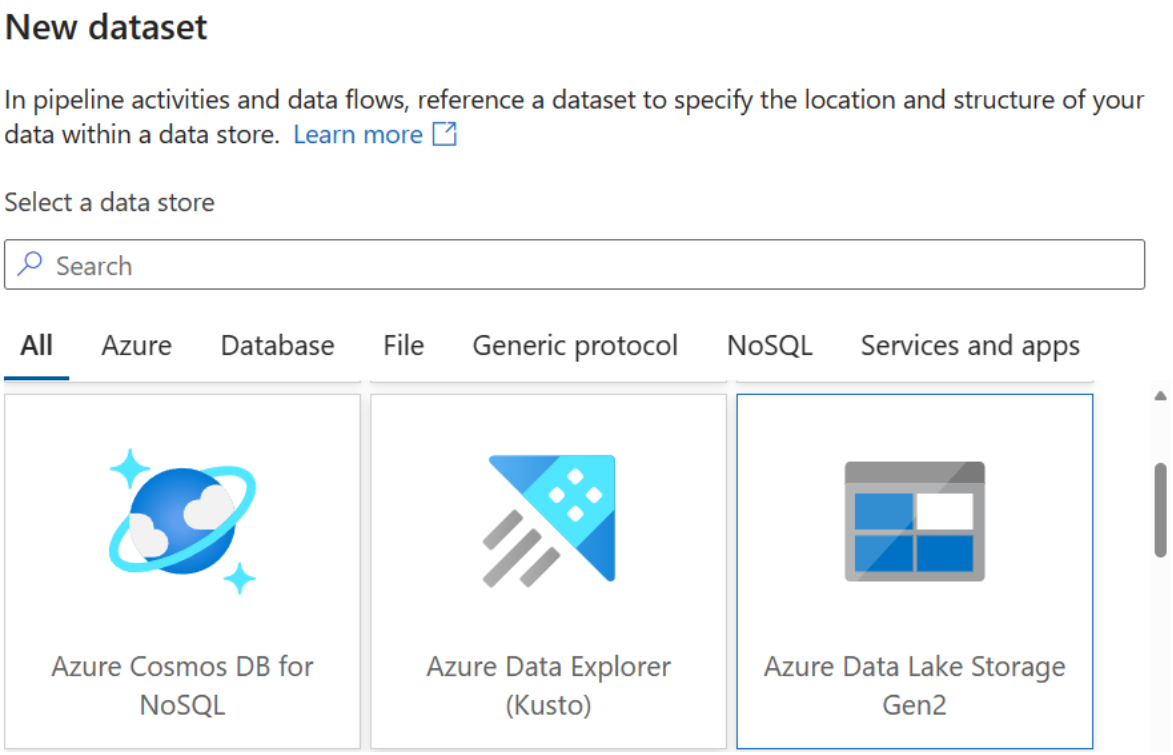
8. Now in source we will choose our source dataset. It means that from where we want to pick up our data.



