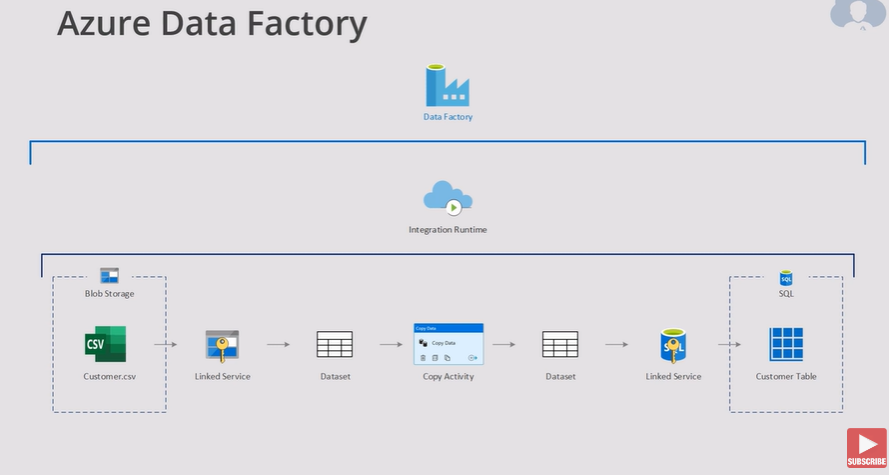
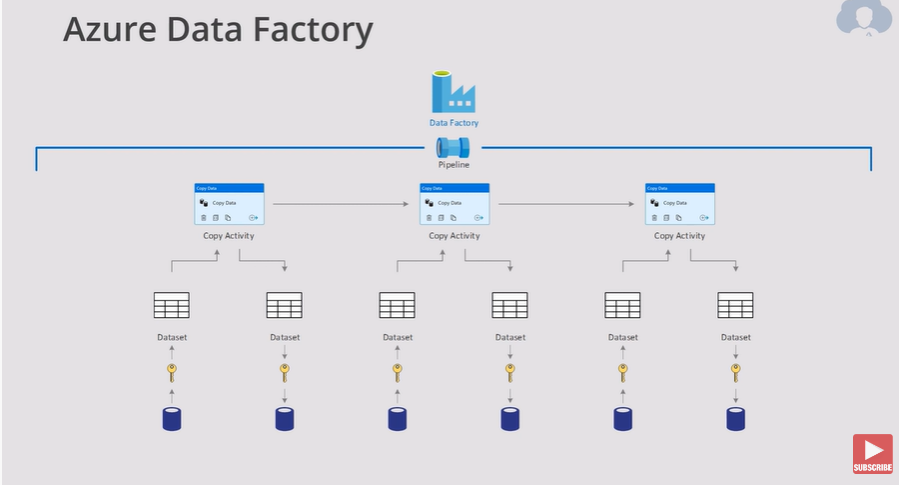
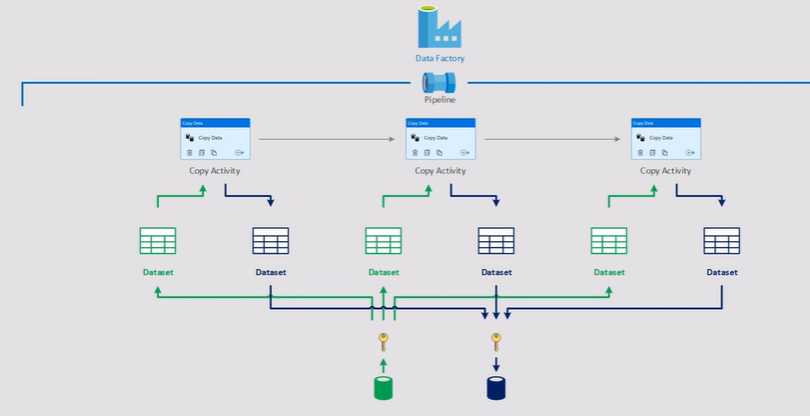
<https://www.youtube.com/watch?v=EpDkxTHAhOs&ab_channel=AdamMarczak-AzureforEveryone>

AZURE DATA FACTORY

Integrate multiple sources of data into the cloud db.



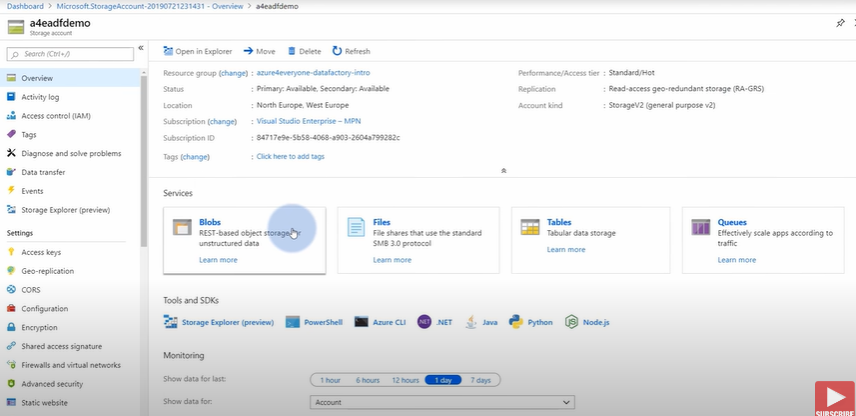




You need, create the azure data factory account

Create a SQL Database (for example)

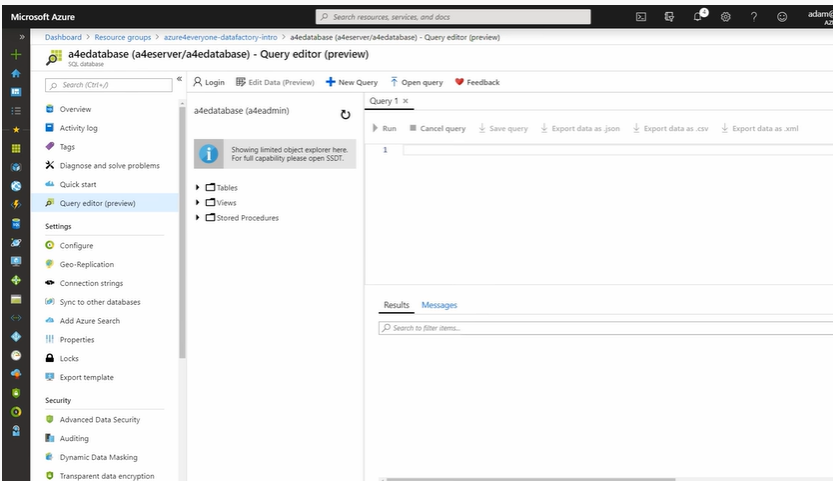
Create an storage account (blob, files, tables or queues)



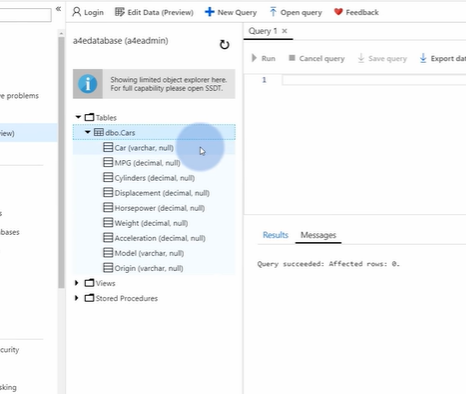
Create a container (Input)

