DXC - RealTime Projects

Name: Lakshmi Naidu Thatikonda

Reg No: dxcab1226

Project1 Name: Smart Vehicles

Date: 10/06/2022
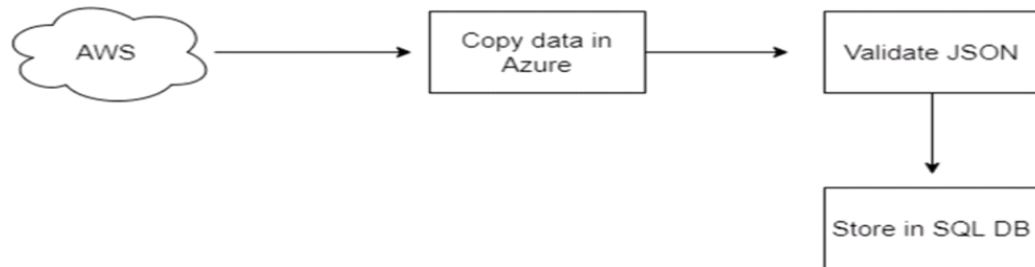
Project 1 : Connected Vehicles

- General Motors is one of the leading heavy vehicle manufacture company. To improve their service they are planning to rollout lot new features based on IoT.

Vehicle   Sends Data in real time   Azure Cloud

Project 1 : Connected Vehicles

- Vehicle has third party IoT device which will send the telemetry data (in JSON format) over the AWS cloud.
- You need to move data from third party AWS to General Motors Azure cloud.
- You need to validate the JSON sometime it could be incomplete or wrong JSON which need to be rejected.
- Once JSON got validated this data would be stored in the SQL database which will be further utilized by data science team.
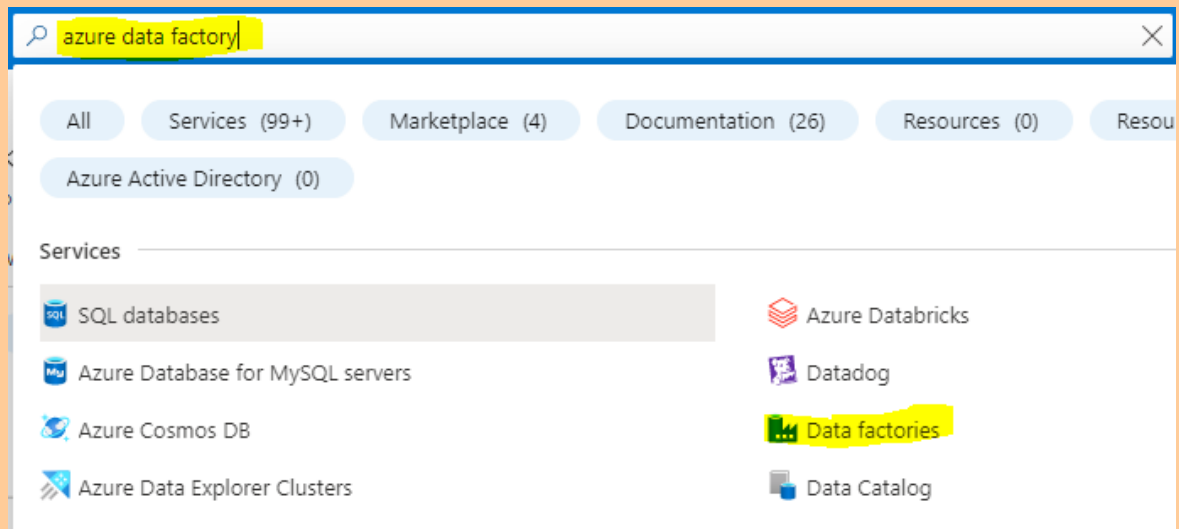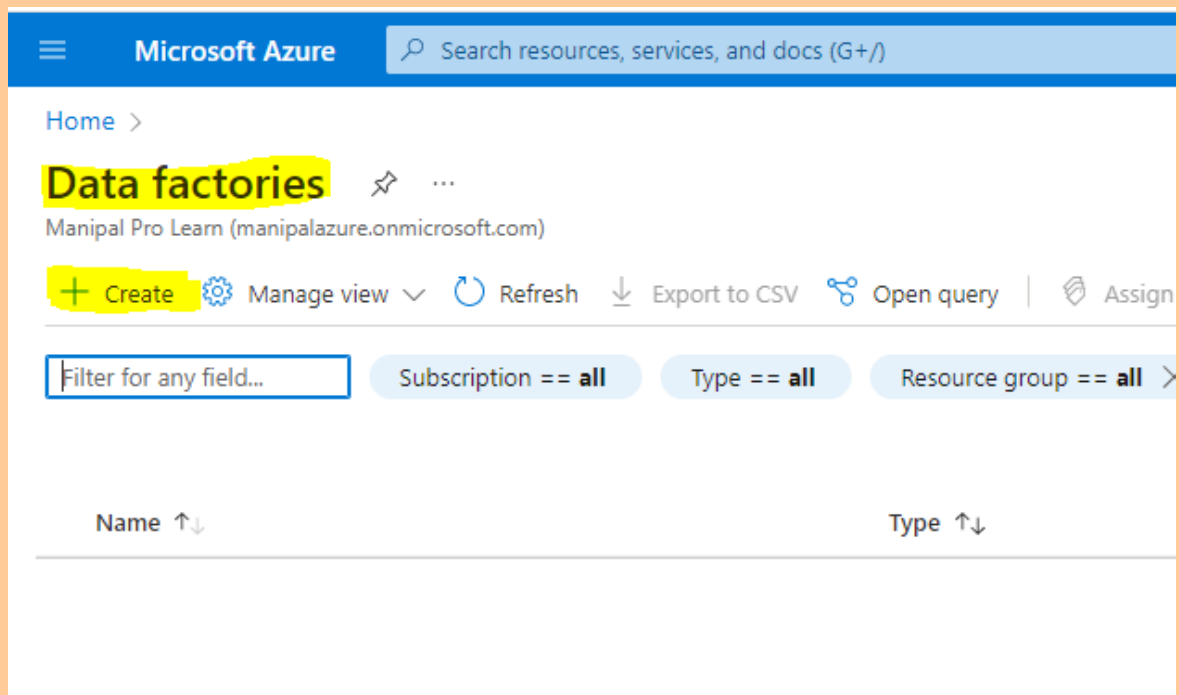
Project 1 : Connected Vehicles
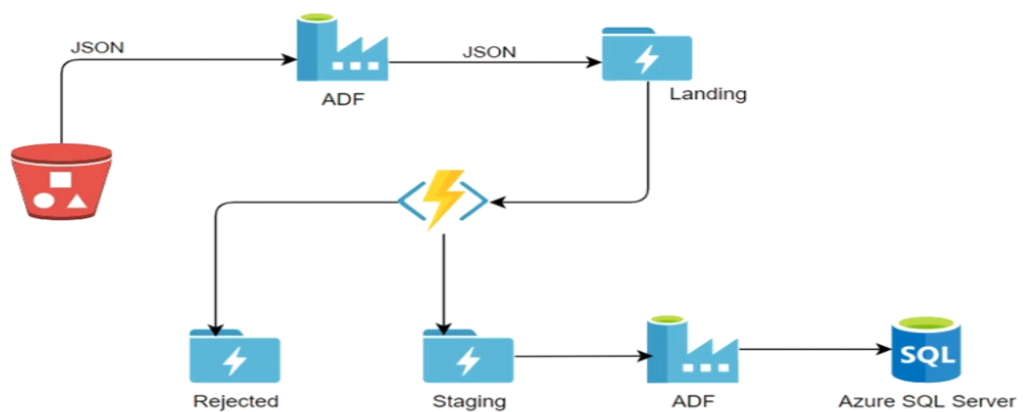
Architecture Diagram for Connected Vehicle Project

Practical Lab: Create **Azure Data Factory** Account For Data pipelines

Step 1: Go to search bar click "Data Factory" and click on it.

Step 2: Click on "Create".

Step 3: Fill the details like "Resource Group" and "Data Factory Name"

Step 4: No need to set all the highlighted portions. Click on "review and Create".

Step 5: And the Data Factory is being Deployed.

Practical Lab: Create **ADF Pipeline** End to end pipeline with triggers enabled

Step 1: Click on "Azure Data Storage" on home screen of Azure.
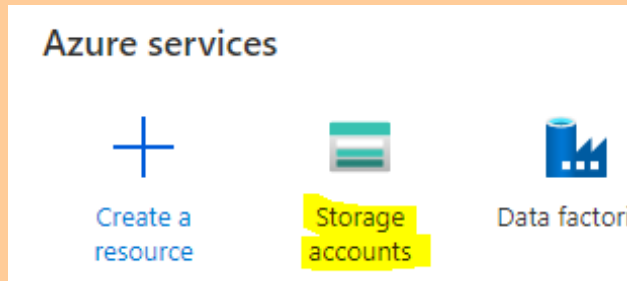
Step 2: Click on "Create".
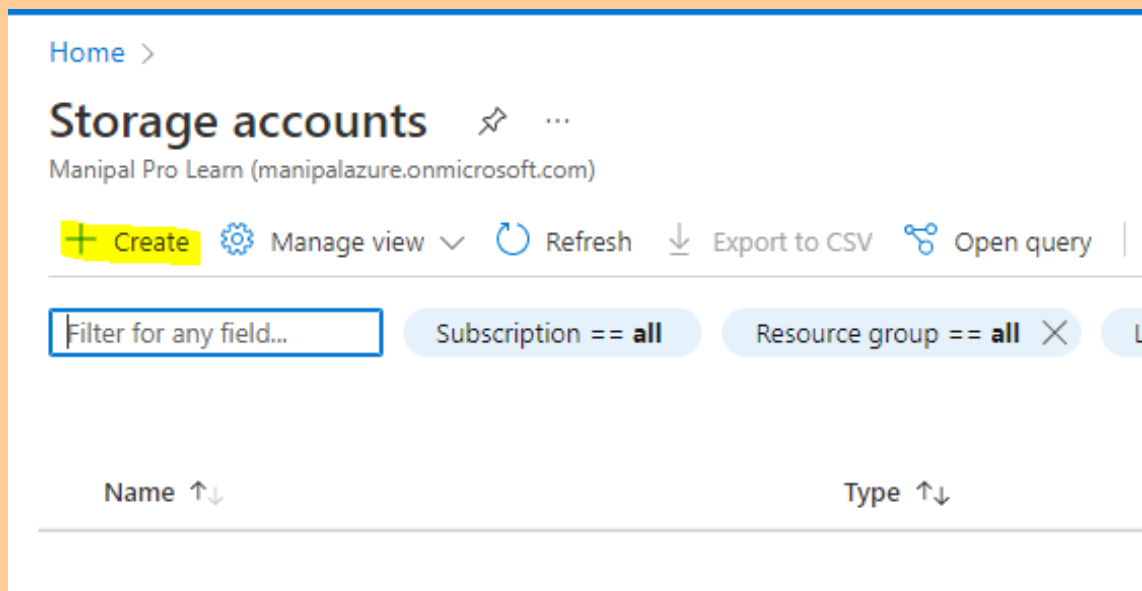
Step 3: On the next screen Give all the details and "Click: Adavanced".