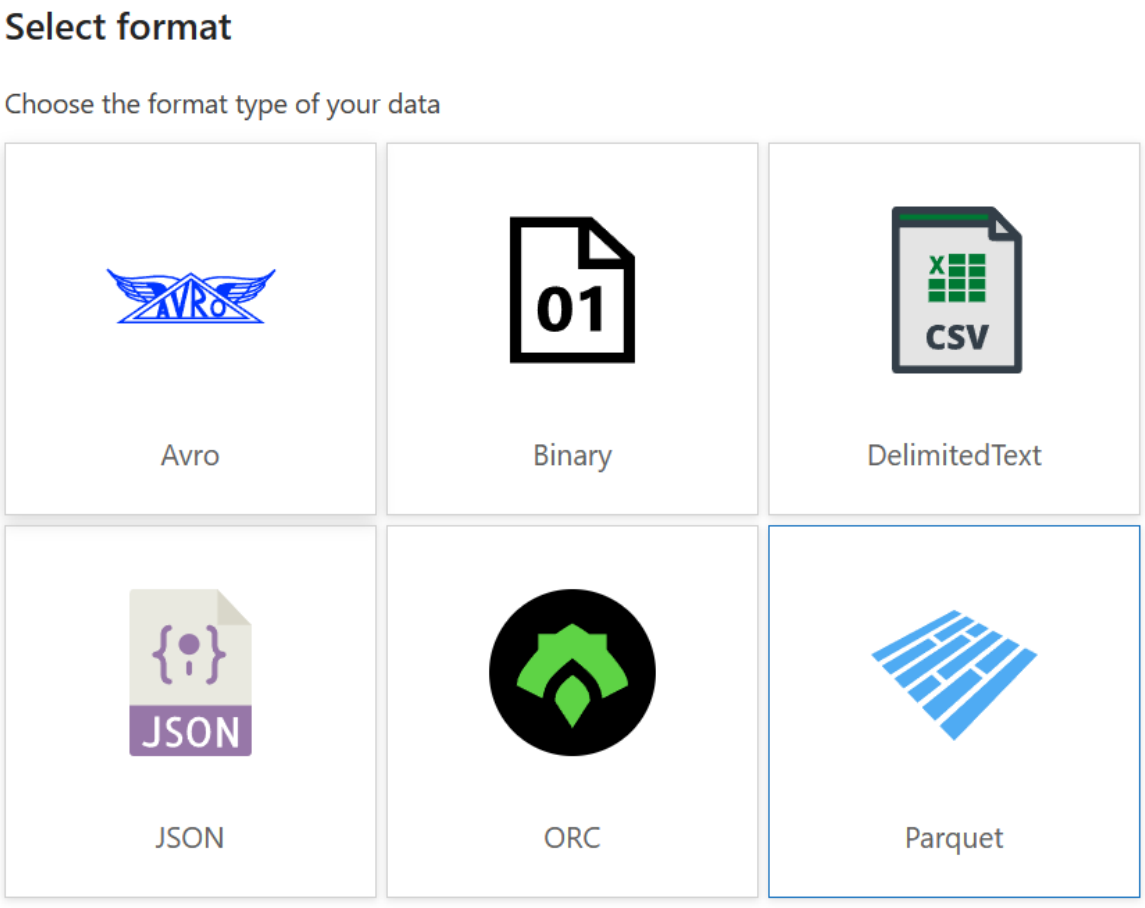
9. Then in the sink you must click on New. Here we will provide the destination which is our parquet-based container. For this, we need to create a new dataset.



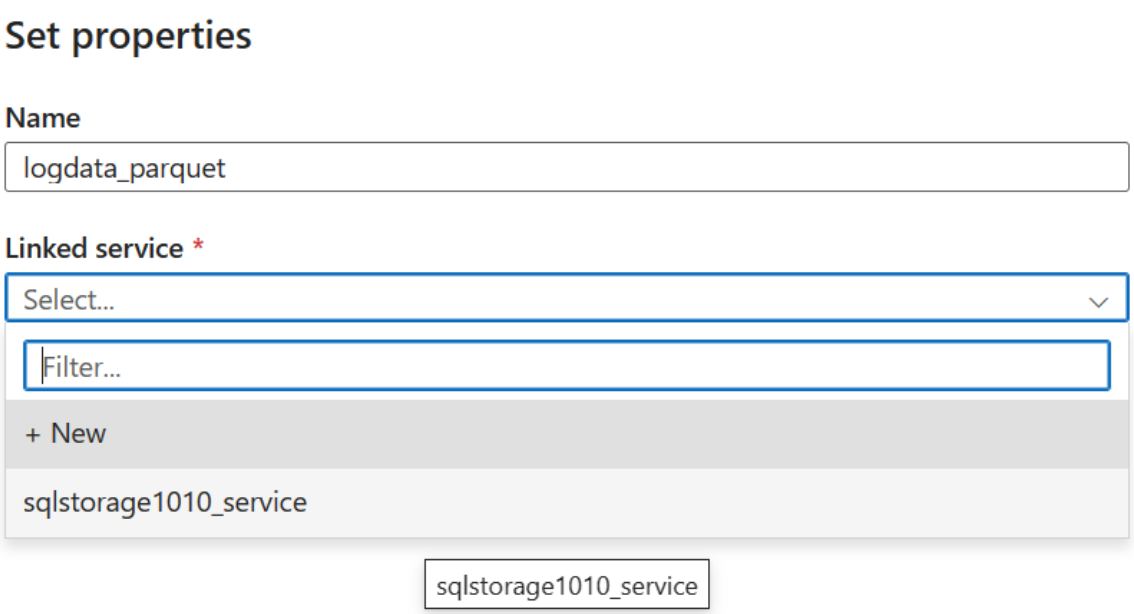
10. Again, you have to choose Azure data late gen2 and click on continue.