Upload the file in the Input container. (cars.sv)

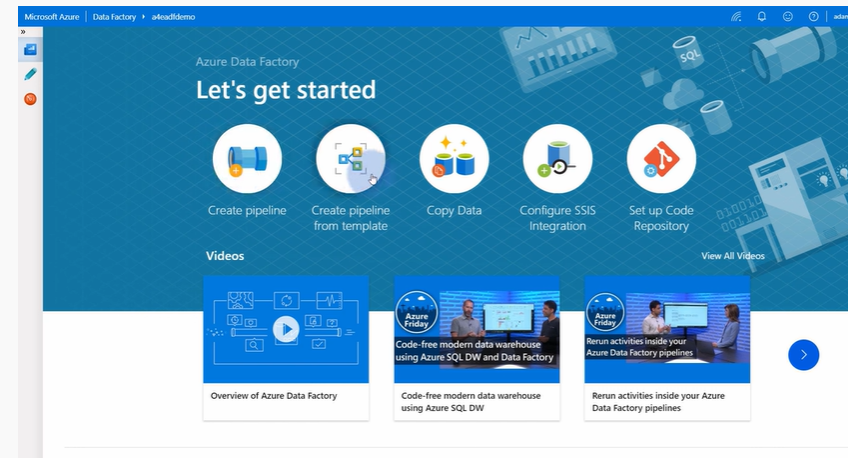
Then prepare SQL for that date. Go to the SQL dtabase (query editor)



Then create the sql to create table to store the cars file

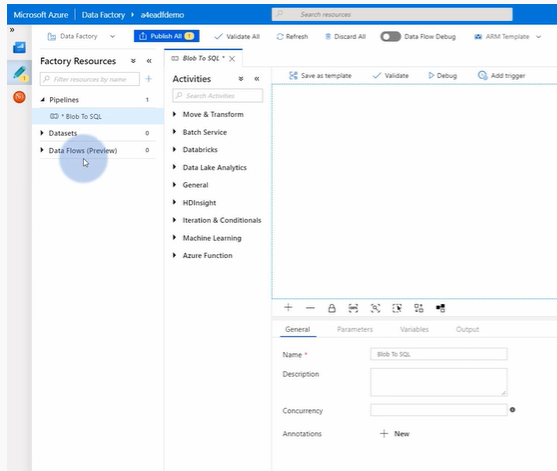


Go to Data Factory



Go to Author tab.

You need a pipeline (everything is contained into the pipeline)



We need to connect to the blob storage (where the cars file is)

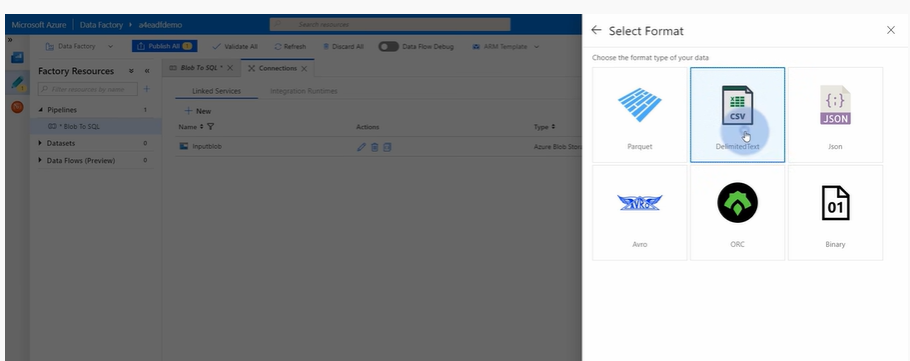
Use Link Service – new – blob storage, continue

Select security

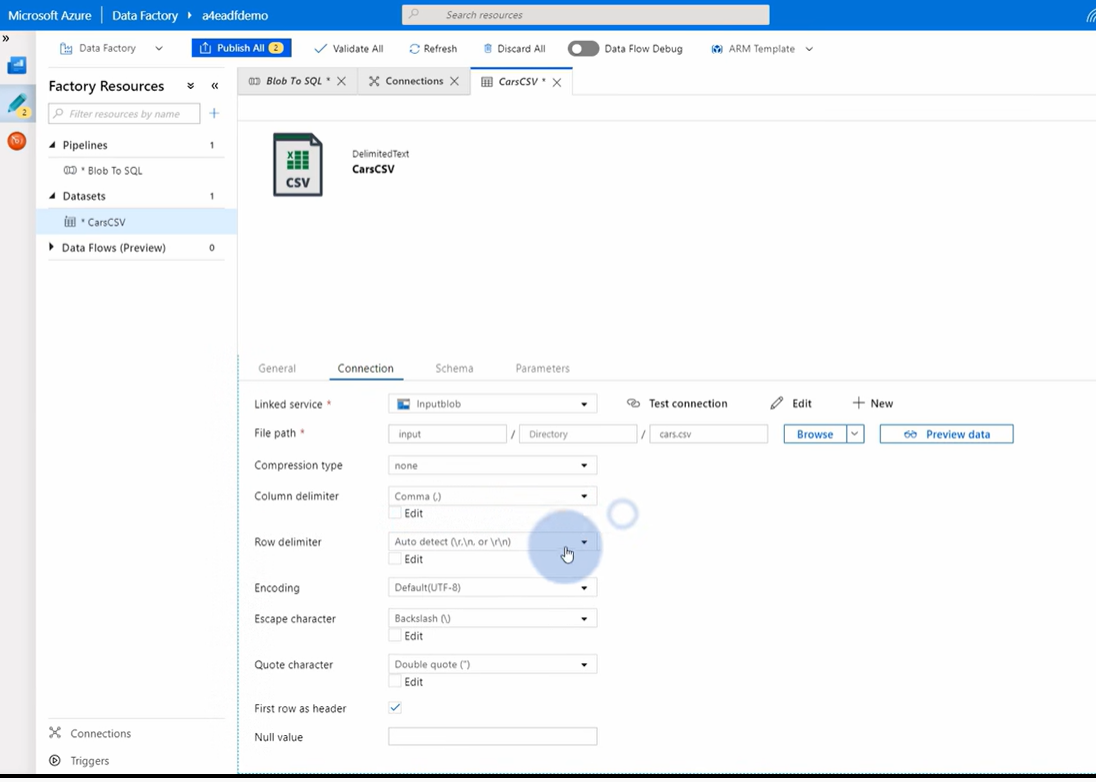
Test the connection

Then we need to know how the data looks from the blob starege.