Step 4: Go on till the "Review and create" tab by filling the details required and click on "Create".
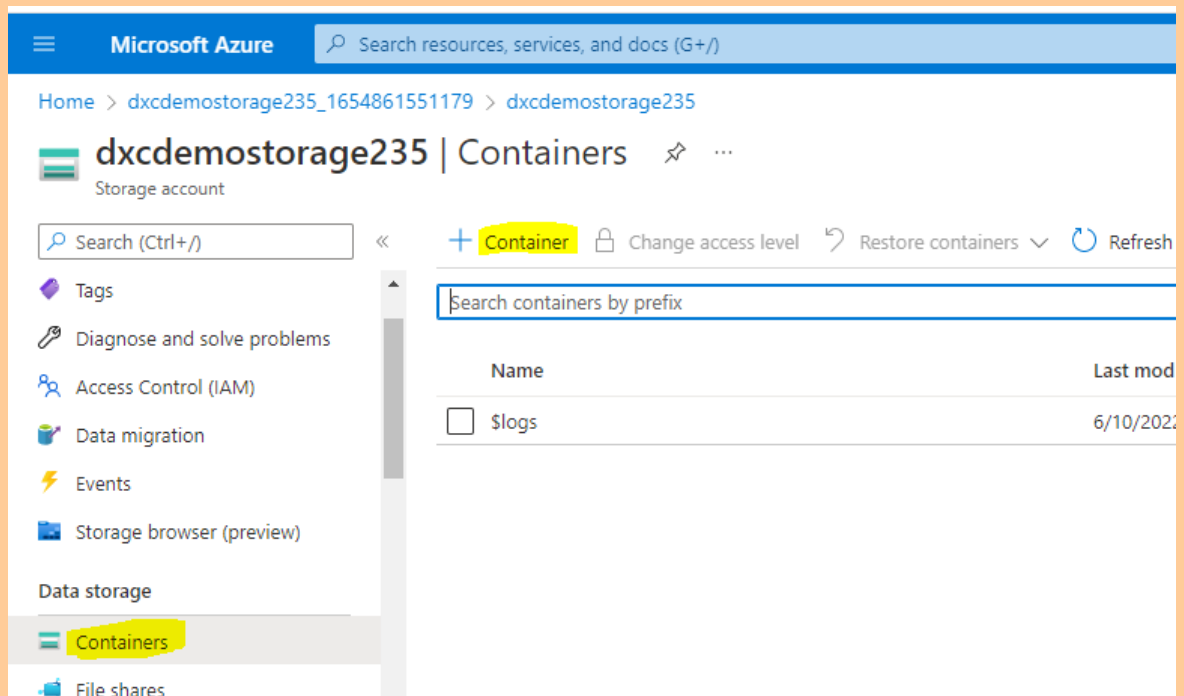
Step 5: Once Deployment is done, Click on "Go to resource".
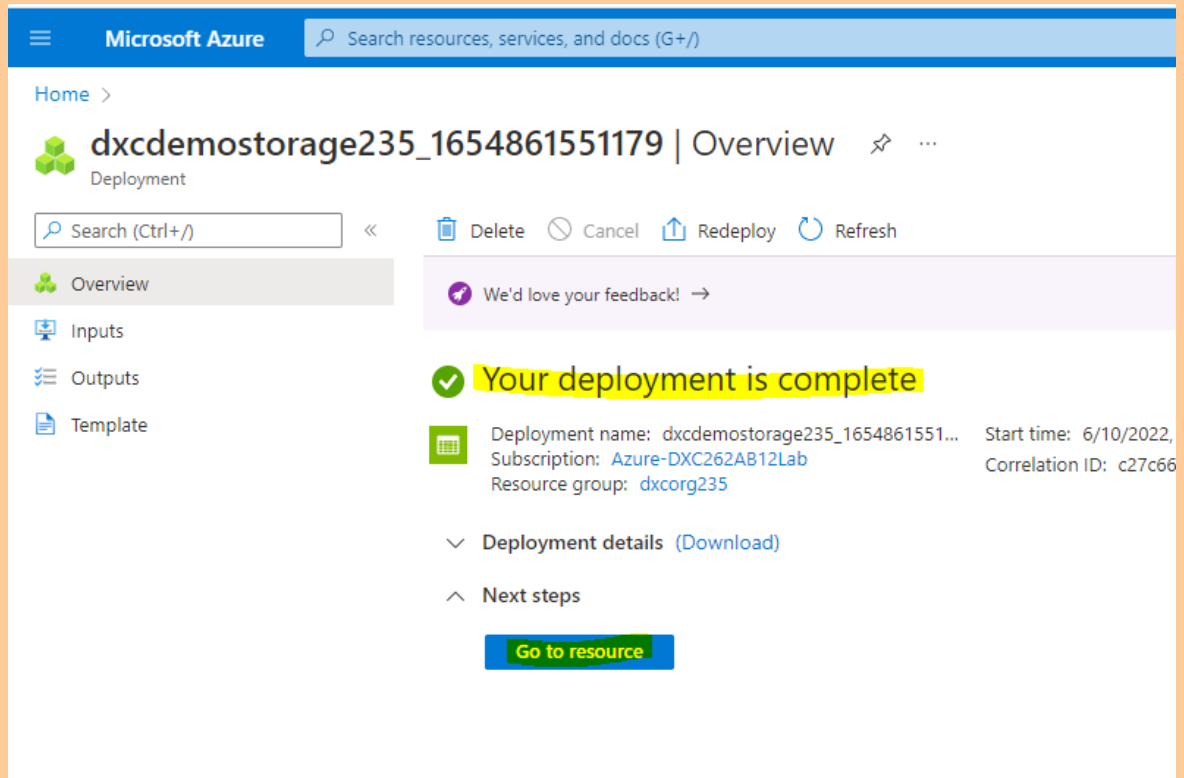
Step 6: After goint to resource click on "Container->+Container".

Step 7: Then give the name for blob as "sourceblob" and click on "Create".

New container

Name *

sourceblob

Public access level ⓘ

Blob (anonymous read access for blobs only)

⚠ Blobs within the container can be read by anonymo
but container data is not available. Anonymous clie
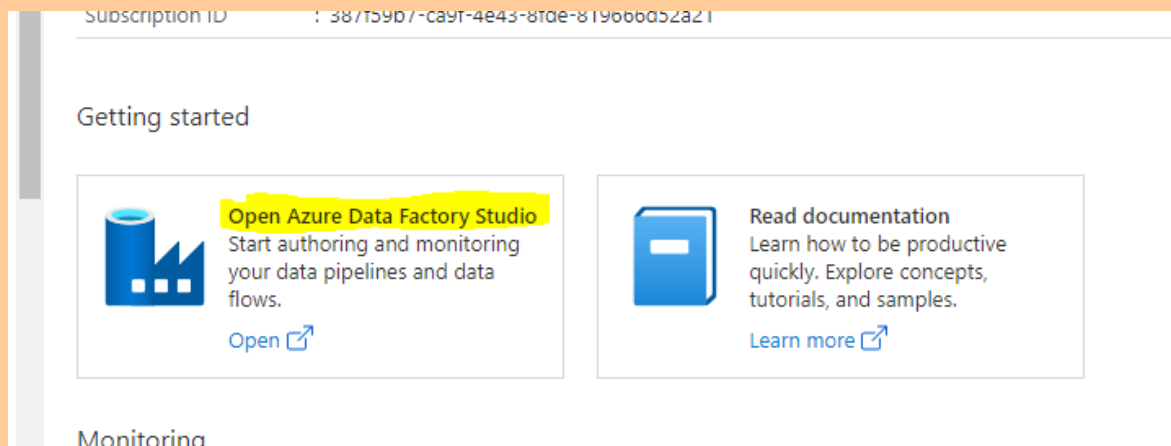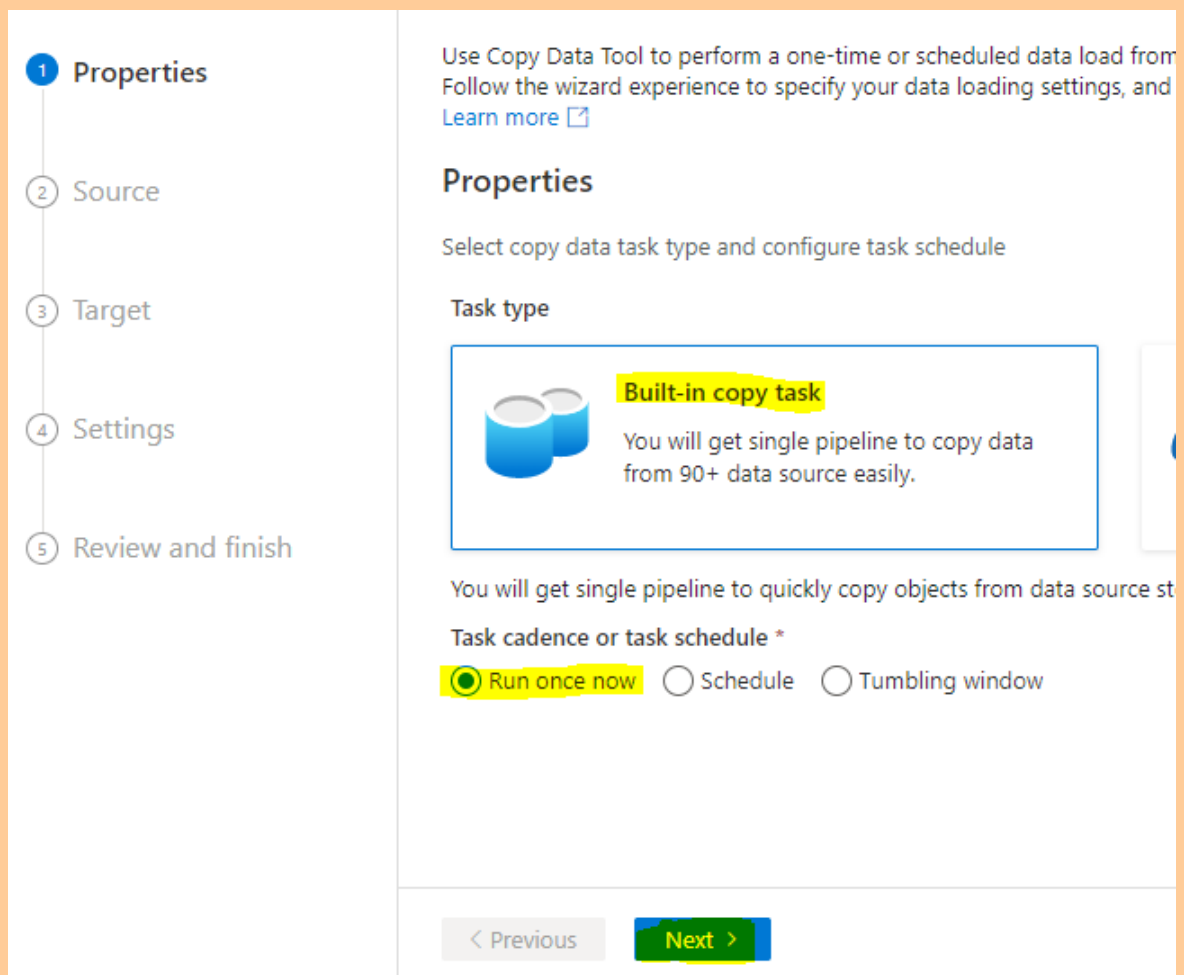enumerate the blobs within the container.

∨ Advanced

Create   Discard

Step 8: Similarly create the destination blob as "destinationblob". Both blobs are created.

Name	Last
☐ $logs	6/10
☐ destinationblob	6/10
☐ sourceblob	6/10

Step 9: Go to Data factory and click on "Azure Data Factory Studio" and then click on "Ingest".

Step 10: Click on "Built-in copy task". And click on "Next".

Step 11: Click on the source as "Azure Blob Storage".

Step 12: Click on "New connection" and create "Connection".

Step 13: Click on "Browse" and "Next".

Source data store

Specify the source data store for the copy task. You can use an existing data store connection or specify a new data store.

Source type   Azure Blob Storage

Connection *   AzureBlobStorage1   ✎ Edit   + New connection

File or folder *

If the identity you use to access the data store only has permission to subdirectory instead of the entire account, specify the path to browse.

📁 Browse

Options

☐ Binary copy ⓘ

☑ Recursively ⓘ

☐ Enable partition discovery ⓘ

Max concurrent connections ⓘ

< Previous   Next >

Step 14: Click on sourceblob and "ok".
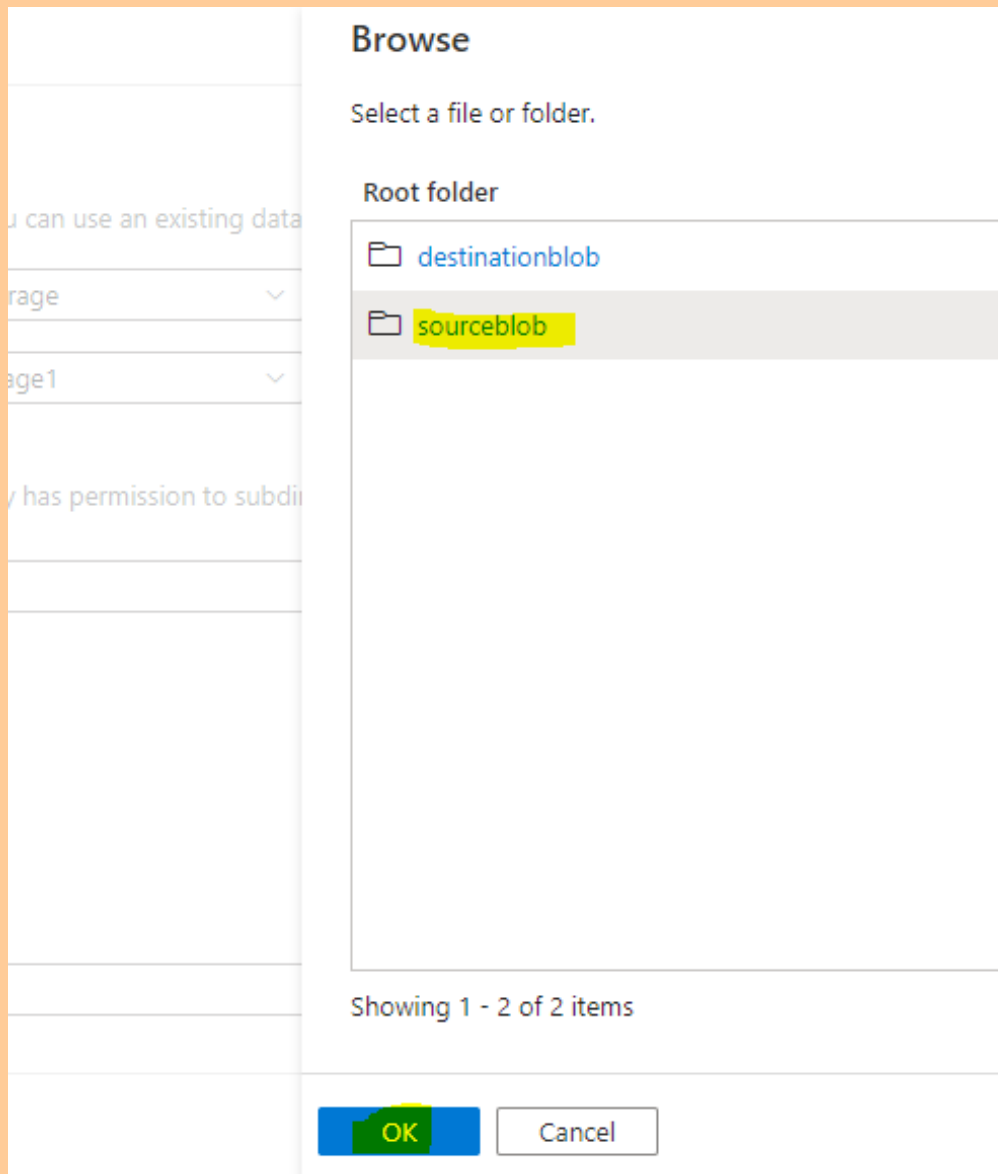
Step 14: In Configuartion follow the steps shown below.