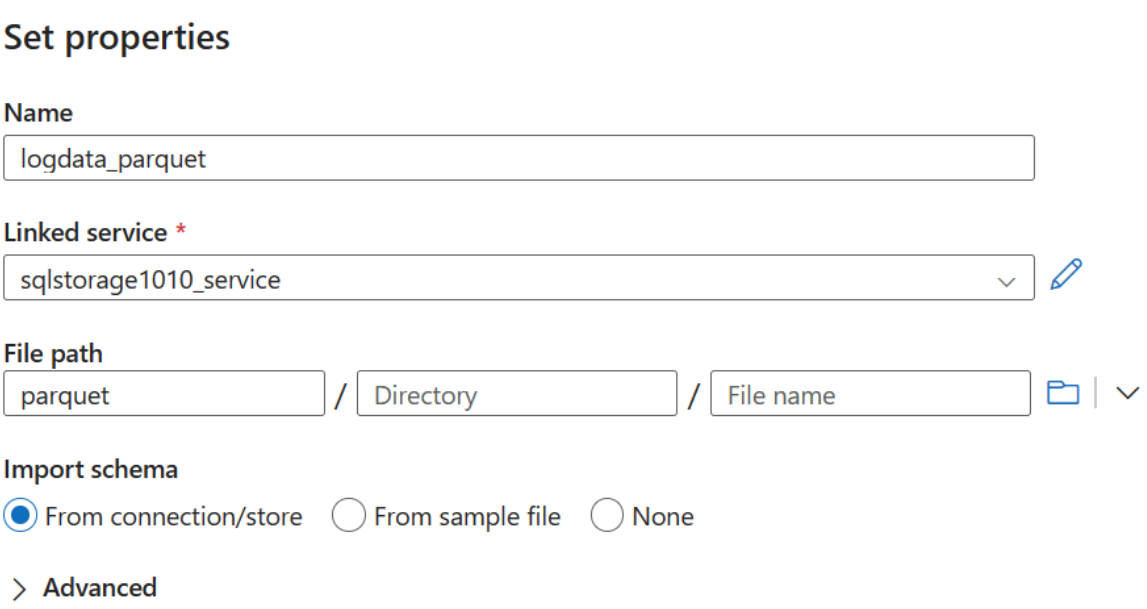
11. Now this time around you have to choose a format for our data. So, we will choose Parquet and click on continue.



