Project-1 (Data Engineering Batch-1)

Ingestion and Transformation of Olympics Data

Vuthur Sriganga

Hexaware Data Engineering Batch-1

vuthursriganga@gmail.com

Ingestion and Transformation of Olympics Data

Title:

Data Ingestion: Configure and run the ADF pipelines to ingest the Olympics data into Azure Data Lake Storage Gen2.

Data Transformation: Execute the PySpark notebooks in Azure Databricks to perform data transformation tasks.

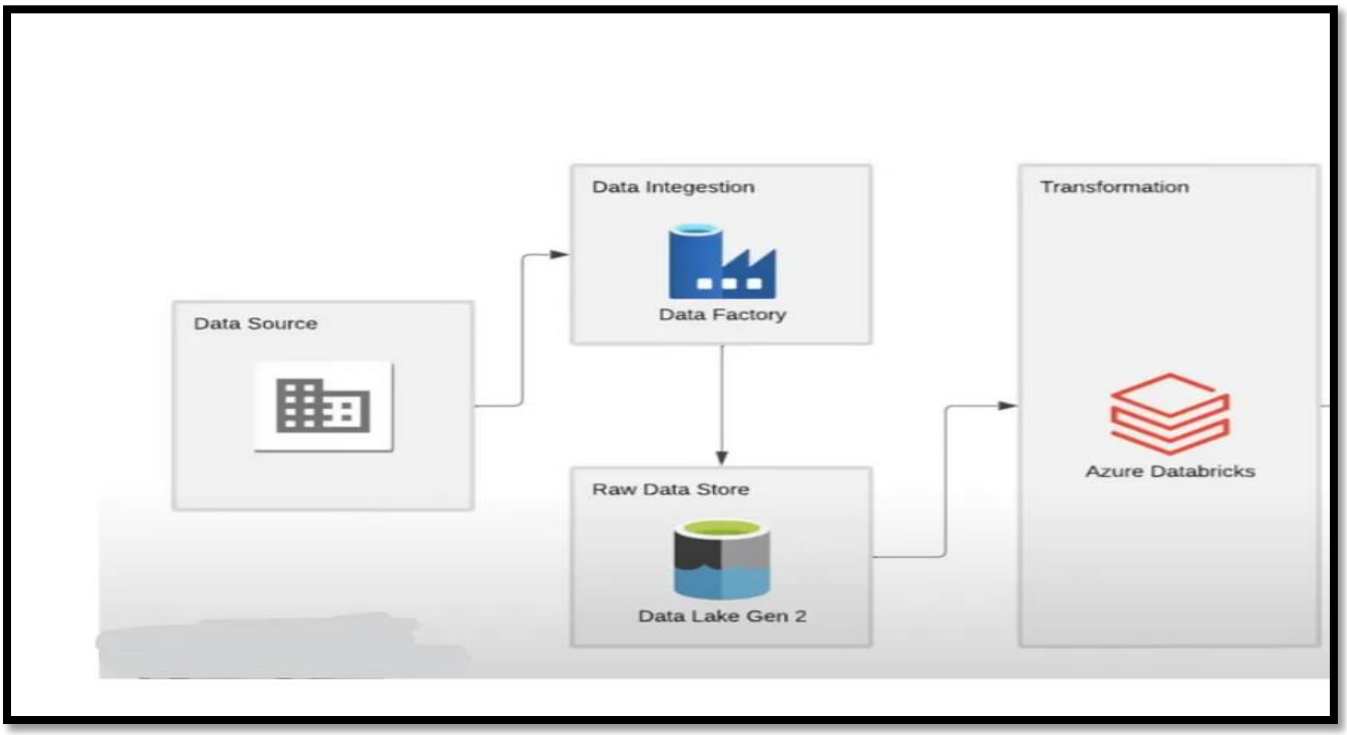
Project Overview:

This project is designed to showcase how various Azure services can be utilized to ingest, store and transform Olympics-related data. This project can configure and run Azure Data Factory pipelines to ingest Olympics data into Azure Data Lake Storage Gen2.

About the Project:

The main objective of this project is to ingest Olympics data into Azure Data Lake Storage Gen2 (ADLS Gen2) using Azure Data Factory (ADF) pipelines and perform data transformation tasks using PySpark notebooks in Azure Databricks.

Architecture Diagram:



This architecture diagram shows about ingesting and transforming the Olympics data using Azure Data Factory , Azure Data Lake Storage Gen2 and Azure Data Bricks.

How it works:

1. Data Ingestion:

1) Source Data Identification:

- Identify the sources of Olympics data.
- These sources could include databases, Azure storage accounts, or any other data repositories where the Olympics data is stored.

2) Azure Data Lake Storage Gen2 Configuration:

- Set up Azure Data Lake Storage Gen2 (ADLS Gen2) as the destination for storing the Olympics data.
- Go to Azure Portal and search for Storage Accounts.
- Click on + to create. Add the Subscription , Resource group name , storage account name
- Enable Hierarchical Namespace to organize data in a hierarchical structure, improving performance and manageability in Advanced setting
- Click on Create
- Deployment is in progress.
- Once the deployment is complete Click on Go to Resource.

3) Azure Data Factory (ADF) Pipeline Configuration:

- Create a Azure data factory by selecting the Azure subscription,Resource group and give a name for data factory and Click on Create.
- Once the deployment is complete, Click on Go to Resource
- Click on Launch Studio.
- Click on Ingest Data
- Give the details of Source and Destination to perform Copy Activity .

4) Run the Pipeline:

- Once the pipeline is configured, trigger the pipeline execution manually or let it run based on the defined schedule.
- Monitor the pipeline execution for any errors or issues and troubleshoot as needed.
- Verify that the data has been successfully ingested into ADLS Gen2.

2. Data Transformation:

1) Azure Databricks Environment Setup:

- Set up an Azure Databricks workspace in the Azure portal.
- Configure clusters within the Databricks workspace with appropriate specifications for running PySpark notebooks.
- Specify the cluster type, instance types, number of instances, and auto-scaling settings based on the workload requirements.
- It is a best practice to use personal compute to reduce the costing.

2) PySpark Notebook Development:

- Create a new notebook within the Azure Databricks workspace.
- Set the configuration to Azure Data Lake Storage gen2 account by providing name of the storage account,name of container and key.
- Develop PySpark notebooks to perform various data transformation tasks on the ingested Olympics data. Tasks may include aggregation, filtering, distinct operations, data normalization, joining datasets, etc.
- Write PySpark code to implement the desired data transformation logic.
- Ensure that the code is well-documented and includes comments for clarity and maintainability.
- Test the PySpark code within the notebook to verify that it produces the expected output.

3) Notebook Execution and Scheduling:

- Execute the developed PySpark notebooks within the Azure Databricks environment to perform data transformation tasks on the ingested Olympics data.
- Schedule the execution of notebooks at specified intervals or triggers to ensure regular updates to the transformed data.
- Configure notebook parameters and input/output paths as necessary for scheduled execution.
- Review the transformed data output to ensure it meets the desired transformation requirements.

Azure Resources / tools used for this Project:

1) Azure Data Lake Storage Gen2 (ADLS Gen2):

- ADLS Gen2 is the storage solution where the Olympics data will be ingested.
- ADLS Gen2 provides a scalable and cost-effective storage solution for big data analytics workloads.

2) Azure Data Factory (ADF):

- Azure Data Factory is a cloud-based data integration service that allows us to create, schedule, and orchestrate data pipelines to move and transform data across various data stores.
- In this project, Azure Data Factory will be used to configure and run pipelines for ingesting Olympics data into ADLS Gen2.

3) Azure Databricks:

- Azure Databricks is a powerful analytics platform that enables organizations to use the capabilities of Apache Spark for processing big data in a collaborative and scalable environment.
- PySpark notebooks are be used in Azure Databricks to perform data transformation tasks like filtering ,group by and aggregations, selecting distinct data, renaming columns etc., on the ingested Olympics data.

4) Python with PySpark:

PySpark is the Python API for Apache Spark, which is a fast and general-purpose cluster computing system. PySpark would be used within Azure Databricks notebooks for developing and executing data transformation tasks.

5) Azure Portal:

The Azure Portal is a web-based interface for managing Azure services. It would be used for provisioning and configuring resources such as Azure Data Lake Storage Gen2, Azure Data Factory, and Azure Databricks.

6) Microsoft Excel :

Microsoft Excel might be used for data preparation or as a data source for creating input datasets for the project.

Project Requirements:

1) Data Ingestion:

- Source Data Identification
- Azure Data Lake Storage Gen2 Configuration
- Azure Data Factory (ADF) Pipeline Configuration for Copy Activity
- Debugging ADF pipeline

2) Data Transformation:

- Azure Databricks Environment Setup
- PySpark Databricks Notebook
- Notebook Execution

Tasks performed:

Data Ingestion with ADF Pipelines:

- Configure Azure Data Factory (ADF) pipelines to ingest Olympics data.
- Define the necessary data sources and sinks, specifying Azure Data Lake Storage Gen2 as the destination.
- Set up activities within the pipeline to orchestrate the data movement.
- Schedule or trigger the pipeline execution as per our requirements.

Data Transformation with PySpark Notebooks in Azure Databricks:

- Utilize Azure Databricks for data transformation using PySpark.
- Create Azure Databricks workspace in Azure portal.
- Develop PySpark notebooks to perform specific transformation tasks like sorting, filtering, aggregations, distinct columns etc., on the ingested Olympics data.
- Leverage the scalability of Databricks for efficient distributed processing.
- Execute the notebooks, ensuring they handle the data according to your transformation logic.

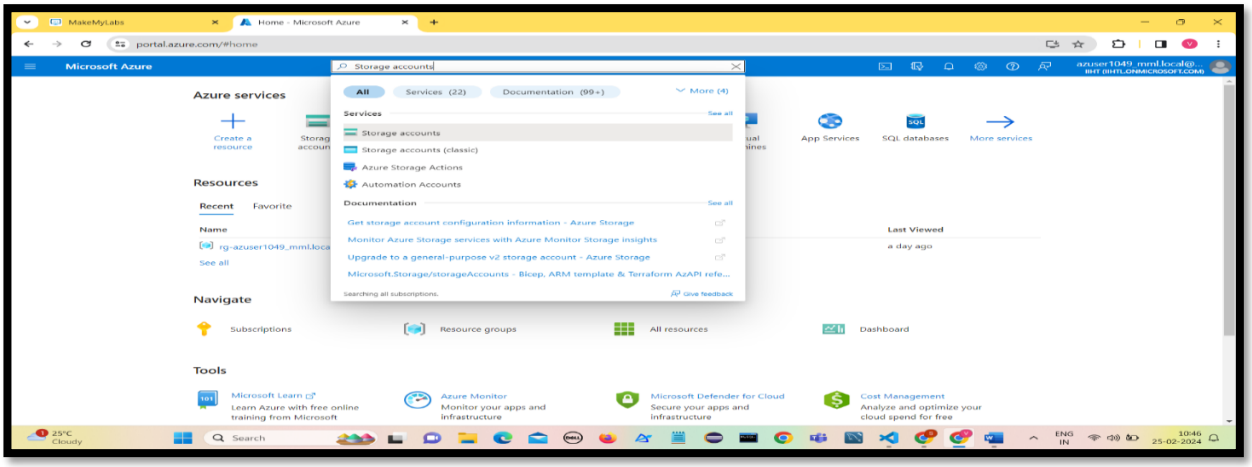
Overall Execution Flow:

- Initiate the Azure Data Factory pipeline to ingest raw Olympics data into Azure Data Lake Storage Gen2.
- Once ingestion is complete, trigger/run the PySpark notebooks in Azure Databricks for data transformation.
- PySpark notebooks read data from Data Lake Storage, apply transformations on the data.
- Monitor pipeline and notebook executions for any errors or performance issues.
- Schedule these processes to run at suitable intervals to keep data up-to-date.