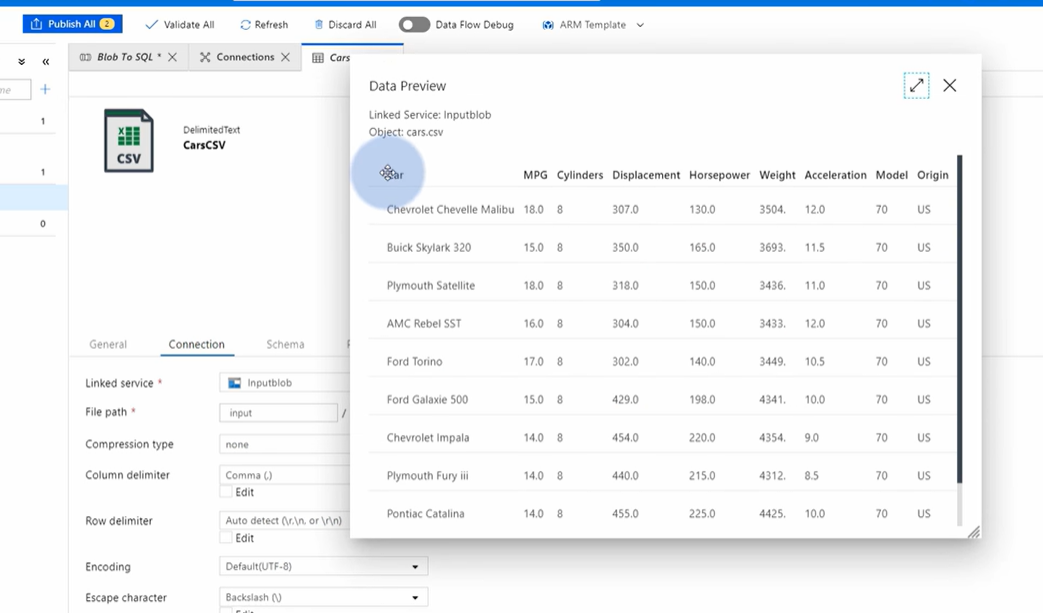
Select the format



Is coming for inputBlob.

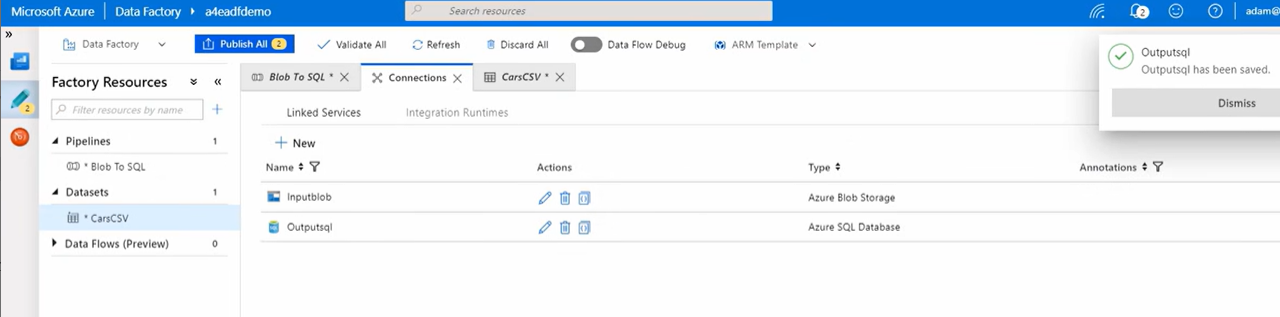


You can change some things about the connection. Check data preview



Then, create another link service to the SQL Database (destiny)

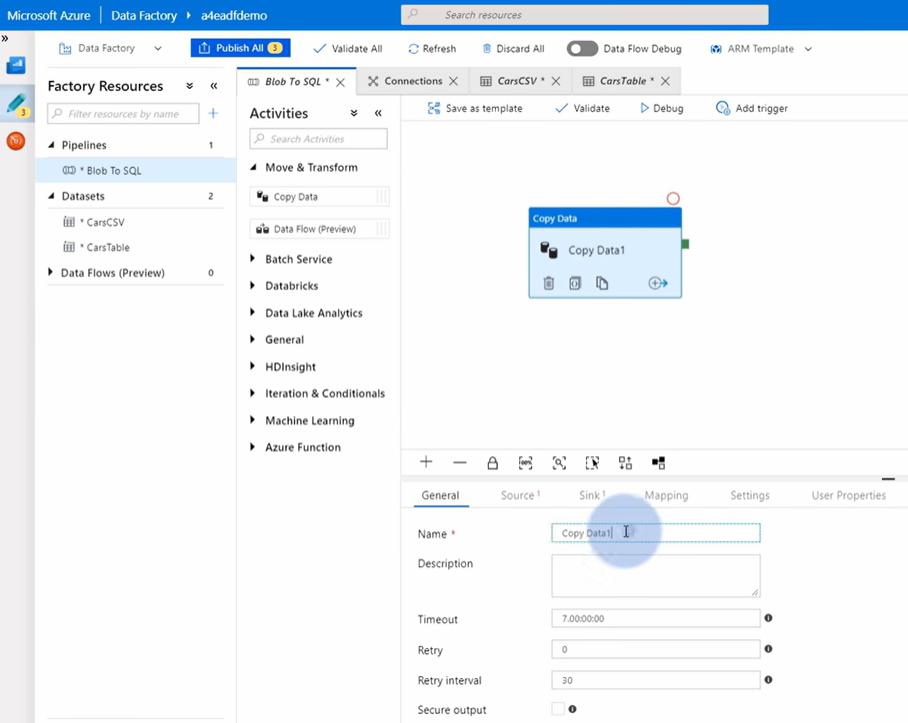




Now you need another dataset. Remember, one dataset per source or destiny.

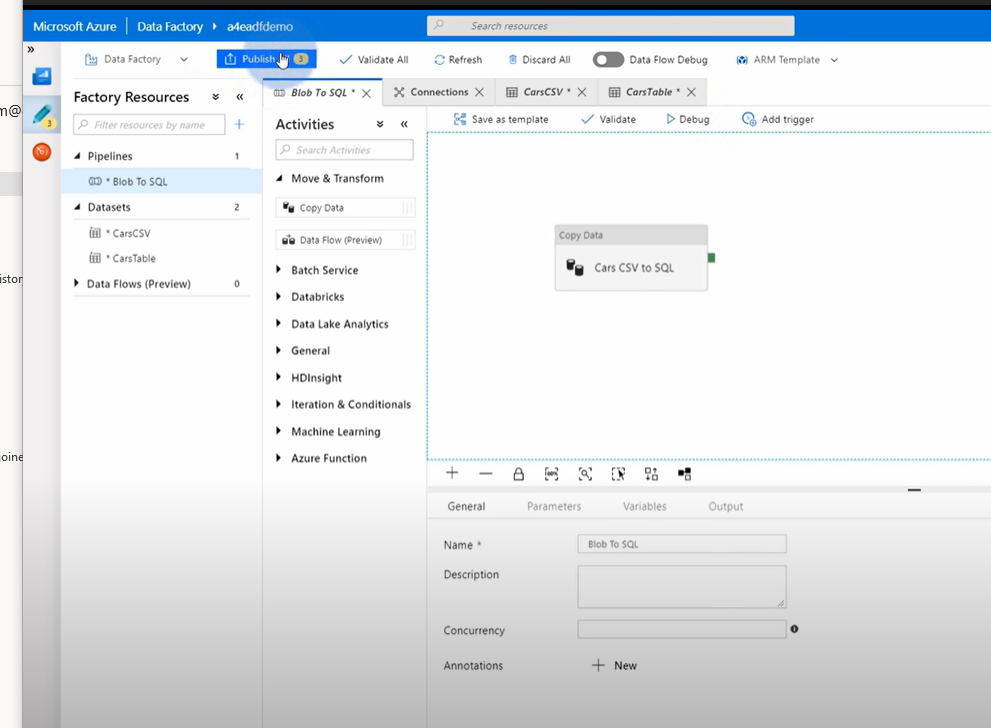
Dataset are the memory locations for the data while is transferred.