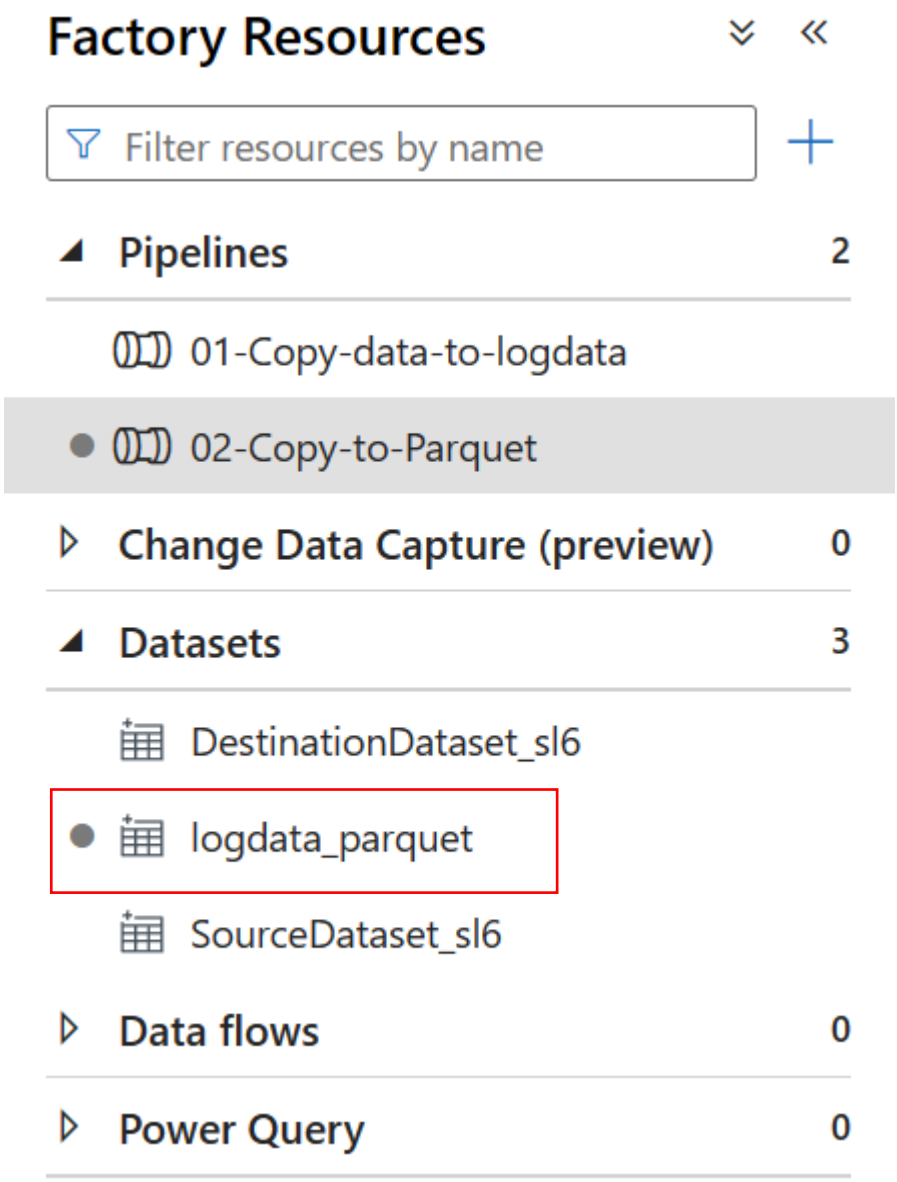
12. Then in the properties first you will give a name to your file after that in the linked service you have to choose your storage account as you can see below.



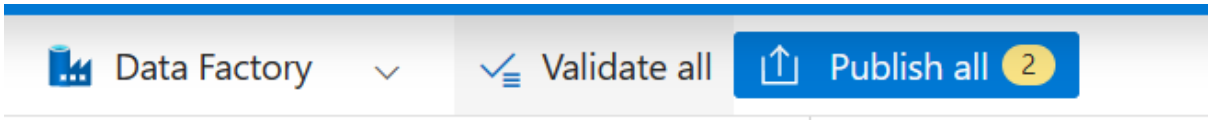
13. Then you have to browse for the container where you want to put your file then click on the Ok button.



14. Now if you look at your factory resources you will see that your dataset has come over here.
15. In the wizard, when we do or choose the copy data activity, there, it does not give you the facility to give a name for your datasets. But here, since we are creating our pipeline from scratch, we can now also mention what is the name of our dataset.

