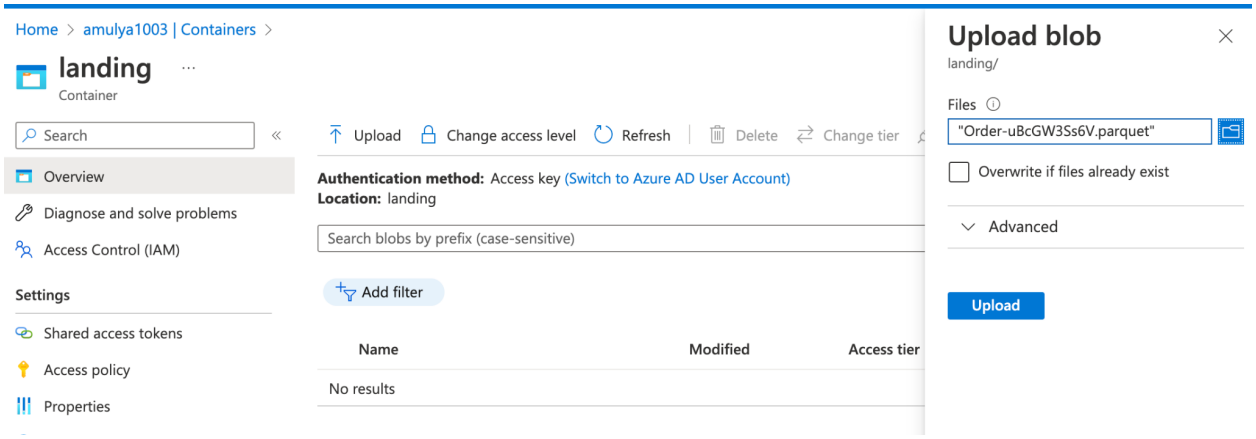
Copy Parquet file to JSON using Copy Activity in Azure data factory.

Create a storage Account if not exists and upload parquet file in a container.

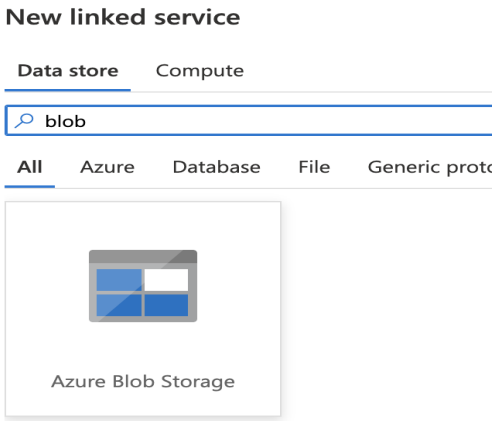


Please refer to this link :

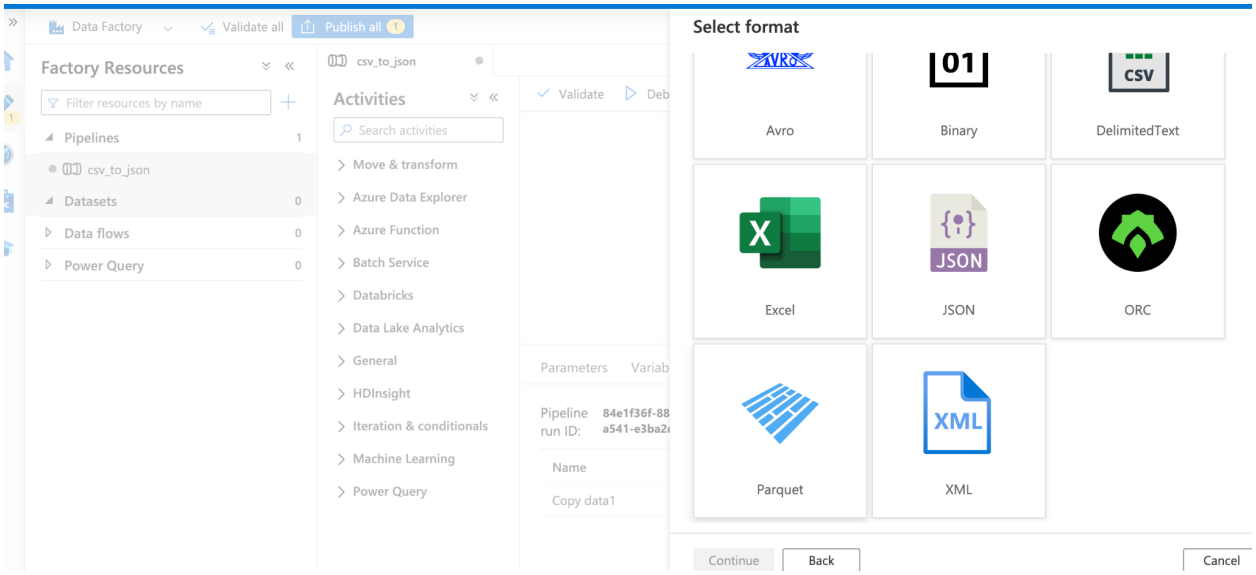
How to create storage account:
https://www.linkedin.com/posts/amulya1003_create-an-azure-storage-account-activity-6998852280389165057-BppX?utm_source=share&utm_medium=member_desktop