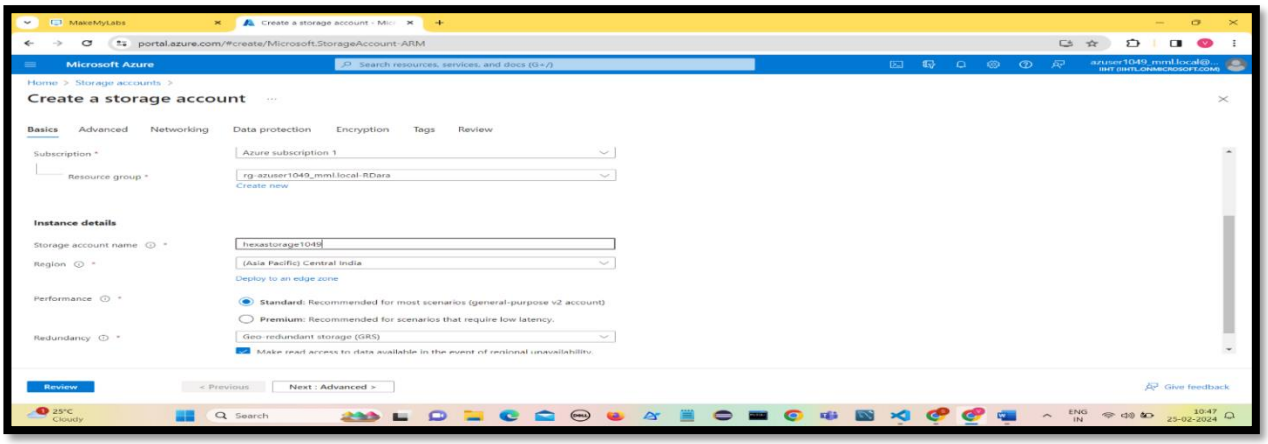
Result :

1) Data Ingestion

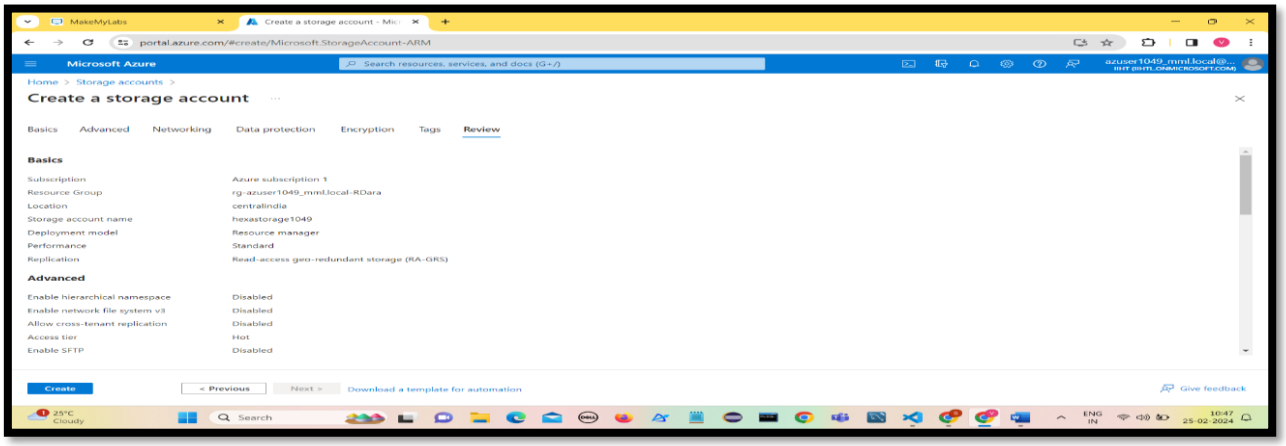
Go to Azure Portal and Search for Storage Accounts.



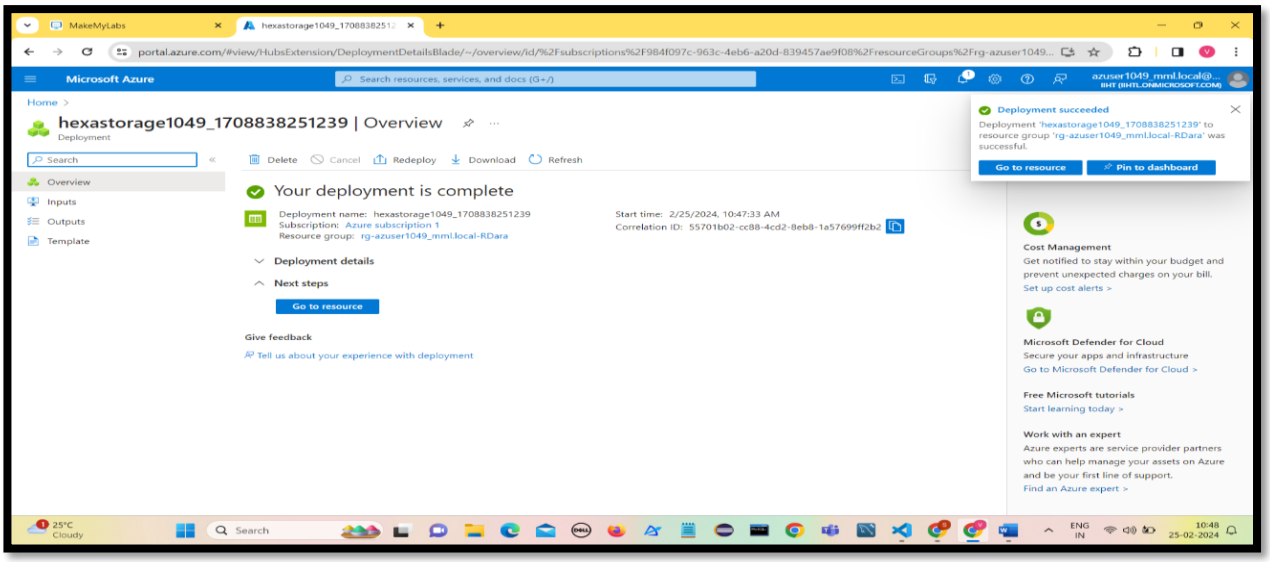
To create a Storage Account fill in all the details and Click on Review.

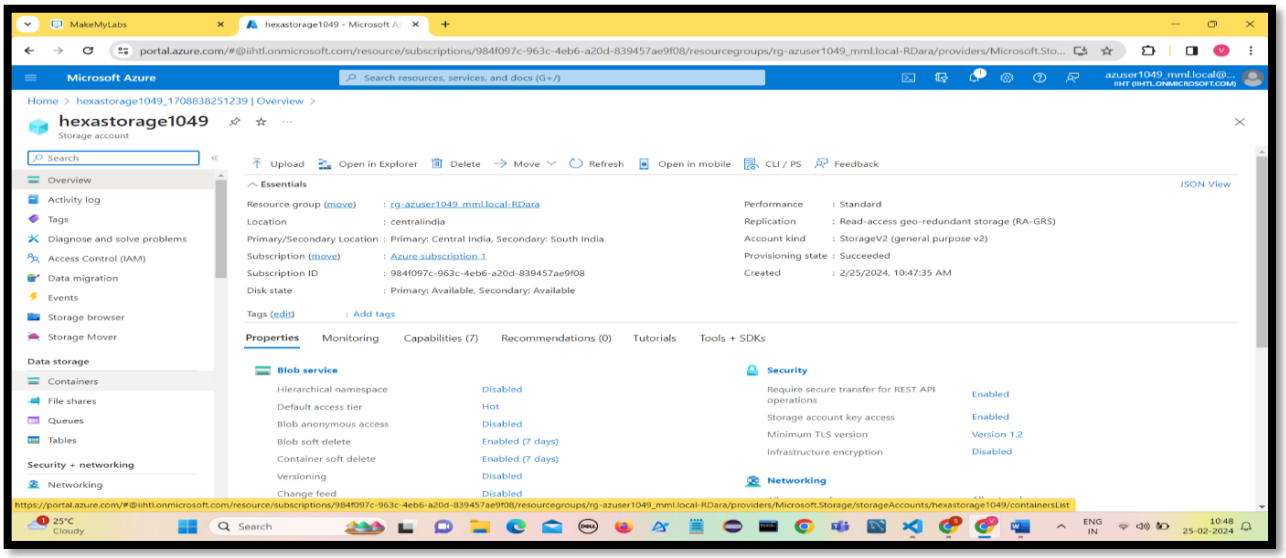


After verifying all the details of Storage Account , Click on Create.

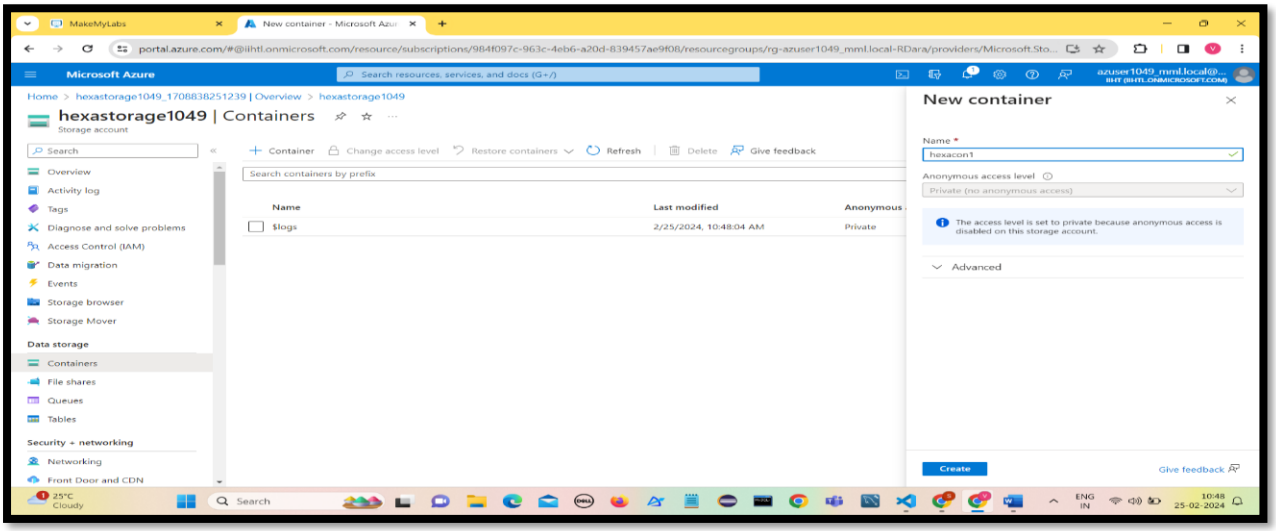


Once the deployment is complete, Click on Go to Resource.