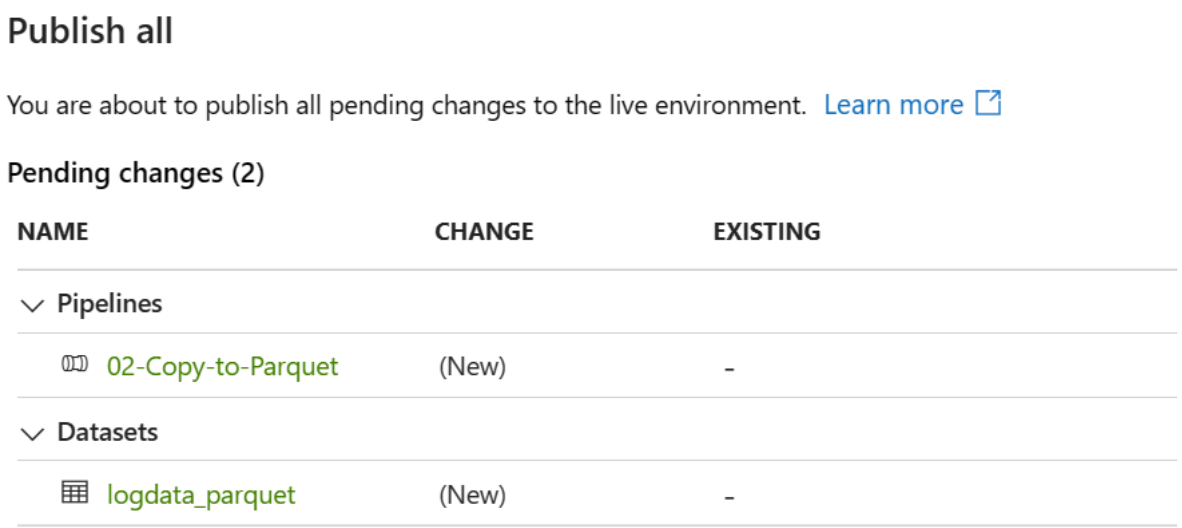


16. So now, we have our source and our sink in place. Before we can trigger this pipeline, or say before we can run this pipeline, so in the wizard, which we have seen in the earlier labs, the wizard would go ahead and create our pipeline, publish our pipeline, and even run the pipeline.
17. But since we are doing everything from scratch here, we need to first publish our pipeline and then run our pipeline.
18. To make sure that our pipeline is correct we have to click on validate all.

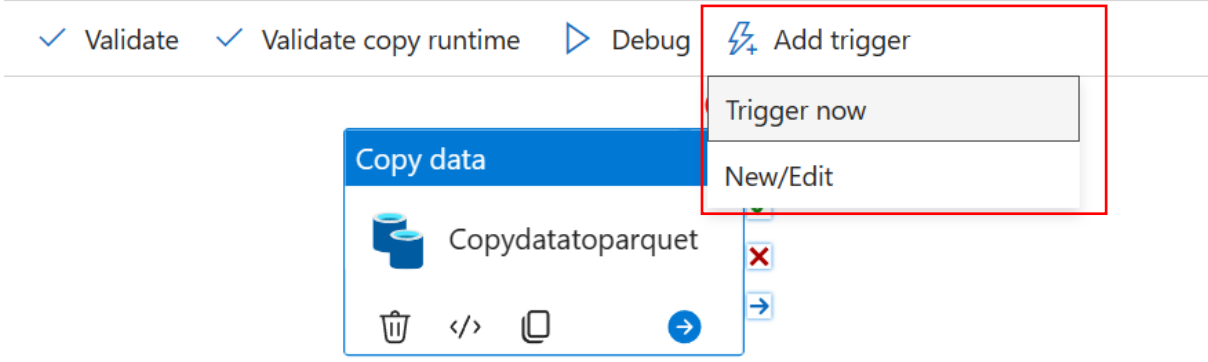


19. Once you have clicked on it then you will see this type of message.

20. Now you have to click on publish all. You can see that you have a pipeline and a dataset to publish click on publish.



21. Once your publishing has completed. Click on add trigger and then on trigger now.



22. This will trigger our pipeline.

Pipeline run

Trigger pipeline now using last published configuration.

Parameters

Name	Type	Value
No records found		

23. Now once you have triggered the pipeline after some time you will see that you are getting an error message. And if you click on view error details then you can see the error message too.

Notifications

Dismiss all

Run Failed

Failed to run 02-Copy-to-Parquet (Pipeline).