How to create linked service to connect to storage account.
https://www.linkedin.com/posts/amulya1003_copy-activity-in-azure-data-factory-activity-6999064498095476736-HivU?utm_source=share&utm_medium=member_desktop

As csv file is in blob storage account, we need to select blob storage.

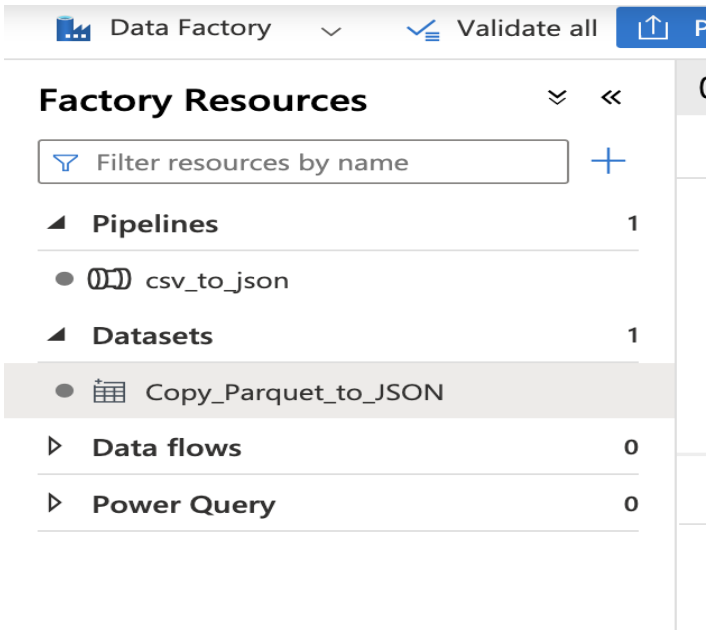


Select Parquet Datastore:

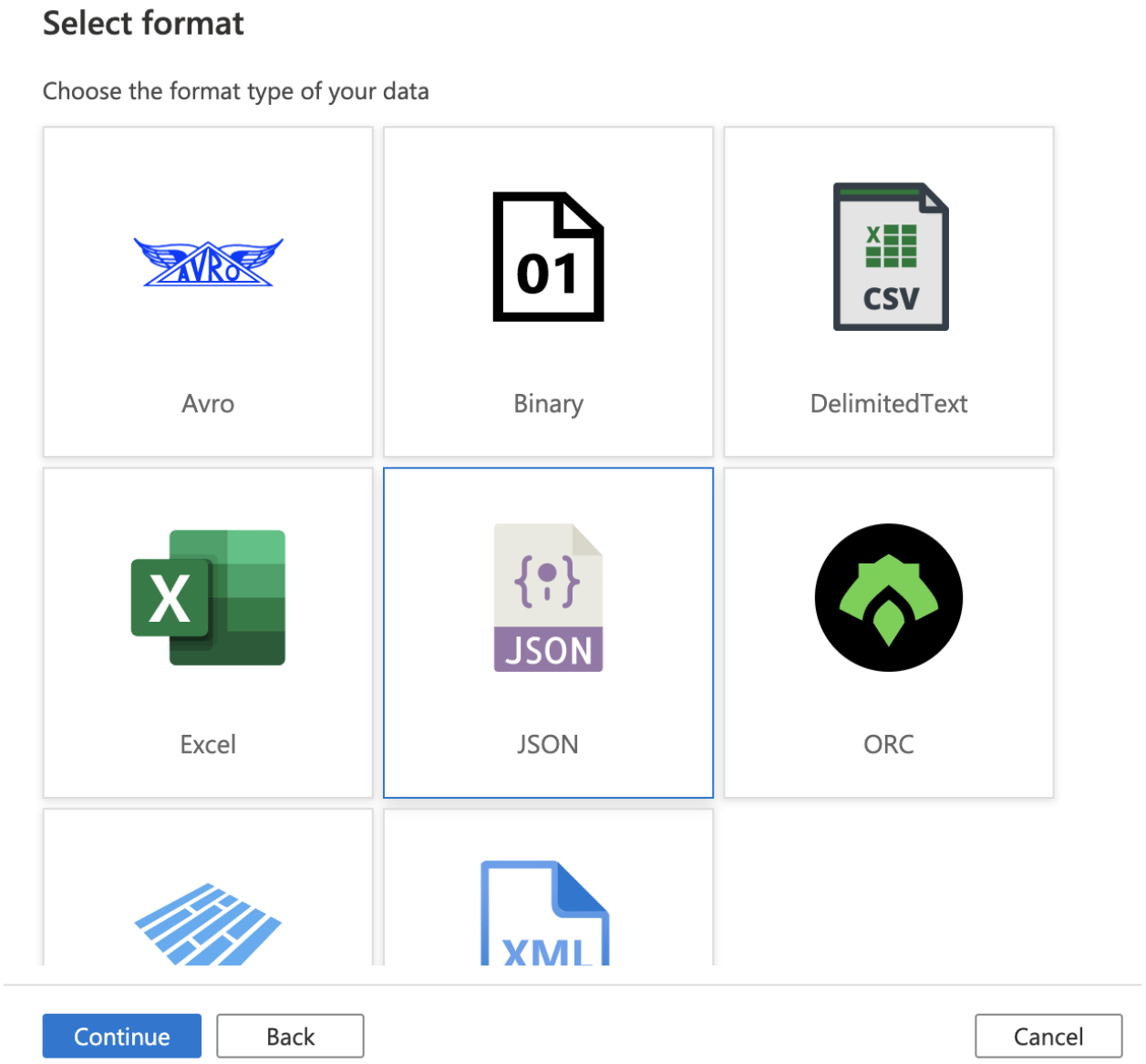


Connect to linked Service and select the root folder and click “OK” to create a dataset.

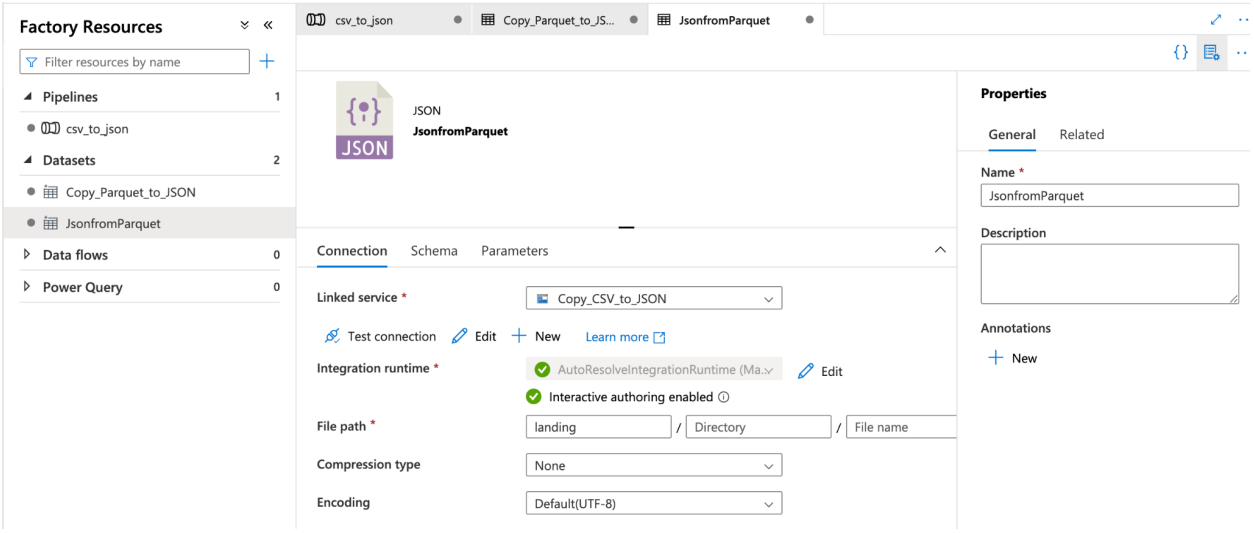
“Copy_Parquet_to_JSON” dataset created as shown below.



Create an other dataset with datasource as JSON and select destination path to land the file.