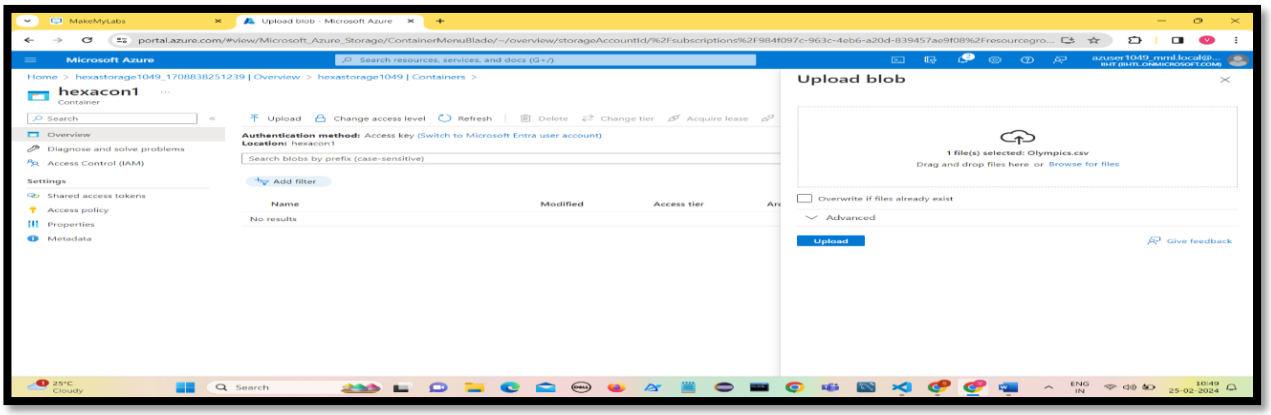




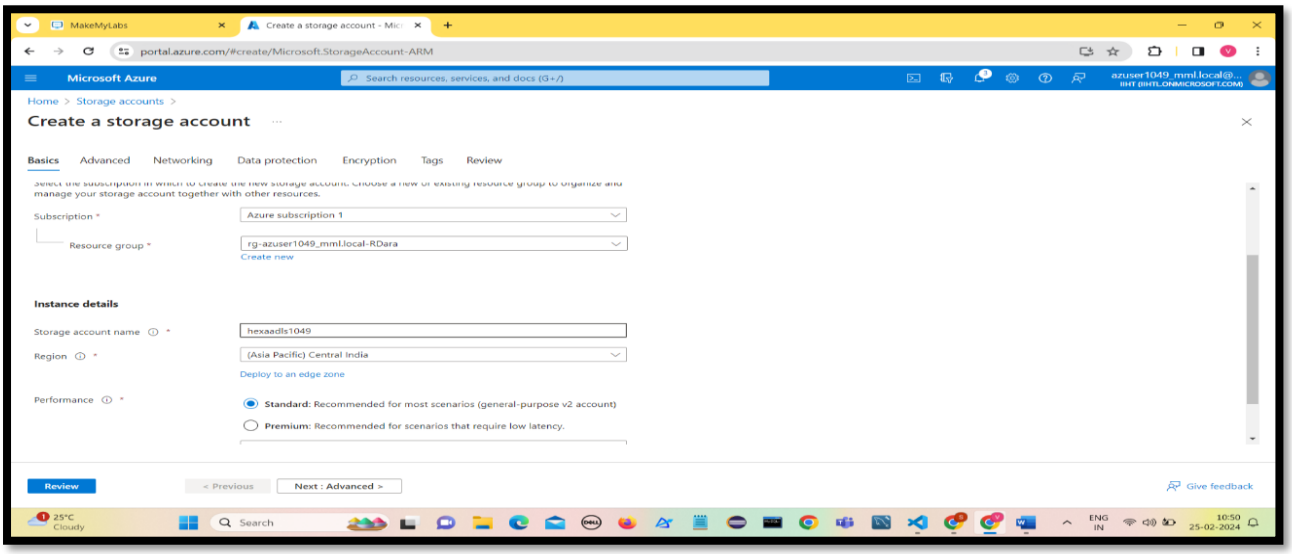
Create a container (hexacon1) in hexastorage1049 storage account



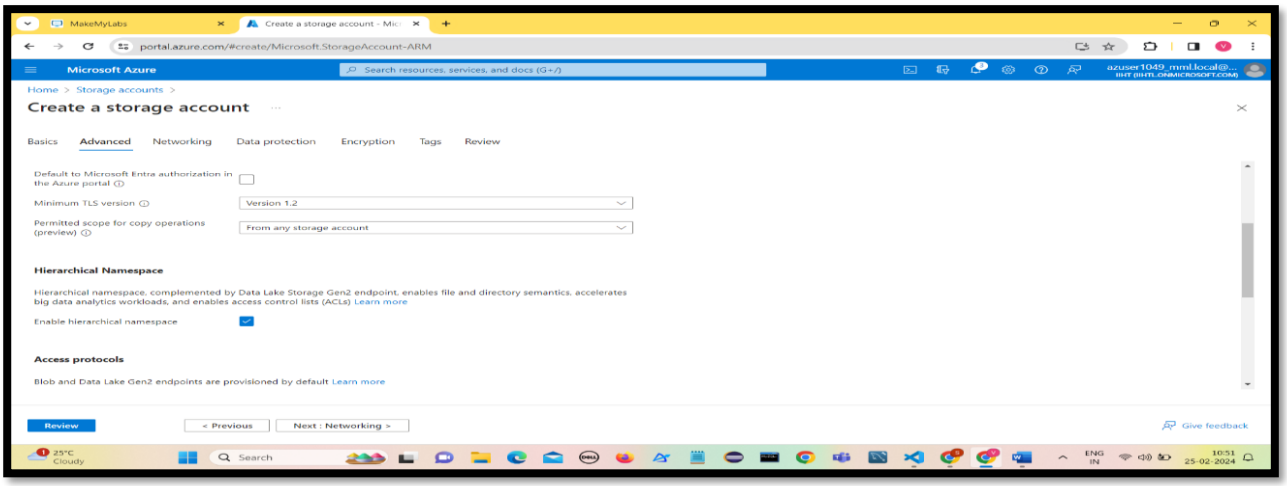
Upload Olympics.csv file (blob) into the container



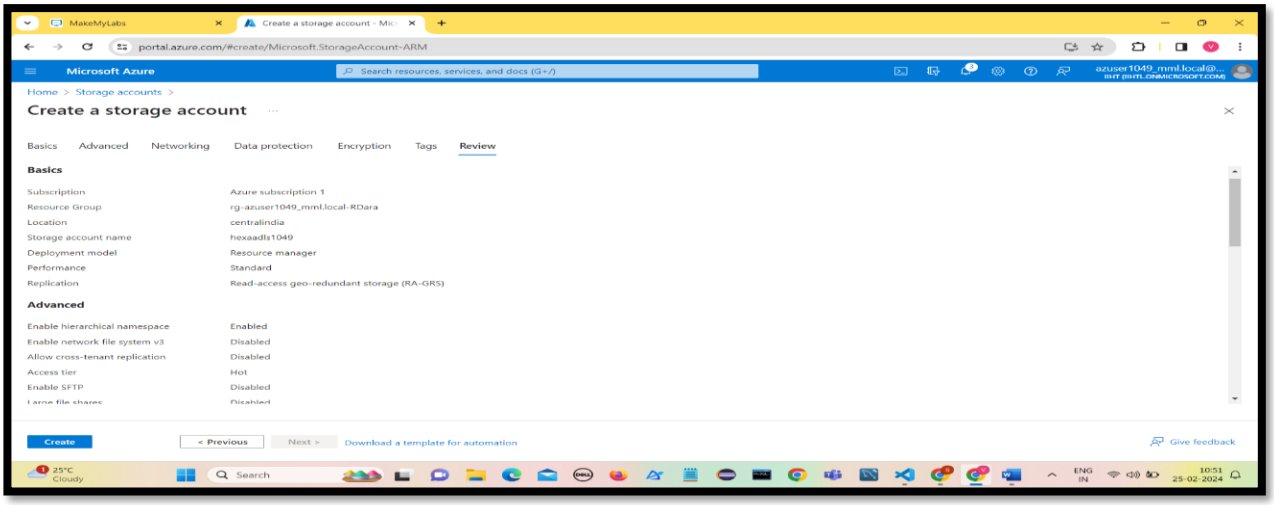
Creating an ADLS gen2 account (hexaadls1049)