View pipeline run

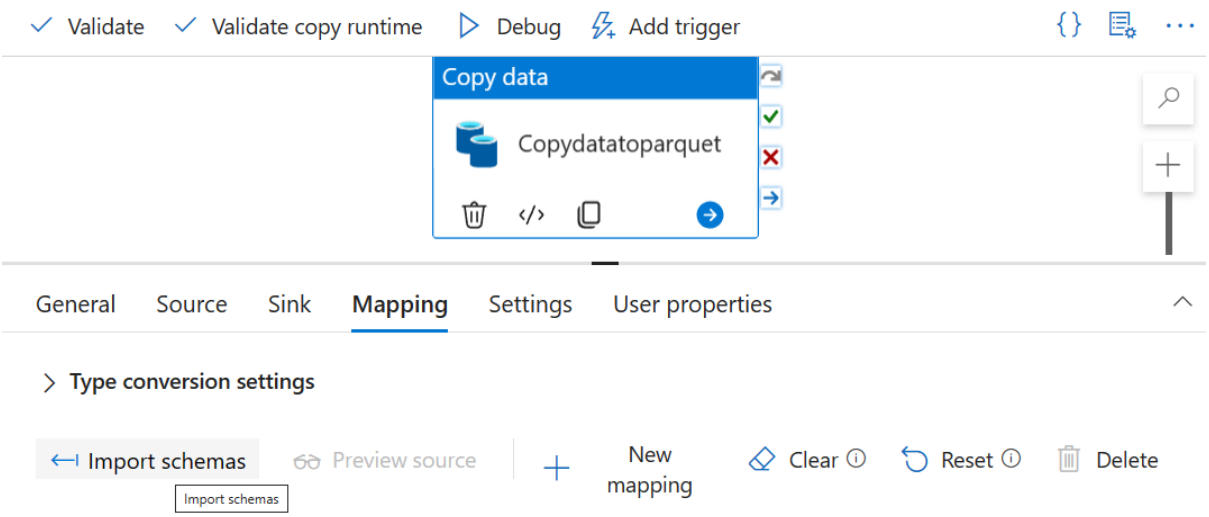
View error details

9 minutes ago

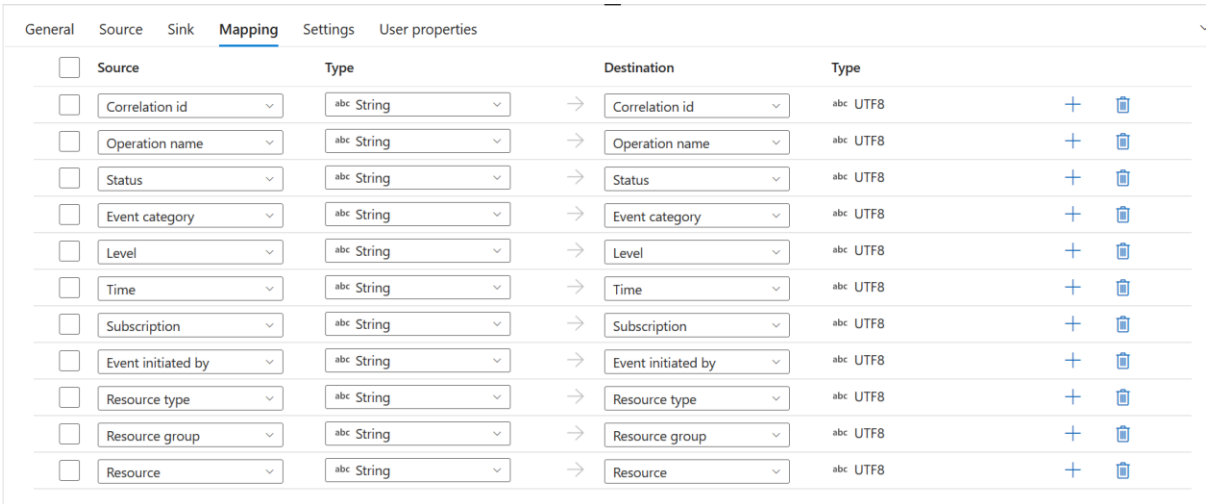
Error

Operation on target Copydatatoparquet failed: ErrorCode=ParquetInvalidColumnName,'Type=Microsoft.DataTransfer.Common.Shared.HybridDeliveryException,Message=The column name is invalid. Column name cannot contain these character:[,;{}()\n\t=],Source=Microsoft.DataTransfer.Common,'