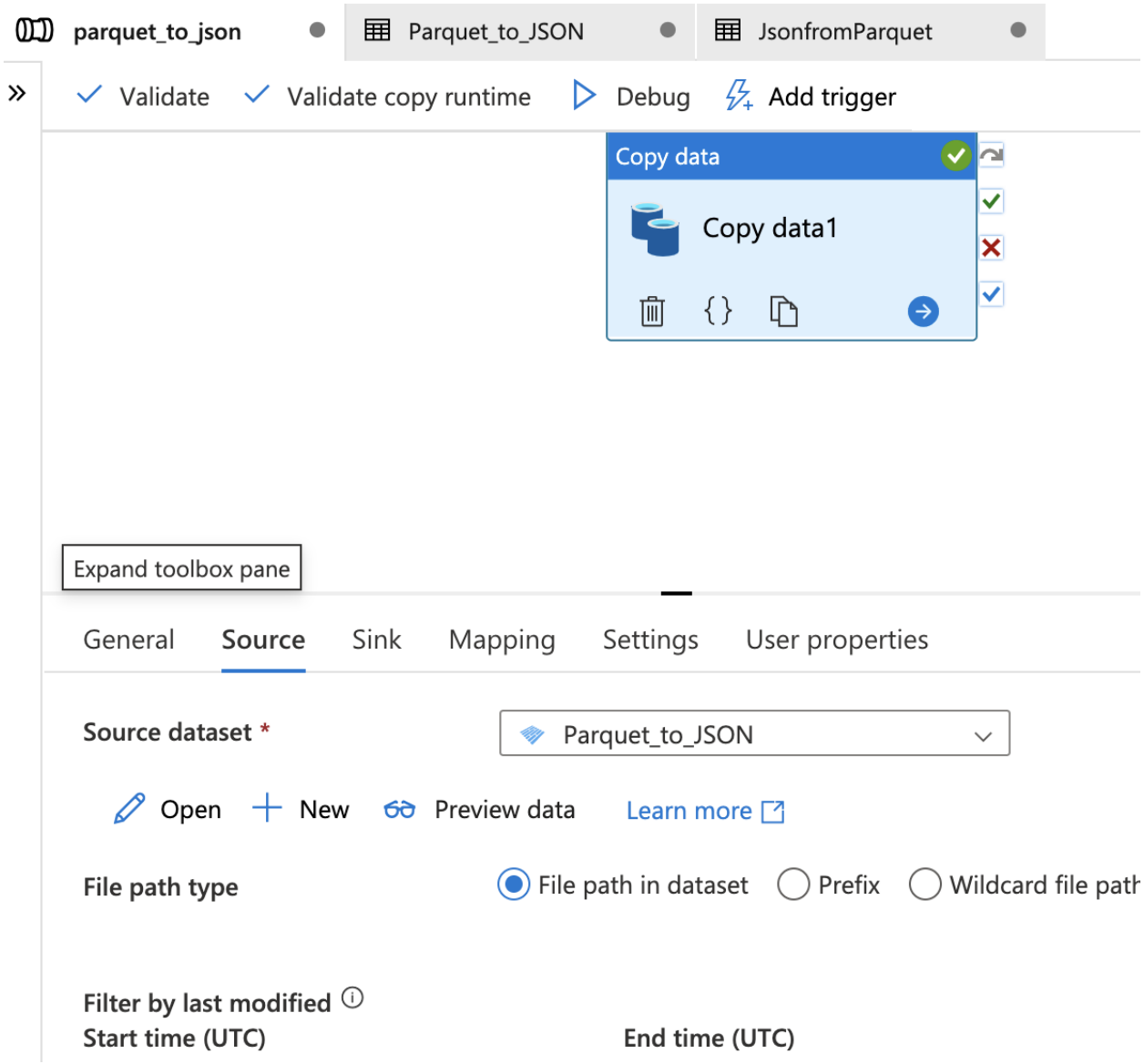
“JsonFromParquet” dataset created as shown below



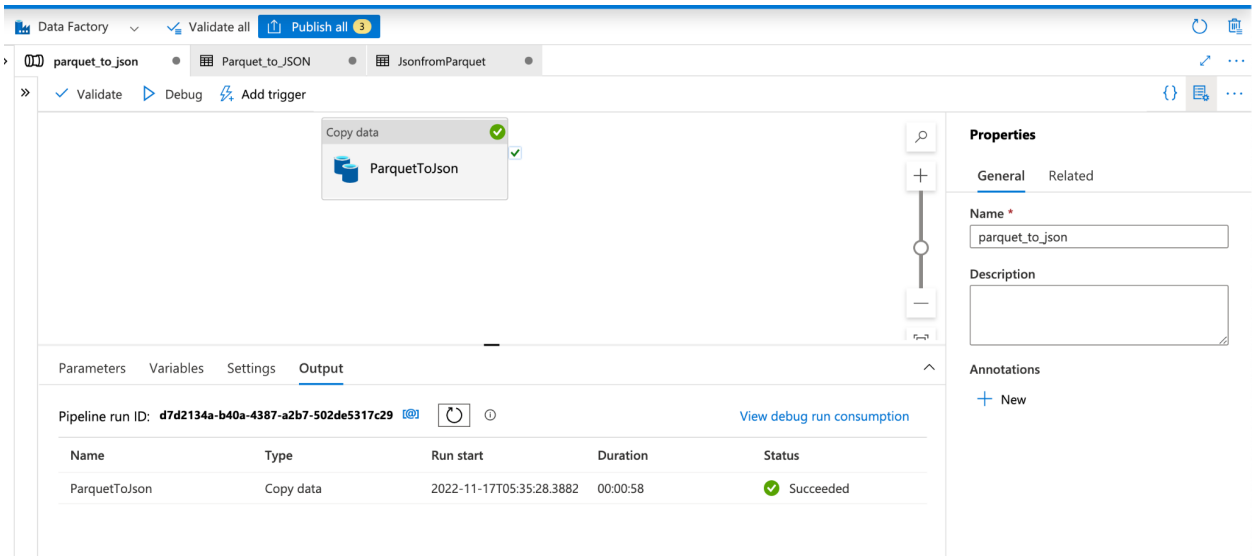
Two datasets Successfully created .Now we need to create a pipeline to pick parquet file and change parquet to json and move it to Azure storage account.