Finally you need the activity that will create the copy task of the data.



Select source, sinks, mappings, settings.

Use debug to see if everything is working fine.



Then you can publish.

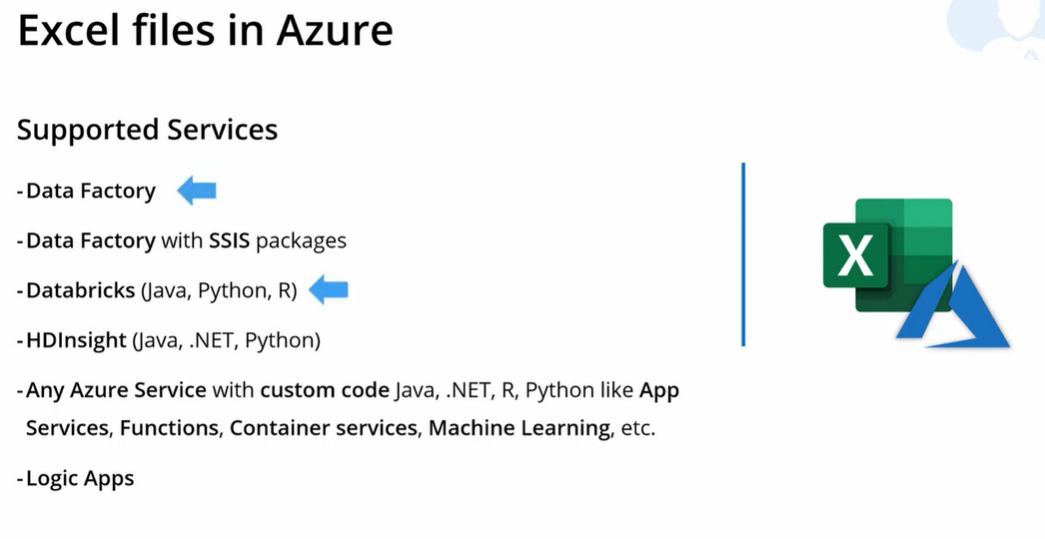
You can then add triggers

Azure Databricks

<https://www.youtube.com/watch?v=pc8Kv-lRD8k&ab_channel=AdamMarczak-AzureforEveryone>

Excel files are standard, easy to use and very versatile, however make development with these files present some challenges. It is better to parse these files in a different format.

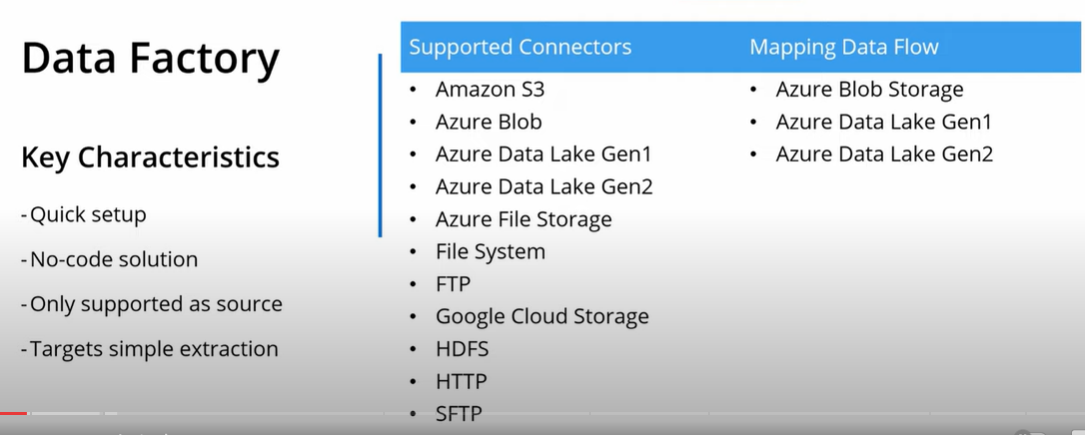
You can use them in Azure:



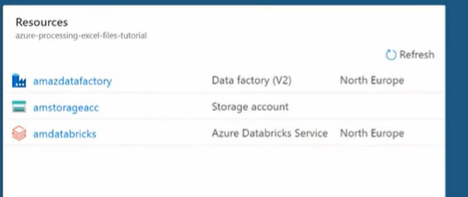
\*avoid Logic Apps to work with Excel Files, not well implemented in its core yet.

Using Data Factory you have no code solution, quick setup.

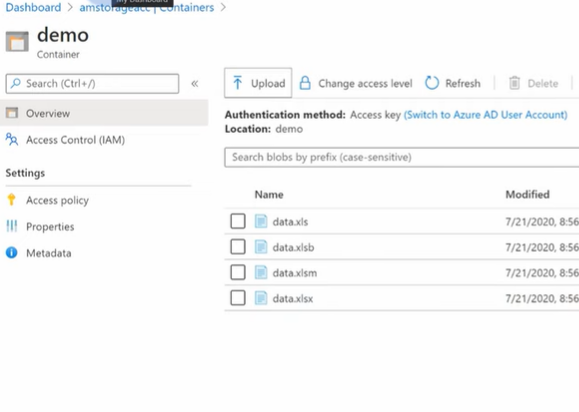
Data Factory may not be the best extraction tool if the excel file is complex.



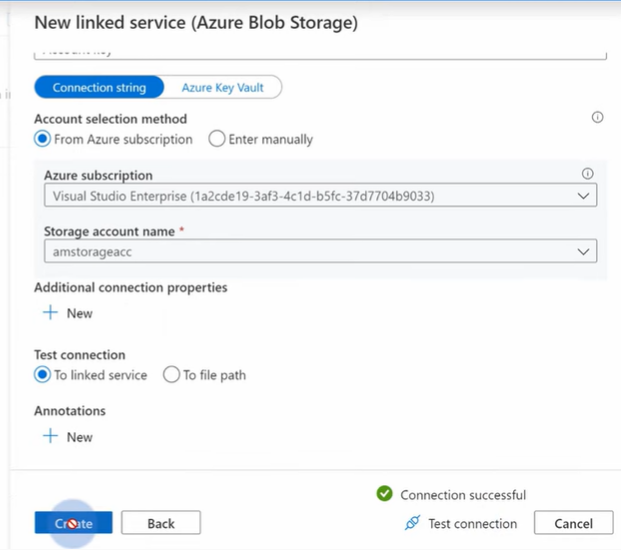
You need:



Inside storage Account you have a container. Upload all files to that container.



Move to Azure Data Factory -> manage and create new link servers to the storage.