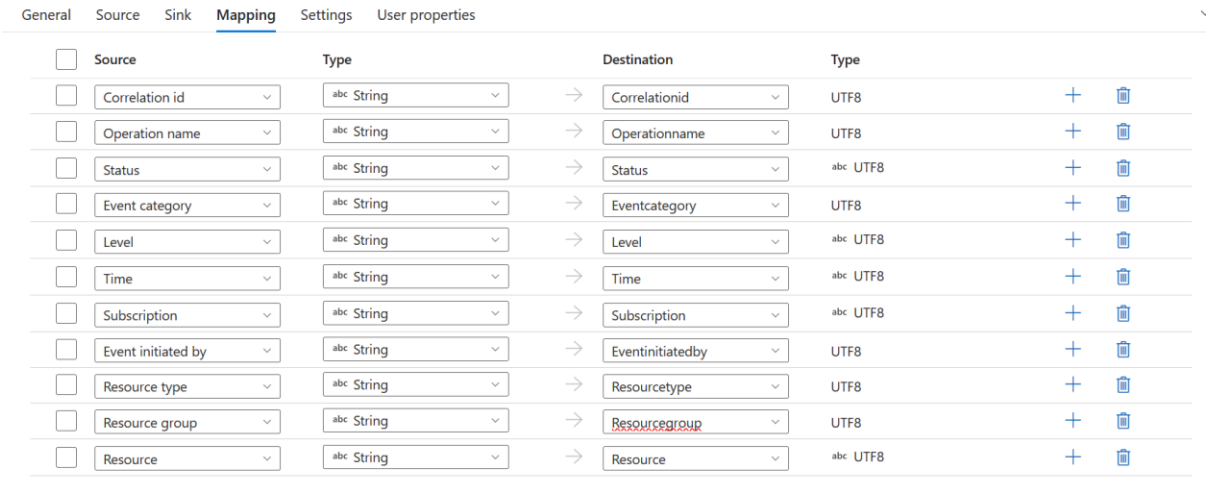
24. We are getting this error message because in Parquet-based files we cannot have spaces in the column names.
25. In a log.csv file, there is a space in certain columns. For example, there's a space in the operation name, there's space in correlation ID, etcetera. This is not accepted when it comes to the Parquet-based file format, so we need to make a change.
26. In your pipeline you can see the mapping section, here you have to click on Import Schemas.



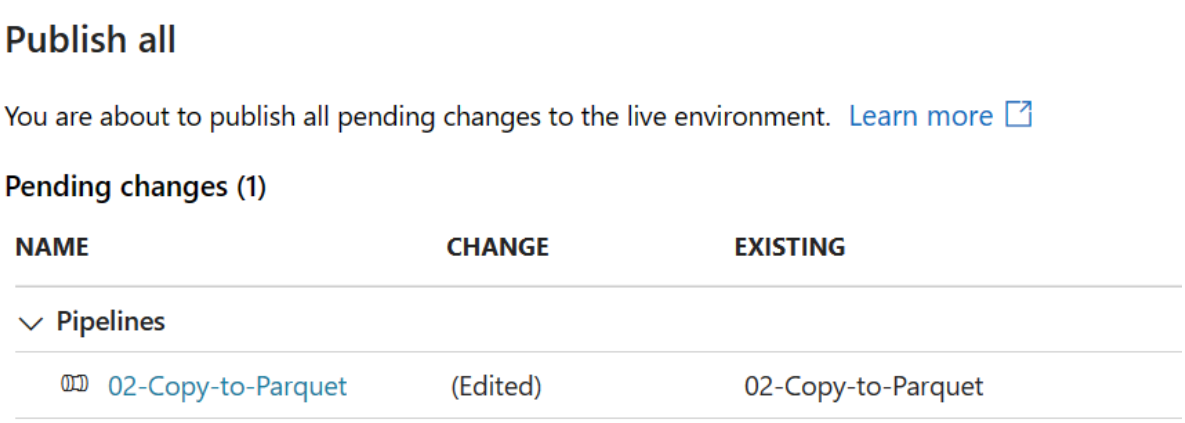
27. Below you can see the mapping.
28. Azure Data Factory is doing an automatic mapping based on its knowledge. It doesn't know about understanding whether columns have spaces in the Parquet-based destination file format. Hence, we are now making a change saying that when you're copying onto the destination, don't have spaces in the column names.