Data Factory uses the Copy activity to move source data from **source** data location to a **sink** data store.

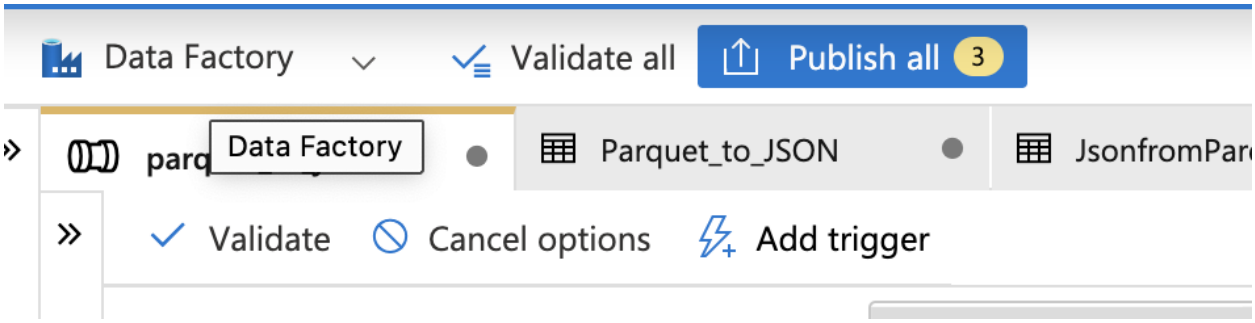
Select source and sink in toolbox pane and select source and destination dataset


parquet_to_json
Parquet_to_JSON
JsonfromParquet
Validate
Validate copy runtime
Debug
Add trigger
Copy data
Copy data1
Expand toolbox pane
General
Source
Sink
Mapping
Settings
User properties
Source dataset *
Parquet_to_JSON
Open
New
Preview data
Learn more
File path type
File path in dataset
Prefix
Wildcard file path
Filter by last modified
Start time (UTC)
End time (UTC)

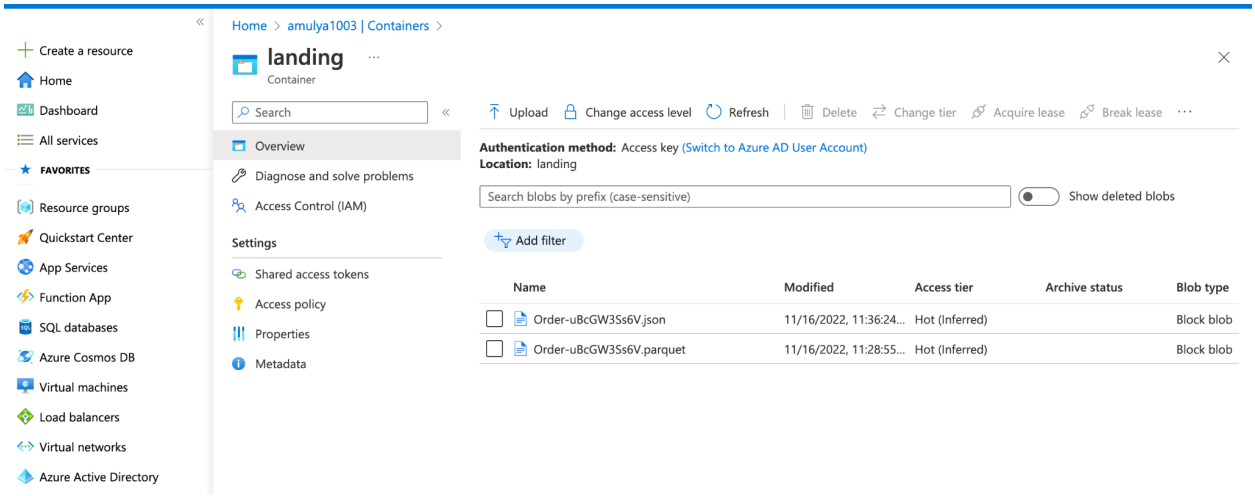
Click on Debug to run the pipeline.