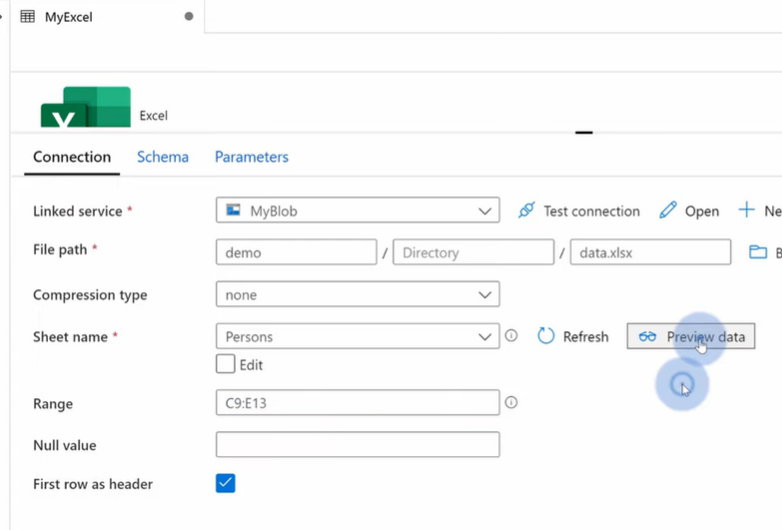


Create new dataset:

Select blob storage

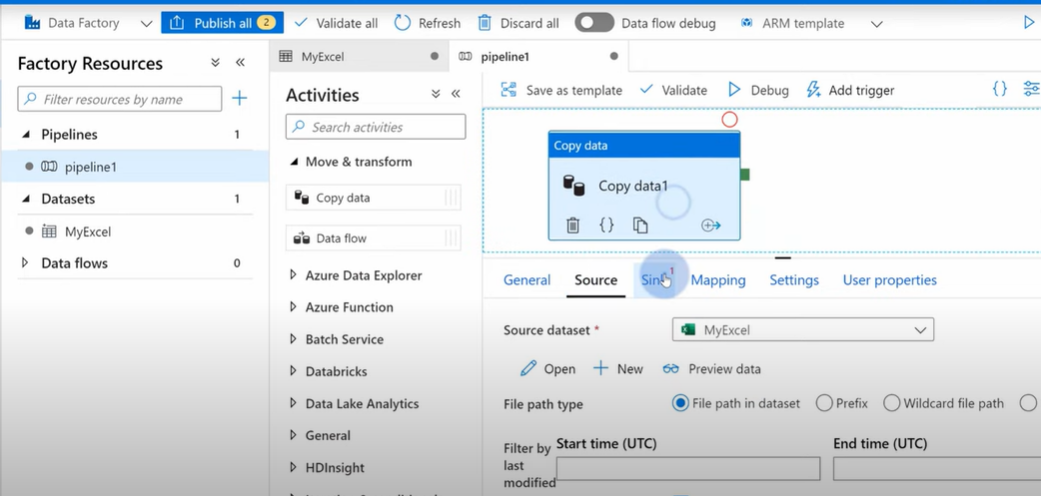
Pick Excel format.

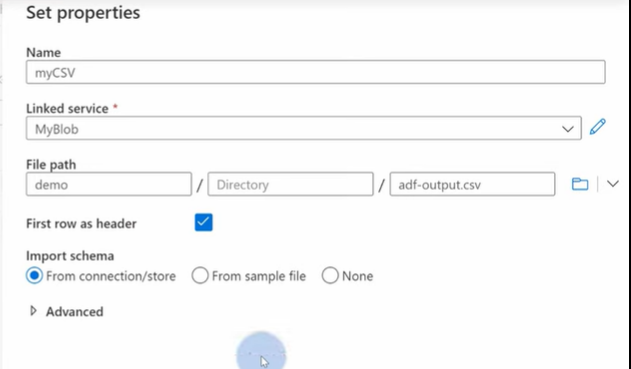
When you have data with empty cells, you can select a range in the cells:



Select the output when you are happy with the preview results.

Create a pipeline to copy the data

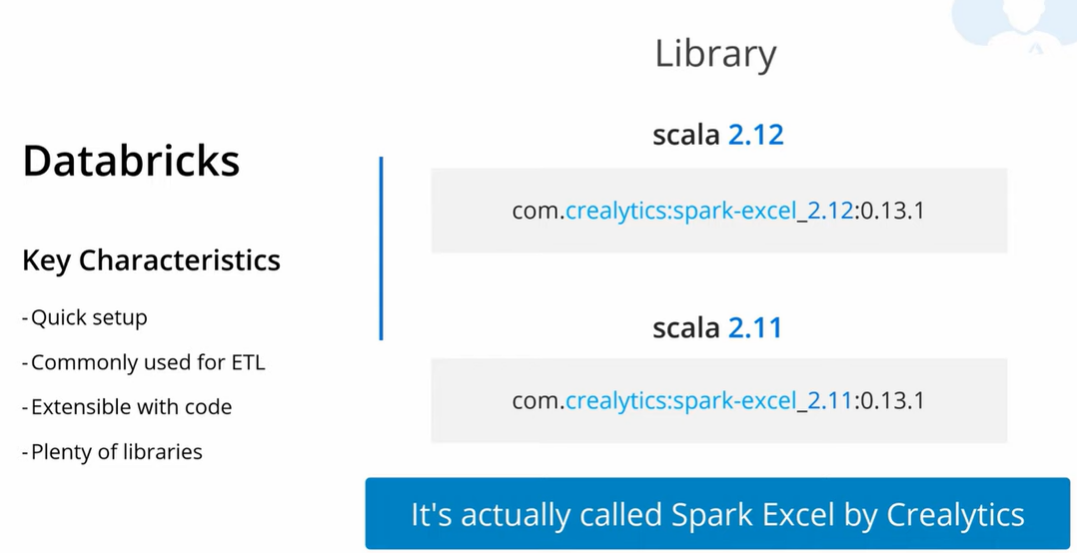




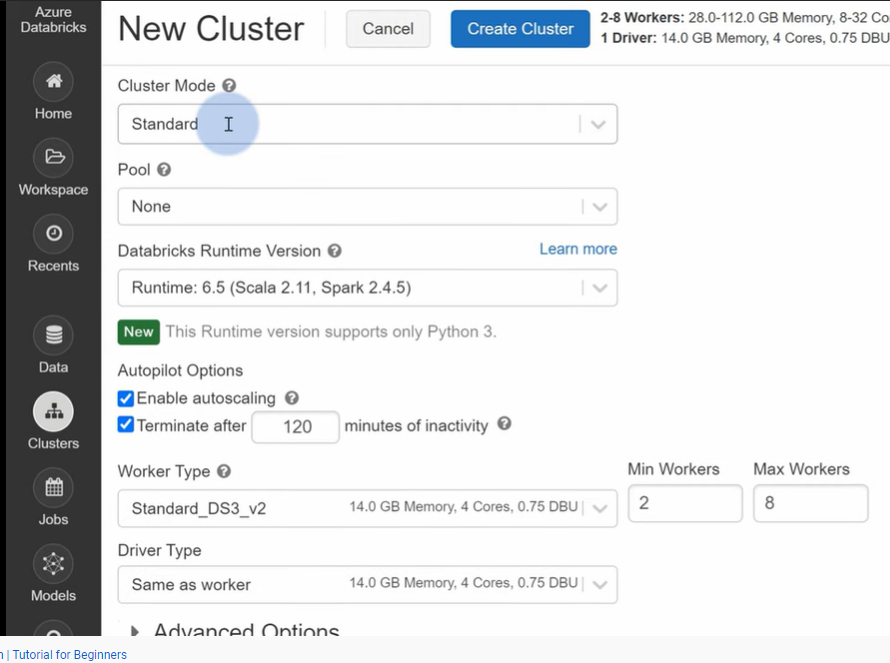
Csv file in this case to store it.