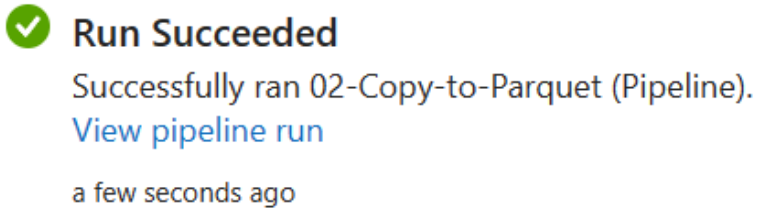
29. Below you can see that we have removed all the spaces.



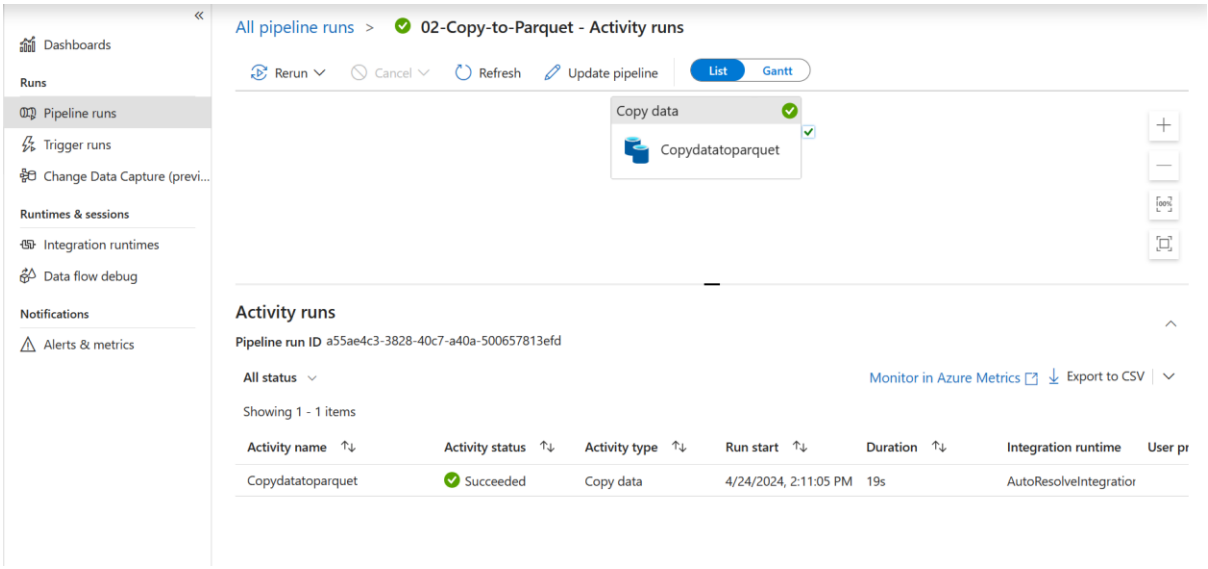
30. Once you are done now you have to click on public again.



31. Once the publishing is completed then you have to click on trigger now. Below you can see that the run was successful.



32. Now go to the monitor section and look for your pipeline it has been succeeded.



33. Now go to your storage account and open your parquet-based container. There you will see your parquet file in place.