At the end. Publish all to save all changes

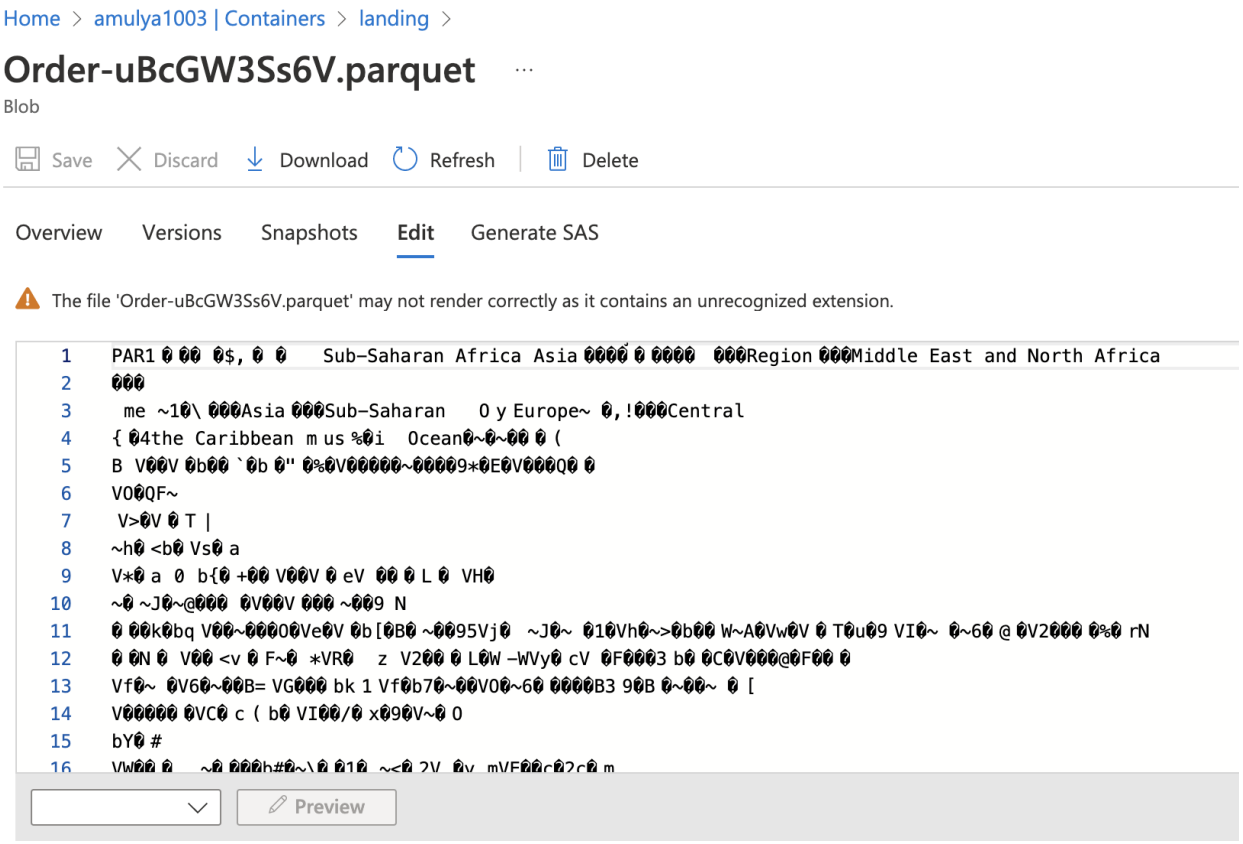


Now we can see JSON file created :