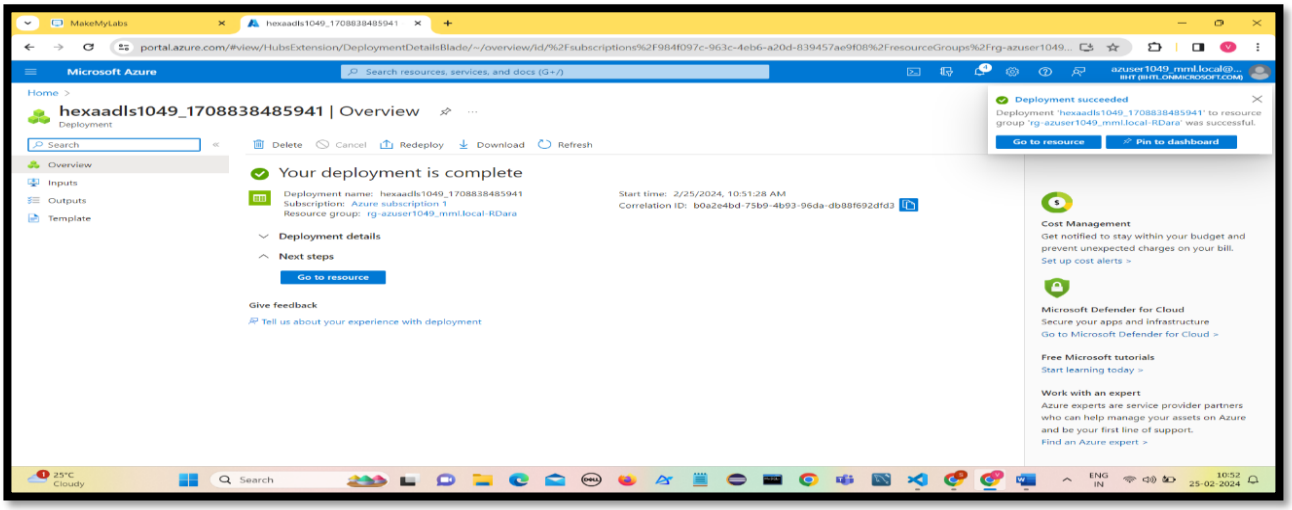


In Advanced , Enable Hierarchical Namespace to make it ADLS gen2 Account

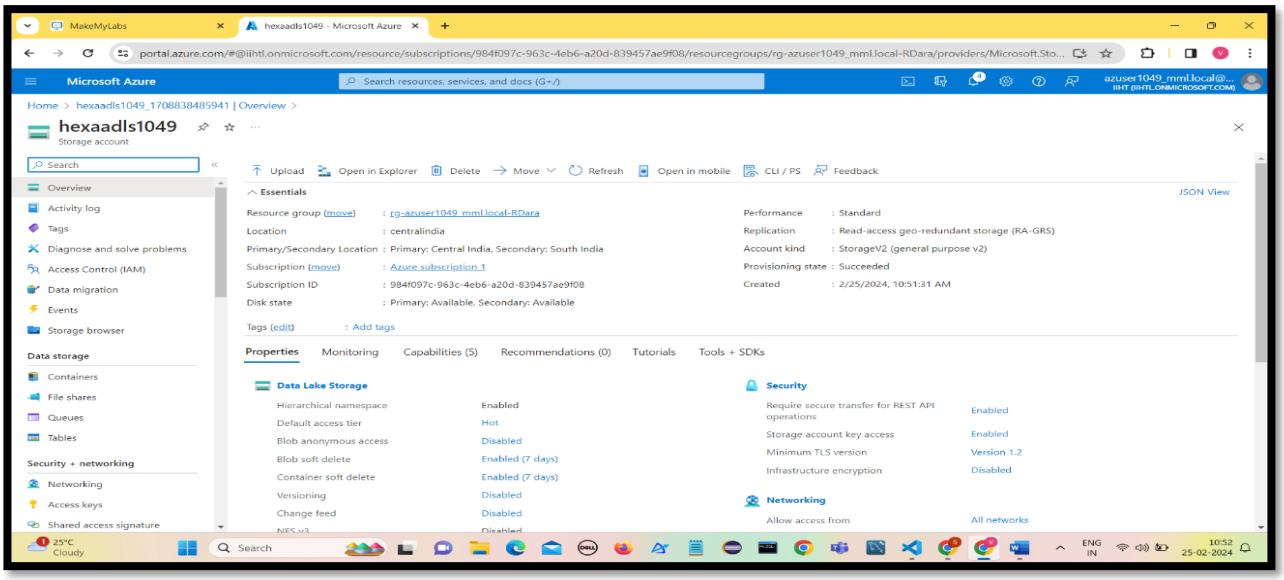


Verify all the details of Storage account and Click on Create

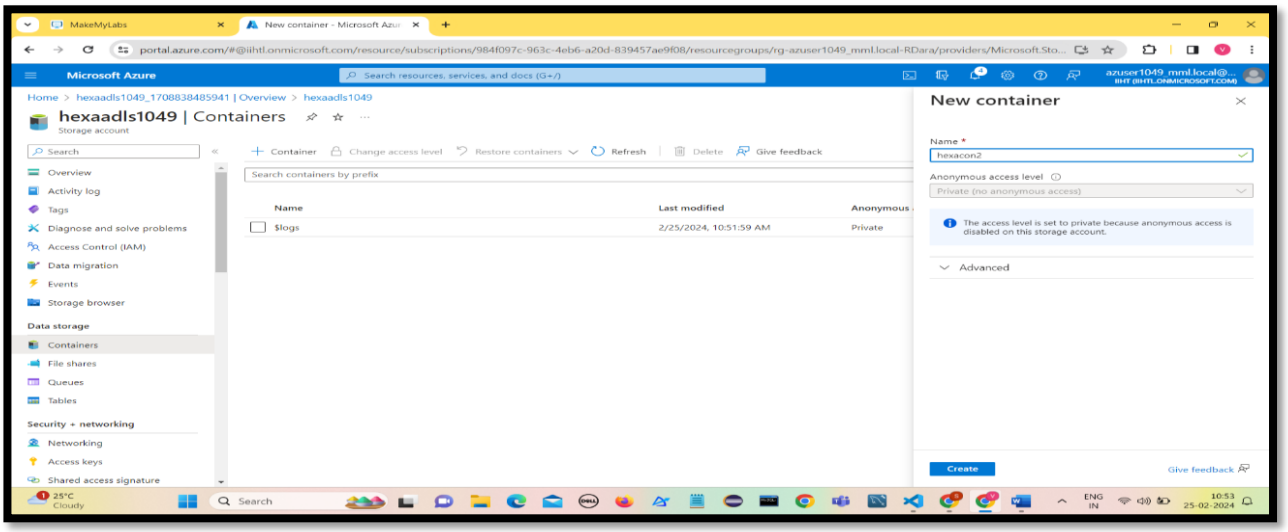




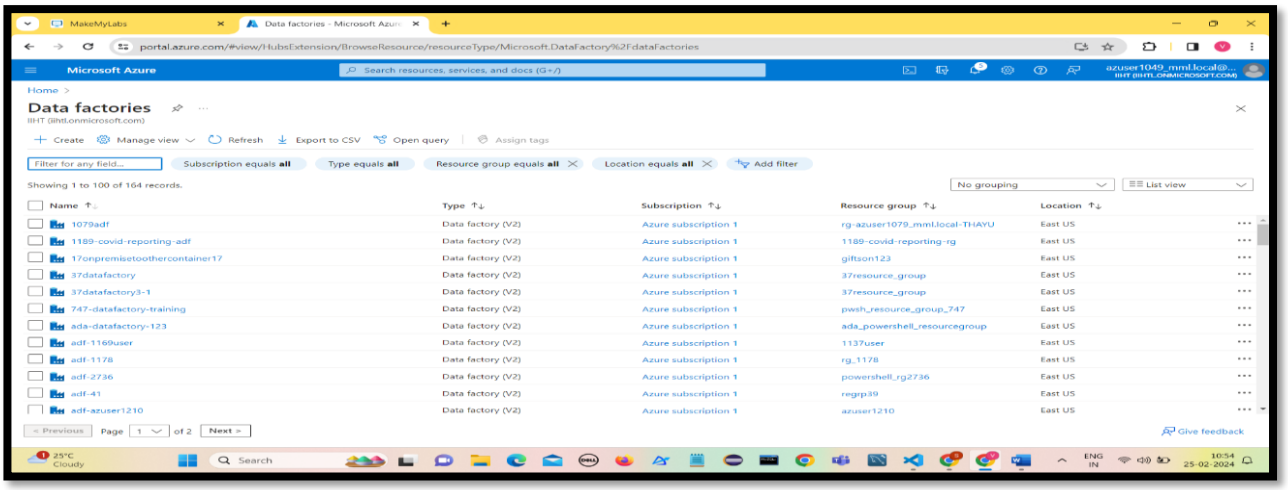
Once the deployment is complete , Click on Go to Resource.



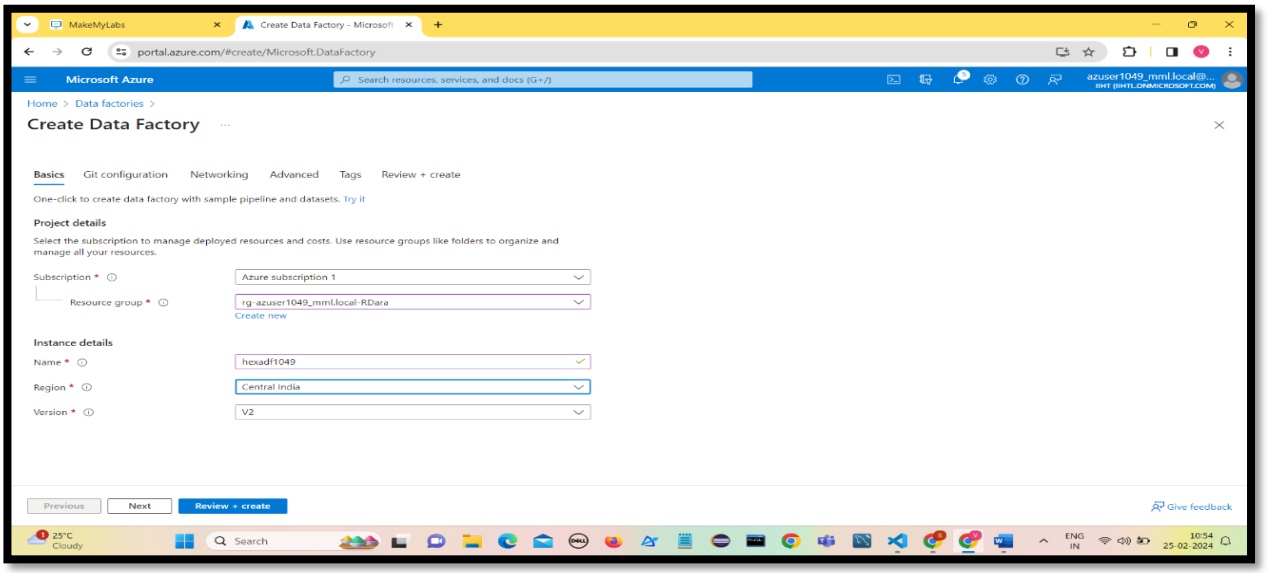
Create a container hexacon2



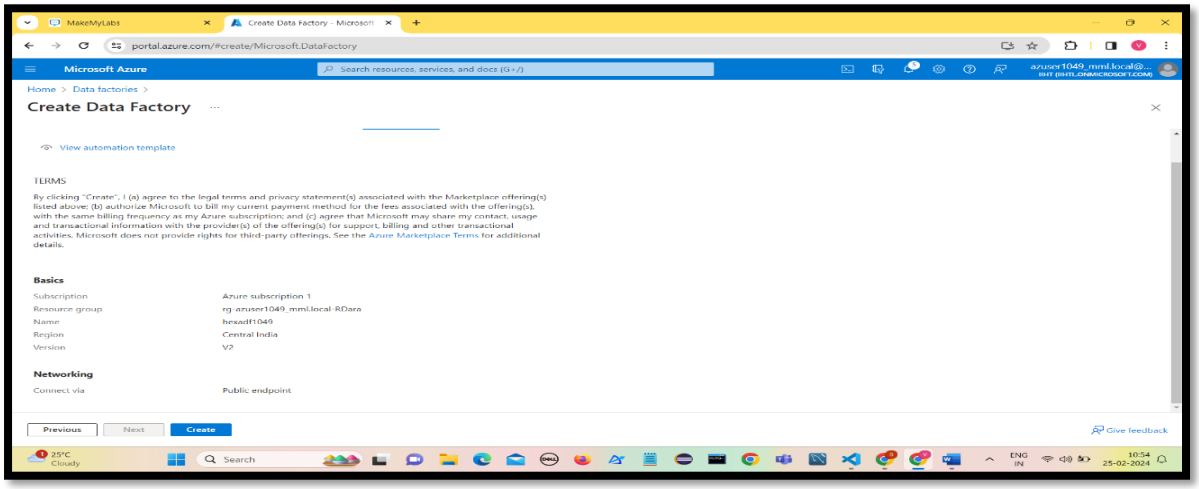
Create a Azure Data Factory (hexadf1049)