Step 15: For target data set follow the steps shown.

Step 16: Select destinationblob for the target.

Step 17: Follow this for Return Configuration.

Step 18: In the settings click on "next".
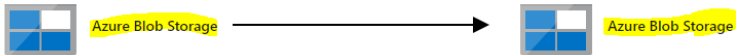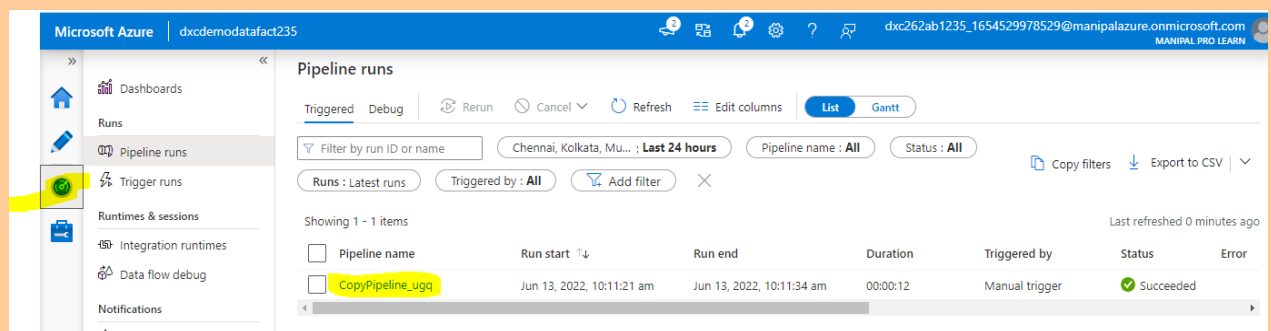
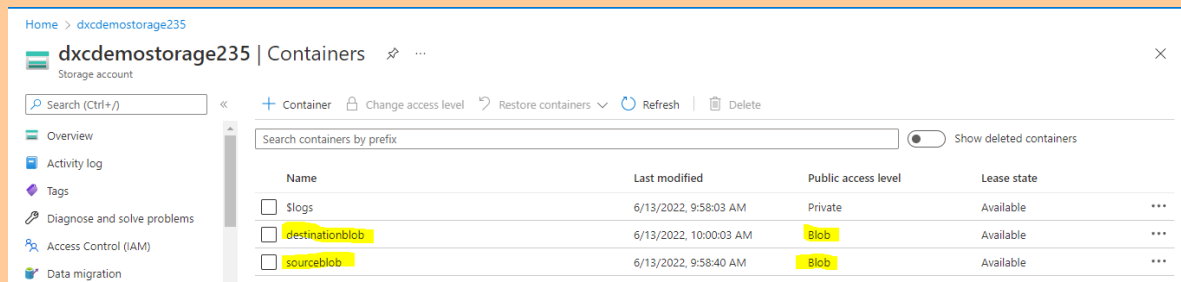Step 19: Review and click on "Ok".

Step 20: Pipeline is created click on "Finish".
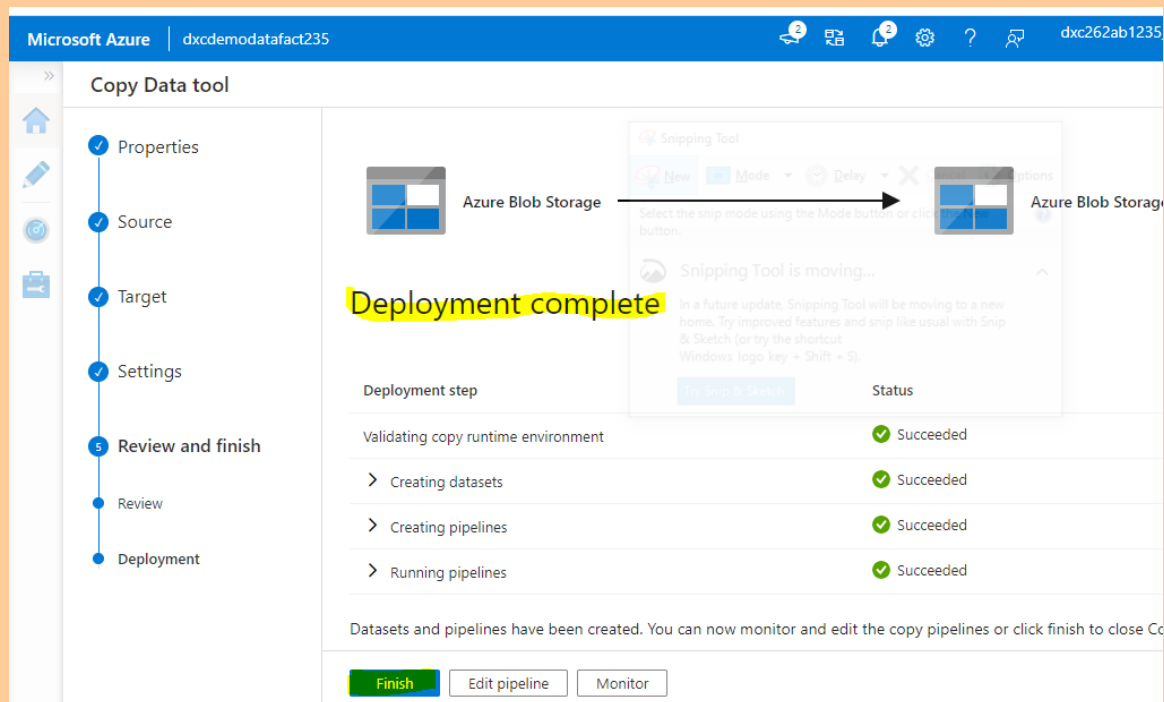
Step 21: Now, enable the pipeline.
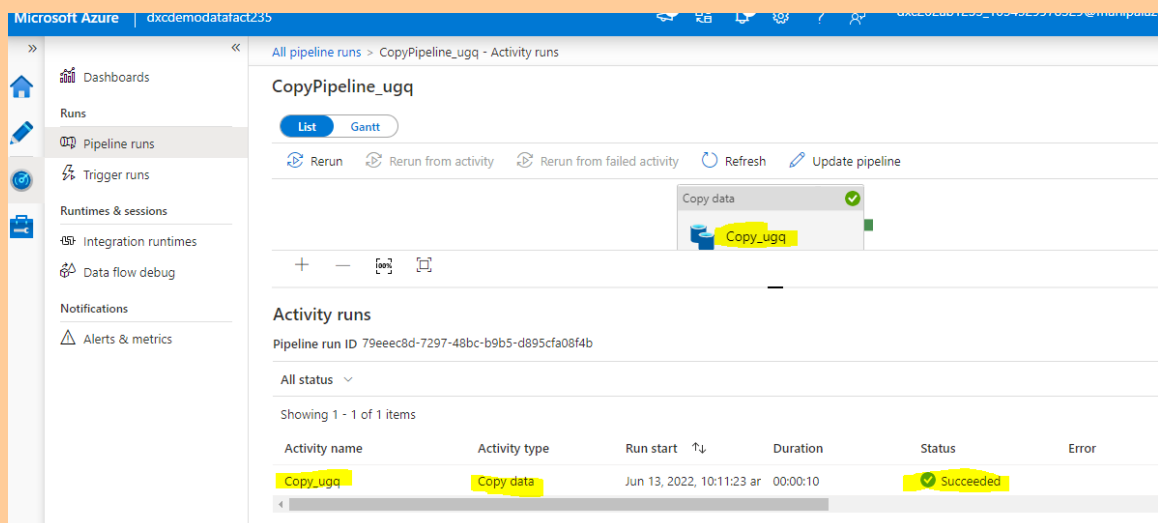

Practical Lab: Create **Azure blob trigger** logic

Step 1: Create a Blob Storage Container as shown.

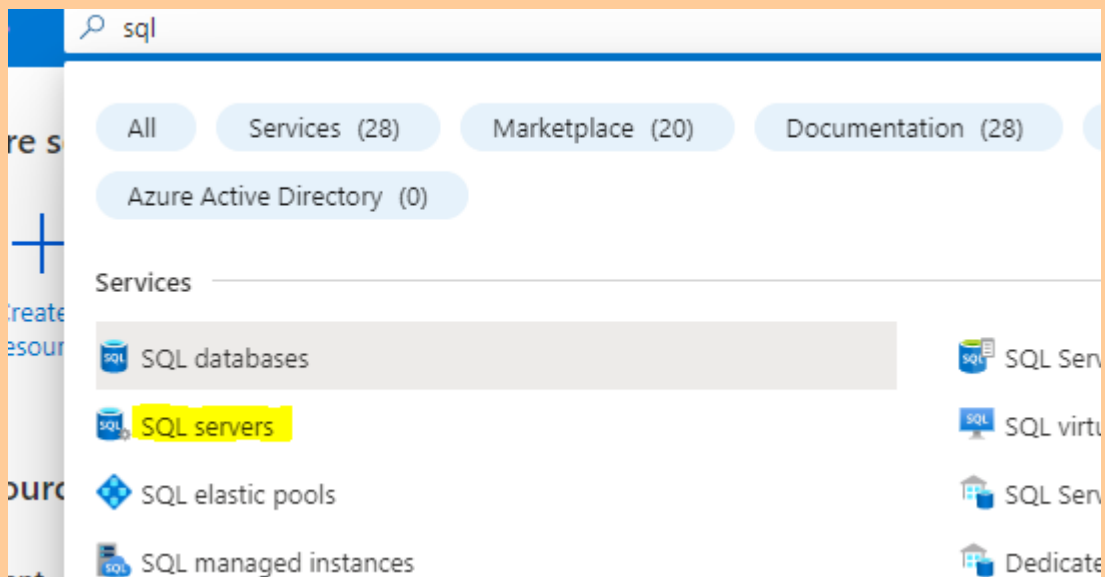Step 2: Click on "Pipeline Runs" and then click on the the pipeline with "Blob trigger".
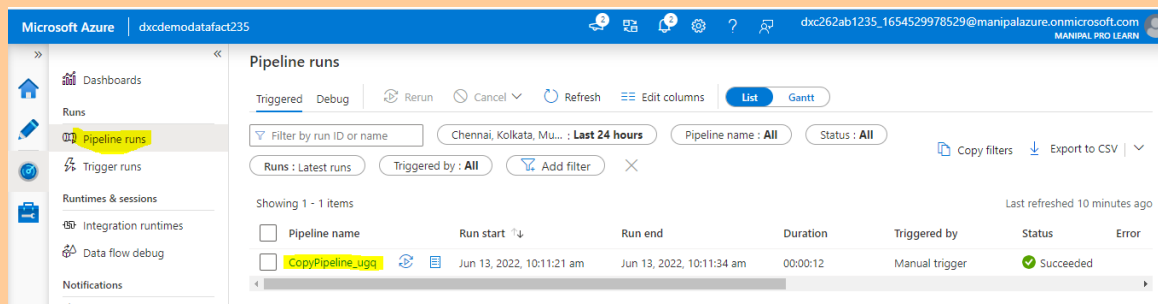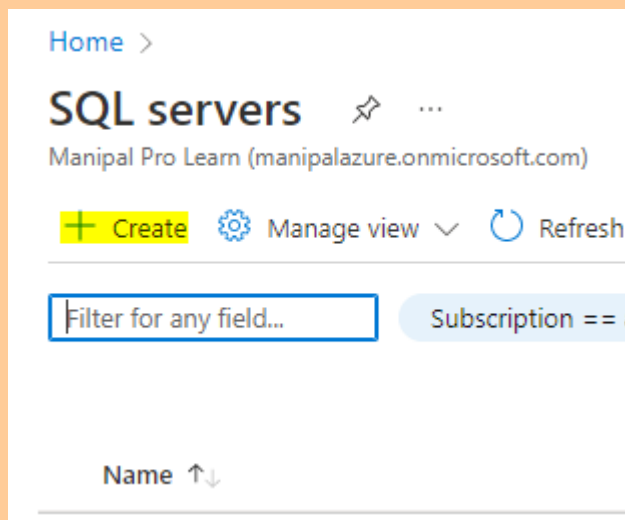

Step 3: And the Blob trigger is ran.