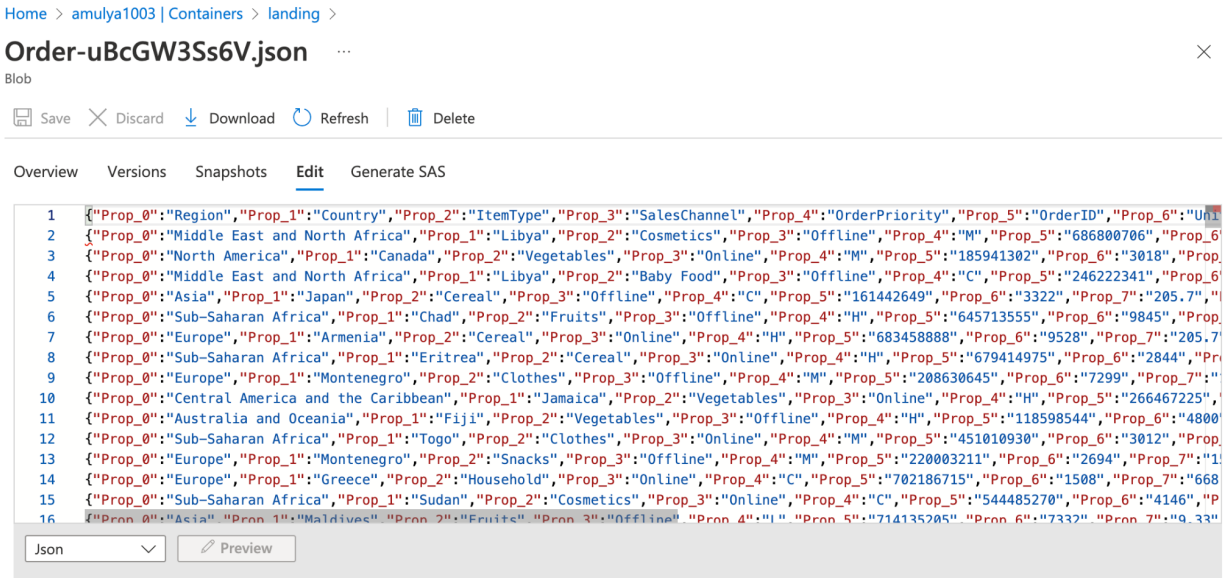


Parquet is an open source, column-oriented data file format designed for efficient data storage and retrieval.Parquet is optimized to work with complex data in bulk and features different ways for efficient data compression and encoding types. This approach is best especially for those queries that need to read certain columns from a large table. Parquet can only read the needed columns therefore greatly minimizing the IO.

Parquet is a binary-based (rather than text-based) file format optimized for computers, so **Parquet files aren't directly readable by humans**. You can't open a Parquet file in a text editor the way you might with a CSV file and see what it contains.

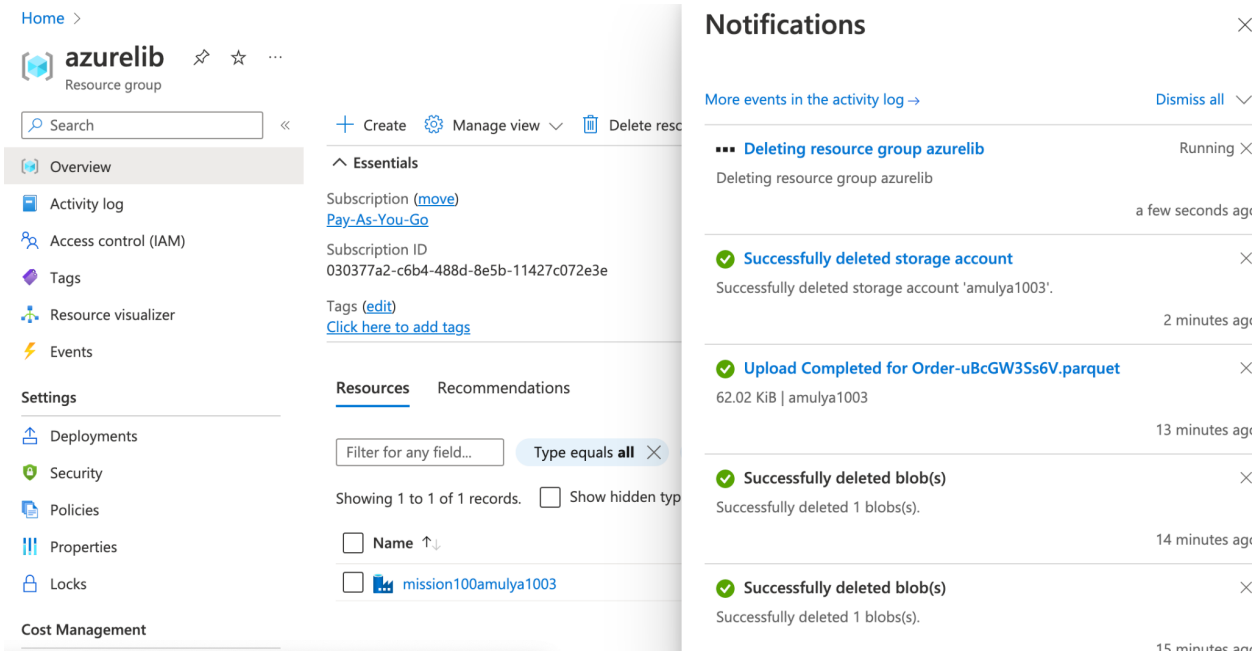


The JSON stores key-value format. In the opposite side, **Parquet file format stores column data**.