**That’s easy, now let’s use Databricks. With Databricks you can use third party libraries to handle more complicated scenarios.**

**Its commonly use for ETL and extend with code.**



Create/Launch Workspace

You need to create a new cluster. 

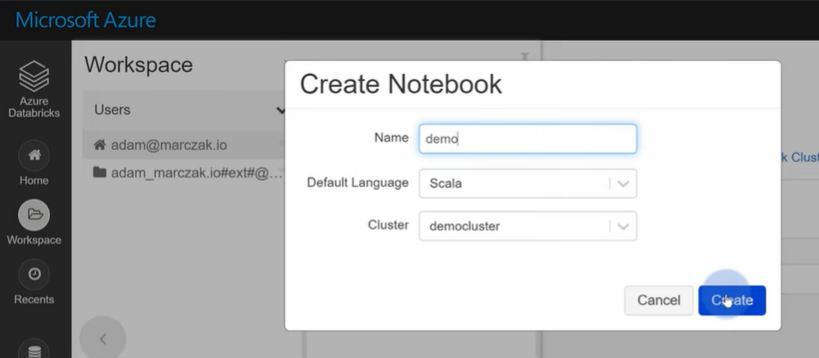
Disable autoscaling for cost.

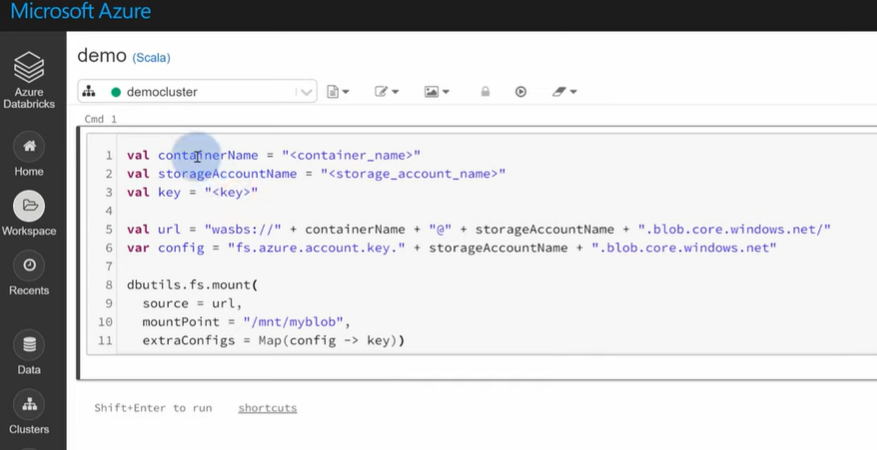
Change workers to 1 also to save money. Choose the least expensive VM.

Install new library inside the Cluster after creation, Libraries tab ans install Maven –



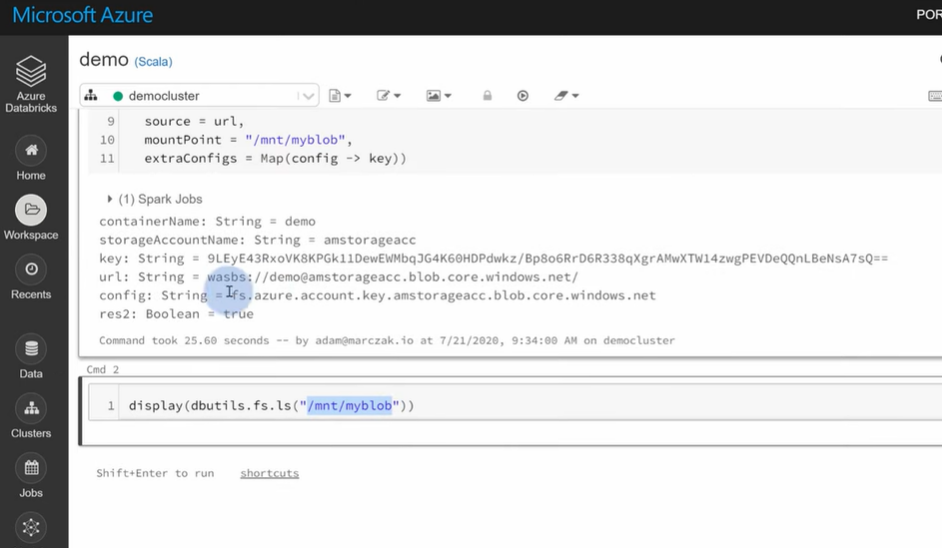
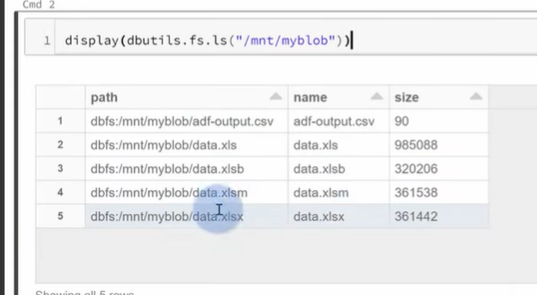
Go back to Workspace and create a new notebook.