Practical Lab: Create Azure **SQL Server and Database**

Step 1: Click on "Sql Server" in Azure open page.

Step 2: Click on "Create".

Step 3: Fill all the details and click on "Next".

Step 4: Now click on "Create".

Step 5: After clicking on "Deployment" will be done. Now click on "Go to Resource".
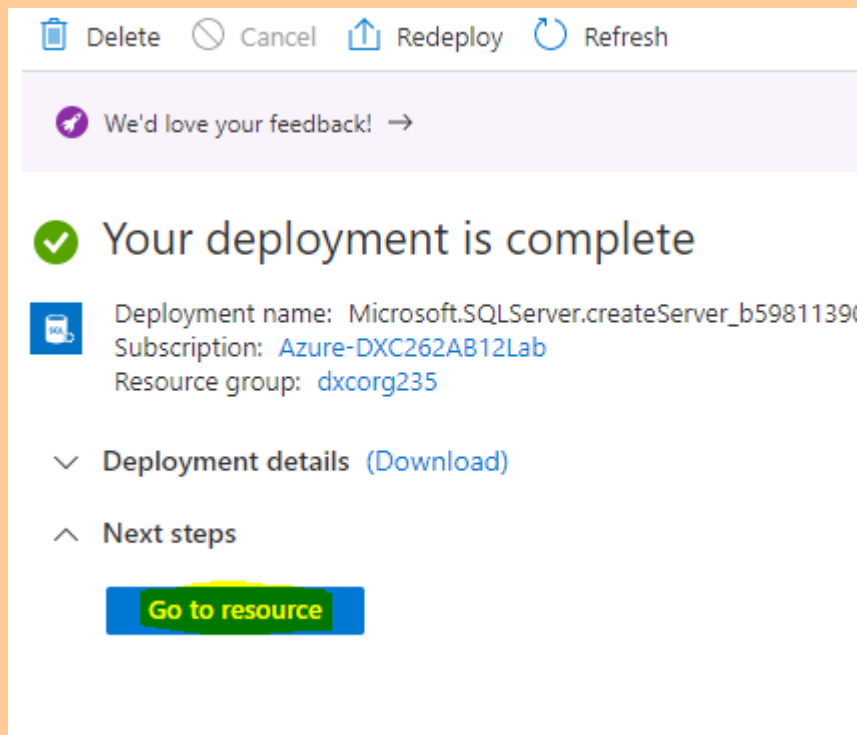
Step 6: Now after going to the resource, Now click on "Create Database".
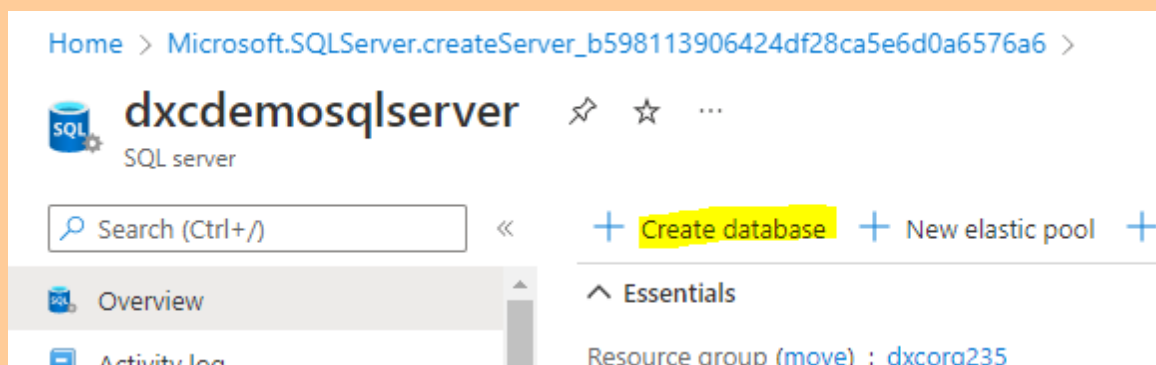
Step 7: Fill all the data and click on "Next".

Step 8: And now click on "Create".

Step 9: Now the deployment is done.

Practical Lab: Add another pipelines for moving data from Staging to **SQL DB**

1. To solve this firstly we need a data storage account, data factory and a sql database. we can create them as shown in above steps.
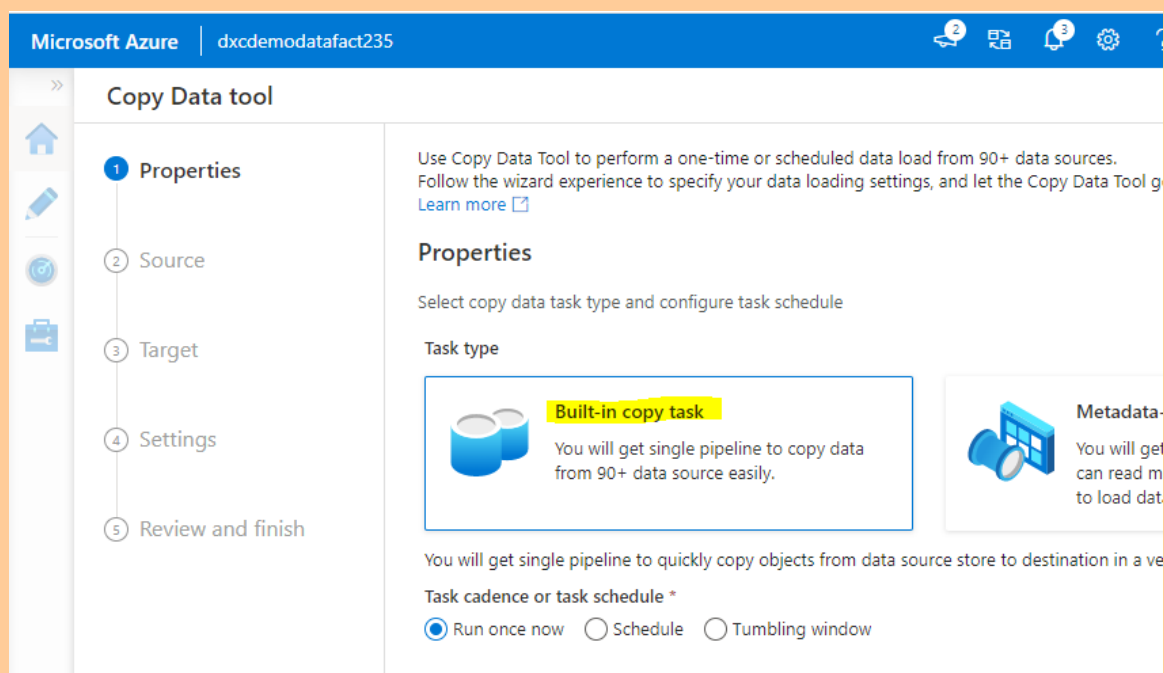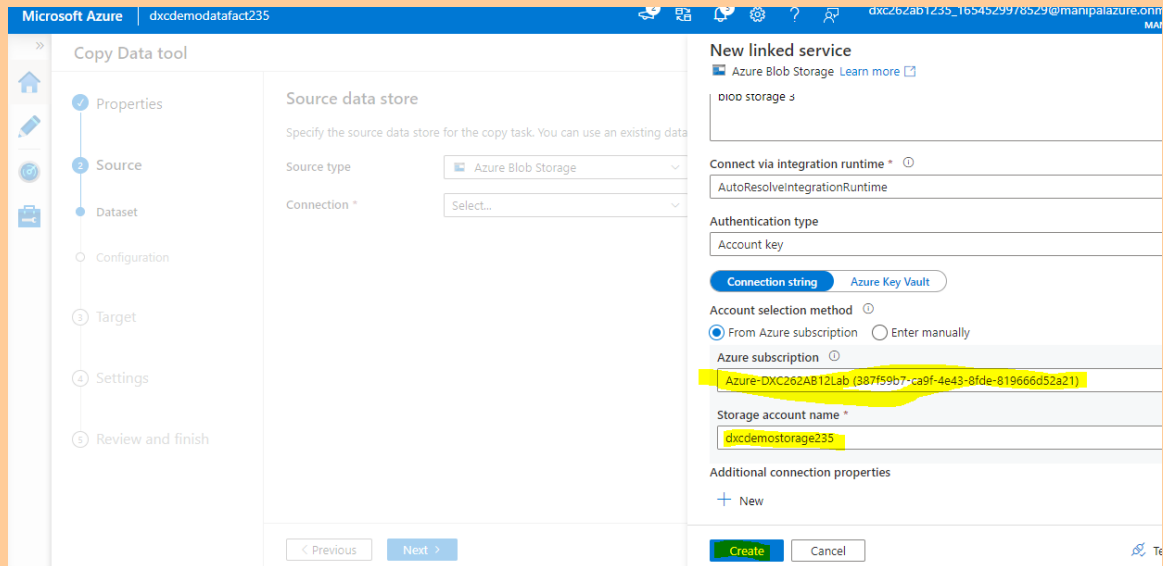
2. After that we need a source storage container to hold blob data. And to move this data to SQL db we need to use pipeline.So we can use Data factory to do this
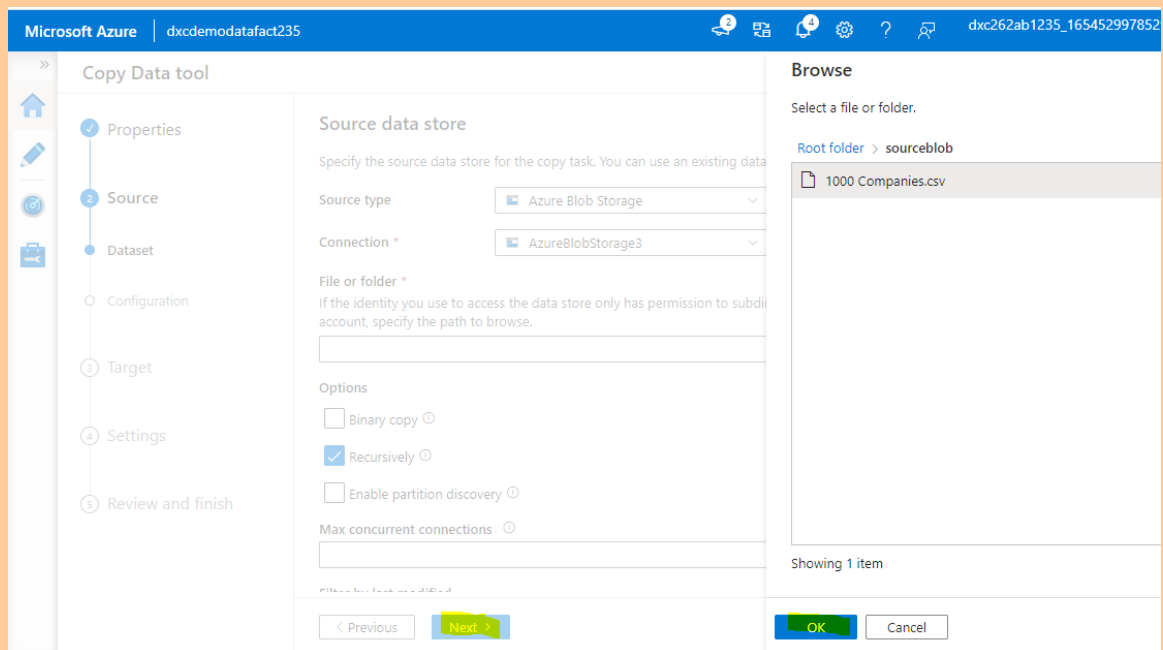
Step 1: Go to Data Factory->Ingest->Built-in Copy.

Step 2: Fill the source details of the source container in the stoarge account. And Click on "Create".

Step 3: Browse the source file and then click on "Ok" and click on "Next".

Step 4: Now select target as Sql Db and fill details and click on "Ok" and click on "Next".

Step 5: Check Summary and click on "Next".

Step 6: Now the pipeline is created and now review it and click on "Finish" it. And the it is Deployed.