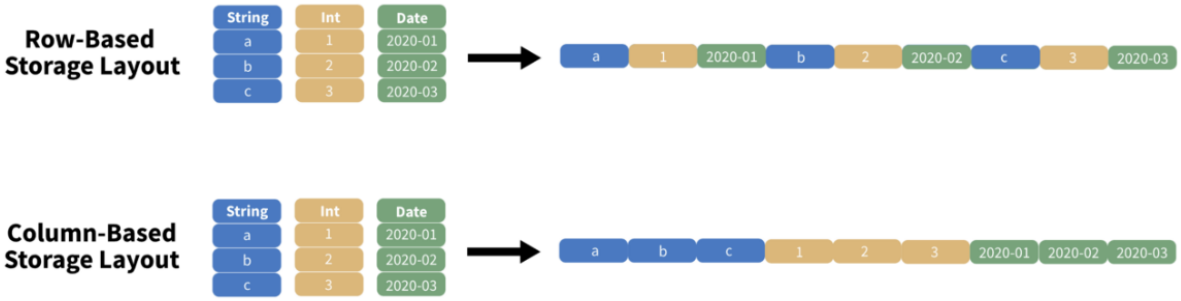


!!!!!Successfully moved PARQUET file to JSON file using COPY activity in pipeline!!!!!

To avoid cost, you can delete all the resource groups if no more needed.



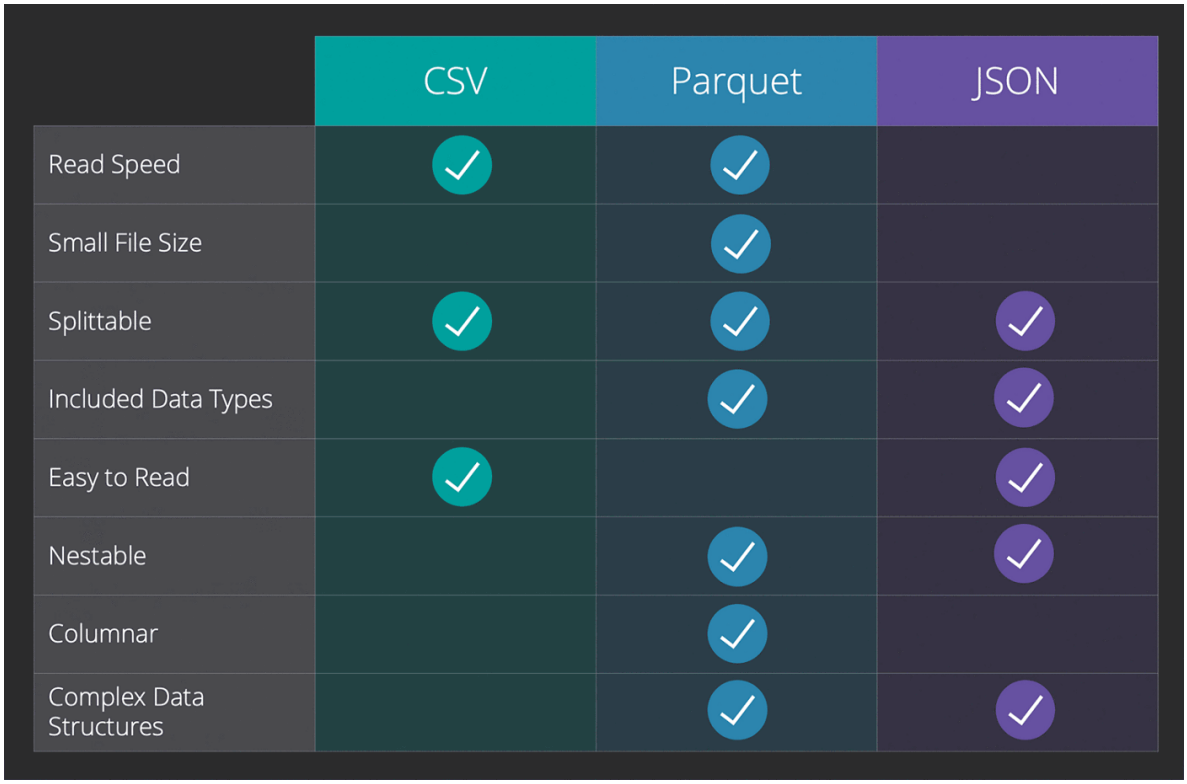
In Parquet, each column is stored independently.



Row storage	
Row 1	1
	US
	Free
Row 2	2
	UK
	Paid
Row 3	3
	ES
	Paid

Column storage	
user_id	1
	2
	3
country	US
	UK
	ES
subscription_type	Free
	Paid
	Paid

For more information on csv, json, parquet file formats:



JSON or **JavaScript Object Notation** is an open data format and is widely used by APIs (Application Programming Interface — basically how servers talk to each other) and several databases (like MongoDB).

JSON is an object notation and so it's in a sense **row-based**, not column-based. This makes it a little slower to work with.

For more updates . Follow me on LINKEDIN: LinkedIn.com/in/amulya1003