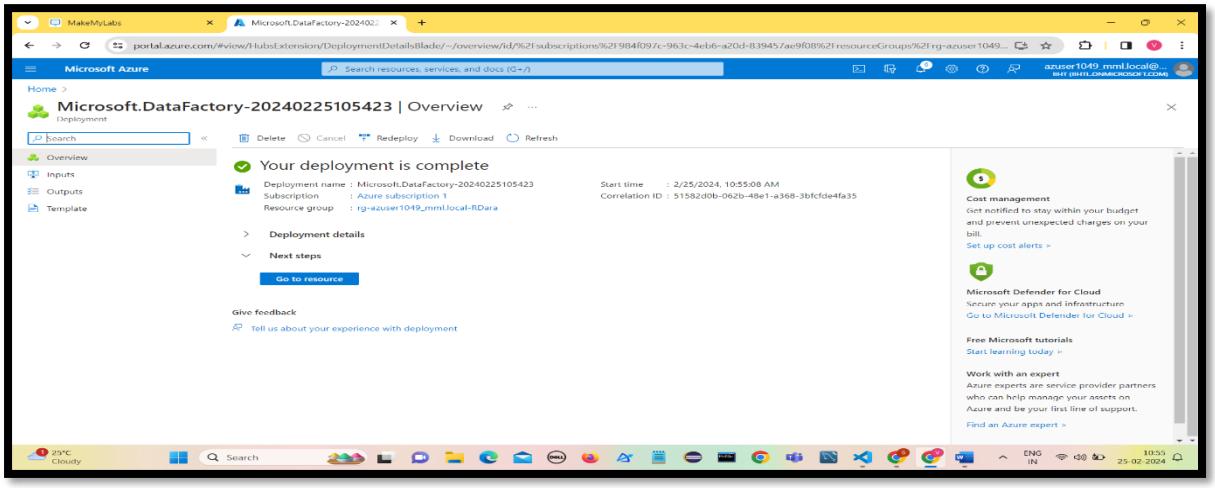
Click on Review+Create



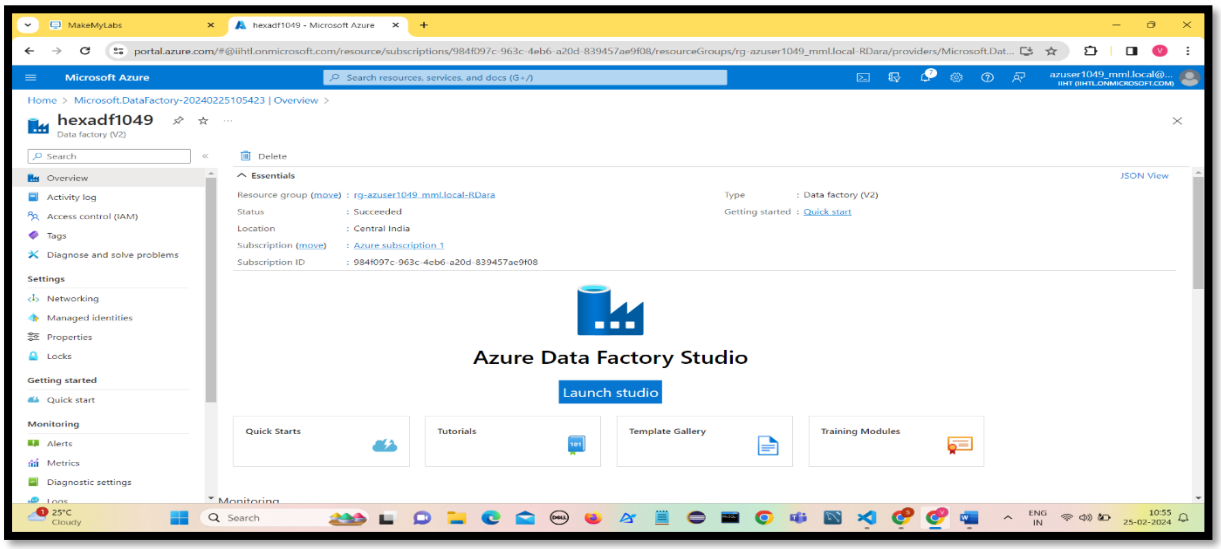
Verify the details and Click on Create



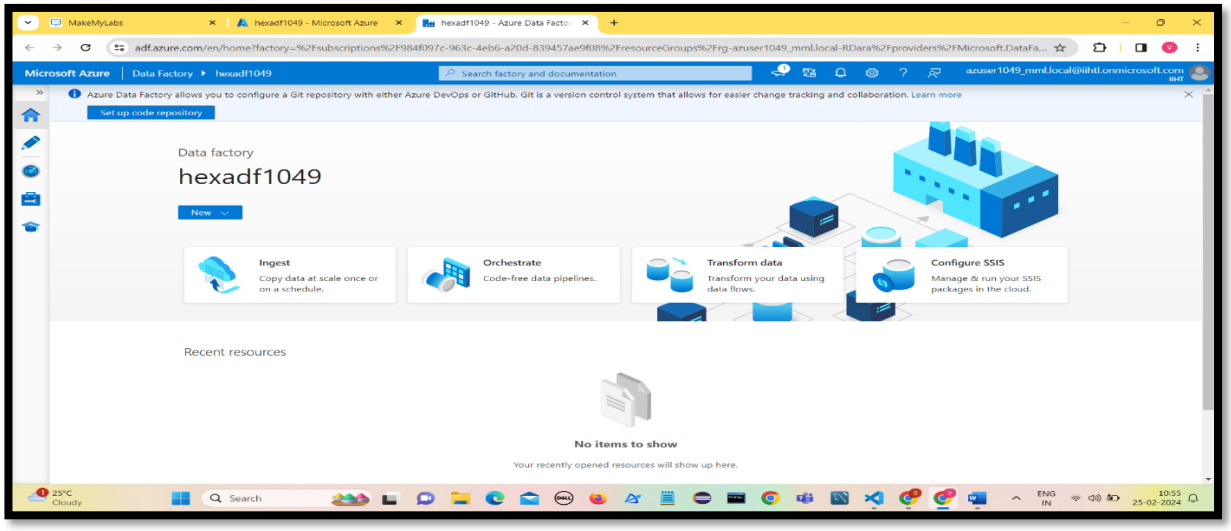
Once after deployment is complete, Click on Go to Resource



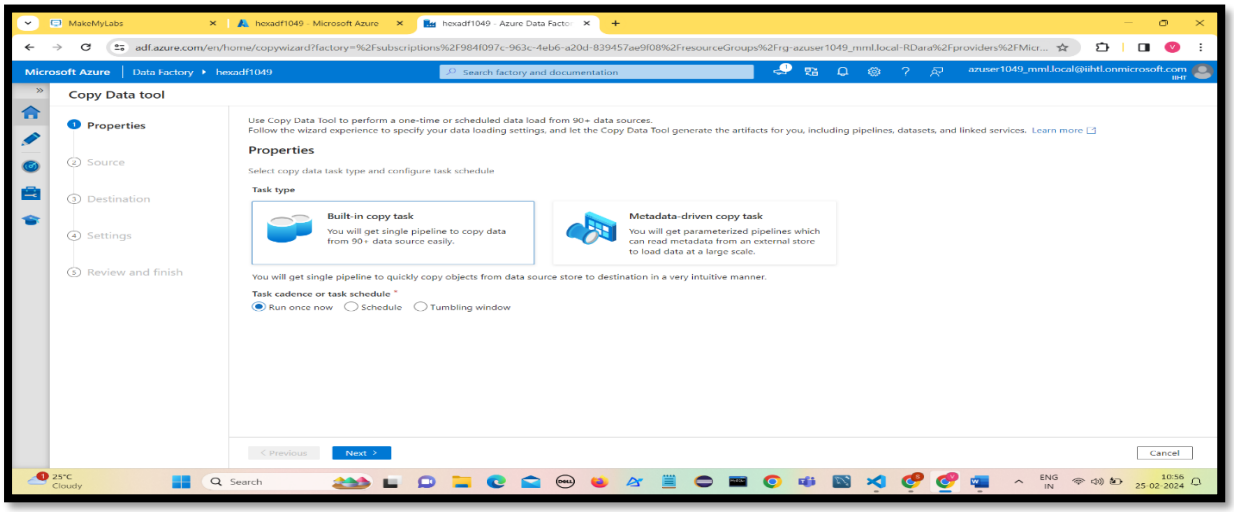
Click on Launch Studio



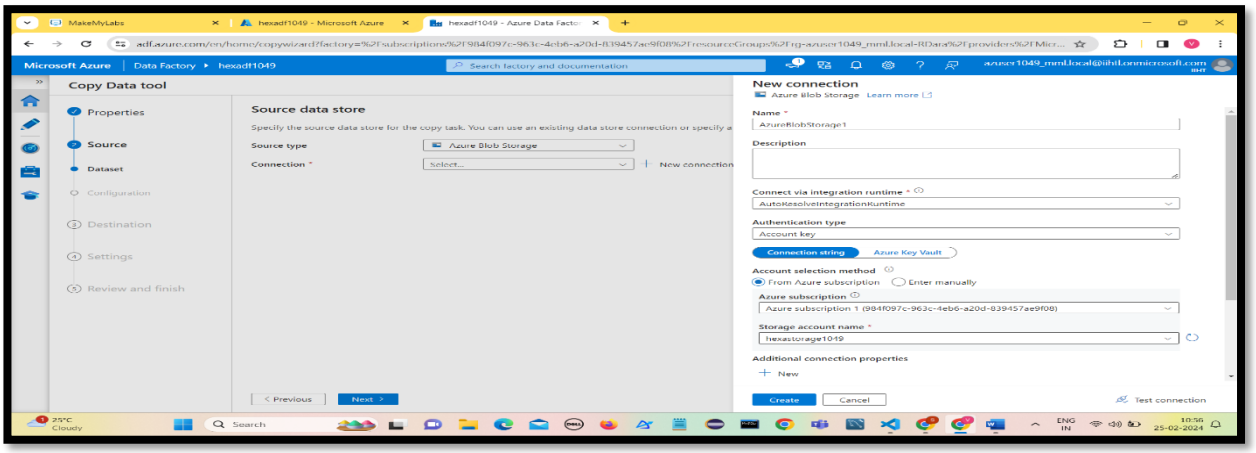
Click on Ingest to create a data factory pipeline to ingest data into ADLS gen2 storage.



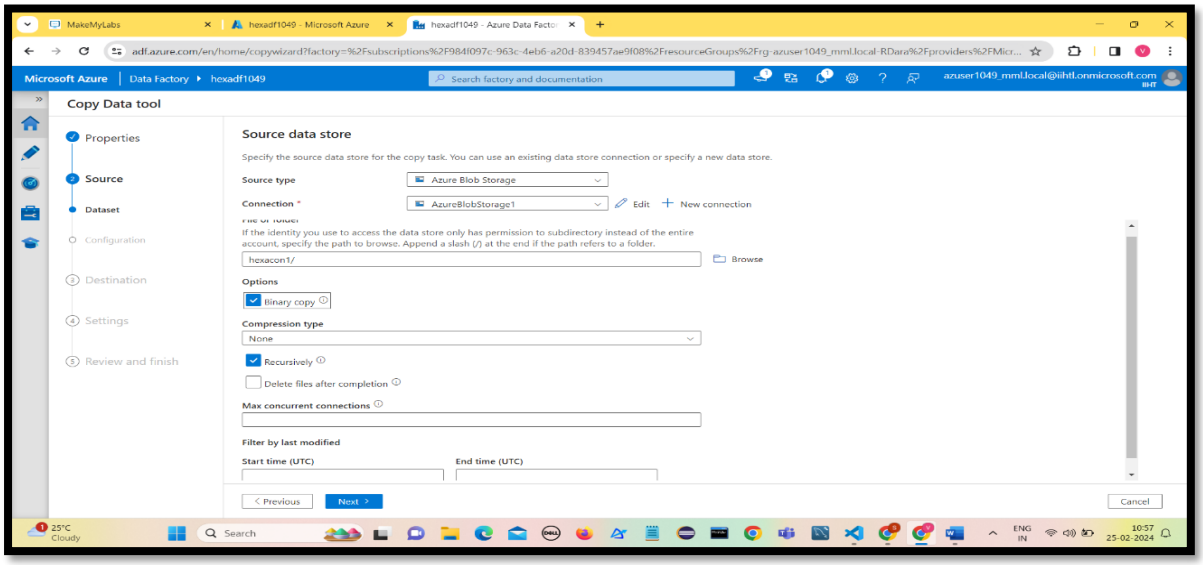
Click on built-in copy task and Enable Run Once now.



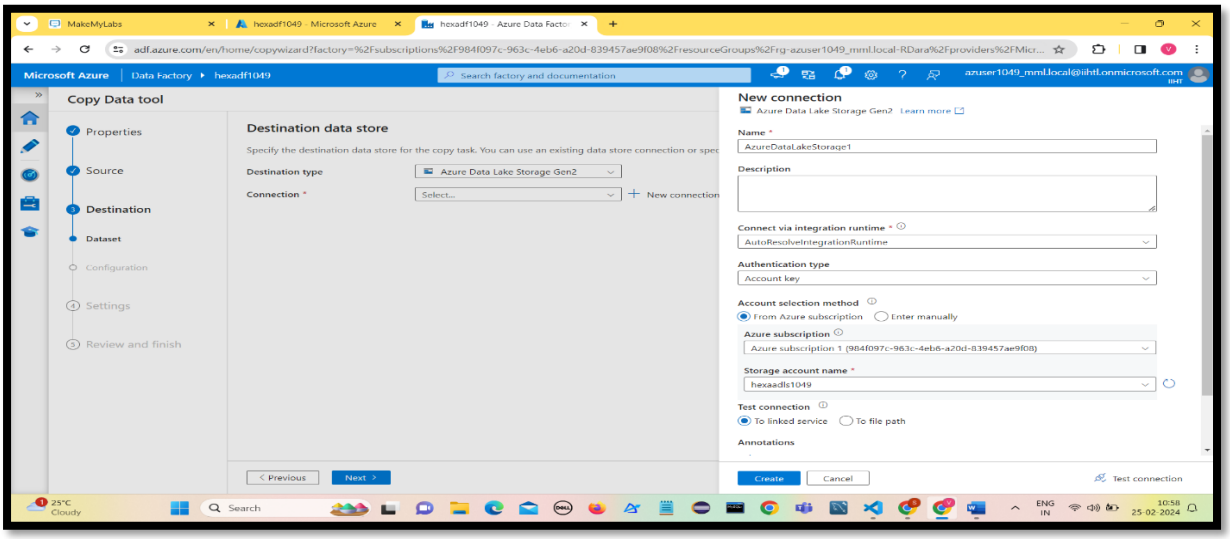
Create Source Connection by specifying the required details.