Step 7: Now navigate to sql database Query page to check the data.
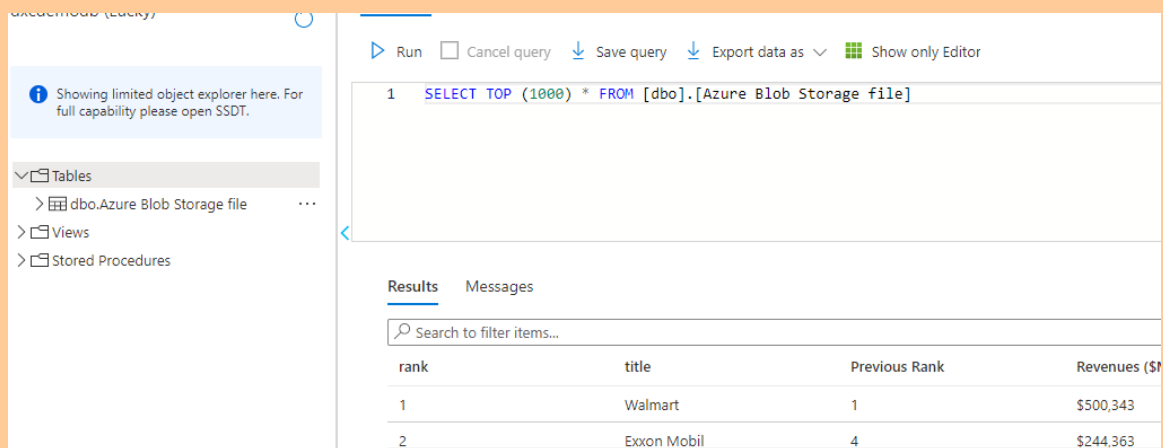
Result: By using all the above shown steps and tools we can create pipeline with the help of a Data factory and can use them to validate and copy data into a Sql Database.

Conclusion: Here, we have stored, validated and copied Blob data to the SQL Database.

Name: Lakshmi Naidu Thatikonda

Reg No: dxcab1226

Project2 Name: AP Morgan Data Platform

Date: 10/06/2022

Project 2: AP Morgan Data Platform

Project 2 : AP Morgan

- Multiple Internal applications sends the data(huge size) in CSV format on daily basis in the cloud storage location. There are couple of Data/schema validation needed to be performed on this incoming data. Once everything is passed data to be persisted as Delta table in Databricks for downstream system.

CSV Data   Cloud Storage   Delta Table
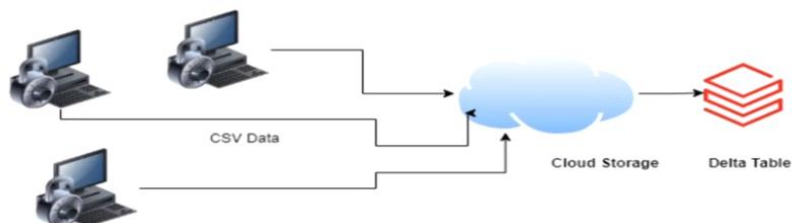
## Project 2 : AP Morgan- High Level Detail

- Internal Application sends CSV file in Azure data lake storage.
- Validation needed to apply on this follows:
  - Check for duplicate rows. If it contains duplicate rows, file need to be rejected.
  - Need to validate the date format for all the date fields. Date column names and desired date format is stored in a Azure SQL server. If validation fails file will be rejected.
- Move all the rejected files to Reject folder.
- Move all the passed files to Staging folder.
- Write the passed files as the Delta table in the Azure Databricks

Project 2 : AP Morgan
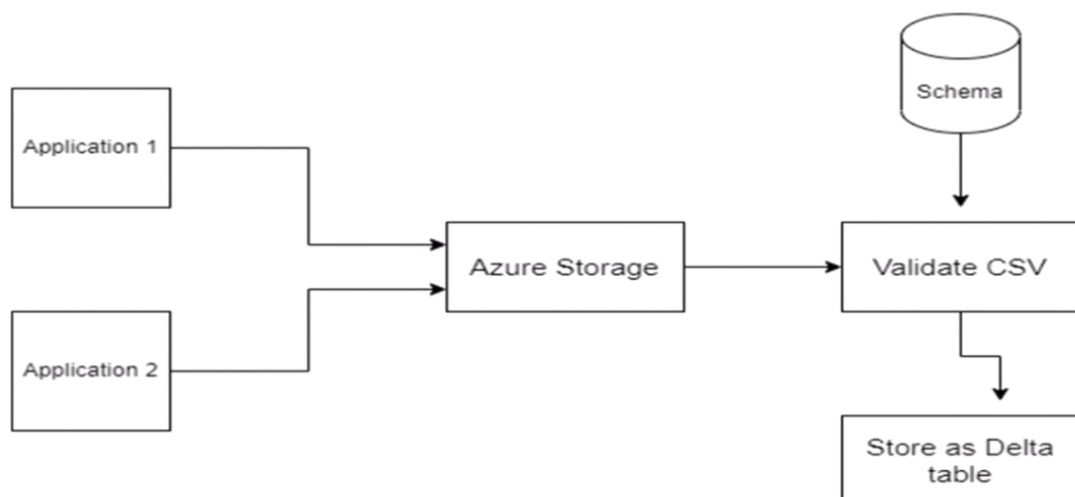
Practical Lab: Create a **Databricks**

Step 1: Login to Azure Portal and select "Data Bricks".
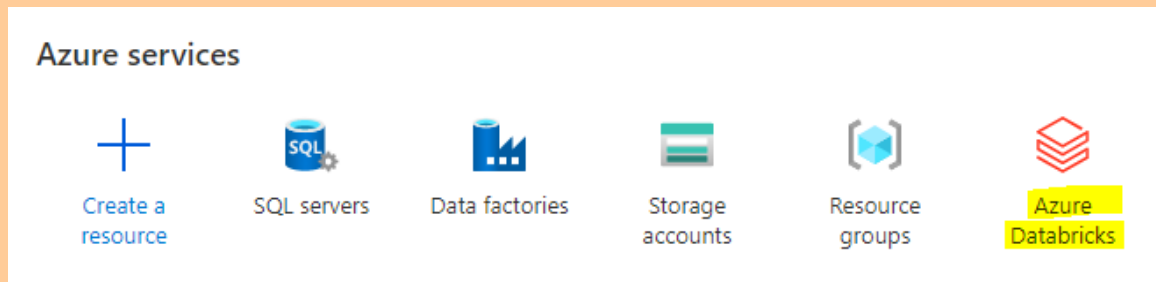
Step 2: Click on "+create".
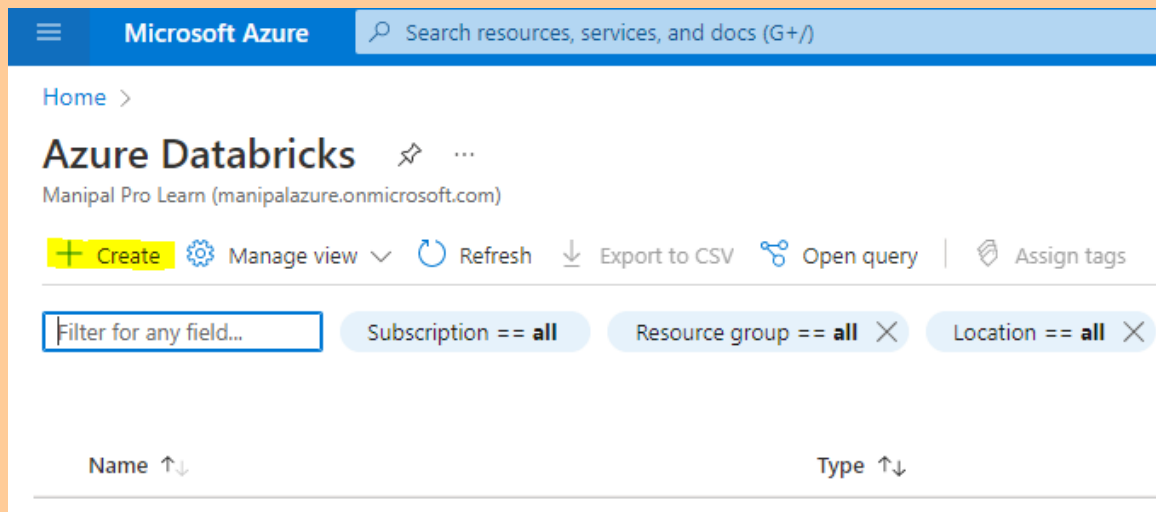
Step 3: Fill all the required details and click on "Next" and finally on Review and Create.

Step 4: Click on "Create".

Step 5: Deployment done. Now click on "Go to Resource".

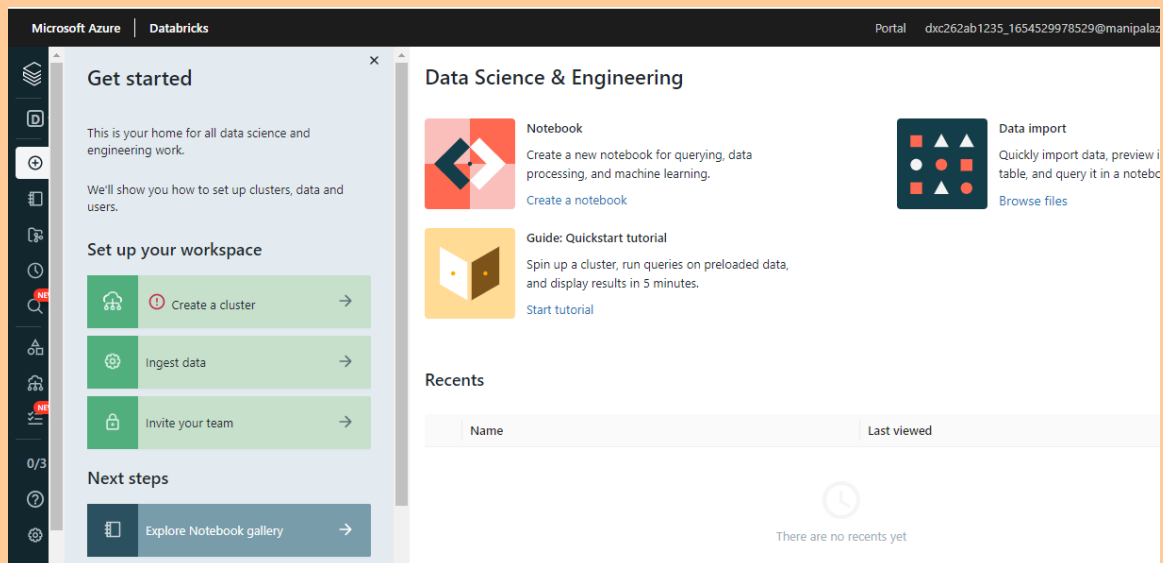Step 6: Click on "Launch Workspace".