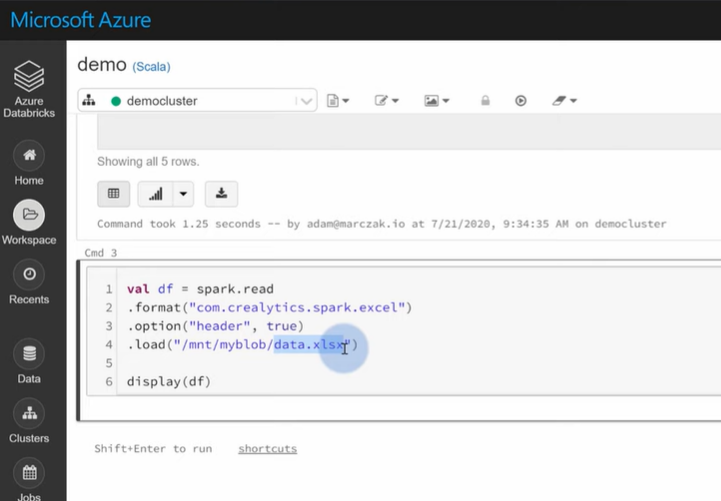


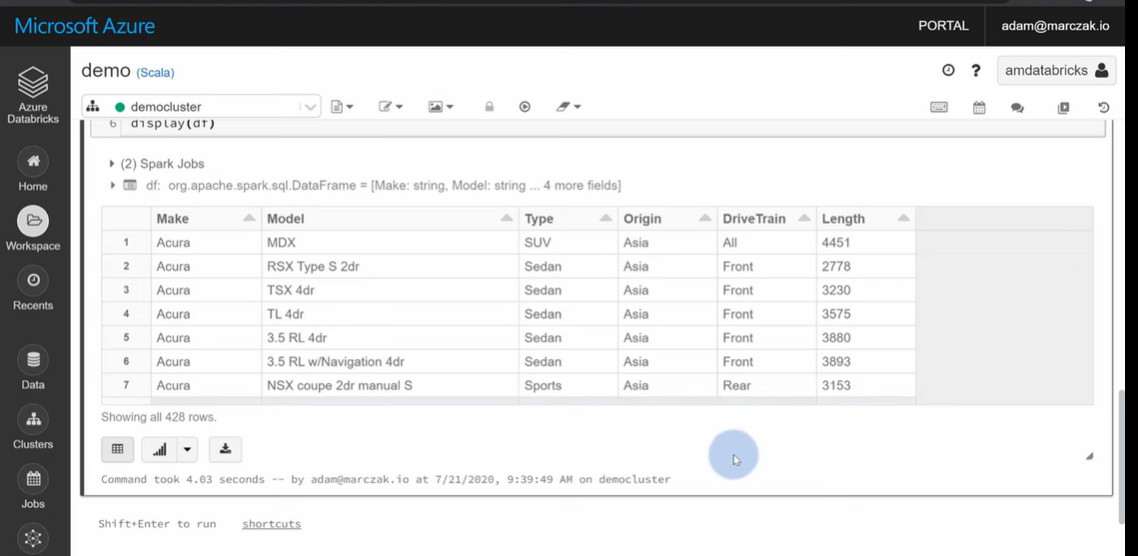


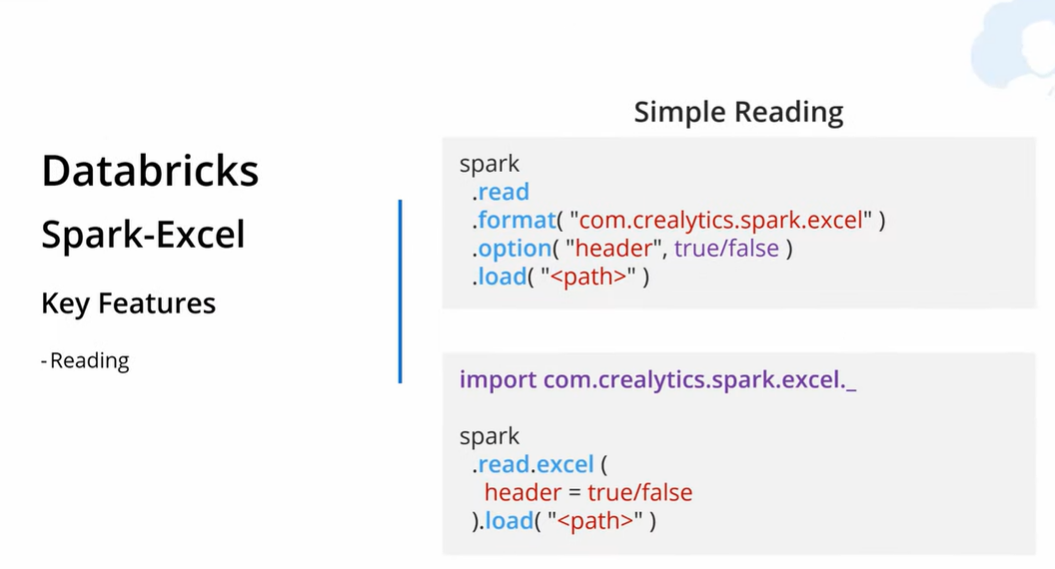
Get the container name and storage account name and key.

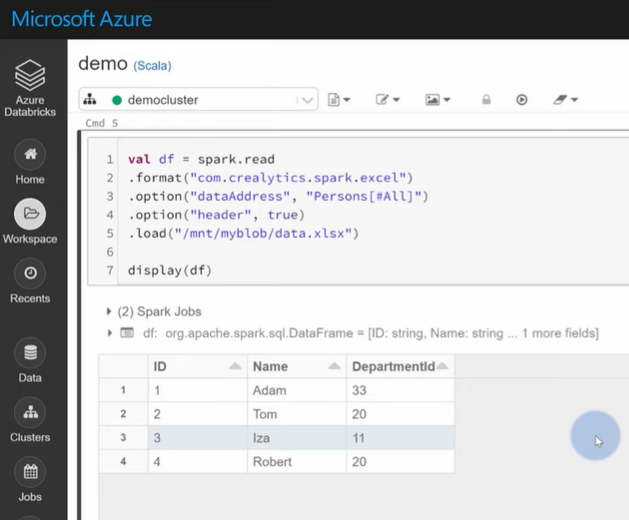
Click Run to mount the storage.