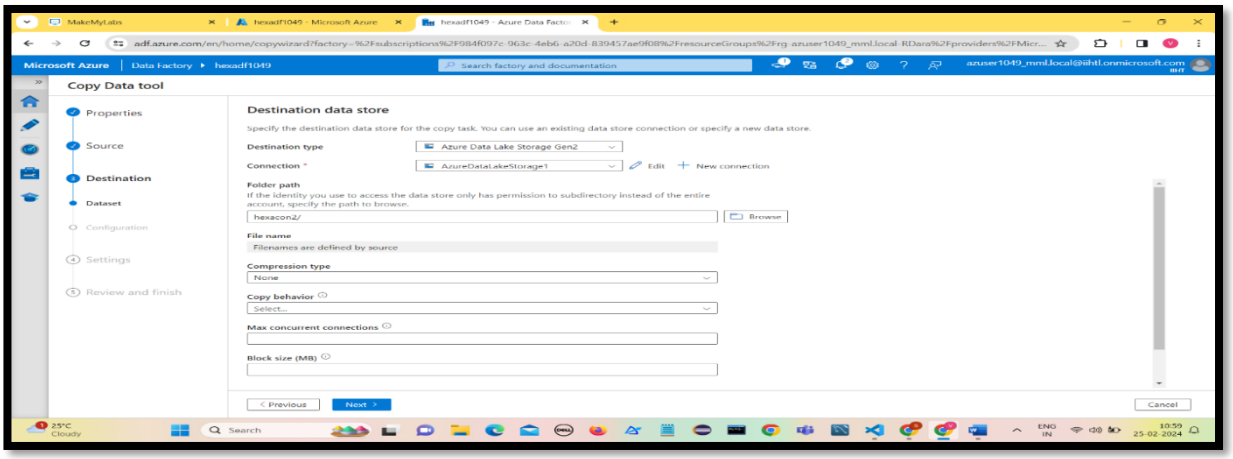


Give the folder details to copy data from storage account

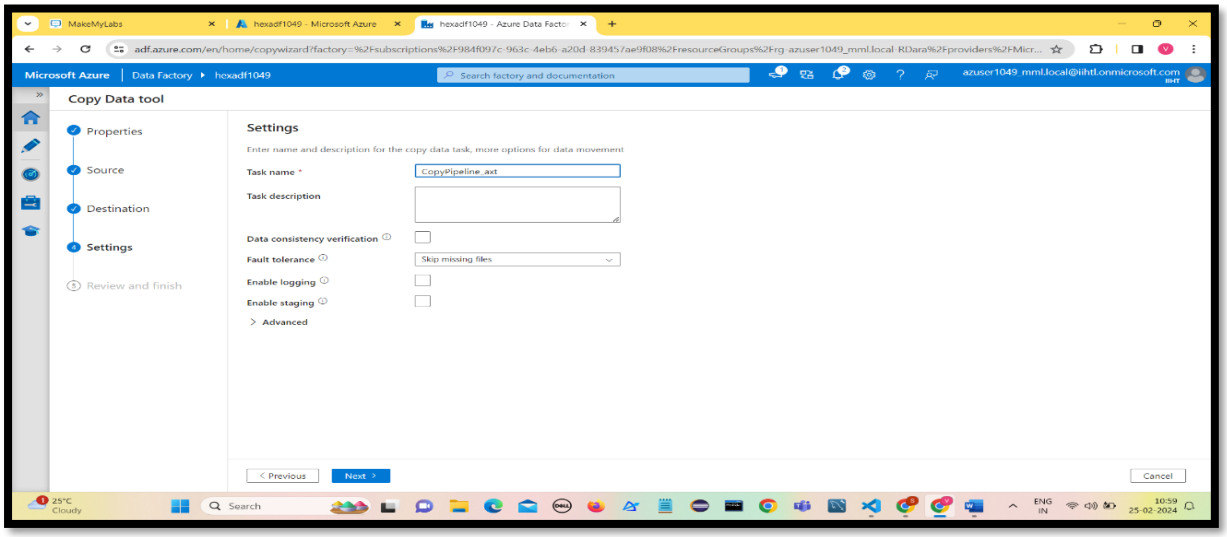


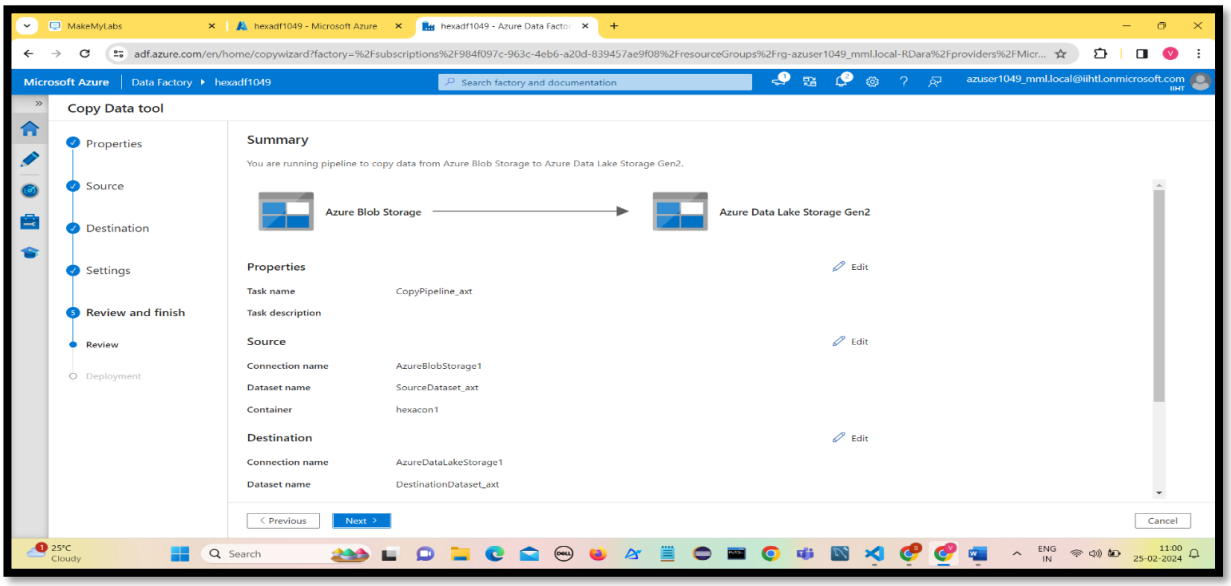
Create Destination Connection by specifying the required details.