Step 7: Sign into Databricks and Azure Data Bricks is created.
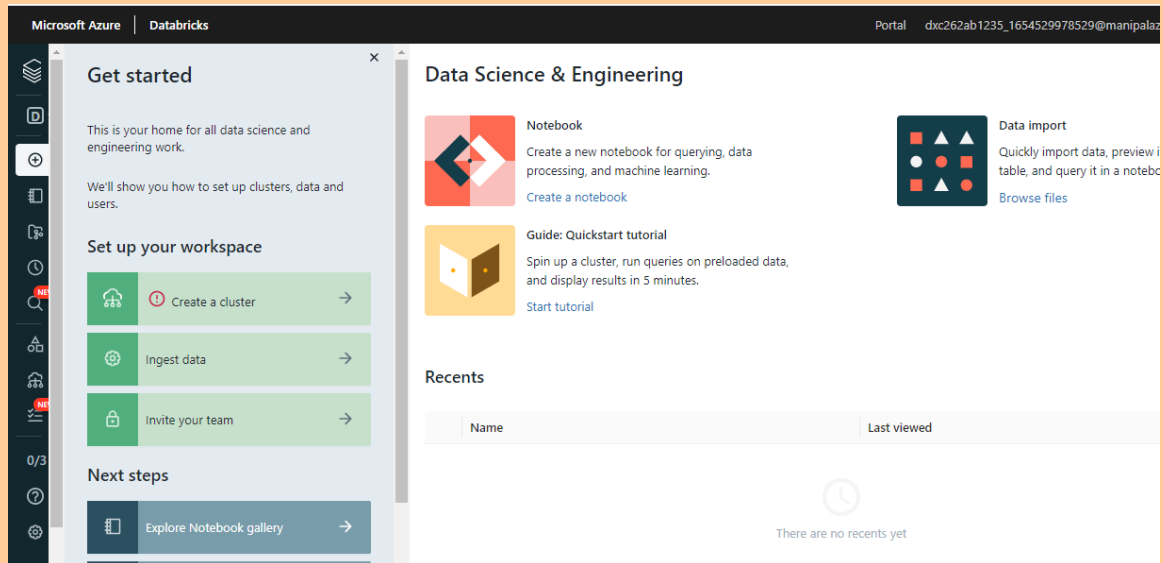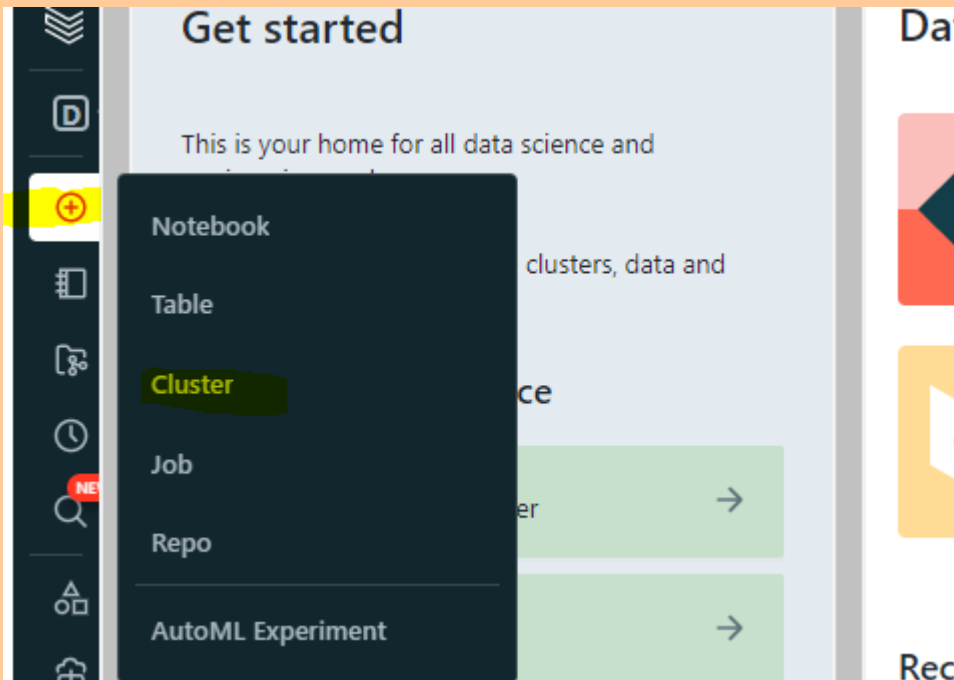
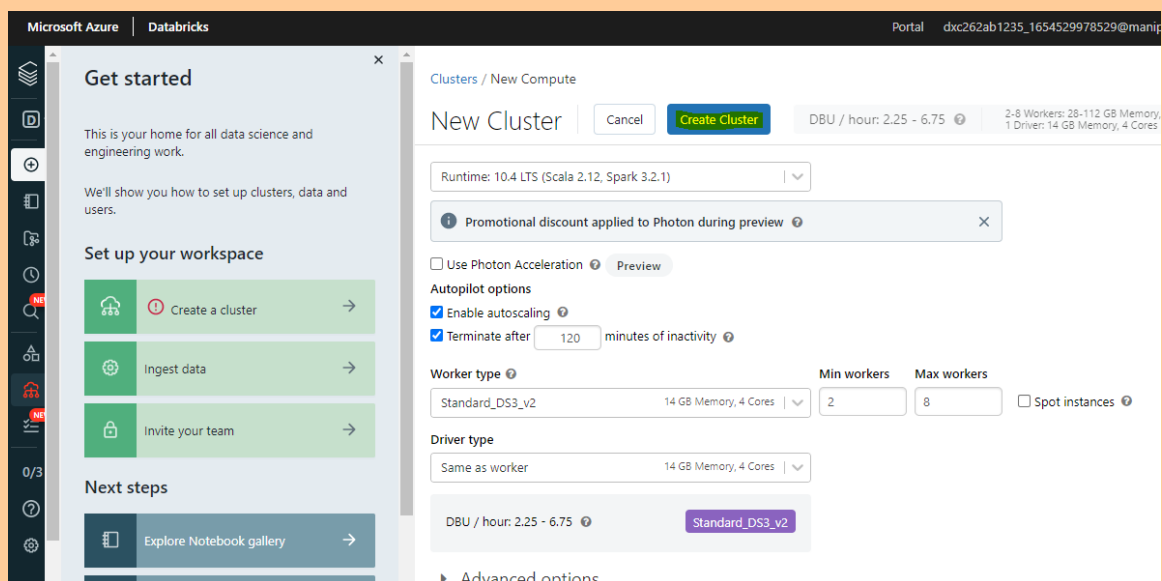## Practical Lab: Create **Cluster in Azure Databricks**
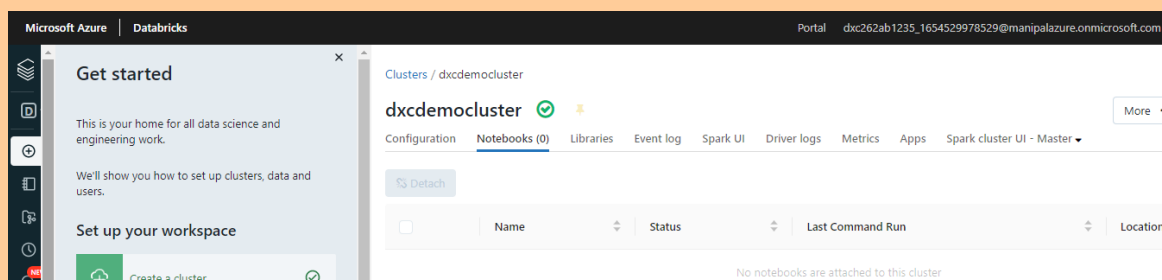
Step 1: Open Data Bricks Workspace.

Step 2: Click on "+Cluster".

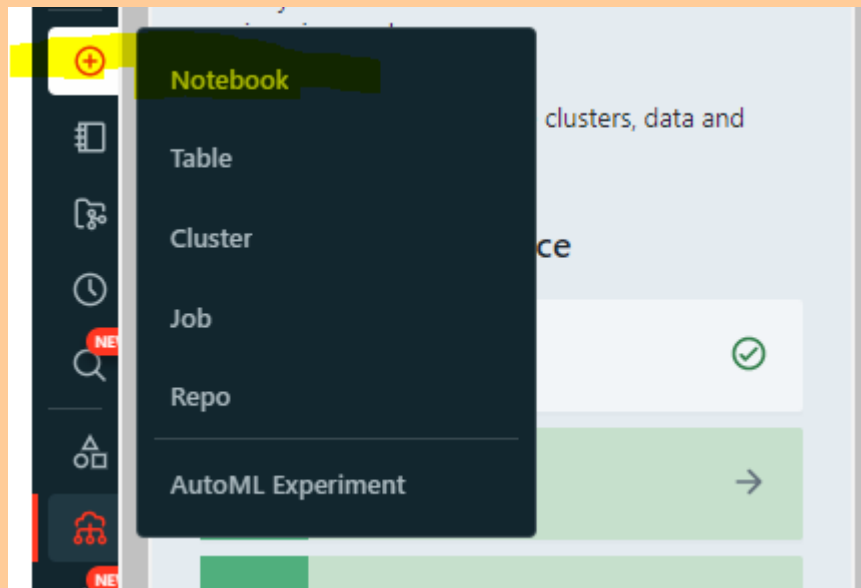Step 3: Fill all the necessary information for the cluster and then Click on "Create Cluster".
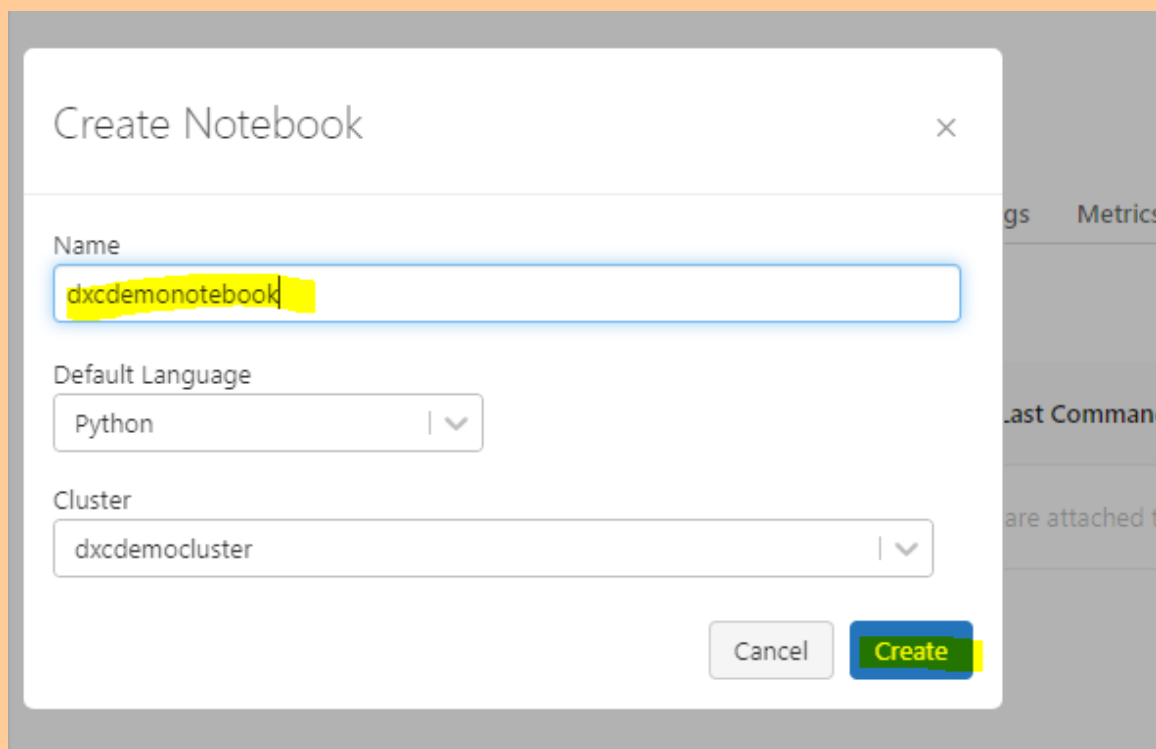

Step 4: Cluster is created.

Practical Lab: **Add notebook in Databricks** and Implement the Business Logic
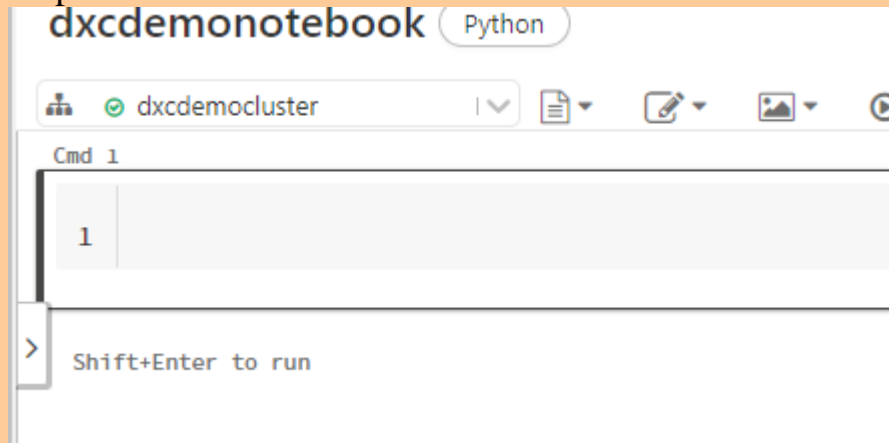
Step1: We need a notebook to perform operations with data available in DataBricks. So Click on "+notebook" in Data Bricks Workspace.
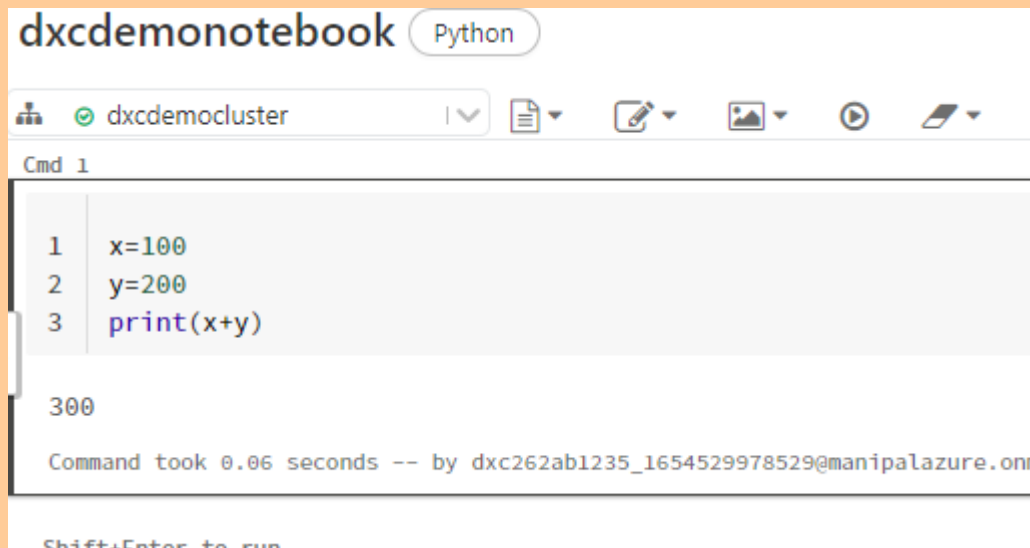

Step 2: Fill the name and Click on "Create".
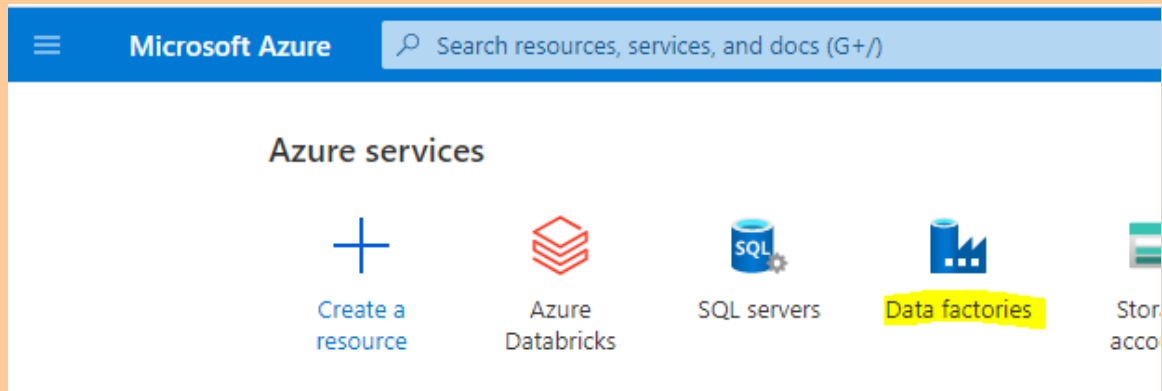
Step 3: Now the notebook is created.

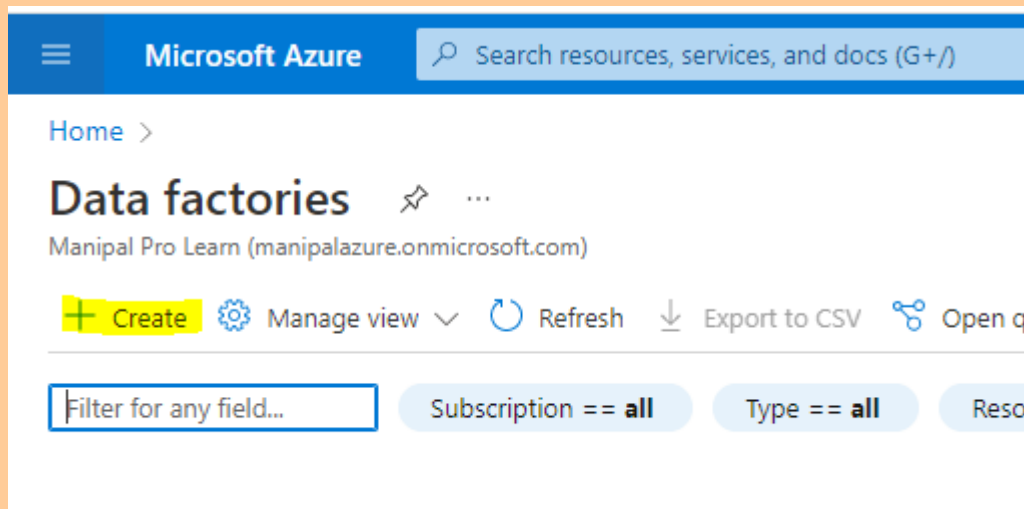Step 4: And then Now perform some operations in the notebook.

And now we have note book which can be attached to Data Factory and execute.

Practical Lab: **Azure Data Factory** For AP Morgan

Step 1: Go to Azure Portal and click on "Data Factories".

Step 2: Click on "+Create".

Step 3: Fill all required fields and then "Review+Create".

Step 4: Click on Create to start Deployment.

Step 5: Deployment is done. And click on "Go to resource".

Step 6: Now click on "Open Azure Data Factory".
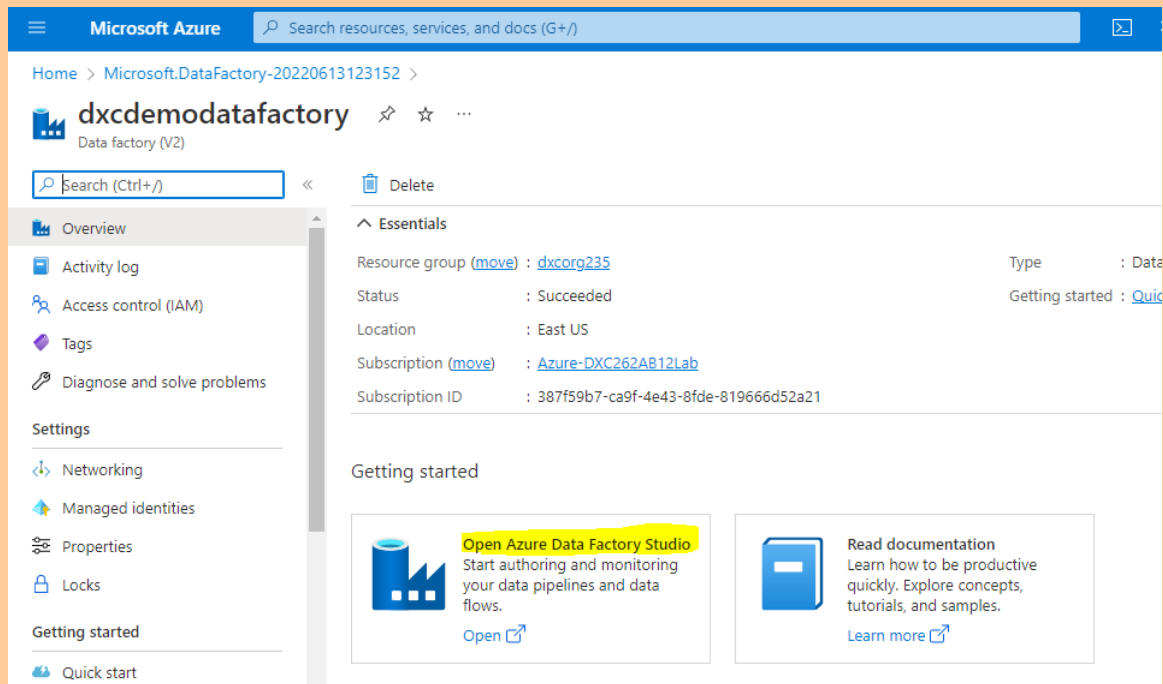
Step 7: And the Azure data Factory is created.

Practical Lab: **Create Azure Databricks** Linked Service in ADF

1. Here, we should create a pipeline to establish a connection between Data Bricks notebook and Data Factory.

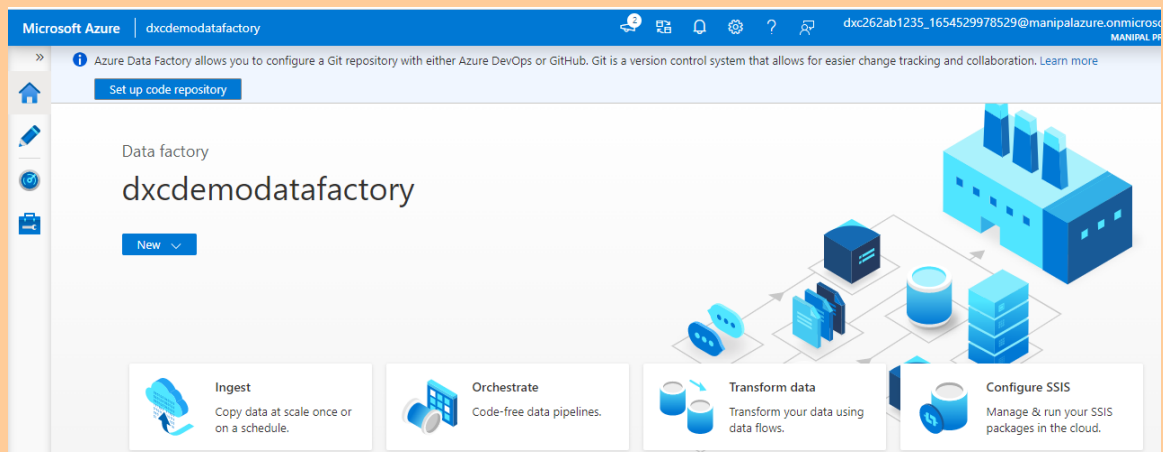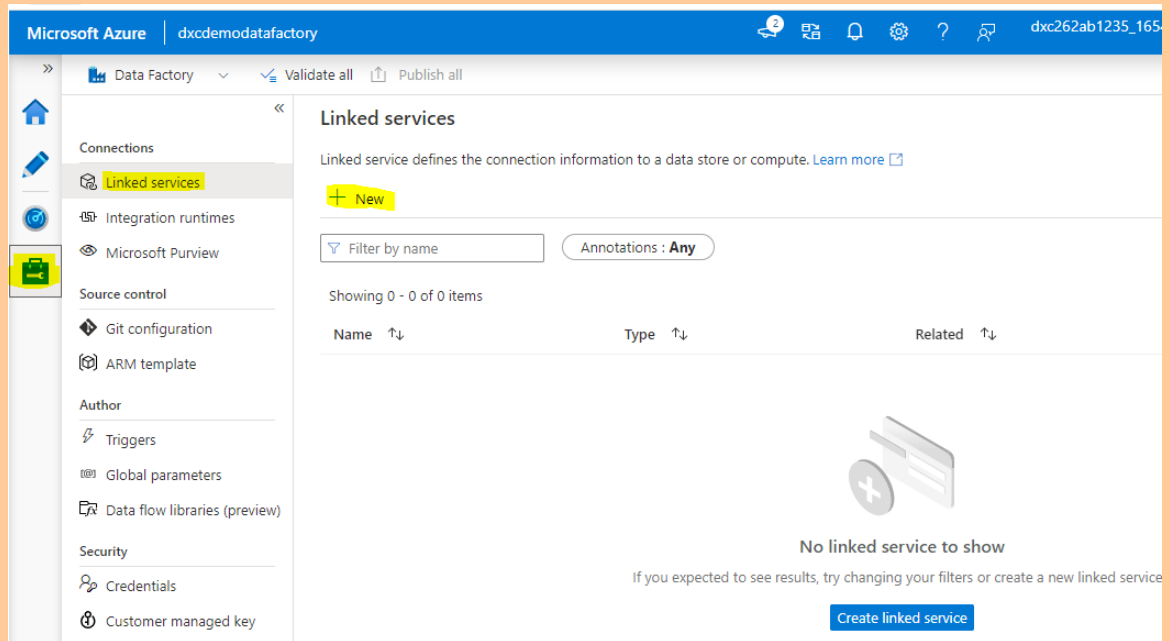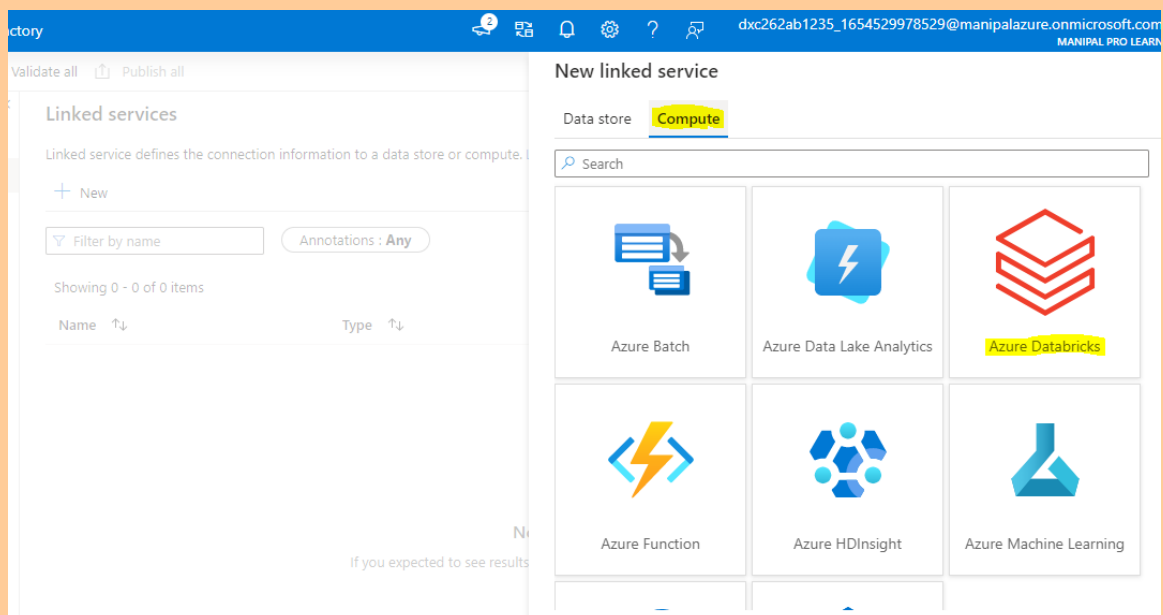Step 1: Click on "Manage" and select "Linked services" under "Connections" and click on "+add".

Step 2: Click on Compute and then select "Azure Databricks".

Step 3: Fill all the required fields to link bata bricks service.

Here to get the access token let's take help from Databricks.

Move to "User Settings" in "Settings" of "DataBricks".
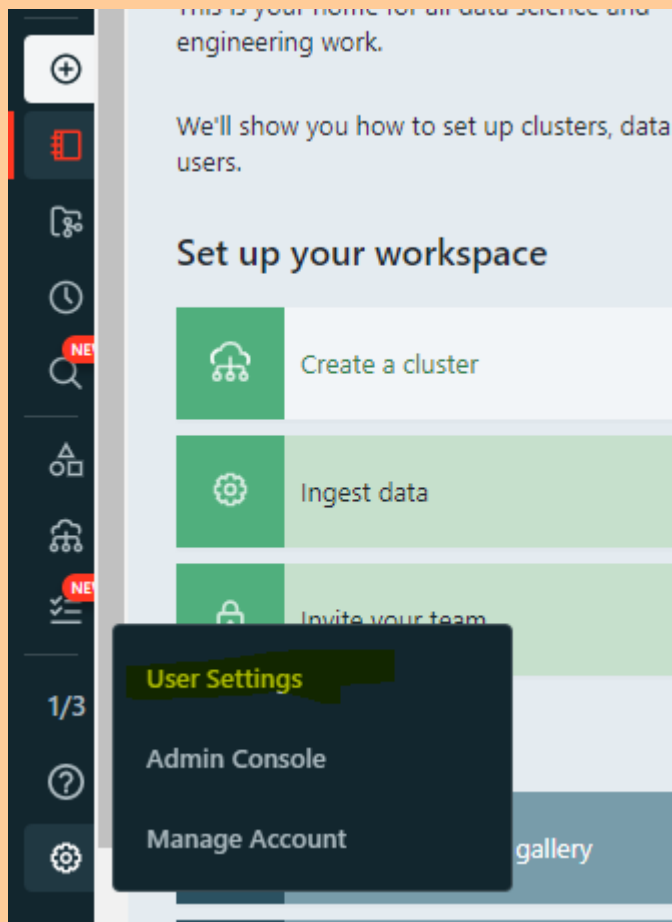
Click on "Generate new Token".

Click on "Generate".

Token is generated.

Step 4: Click on "Create" after filling the required details.

Step 5: Azure data bricks is created.

Step 6: And here we should create a pipeline. To do that, click on "Author" ->"Add Resource" to create a pipeline in Data Factory.

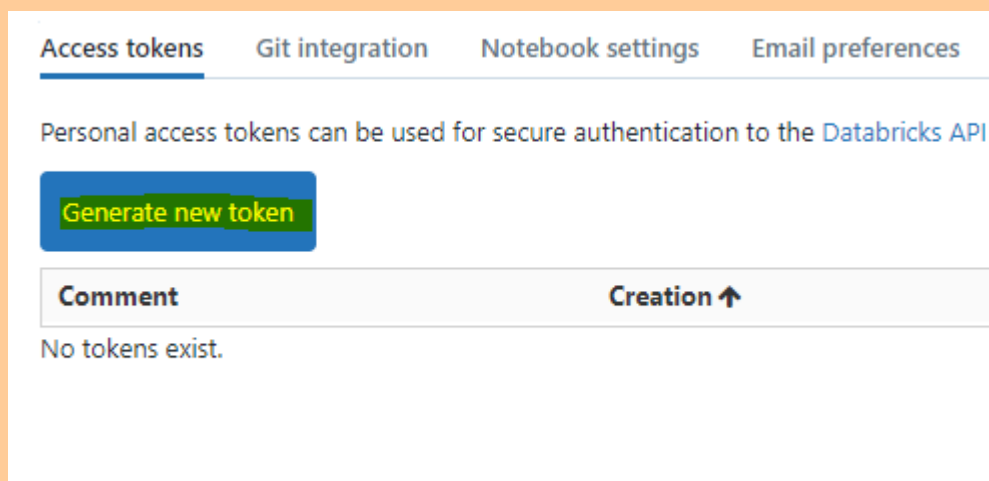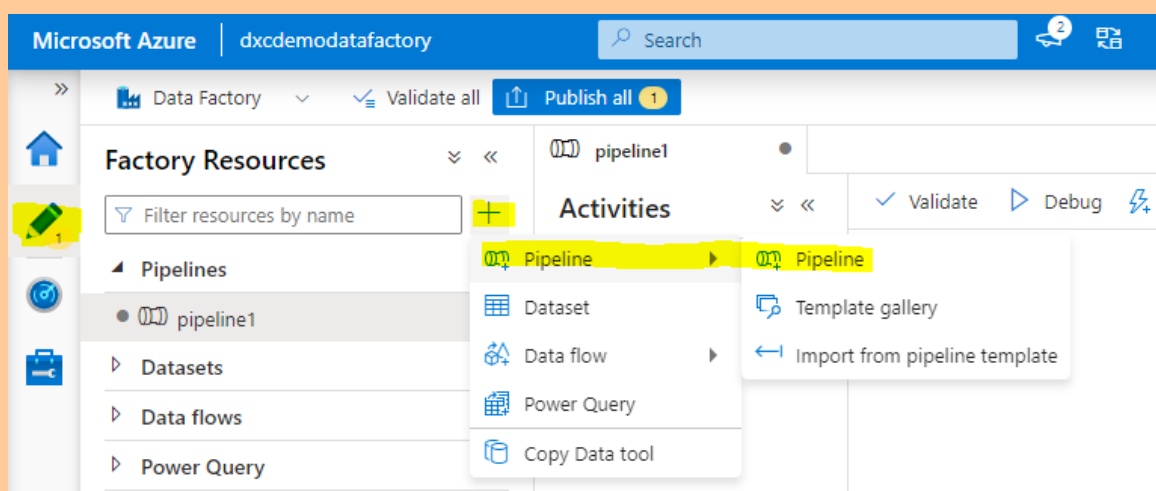Step 7: After that pipeline is created and now click on "Pipelines" and select the pipeline and move to the "Activity" block and then select the "Databricks" drop down and then drag "Notebook" on to the pipeline workspace.
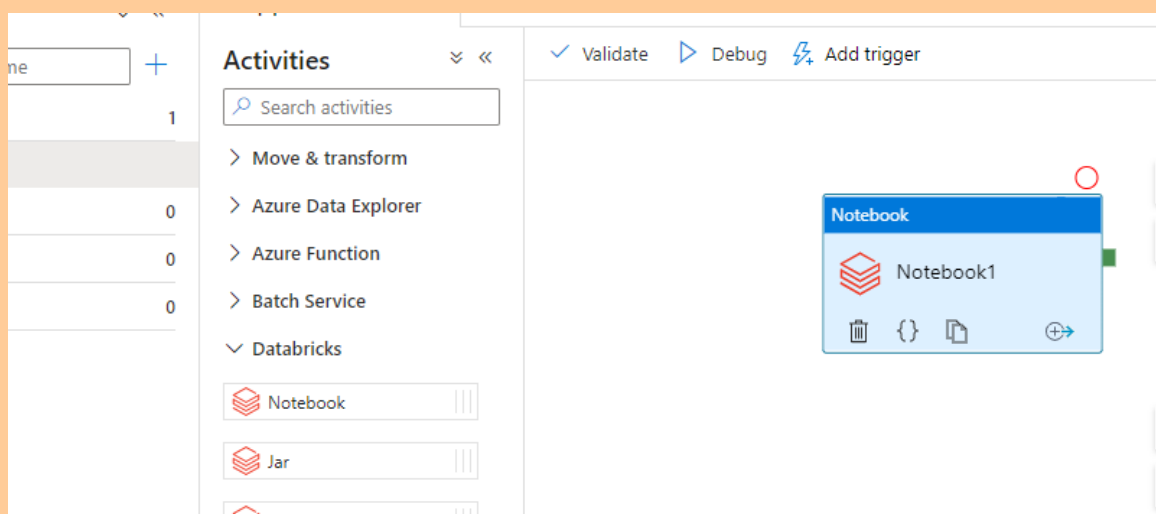
Step 8: Fill all required details and Click on "Publish all" to publish the pipeline and we should run the pipeline by using trigger.
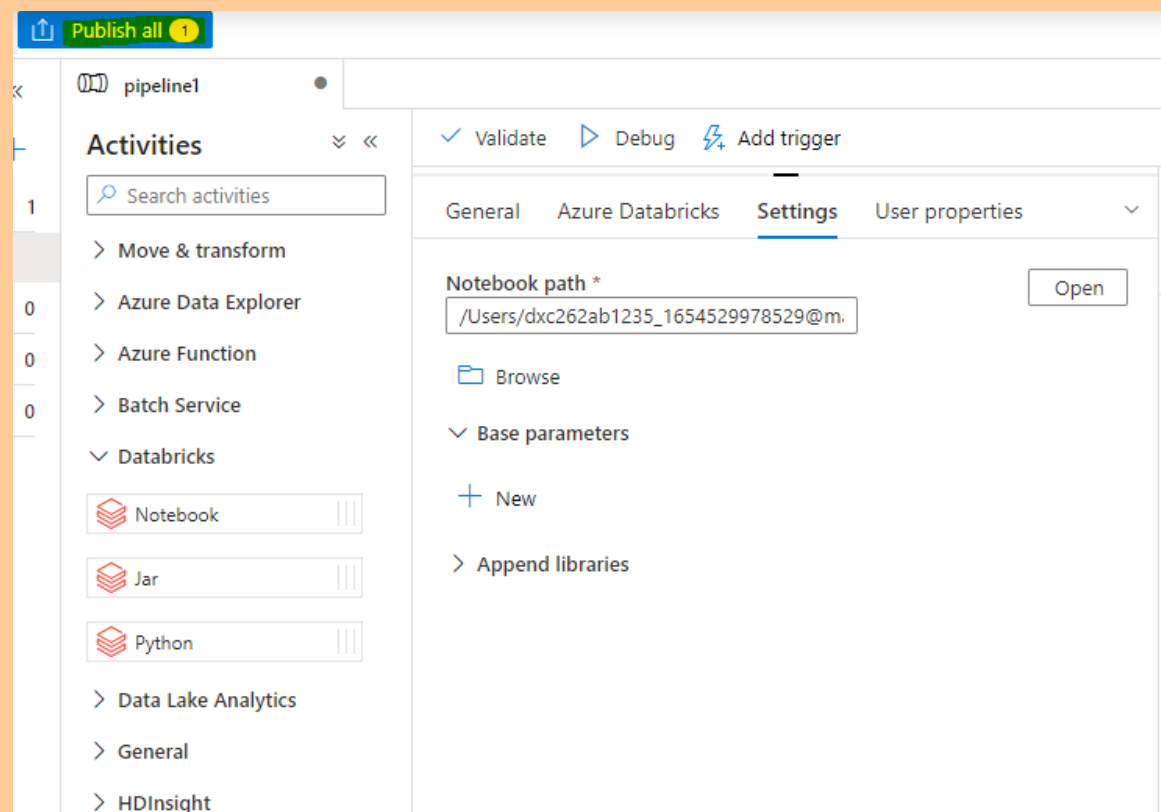
Step 9: Now trigger and moniter the Pipeline created.
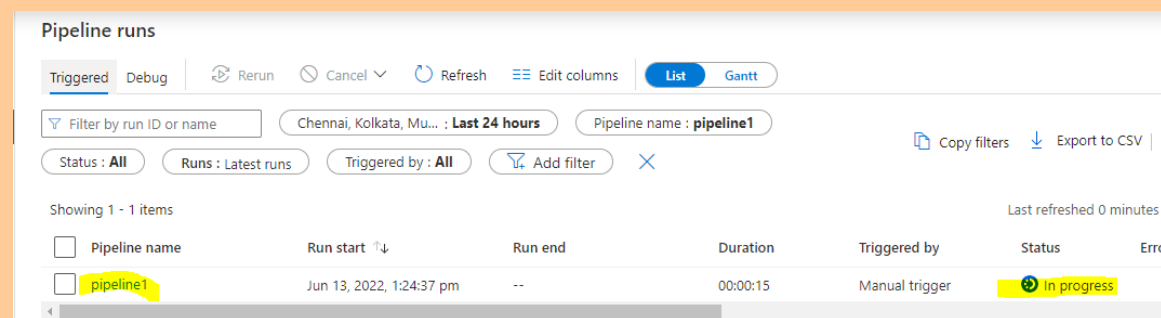
Step 10: Pipelining is successfully executed.

Result: We can observe that, we have successfully link and trigger an Azure DataBricks notebook using Data Factory.
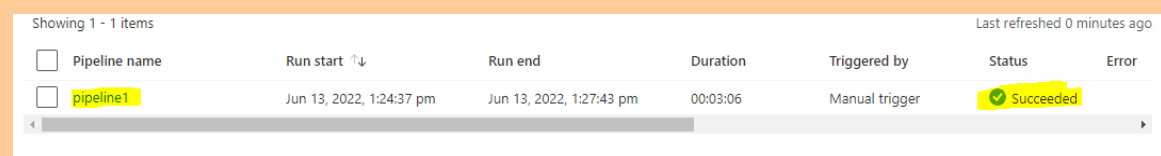

Conclusion: We can see and establish the relation between Azure Data Factory and Azure Data Bricks.

References: https://docs.microsoft.com/en-us/azure/data-factory/concepts-pipeline-execution-triggers