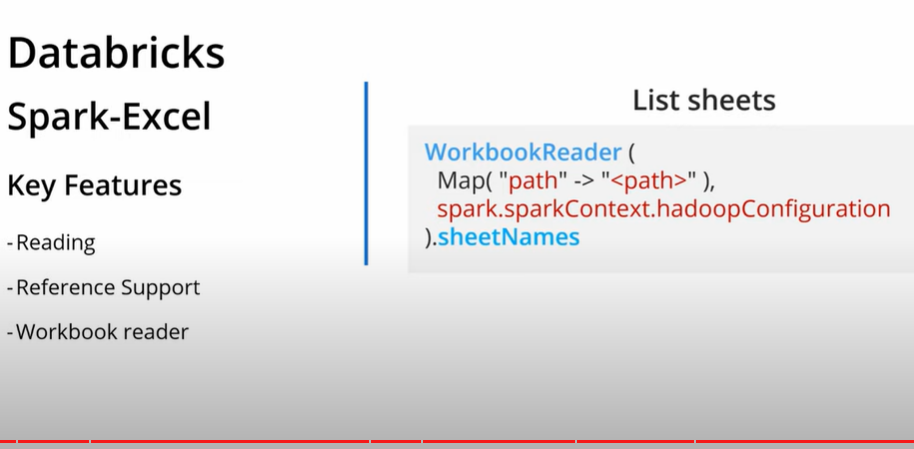
 

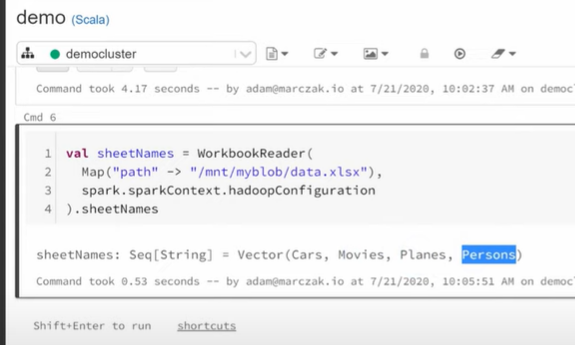




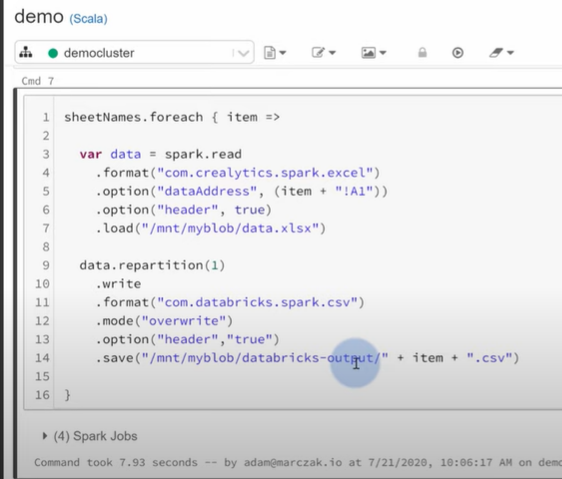


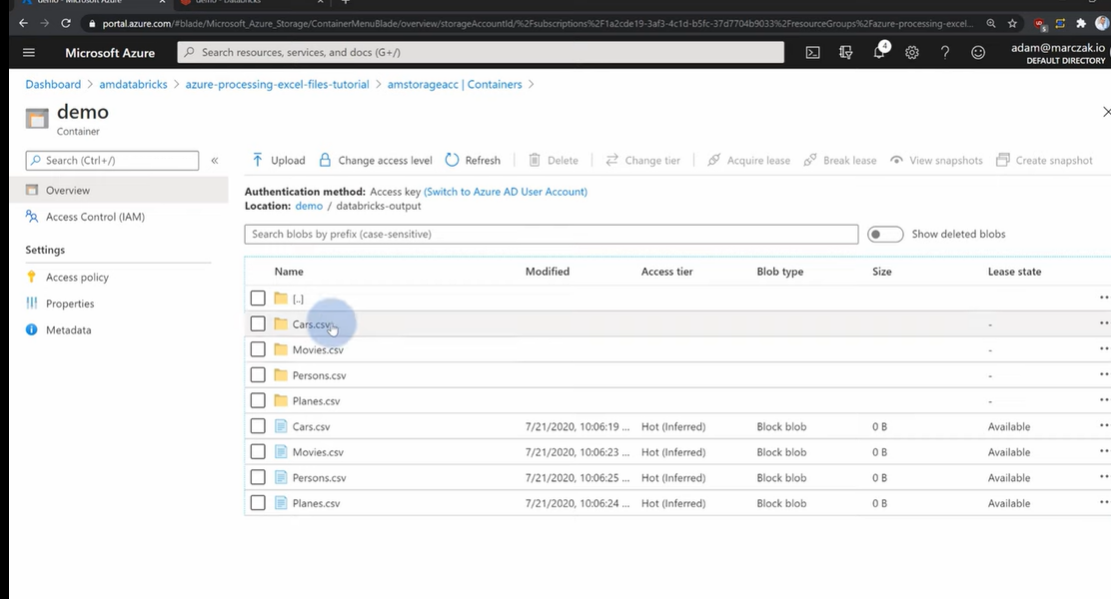




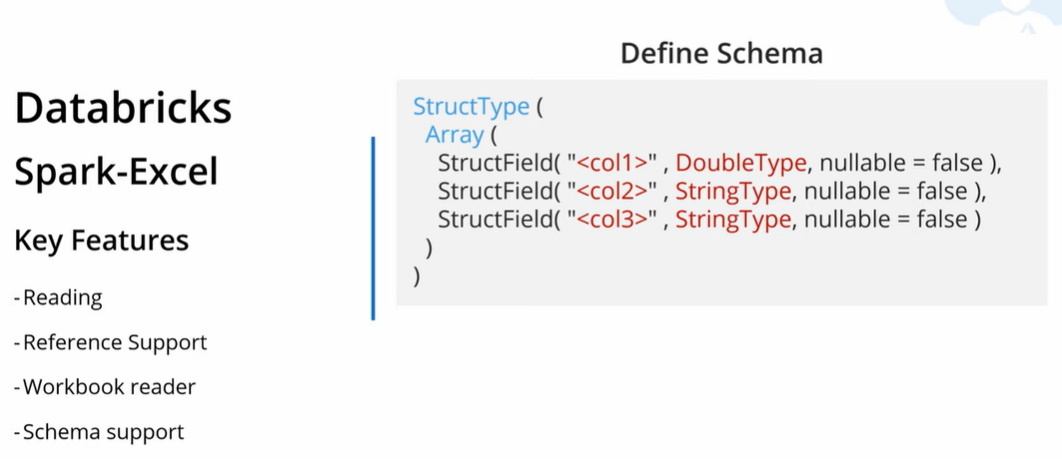
Then you can do a for each sheet:



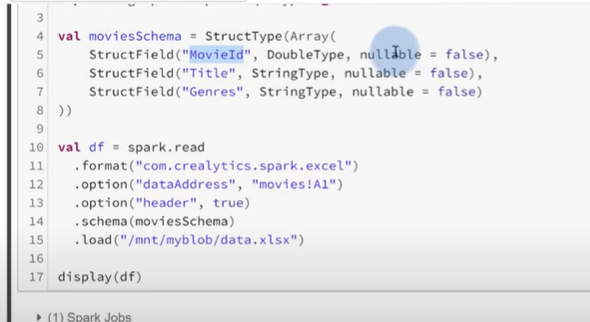
So you can convert one excel file with multiple sheets into a multiple csv files.



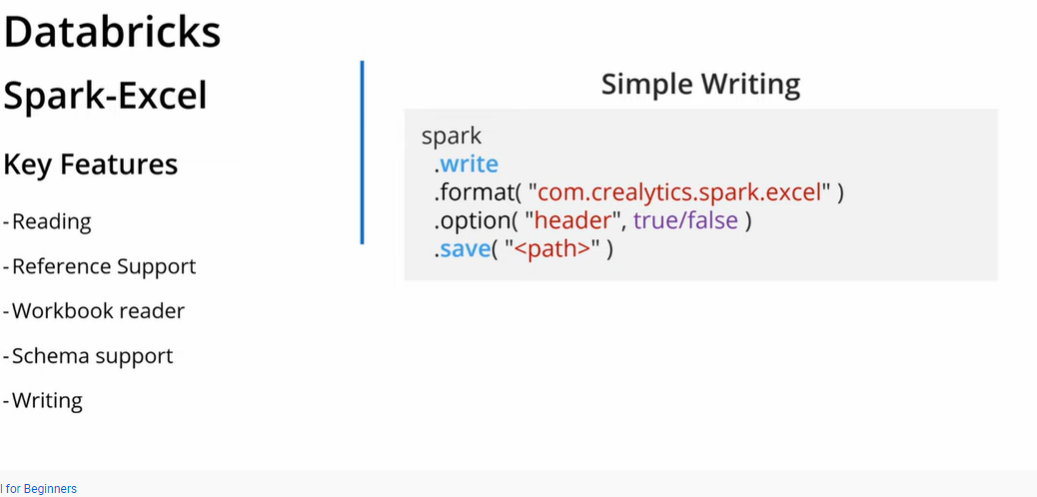
Remember thar Excell it’s a free schema file ? ok you can force a schema when importing. You need to define a schema:



This will avoid issues with the data inside the excel



Crealytics



Do small agregattion:

Read the planes data 

Then create data frame and group by Maker of th eplane:

