DXC REALTIME PROJECTS

AZ-900, DP - 203

JUNE 10, 2022

DXC

Name: Nandi Vomkara Aditya Mohan

Reg No: DXCAB1203

Project1 Name: Smart Vehicles

Date: 10-06-2022

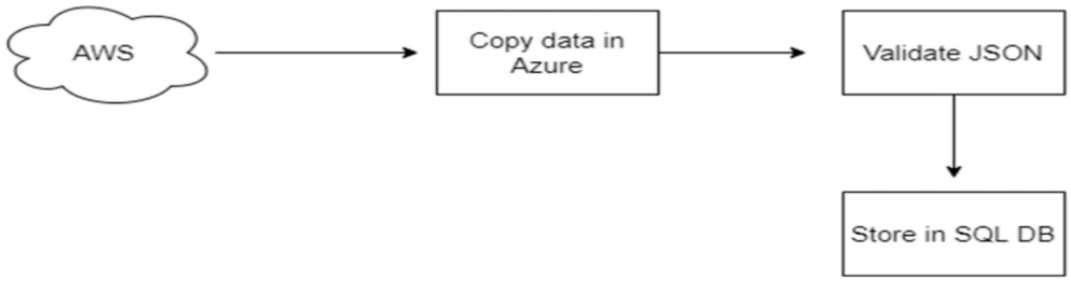
Project 1 : Connected Vehicles

- 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.

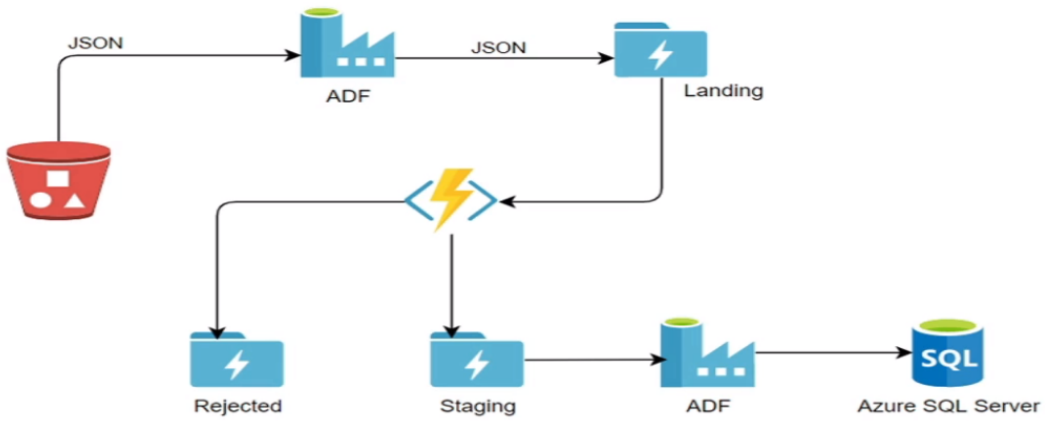
Project 1 : Connected Vehicles

- 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



Project 1 : Connected Vehicles



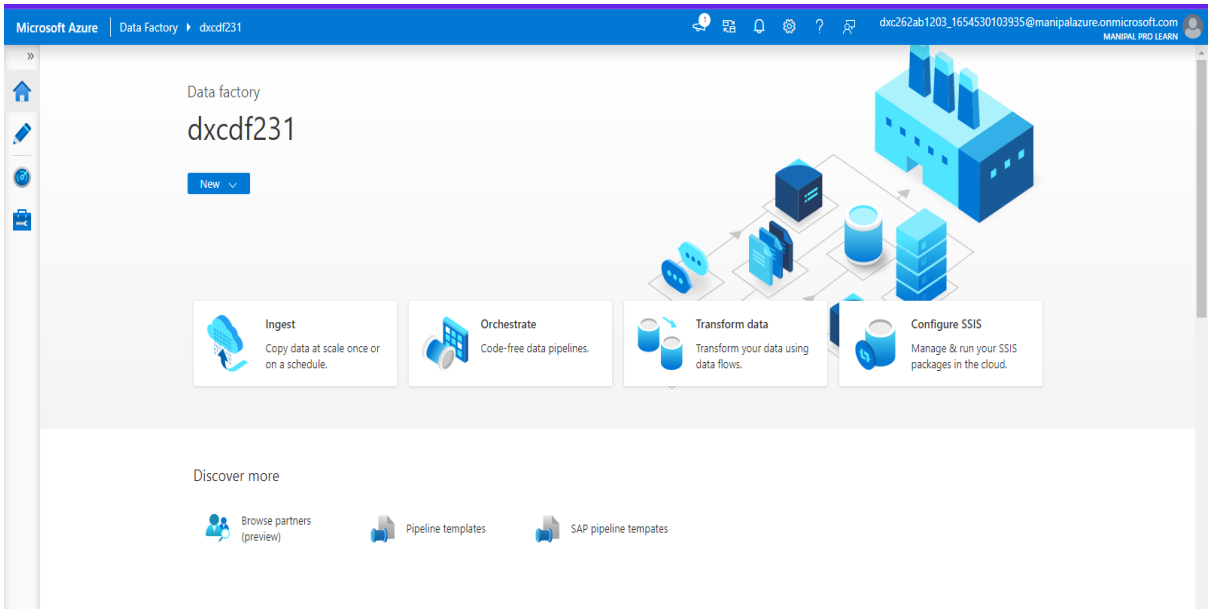
Architecture Diagram for Connected Vehicle Project

Motivation:

Motivation of this project is to connect and migrate the data from IOT connected AWS cloud to our clients Azure Cloud. This action includes the linking of AWS data using azure data factory which provides insights of the data as well as migration in a same step with in no time and cost effective.

Resource used:

Azure data factory



Practical Lab: Create Azure Data Factory Account for Data pipelines

Azure data factory can be created by using the following steps

Step-1: login to the azure portal and search for azure data factory as shown in fig 1.1

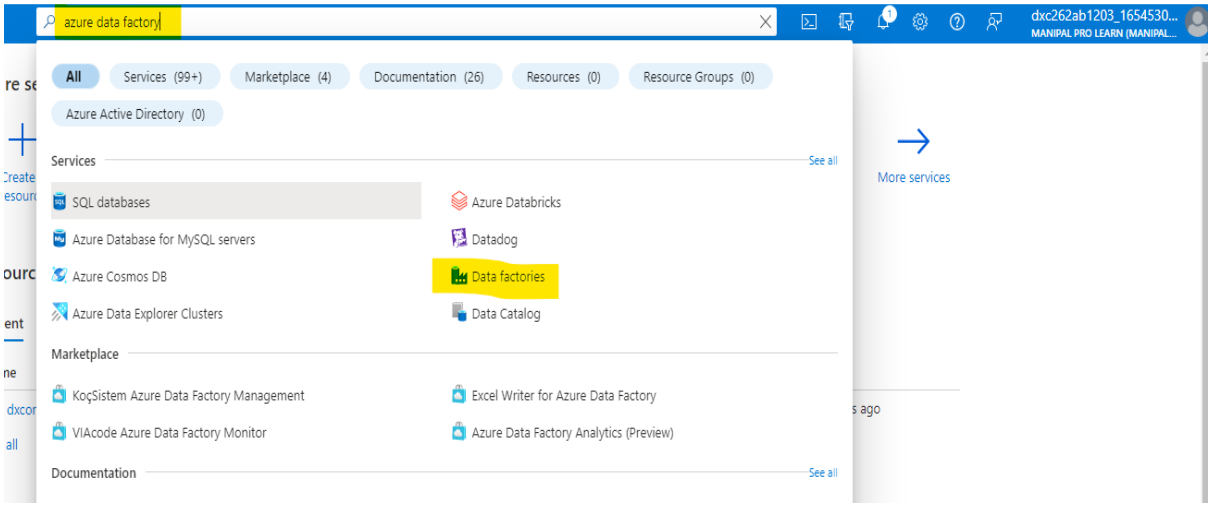


Fig 1.1 shows search for data factory in azure portal

Step-2: after navigating to the azure data factory page click on create and select the resource group, name, region, version after that click on next follow the fig 1.2

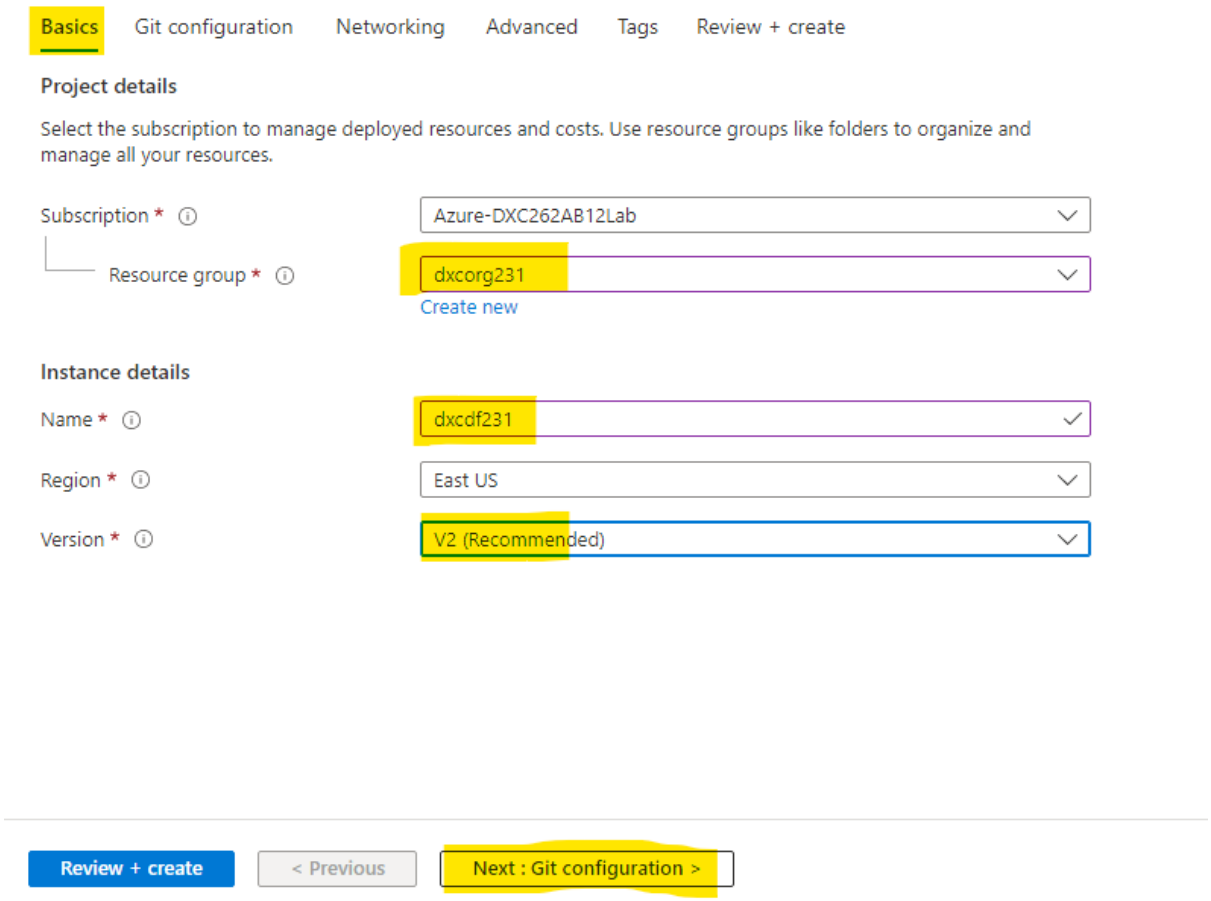


Fig-1.2 shows the creation procedure of data factory

Step-3: click on git configurations and checkbox it as configure git later as shown in fig 1.3

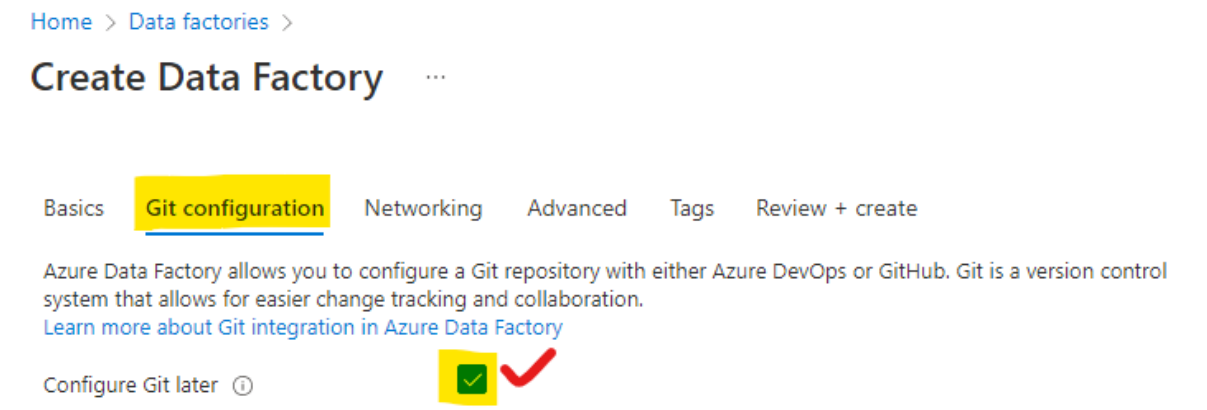


Fig-1.3 shows the git configuration during data factory creation

Step-4: Go through the next steps followed by succesfull completion of validation clikc on create as shown in fig-1.4

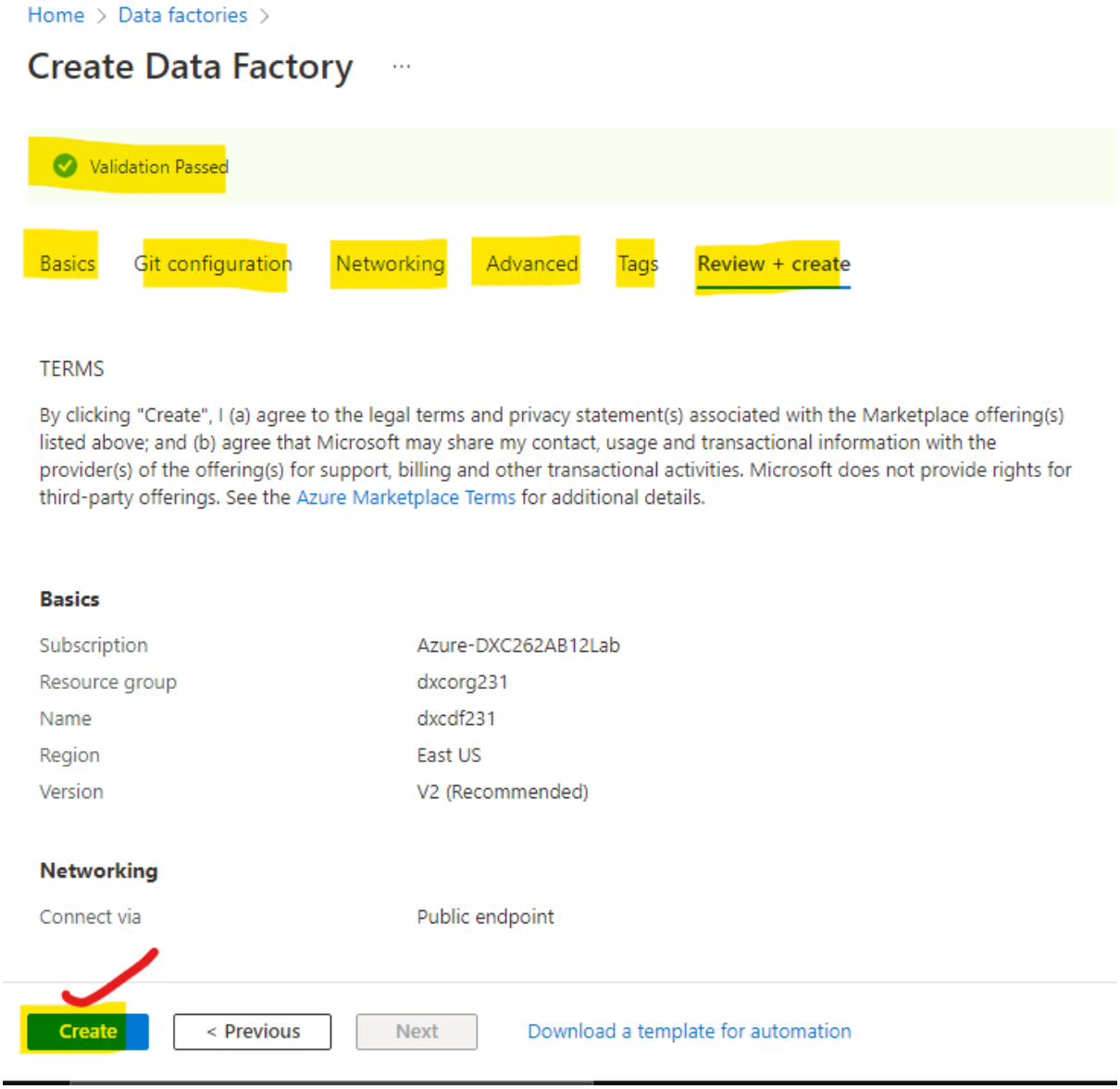


Fig-1.4 shows the steps in creation of data factory

Step-5: after clicking on create it takes some time for deployment after deployment you can see the following as shown in fig-1.5

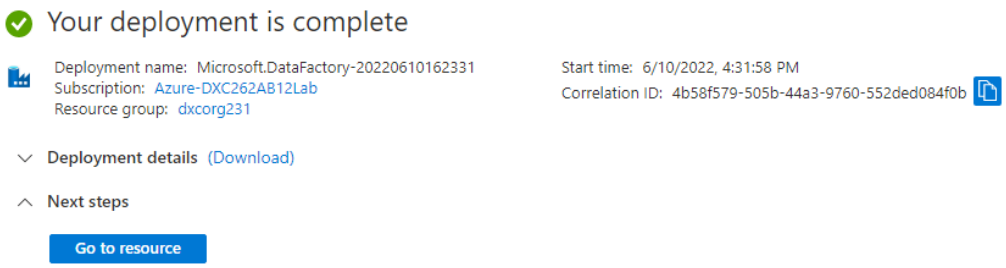


Fig1.5 shows the successful deployment of azure data factory

Step-6: click on go to resources after that click on open as shown in the figure 1.6

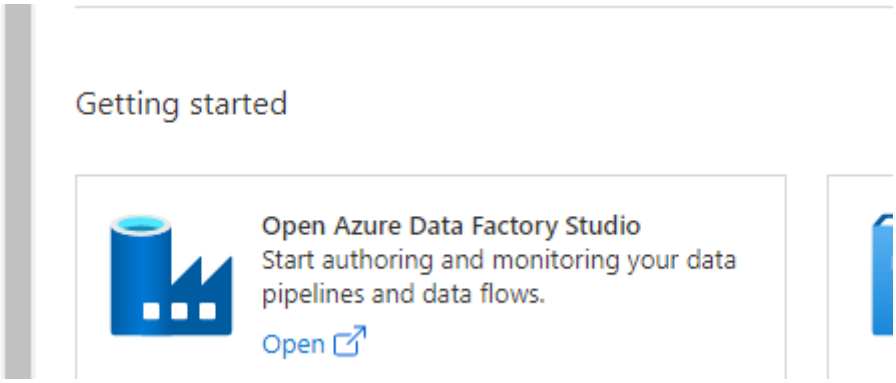


Fig-1.6 helps to navigate to the DF studio

Step-7: After clicking over that it will open azure data factory in new window as mentioned in fig 1.7 and we can use this creation of pipelines

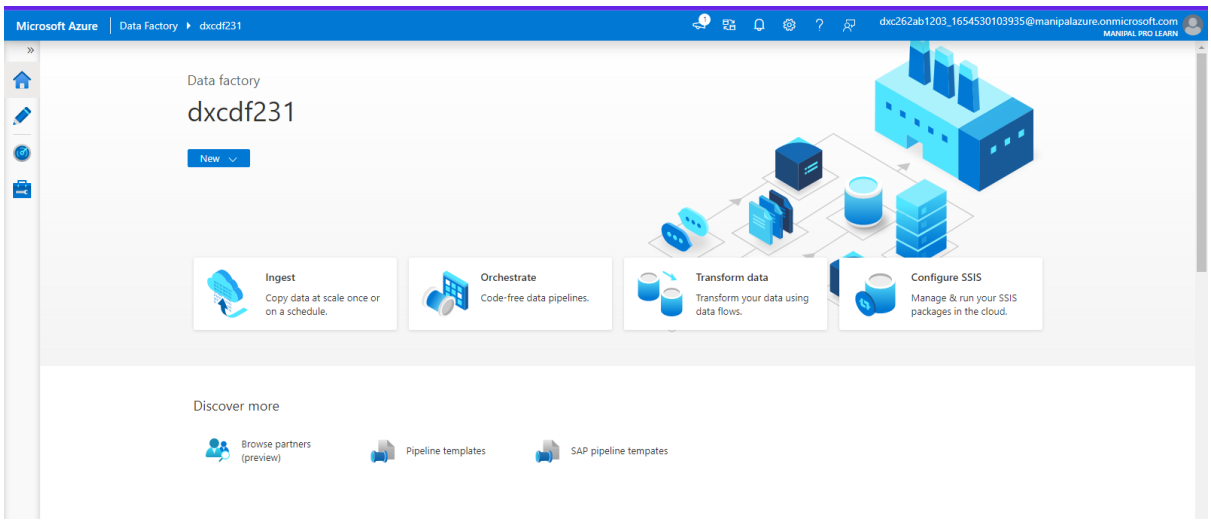


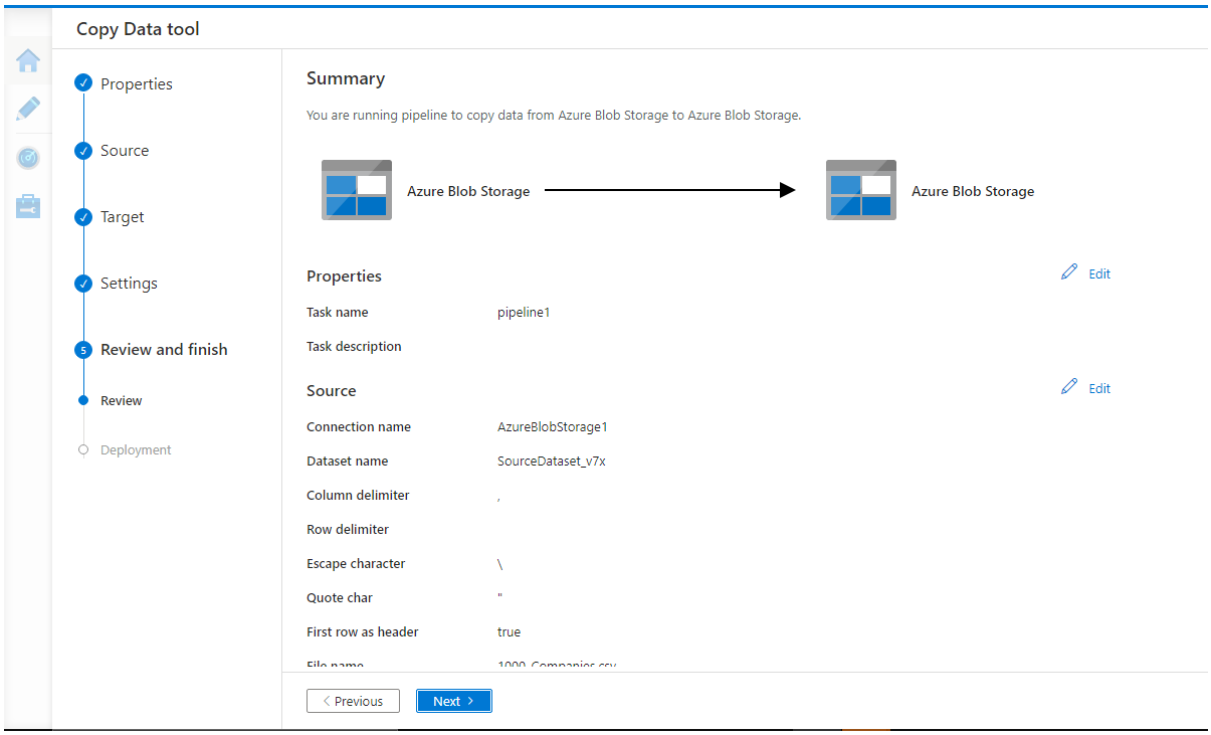
Fig-1.7 shows the main page of DATA FACTORY

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

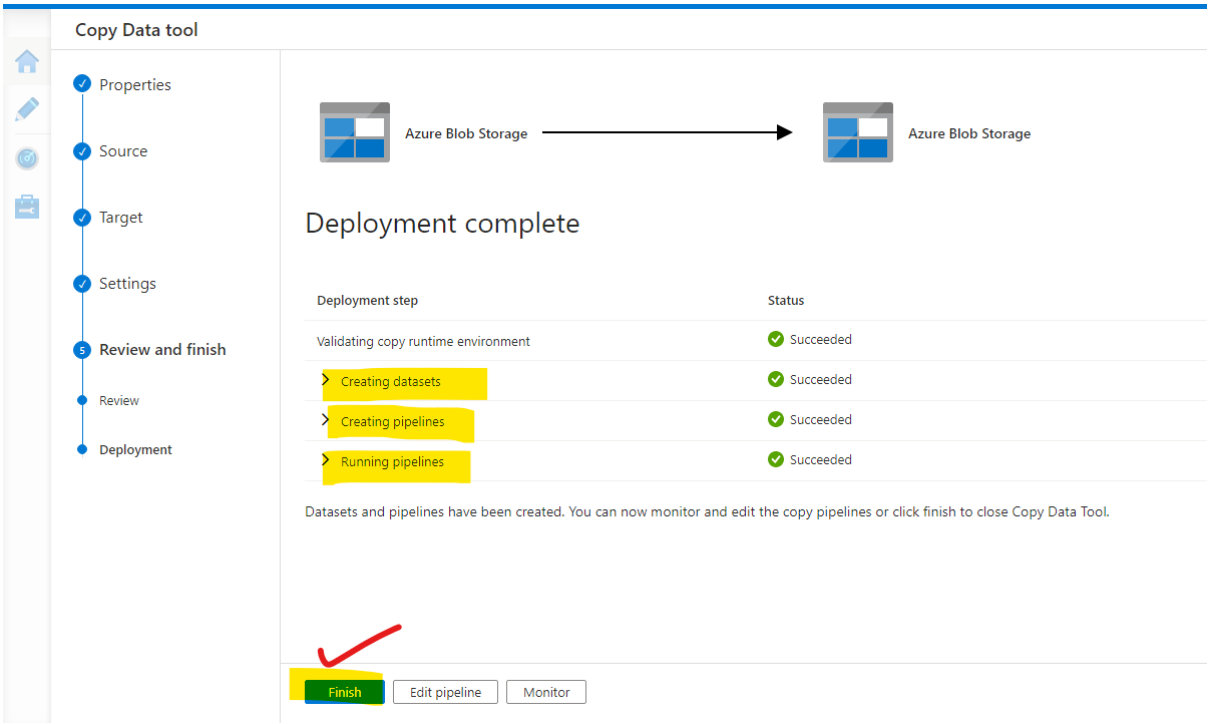
In Azure data factory we use pipe lines for channeling the source to the correct destination which are having set of constraints. In order to create an end-to-end pipe line along with triggers enabled we have to follow the below mentioned steps

Step-1: As per our project the data is in json format we will get data from aws and we are storing data in file and we are moving this data through the source to the destination in our case it is

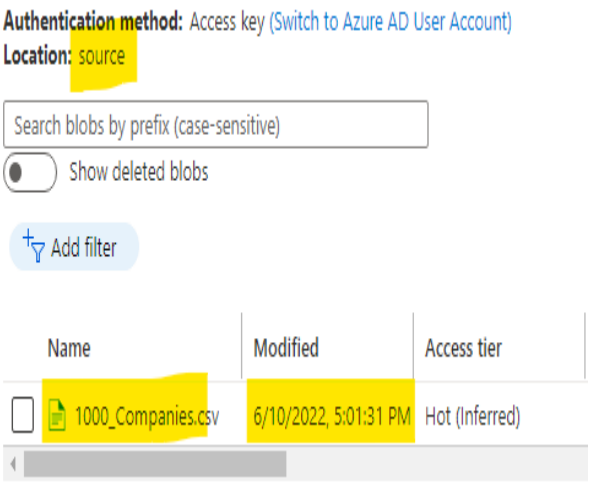
Step-2 : we are moving the data from the source to the destination using the copy data tool and creating a pipeline

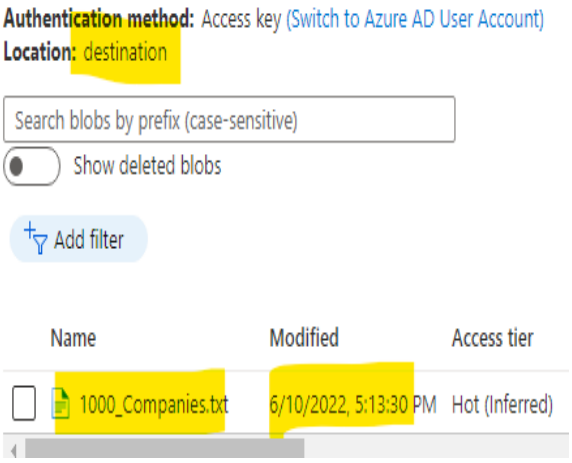


Step-3: The validation and deployment is done and pipeline is created successfully.



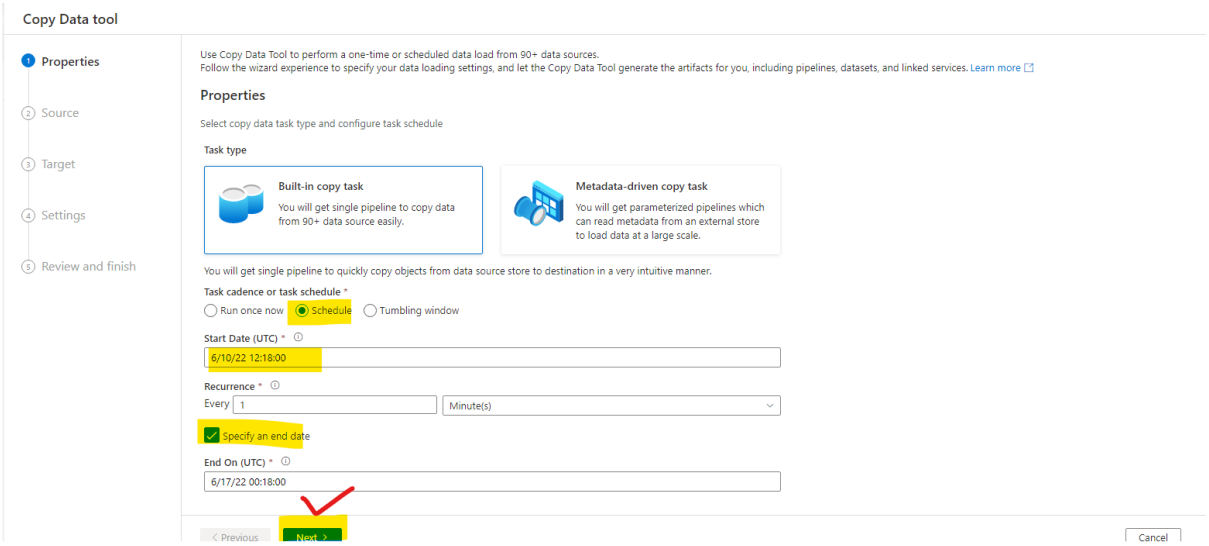
Step-4: The data is successfully copied from source to destination.



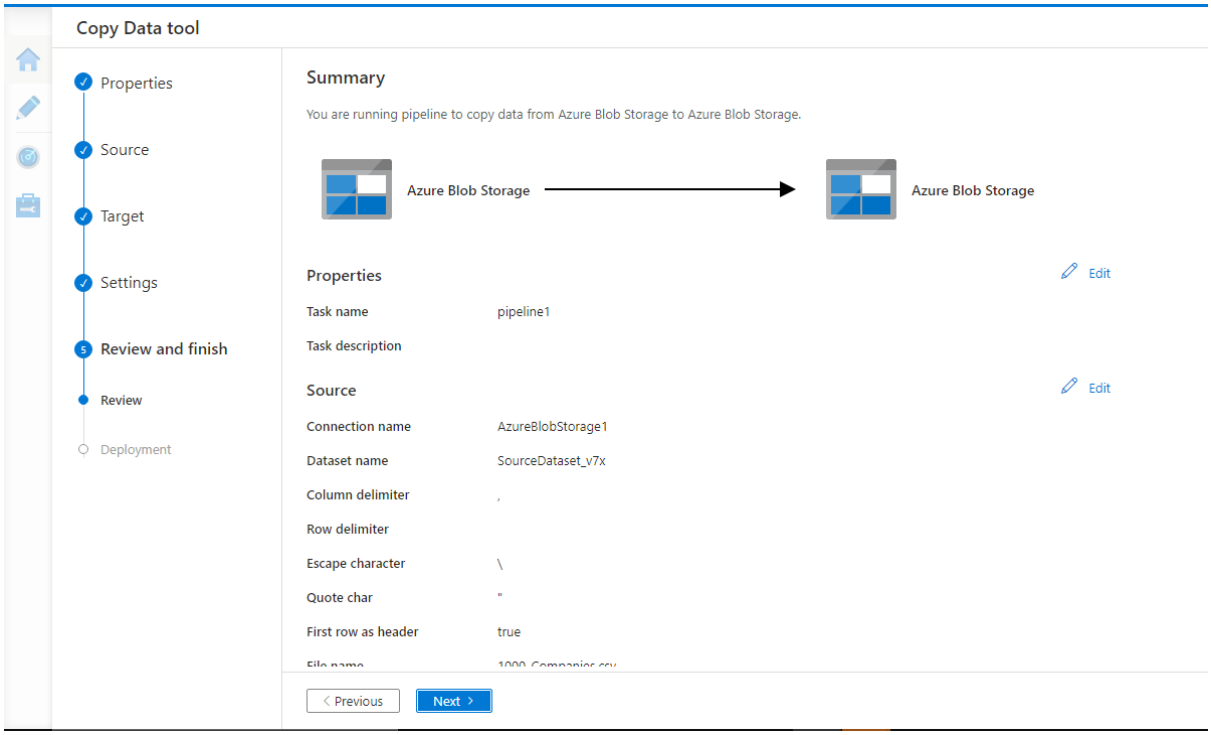


Practical Lab: Create **Azure blob trigger** logic

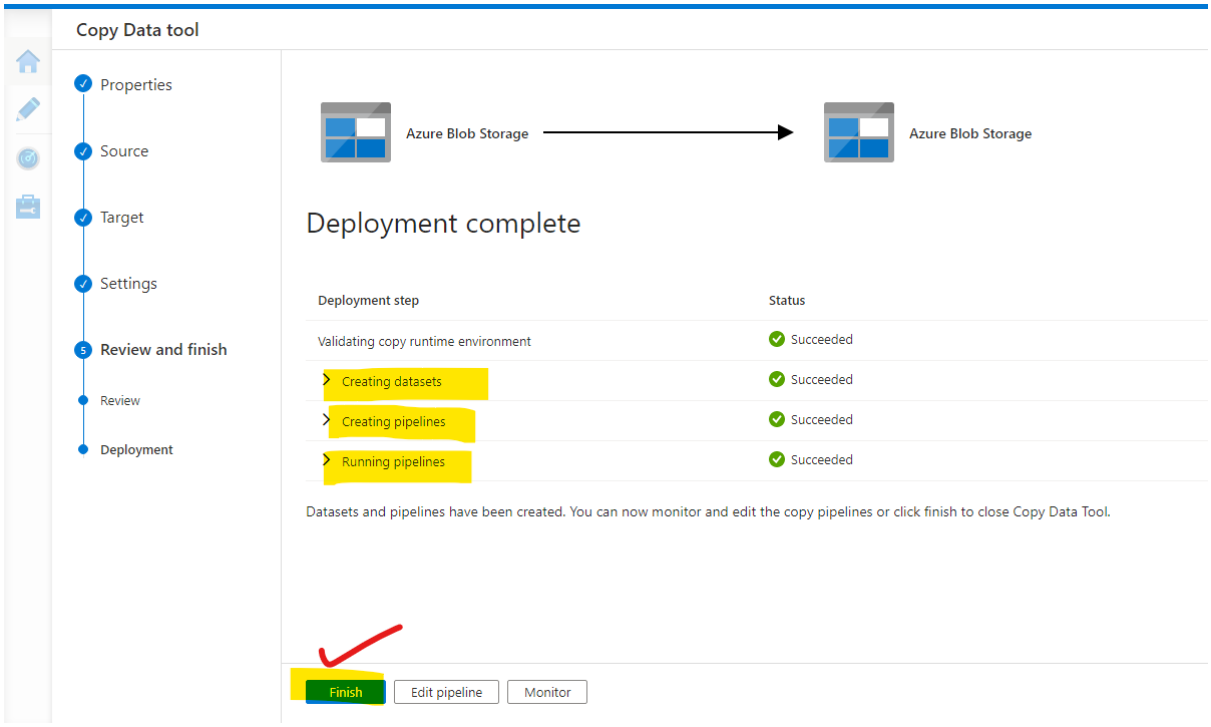
To create a azure blob trigger we have to have to follow the below mentioned steps. We have to keep an idea why we are triggering the data is the data from the device or source is to updated time to time. so in my point of view I used to update the information every minute that's why I set the trigger schedule for every minute.



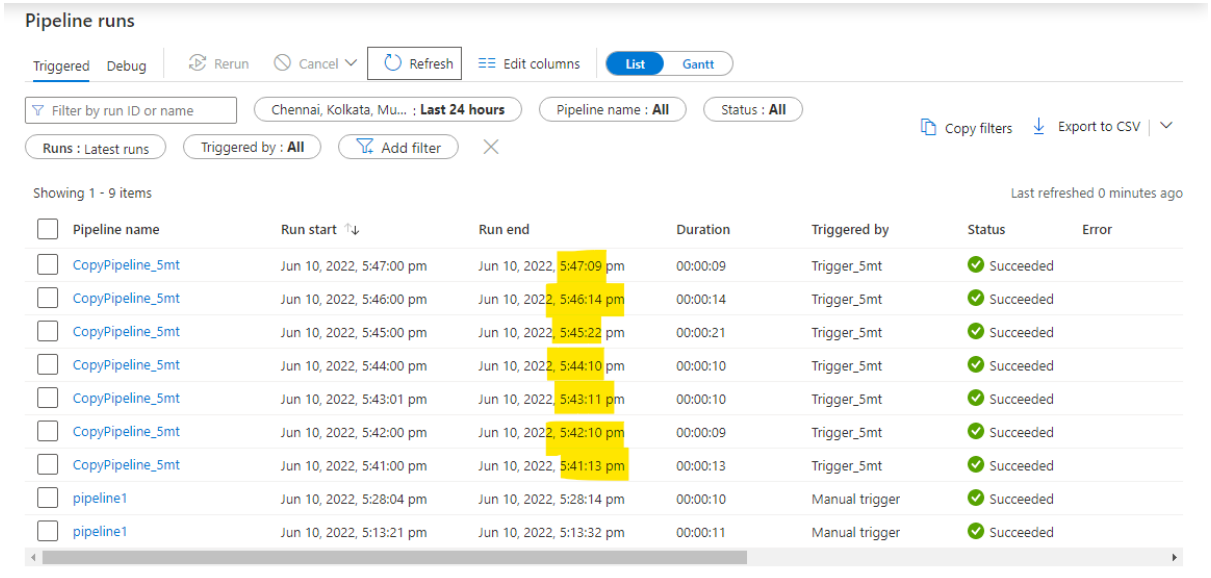
After that we have to follow the steps for setting source and destination for the pipeline



After the pipeline is created wait for deployment and after deployment click on finish



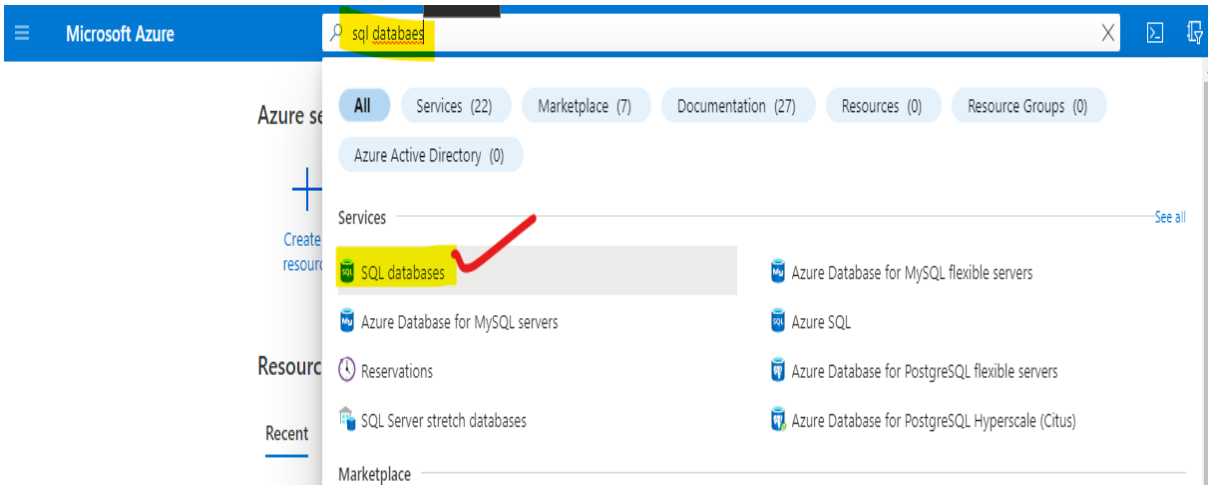
And the pipeline starts data to trigger and below figure shows the triggered runs of the blob storage.



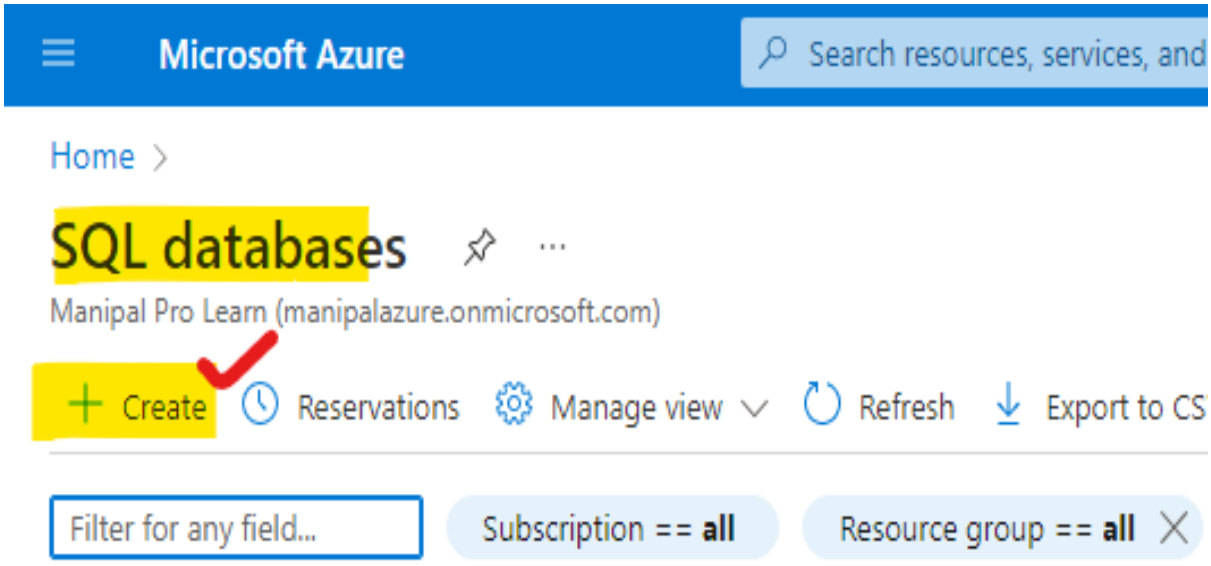
Practical Lab: Create Azure **SQL Server and Database**

To create azure SQL SERVER & DATABASE, we have to follow the steps which are mentioned below

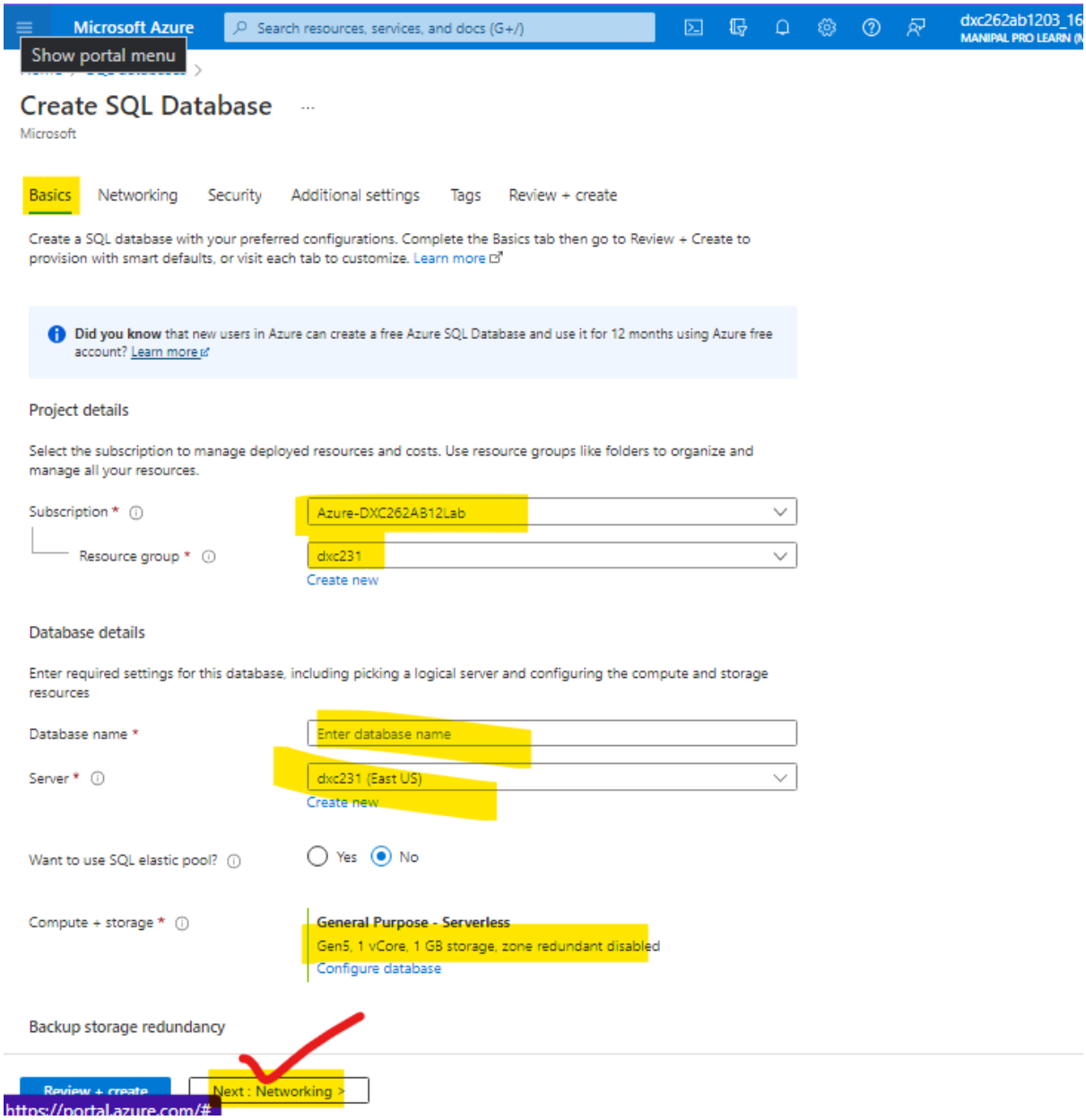
Step-1: search for azure sql database in azure portal and select it and click on create.



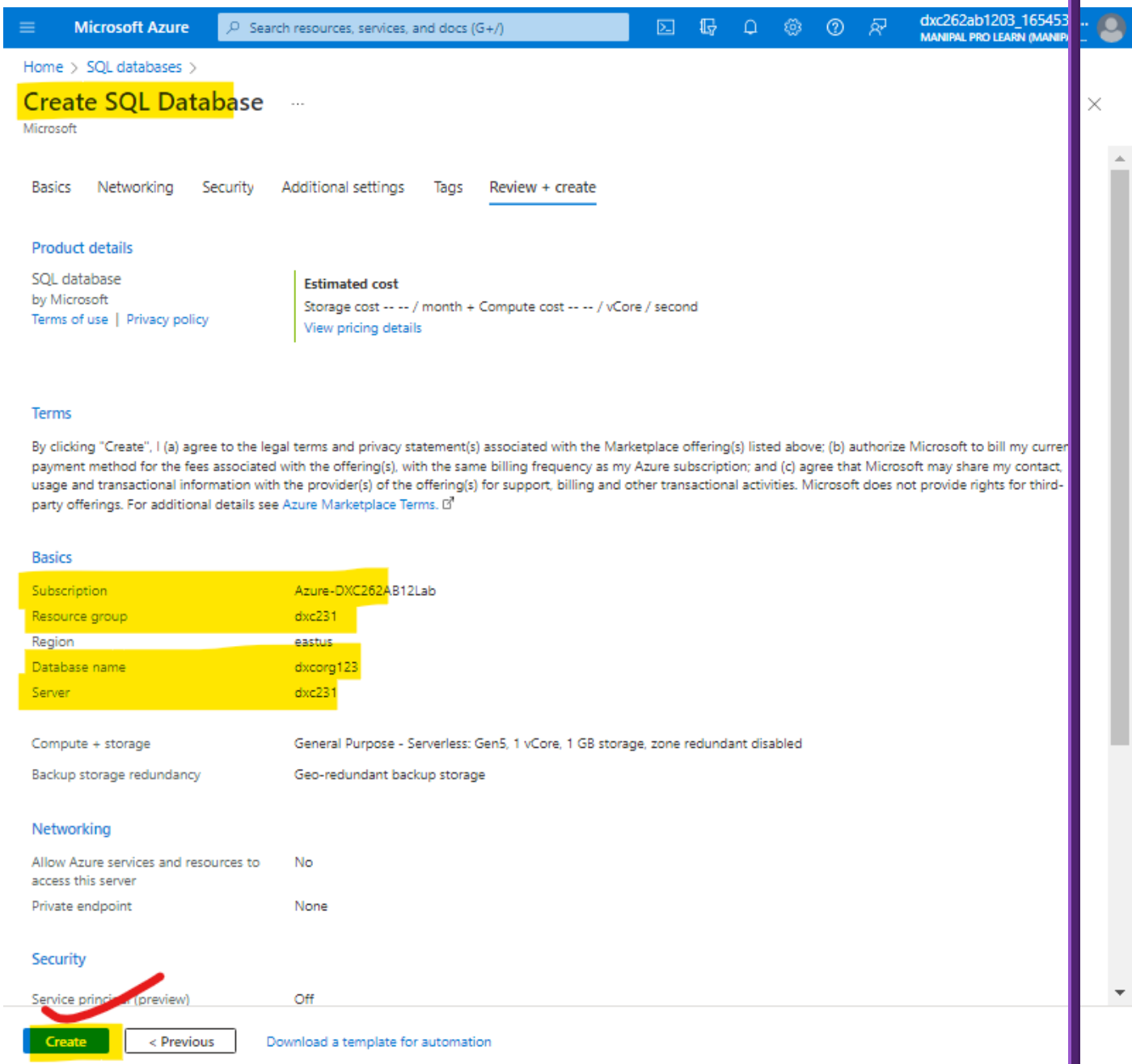
Step-2: After selecting the Sql databases. Click on create button as shown below



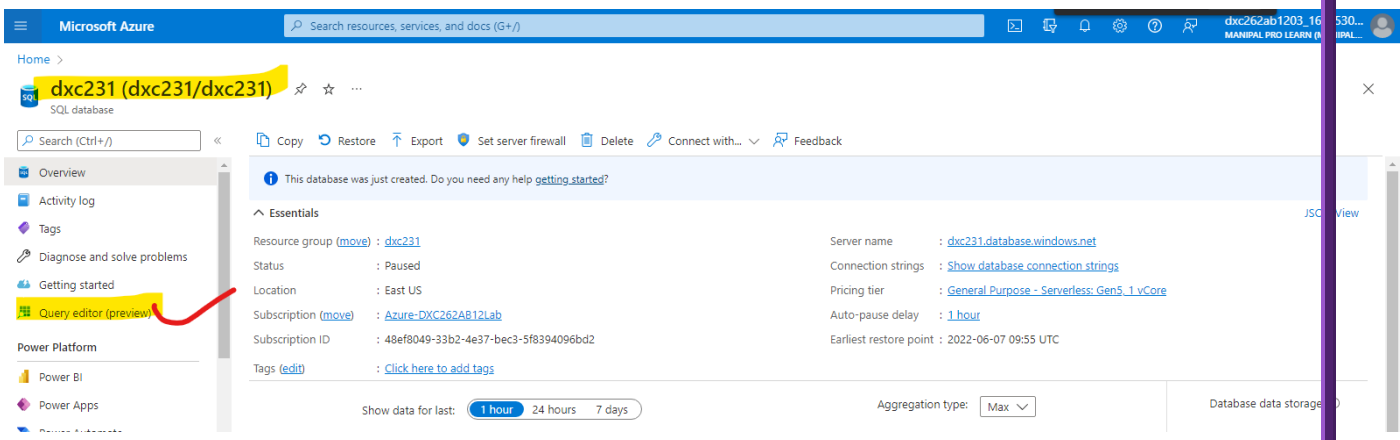
Step-3: select all options as mentioned below



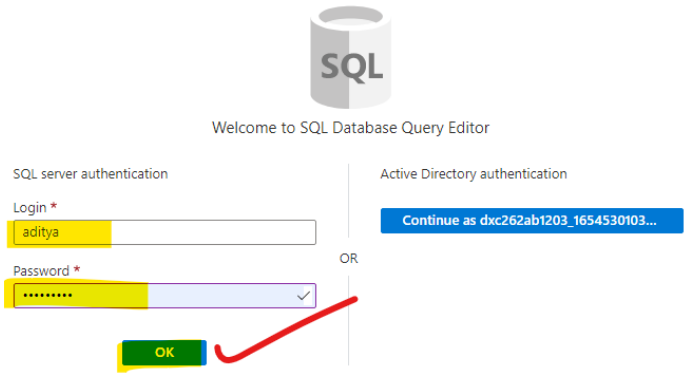
Step-4: after completing this step navigate to the next menus without changing any settings and click on create



Step-5: after clicking on create it takes some time to deploy after that we can access the Database and click on query editor



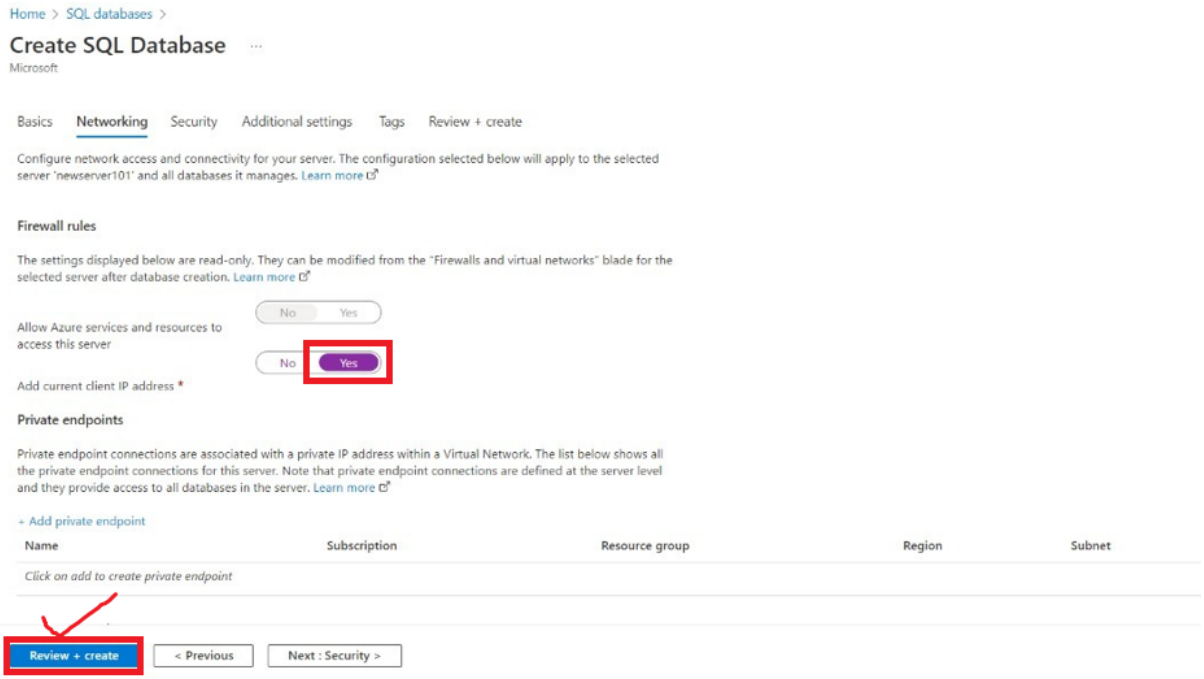
Step-6: login with your login credentials and creation of Sql data base and server is done.



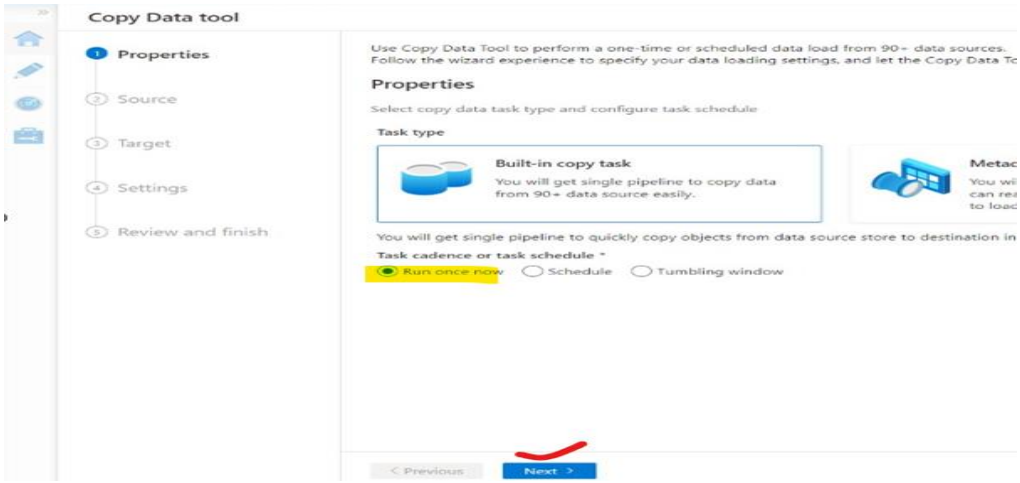
Practical Lab: Add other pipelines for moving data from Staging to **SQL DB**

in this lab we are doing to perform the data from destination to the sql data base to perform this we have to link the blob trigger to the sql data base to perform that task we have to follow the below mentioned steps

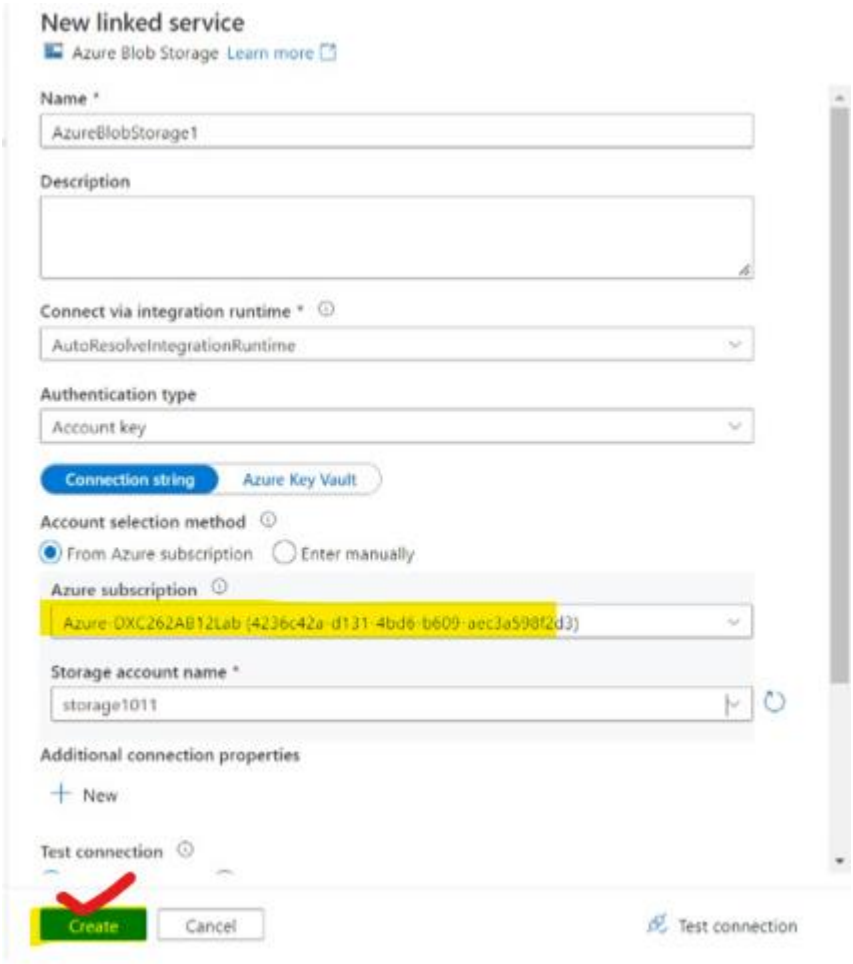
Note: to perform this task we need to enable firewall for the current IP and for the same azure cloud otherwise connecting sql database through data factory is difficult.



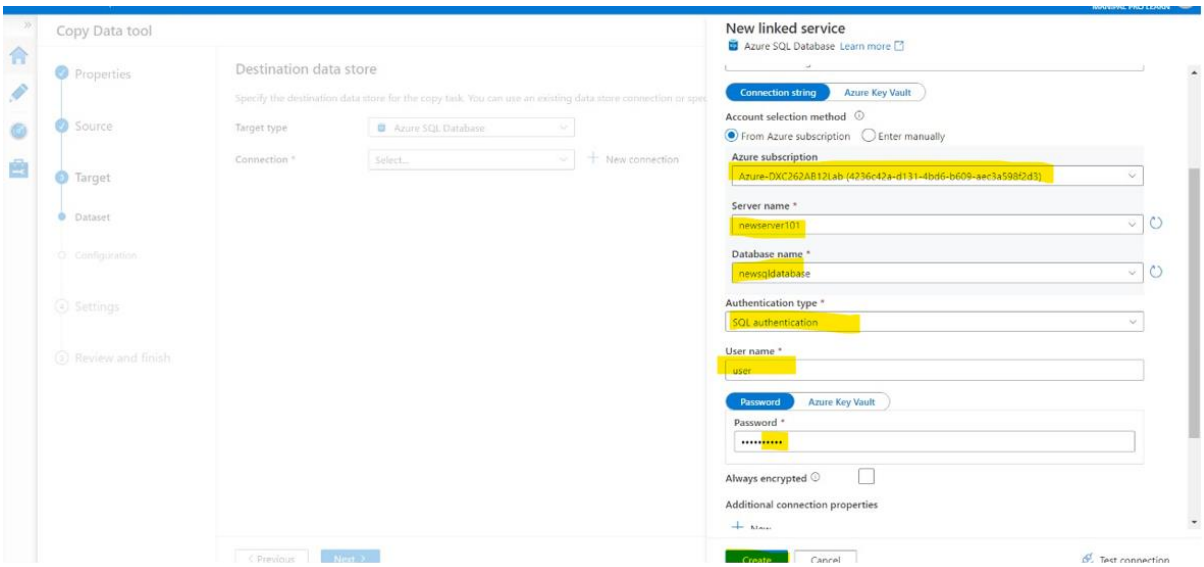
Step-1: go to azure data factory and perform the action of creating a pipeline from blob storage to sql data base.



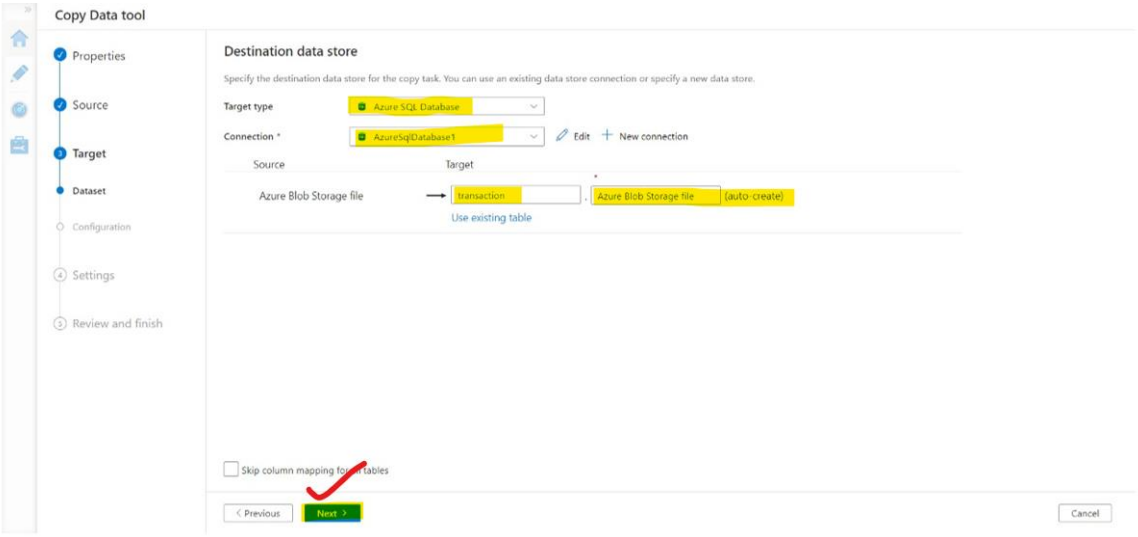
Step-2: now fill the details for the source of the pipeline that is blob storage account name and click on create button



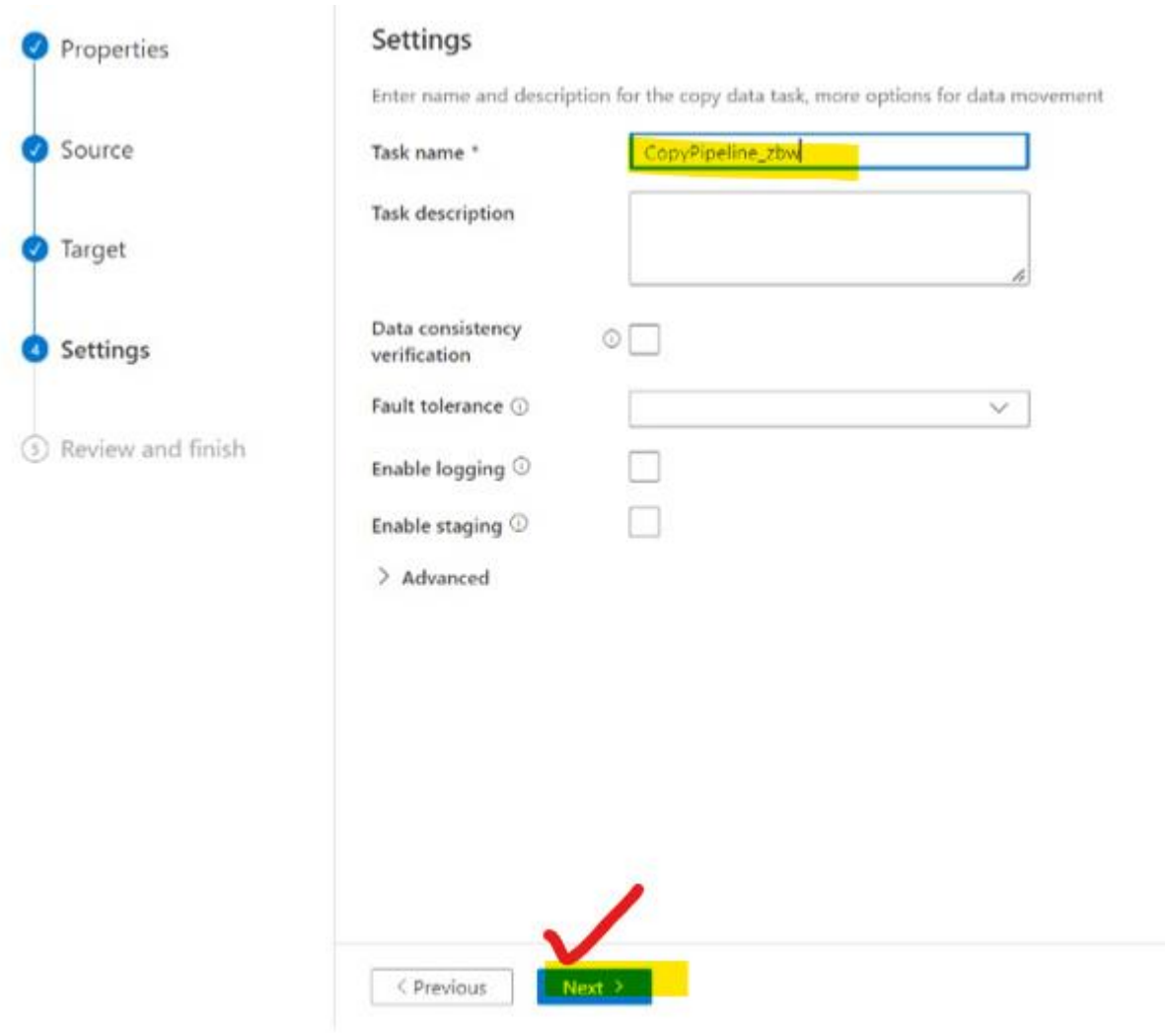
Step-3: After completing the above task we are successfully given the source location of the data and now we have to give the location of destination that is sql database.



Step-4: now, the destination is set and pipeline is ready

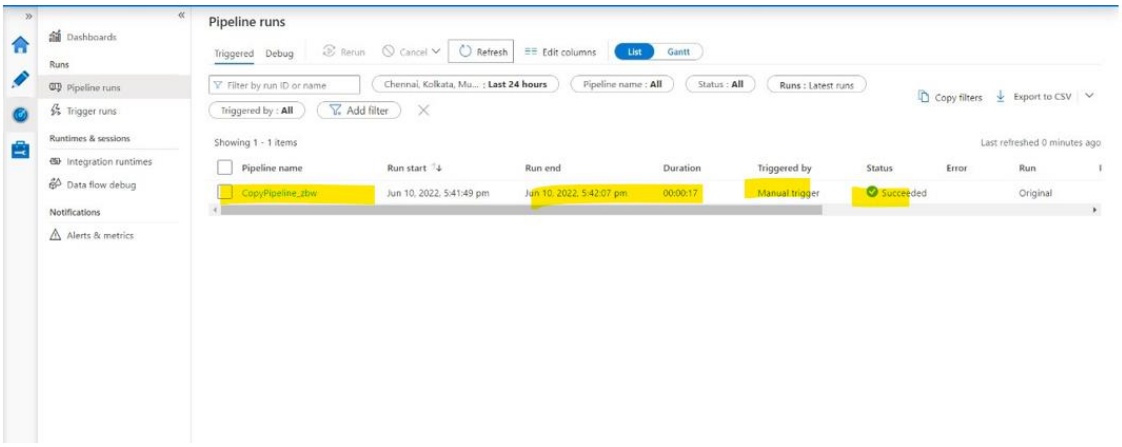


Step-5: enter the task name and description and click on next



Step-6: review all and go ahead by clicking the next button.

Step-8: now trigger the pipeline and the pipe line will be triggered



Step-9: The data was moved successfully to the sql database. Due to limited access and time constraint the lab is closed so, screen shot is not available.

Result:

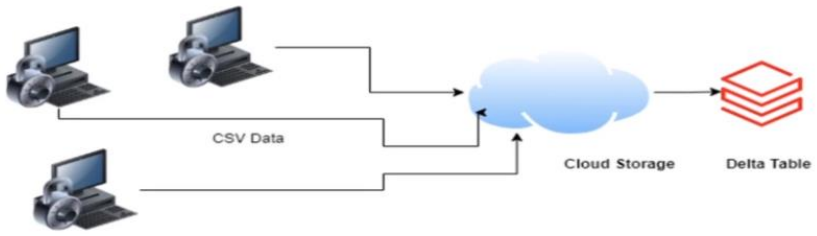
In this project we are successfully created a blob storage account, SQL database, azure data factory and used data to move from source that is AWS to Azure and the data has been moved successfully.

Conclusion: By using the resources from azure we migrated the data from AWS to azure successfully

Project 2: AP Morgan Data Platform

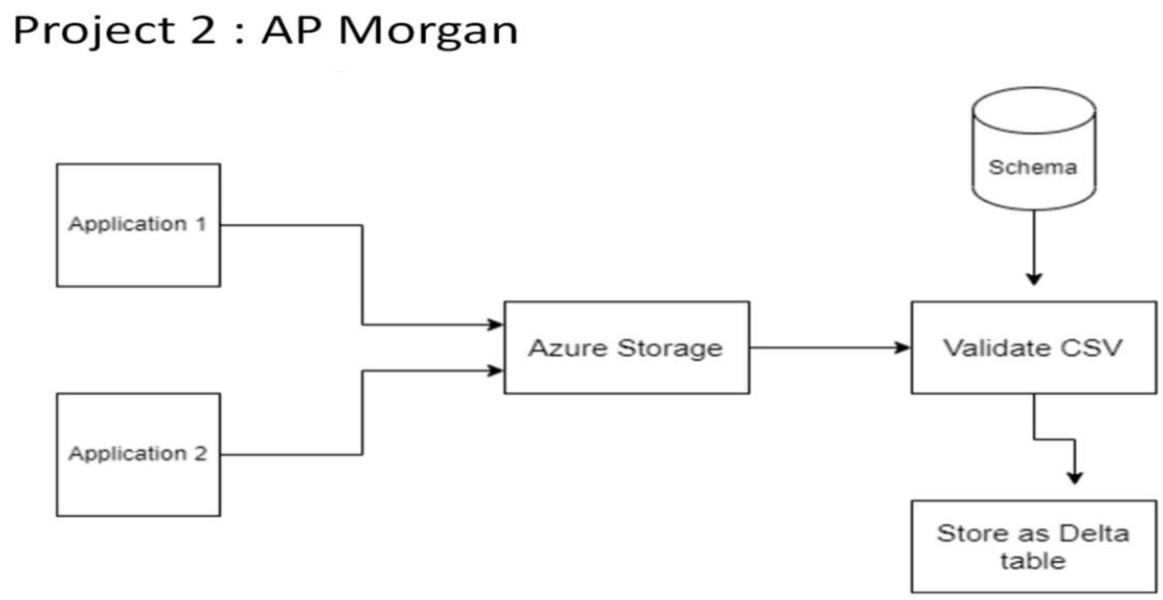
Project 2 : AP Morgan

- 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.



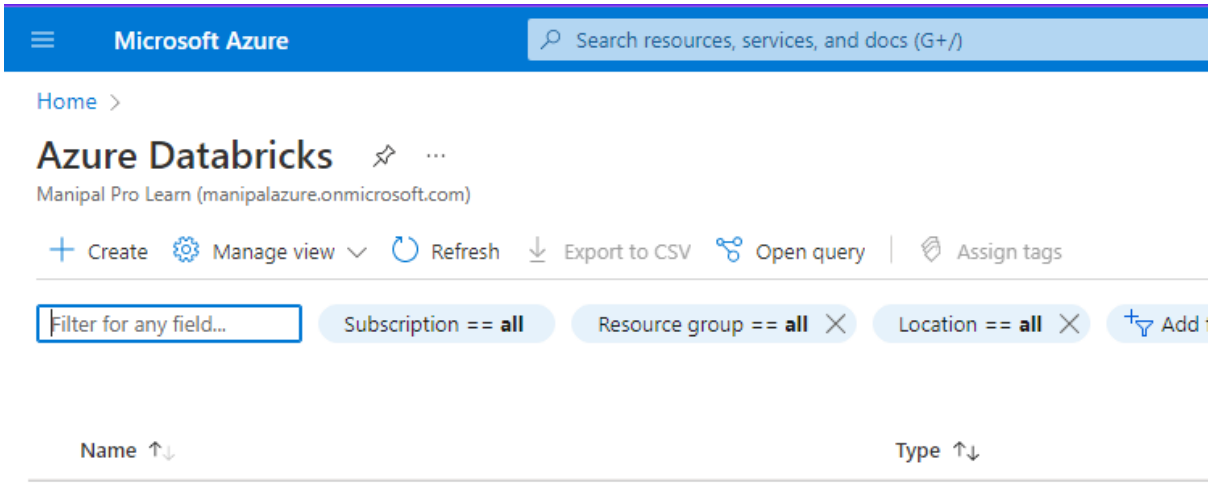
Project 2 : AP Morgan- High Level Detail

- 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

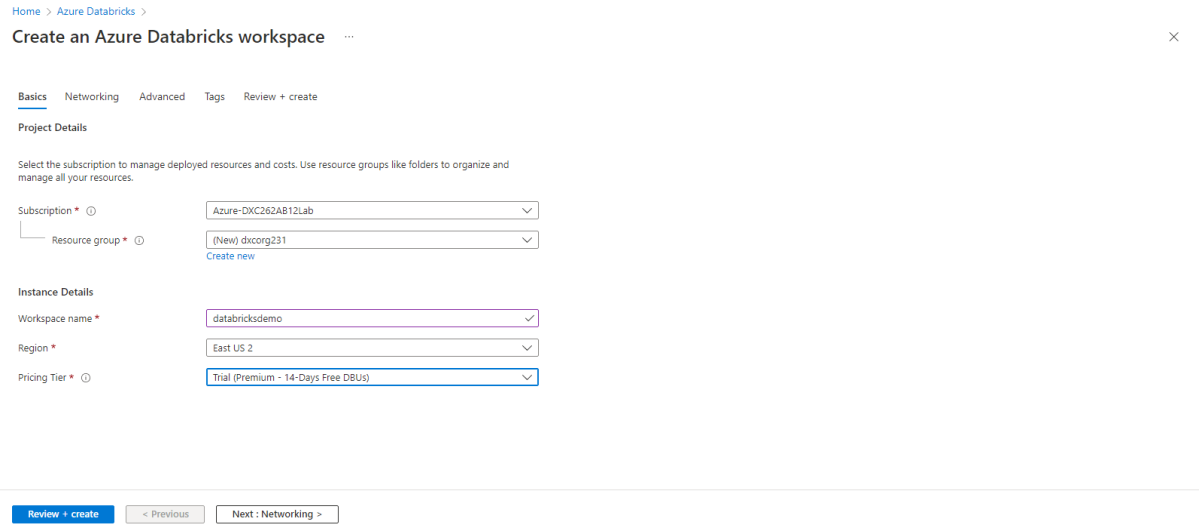


Practical Lab: Create a **Databricks**

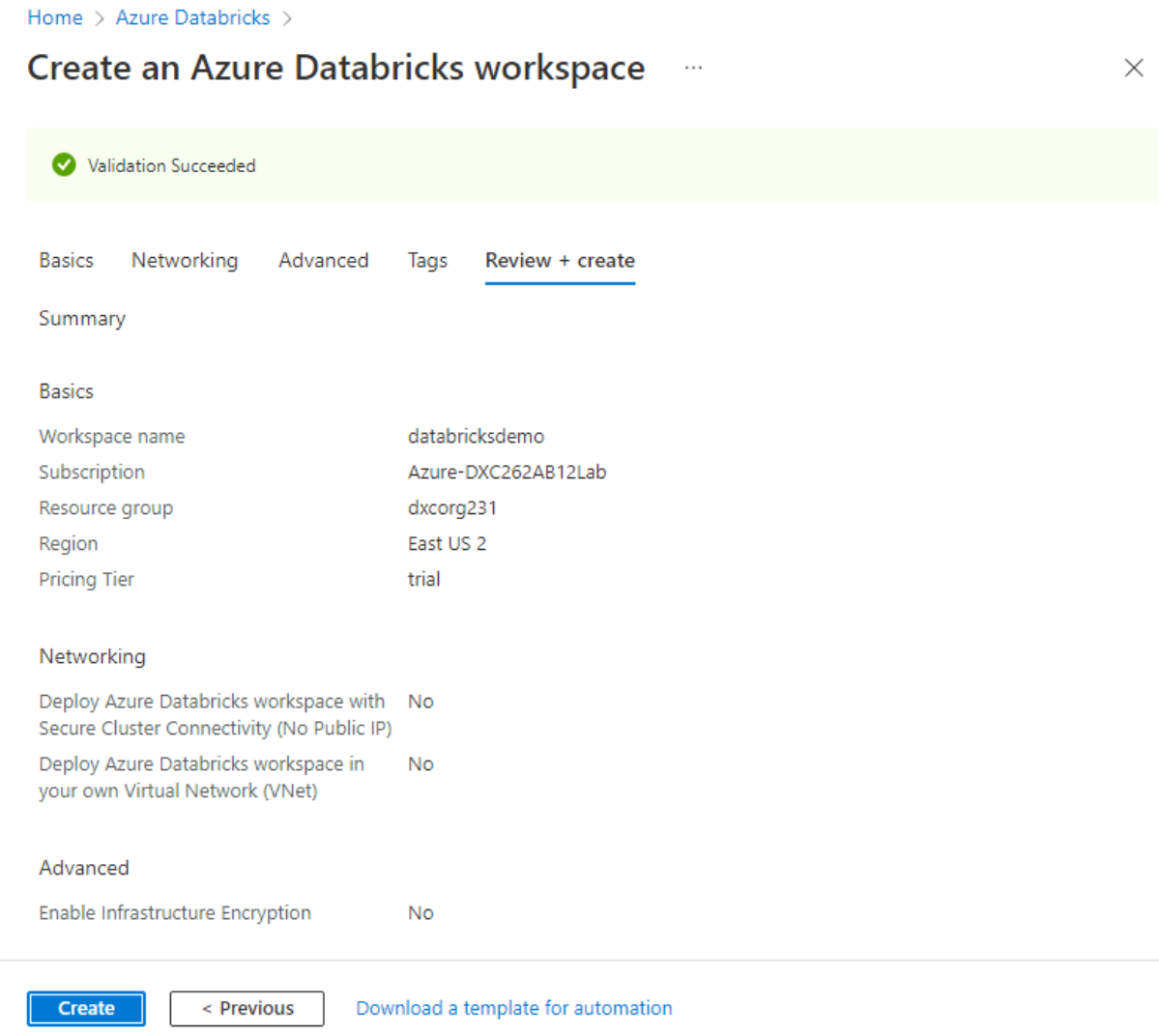
Step-1: to create data bricks we have to search for data bricks and click on create button.



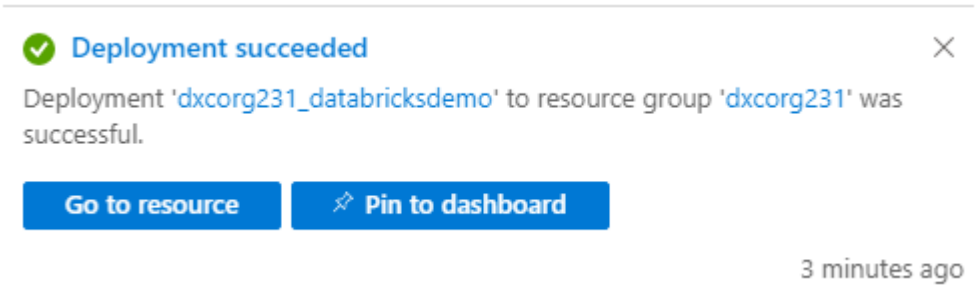
Step-2: we have to create an azure data bricks workspace



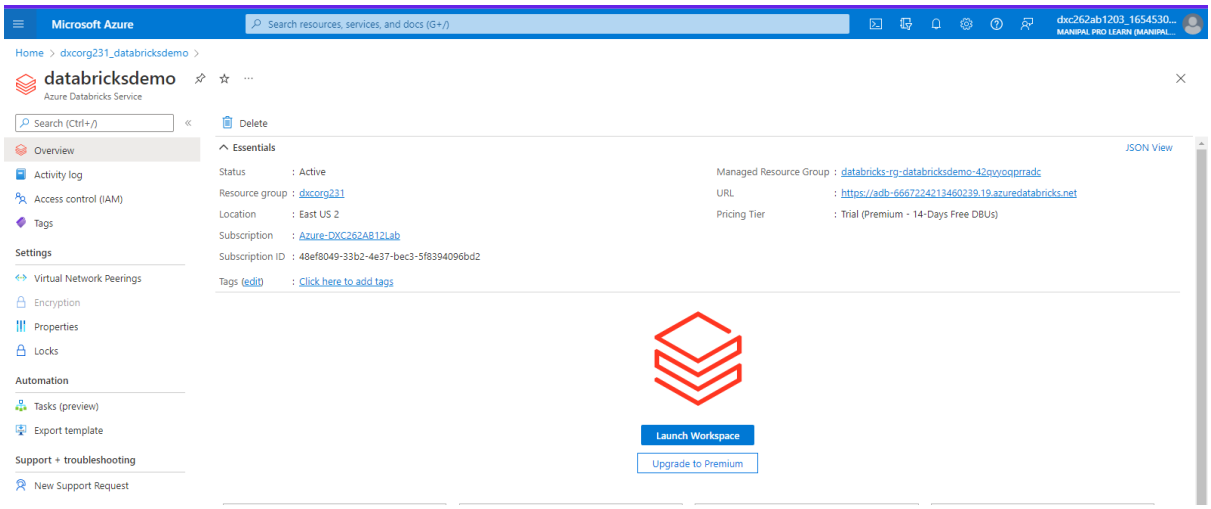
Step-3: after validation and click on create.



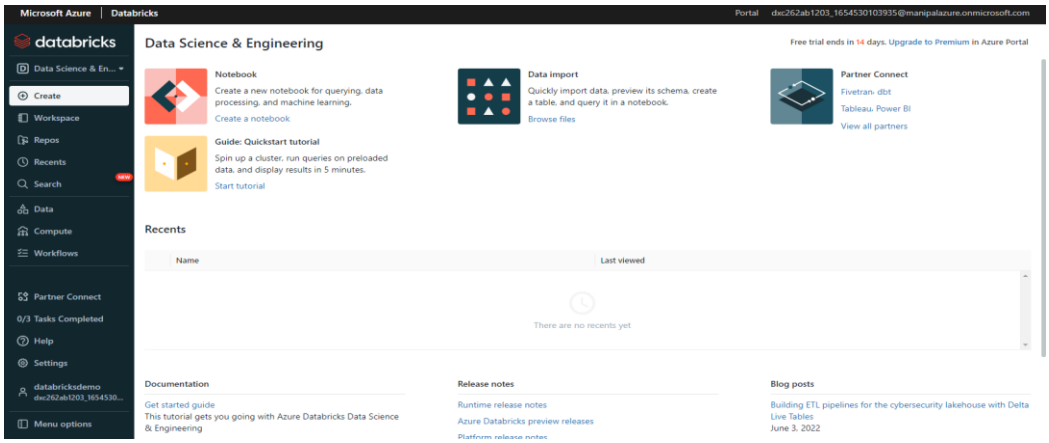
Step-4:after clicking on create wait for deployment & deployment has been done, now click on go to resource button



Step-4: this page will appear and now click on launch workspace



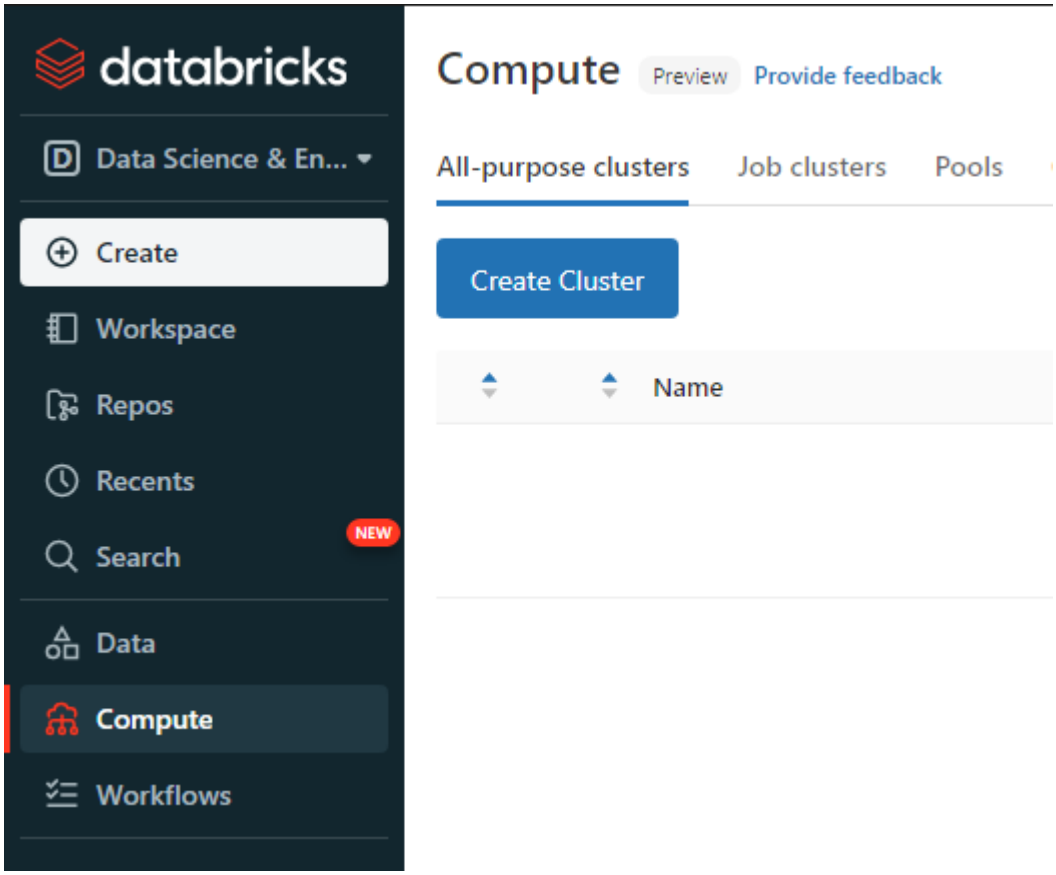
Step-5: after launching the work space we are navigated to the new tab



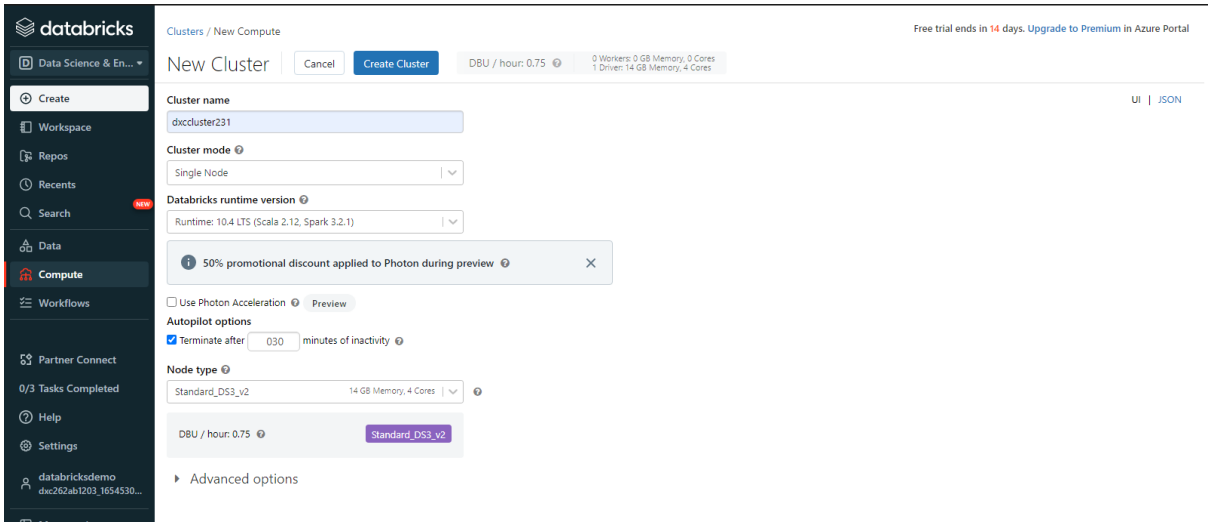
Practical Lab: Create **Cluster in Azure Databricks**

To create a cluster in the data bricks we have to follow the below mentioned steps

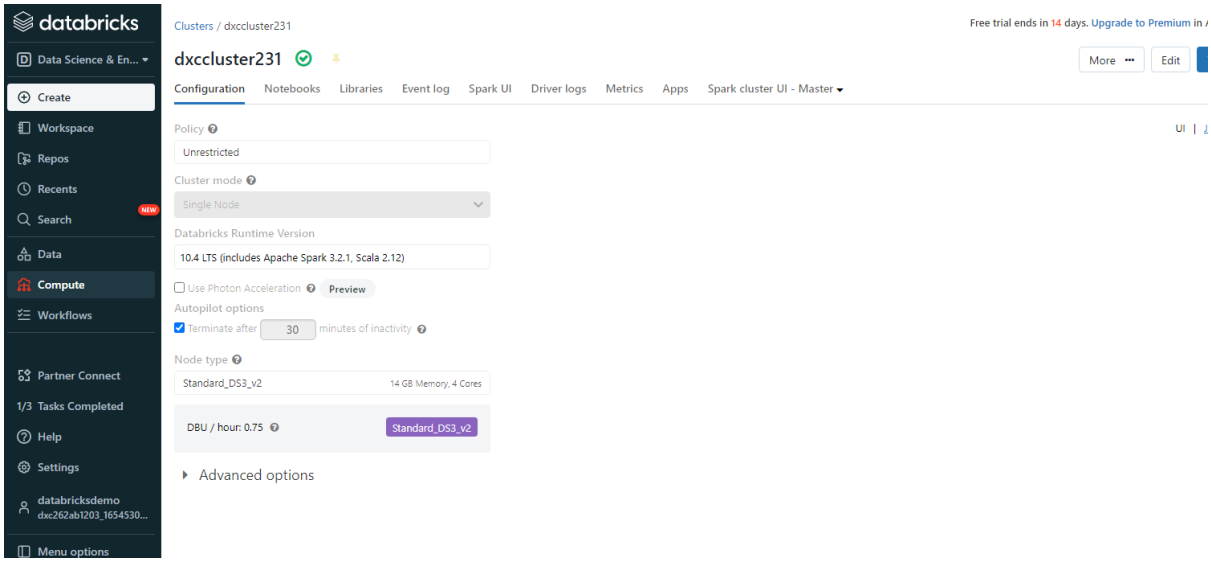
Step-1: after navigating to the home page of azure data bricks click on compute the you can find create cluster button



Step-2: name the cluster and select cluster mode and click create



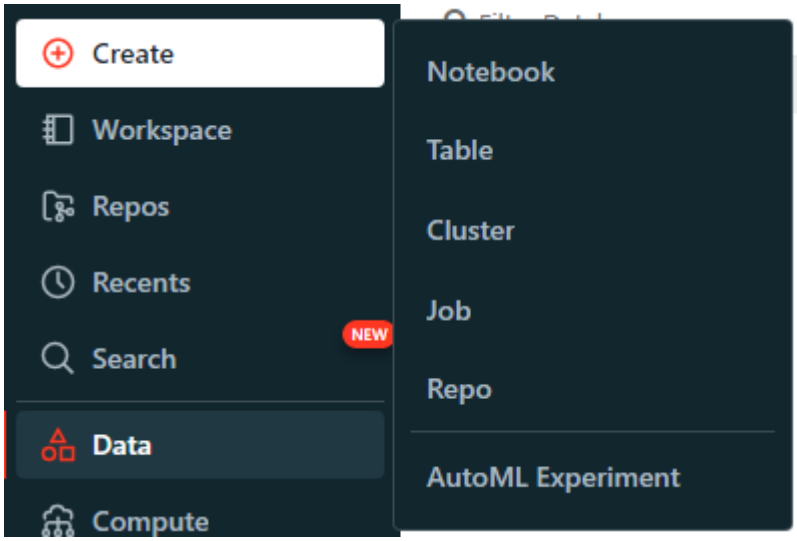
Step-3: after that the cluster is created successfully



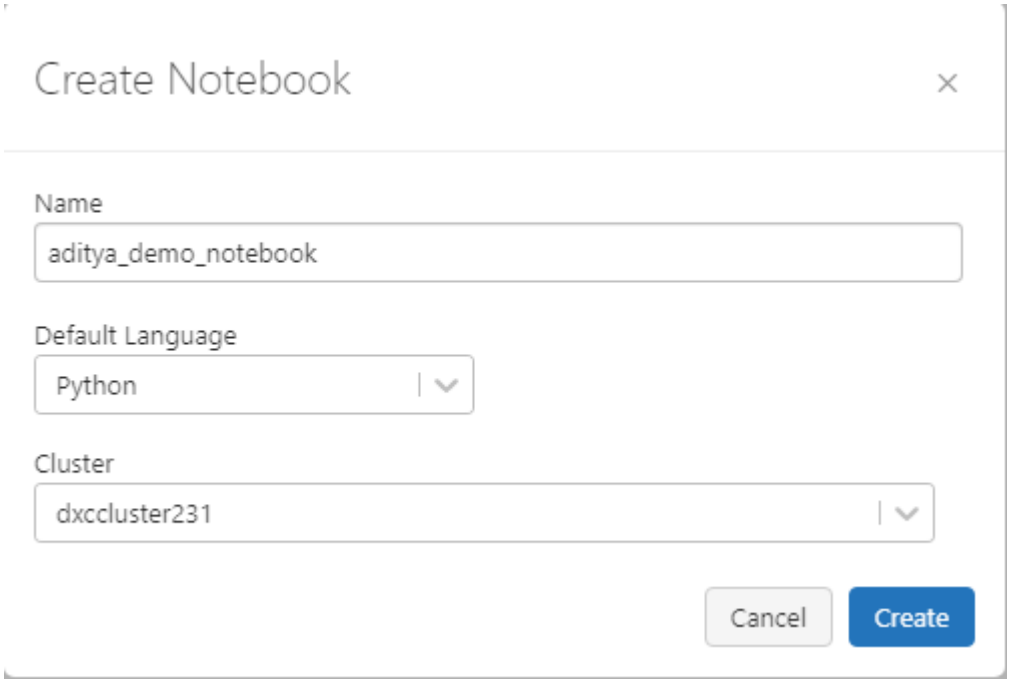
Practical Lab: **Add notebook in Databricks** and Implement the Business Logic

To add notebook in data bricks we have to follow the below mentioned steps

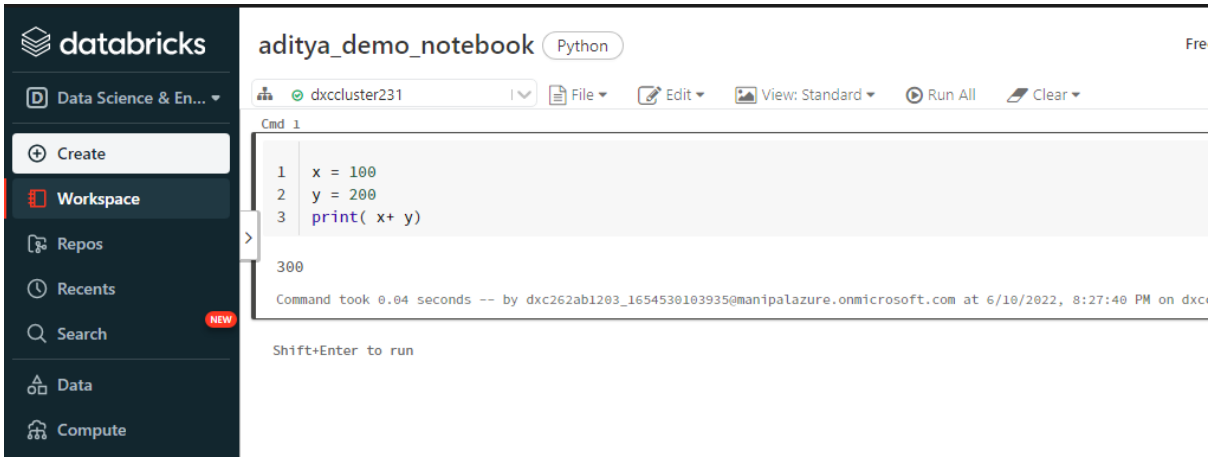
Step-1:click on create and select notebook



Step-2: name the note book, default language and cluster and click on create



Step-3: After clicking on create the note book is created

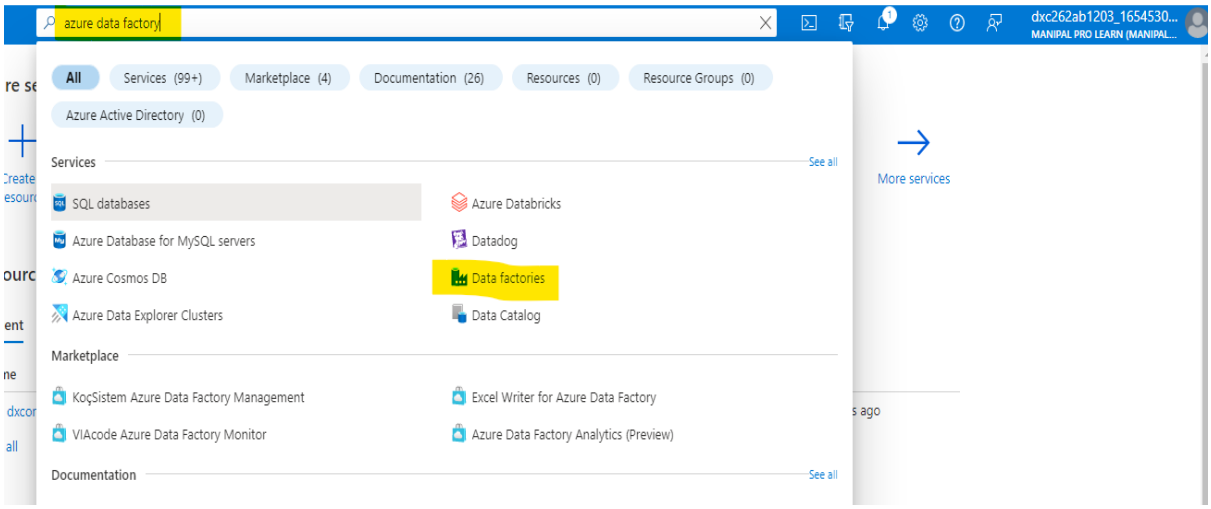


Step-4: And to implement the business logic we need to go to the partner connect and select the power bi (Business intelligence tool by Microsoft) and

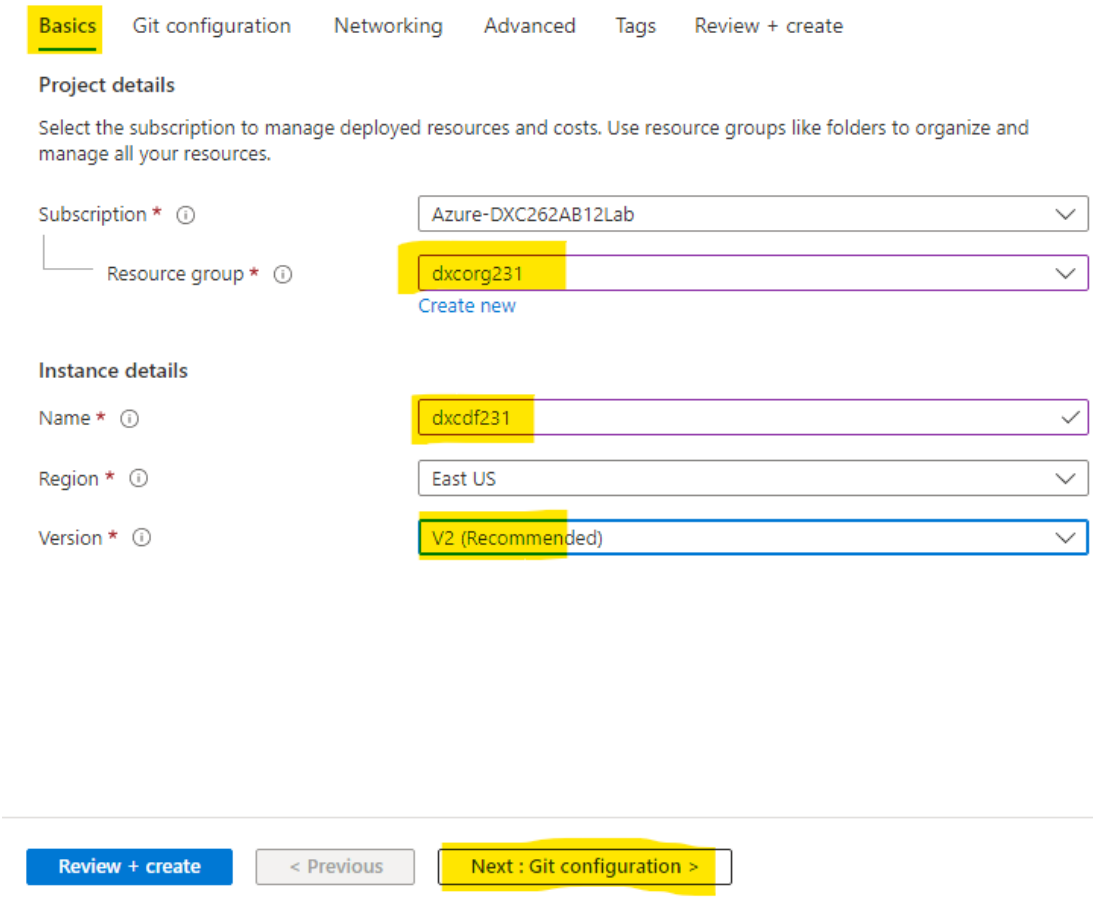
After clicking on Power BI, we have to choose the cluster and download the file to connect using desktop power bi after that go to user settings and generate token and click the file downloaded and enter the access token. And we are done with connecting azure data bricks with the power BI.

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

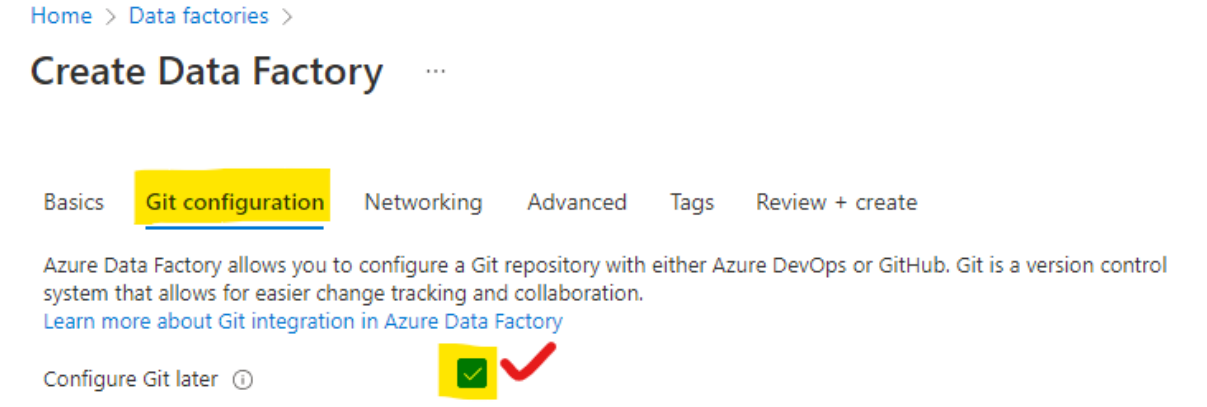
Step-1: login to the azure portal and search for azure data factory as shown in fig



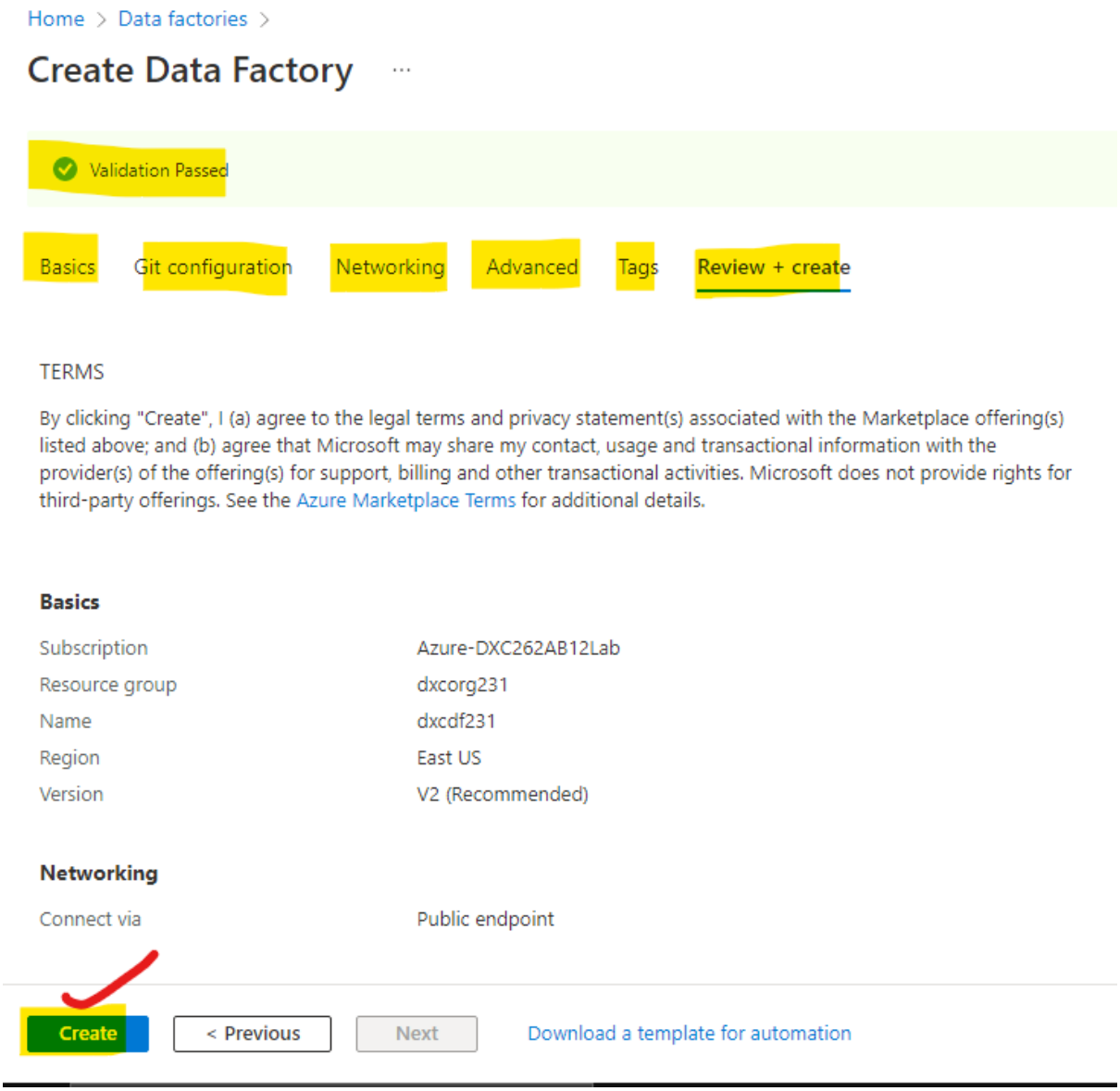
Step-2: after navigating to the azure data factory page click on create and select the resource group, name, region, version after that click on next follow the fig



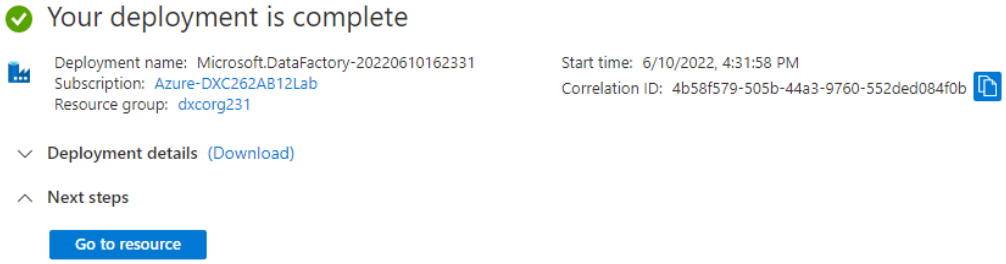
Step-3: click on git configurations and checkbox it as configure git later as shown in fig



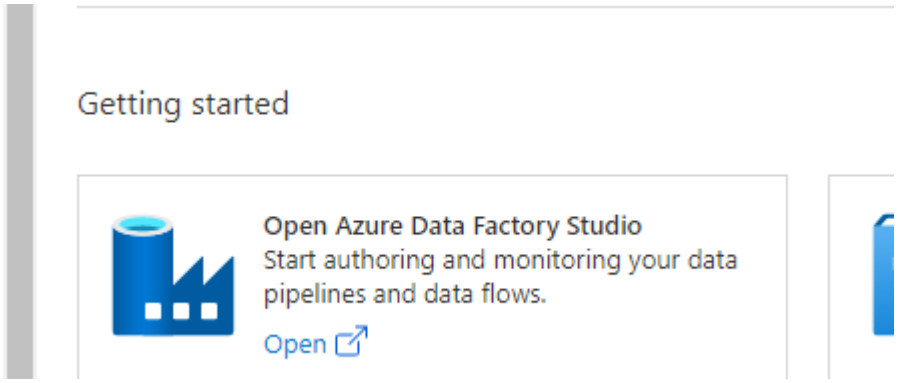
Step-4: Go through the next steps followed by succesfull completion of validation clikc on create as shown in



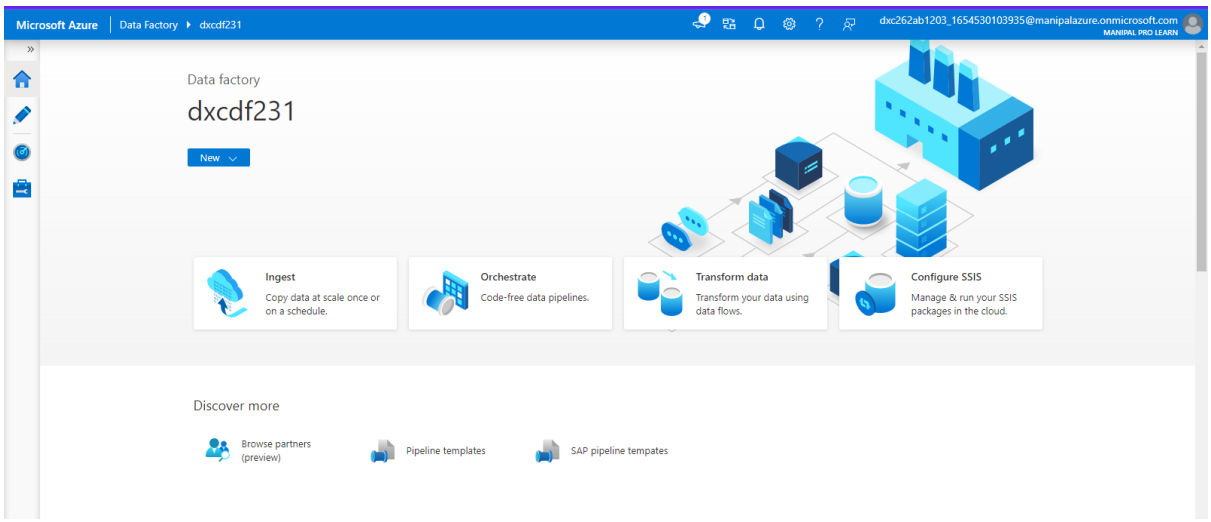
Step-5: after clicking on create it takes some time for deployment after deployment you can see the following as shown in fig



Step-6: click on go to resources after that click on open as shown in the figure



Step-7: After clicking over that it will open azure data factory in new window as mentioned in fig



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

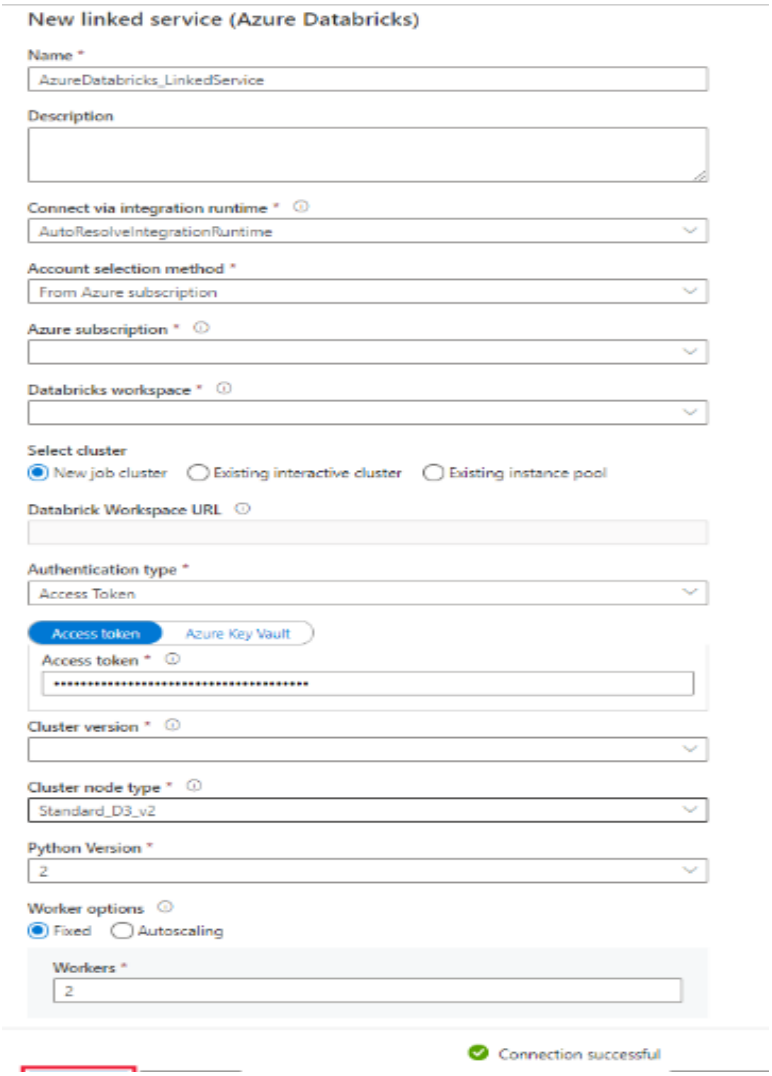
To Create azure data bricks linked service in azure data factory we have to follow the below mentioned steps

Step-1: select integration and then select data factory

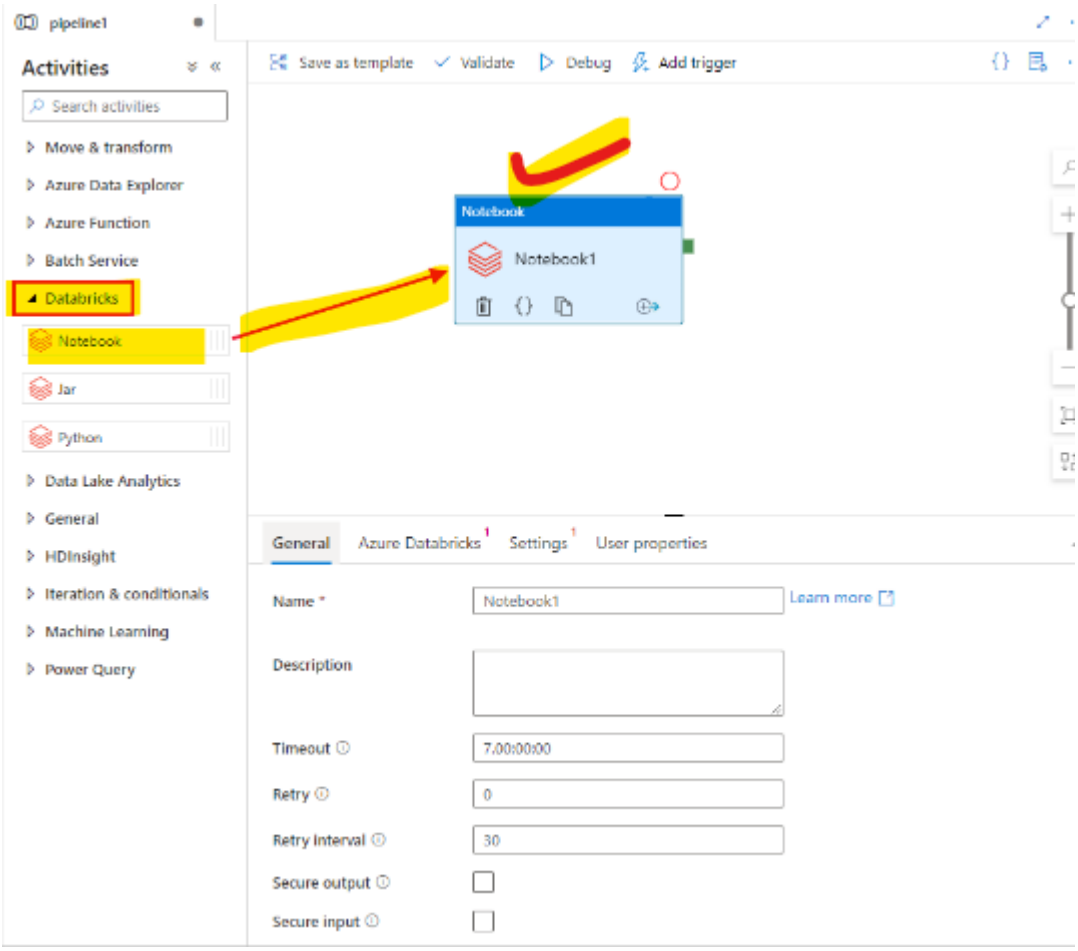
Step-2: now create azure data factory

Step-3: Now create linked services and click on new

Step-4: select compute and azure data bricks and select continue

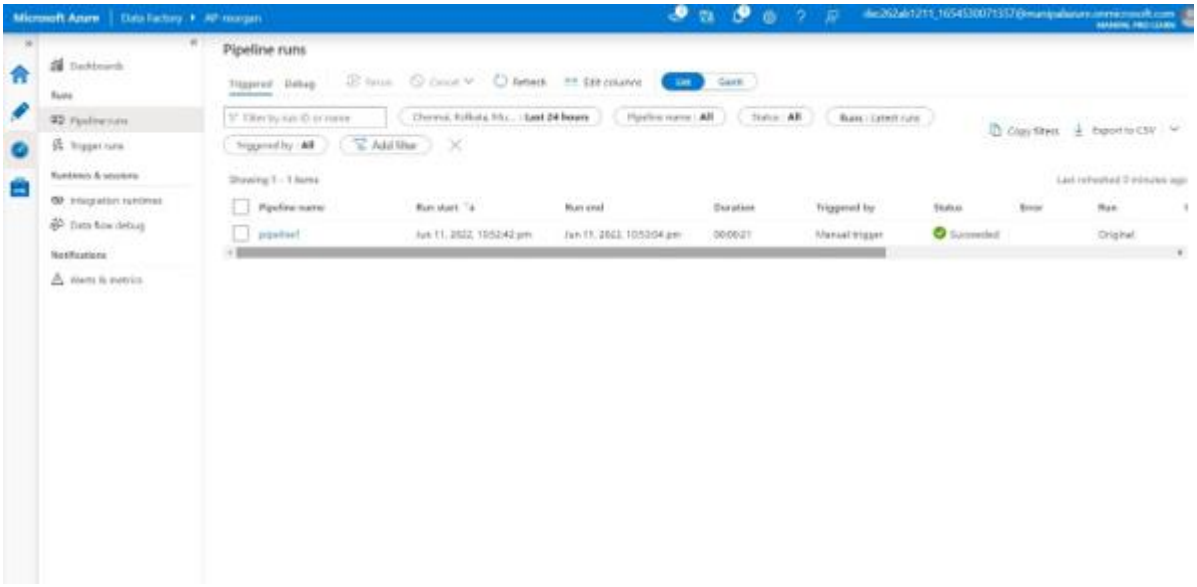


Step-5: create a pipe line and integrate it with the note book



Step-6: now open data bricks and click on workspace and click on create and create a note book

Step-7: now do any action in the note book and perform trigger.



Result: The data from the different application are stored in azure storage and then into the data bricks and we can now access the tables and the data is stored in the form of delta tables.

Conclusion: By using the resources like Data bricks, Clusters, Notebooks,

And created azure data factory for Ap Morgan and linked Azure data bricks and azure data factory

References:

- https://azure.microsoft.com/en-in/blog/operationalize-azure-databricks-notebooks-using-data-factory/
- https://docs.microsoft.com/en-us/azure/data-factory/transform-data-using-databricks-notebook