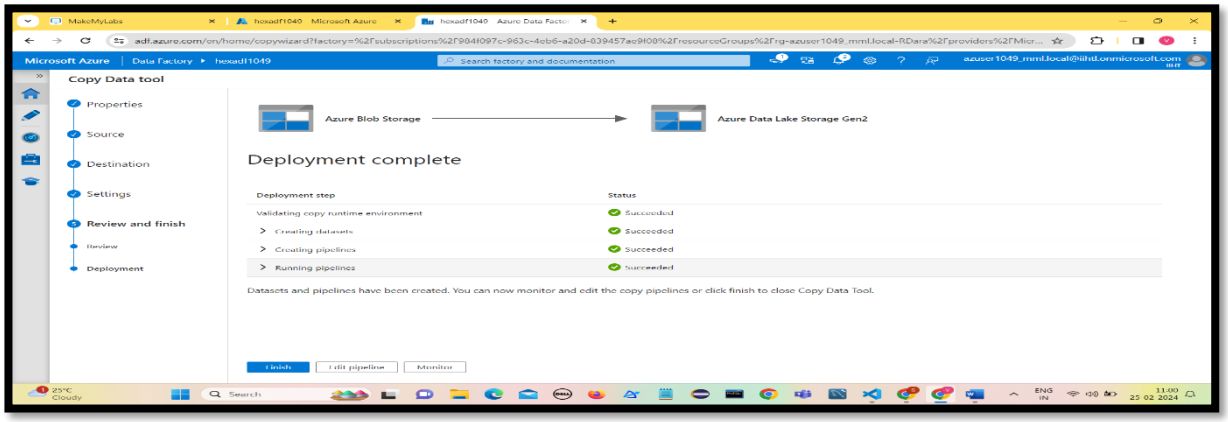


Give the name of pipeline and click on Next

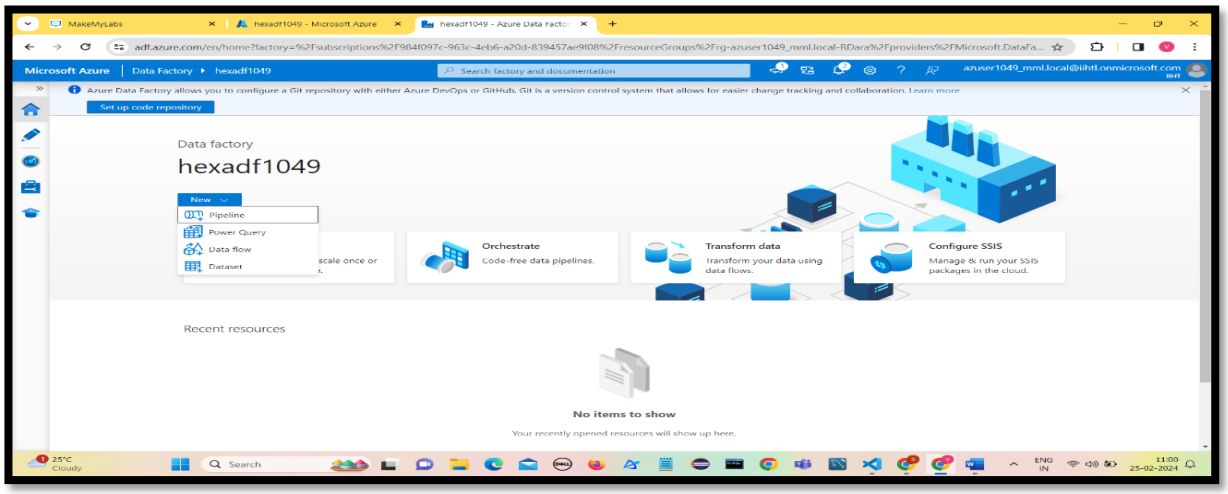




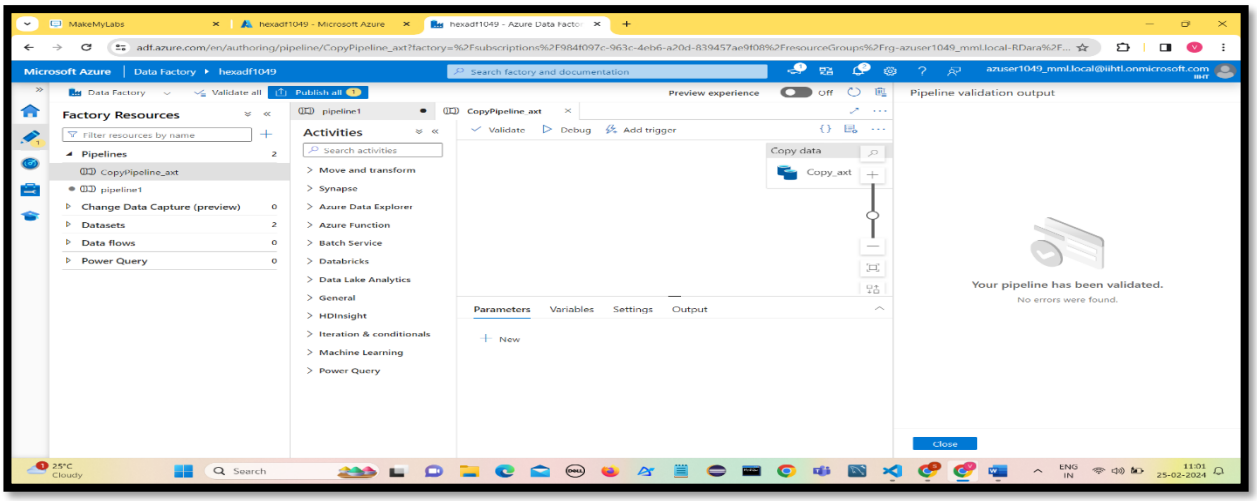
Click on Next



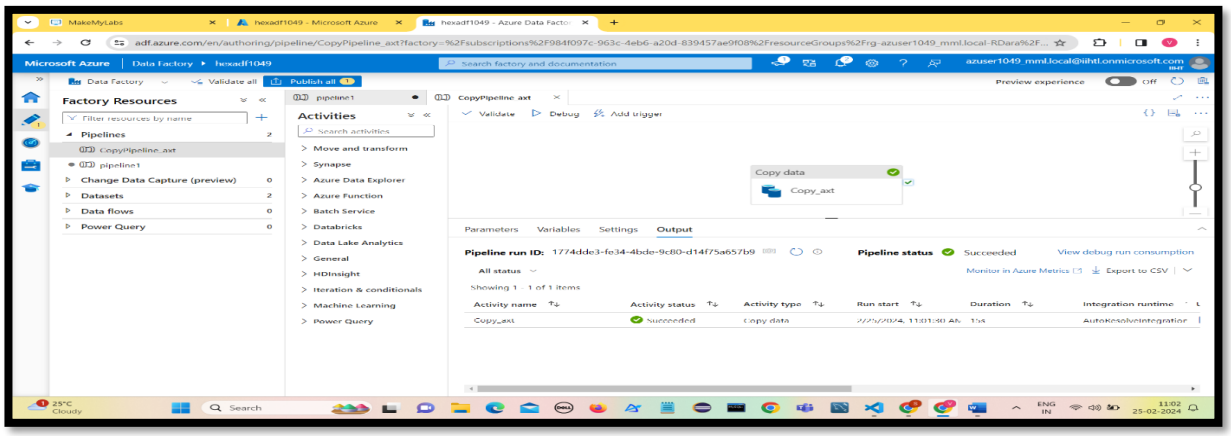
Click on Finish



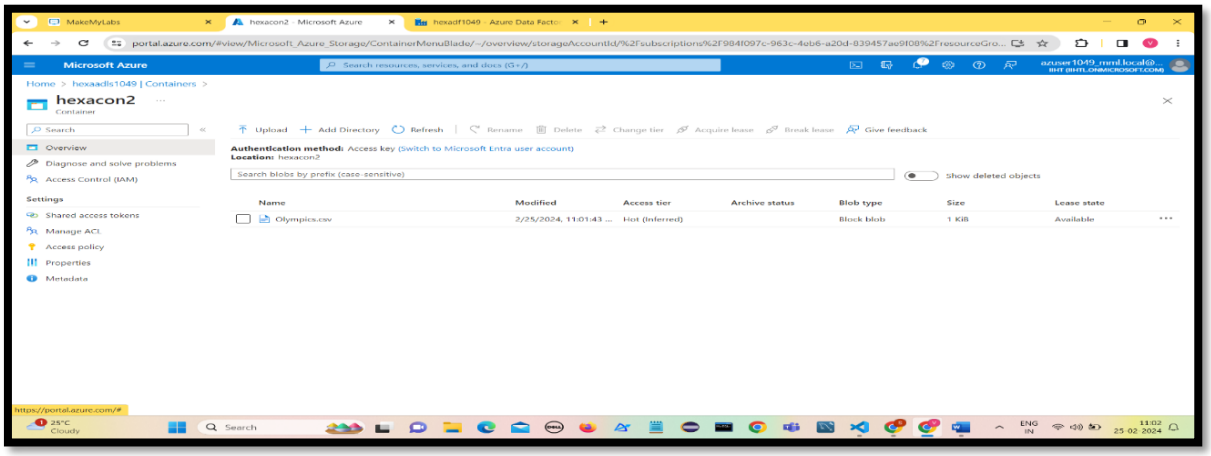
Click on New and then Click on Pipeline
Click on Validate



Click on Debug

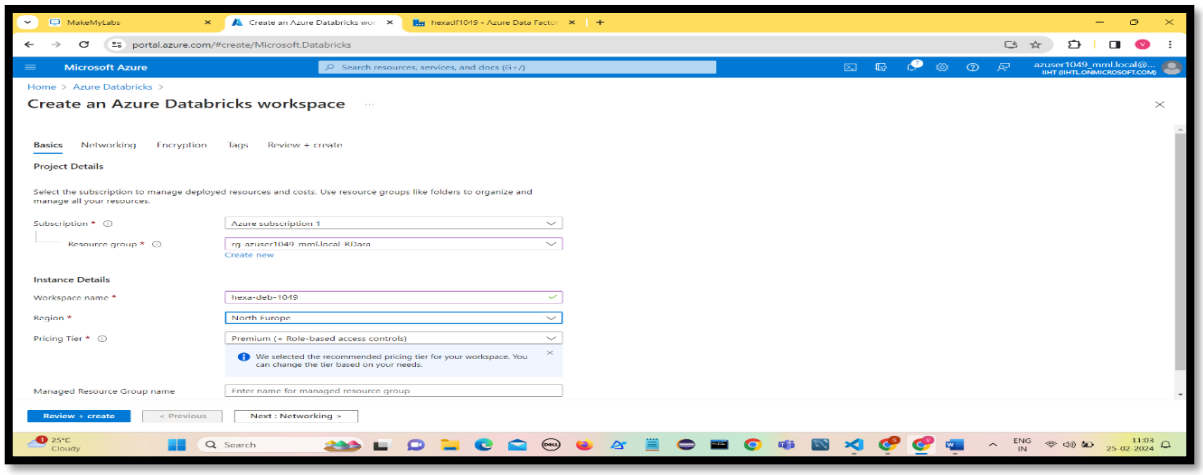


Since pipeline got executed , the file from source is copied into ADLS gen2 container

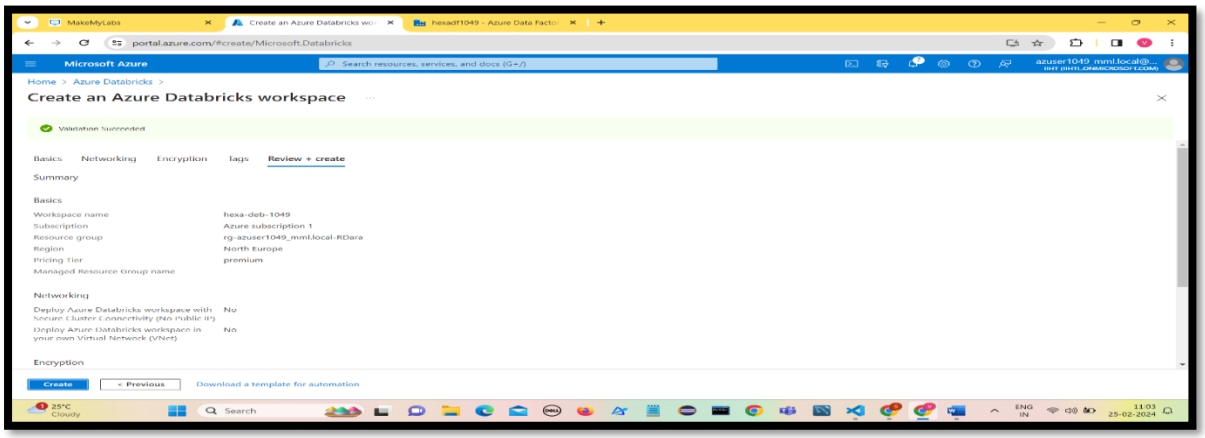


2) Data Transformation

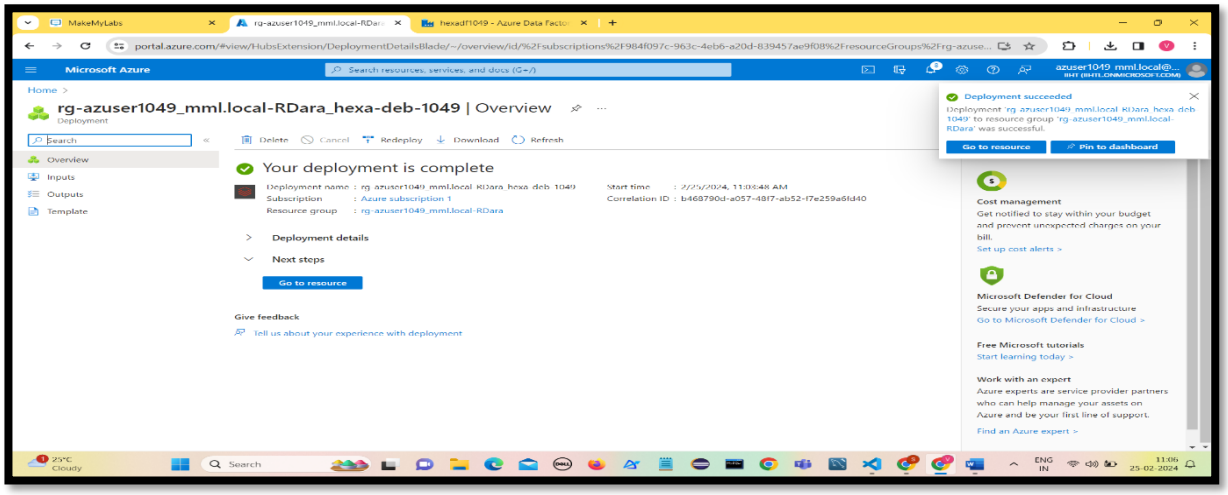
Creating Azure Data Bricks workspace



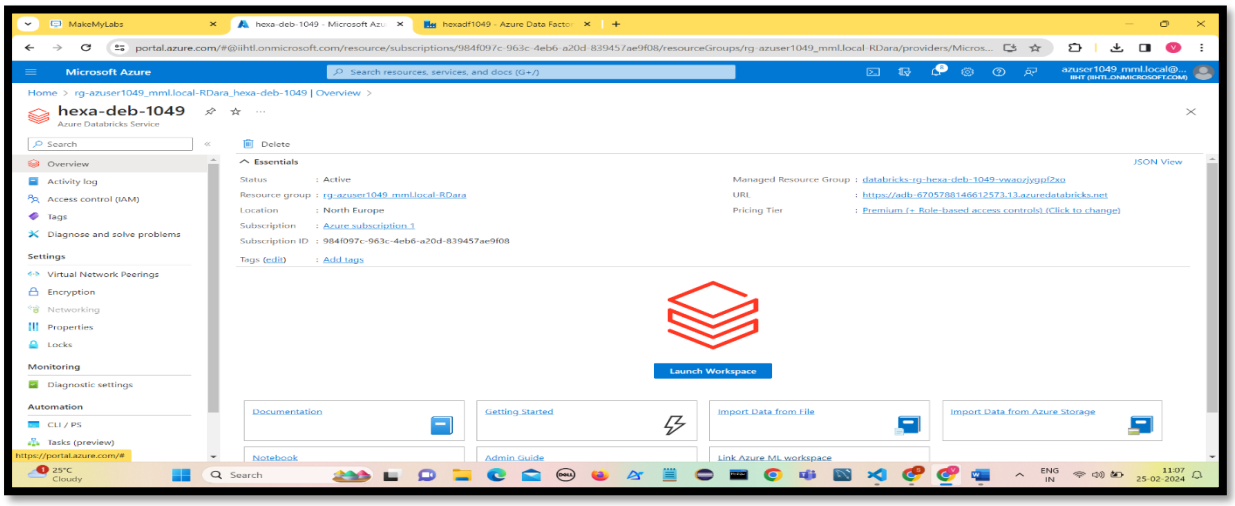
Click on Review+Create

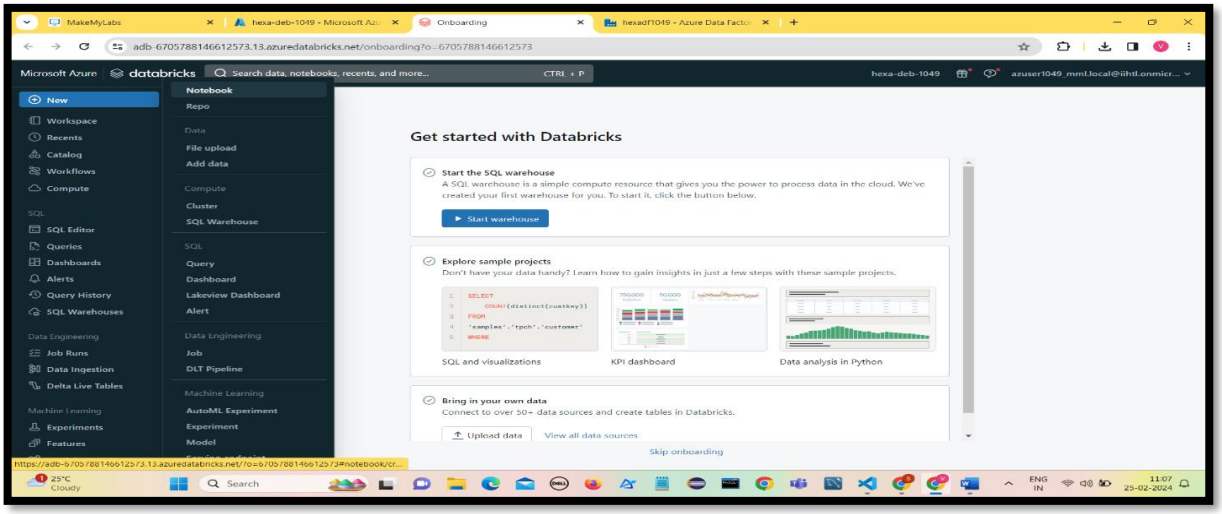


Once the deployment is complete , Click on Go to Resource.

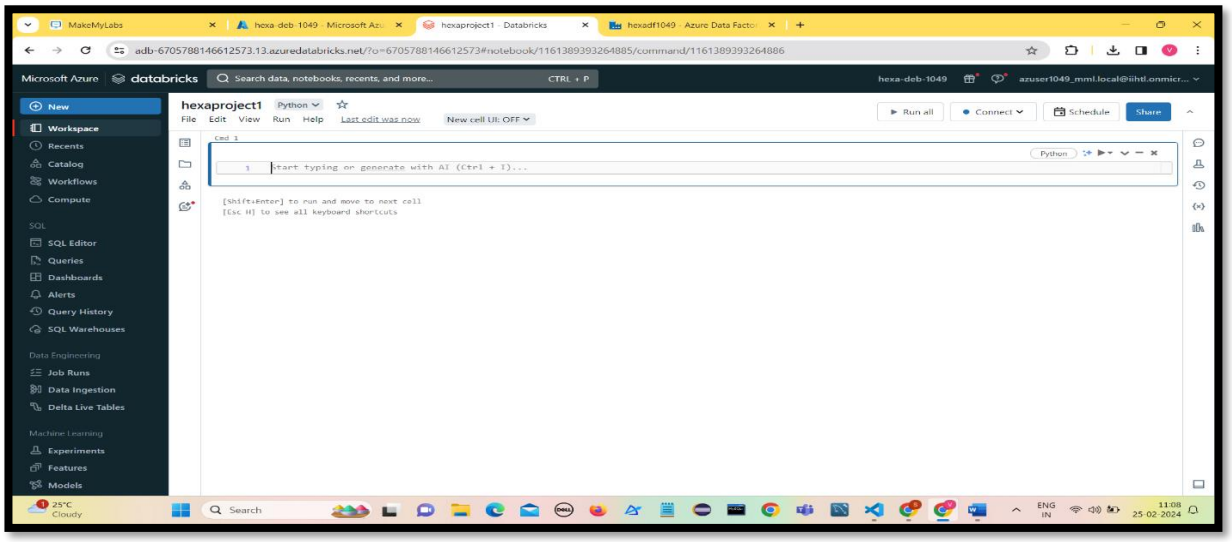


Click on Launch Workspace.

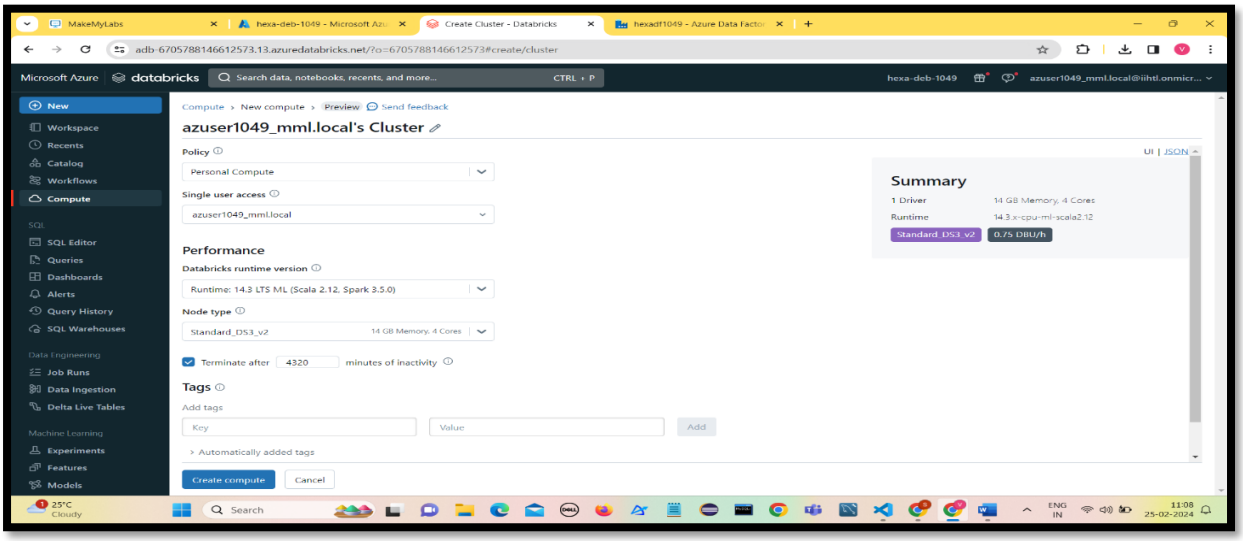


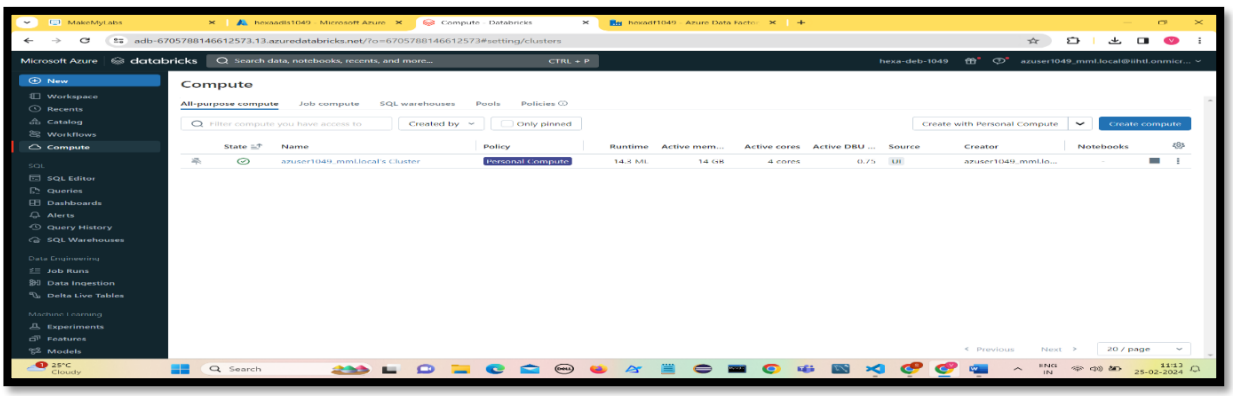


To create a Note Book in Data Bricks, Click on New and then Notebook.



Create a compute (cluster) to run the Data Bricks Notebook .

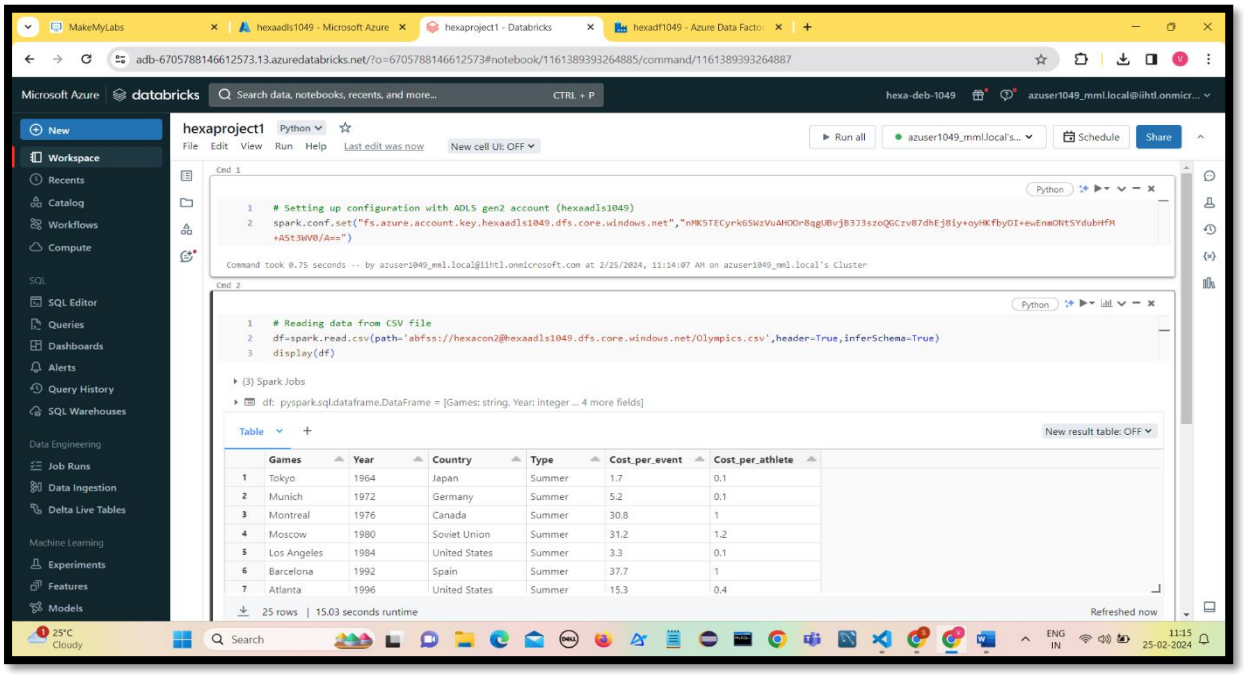




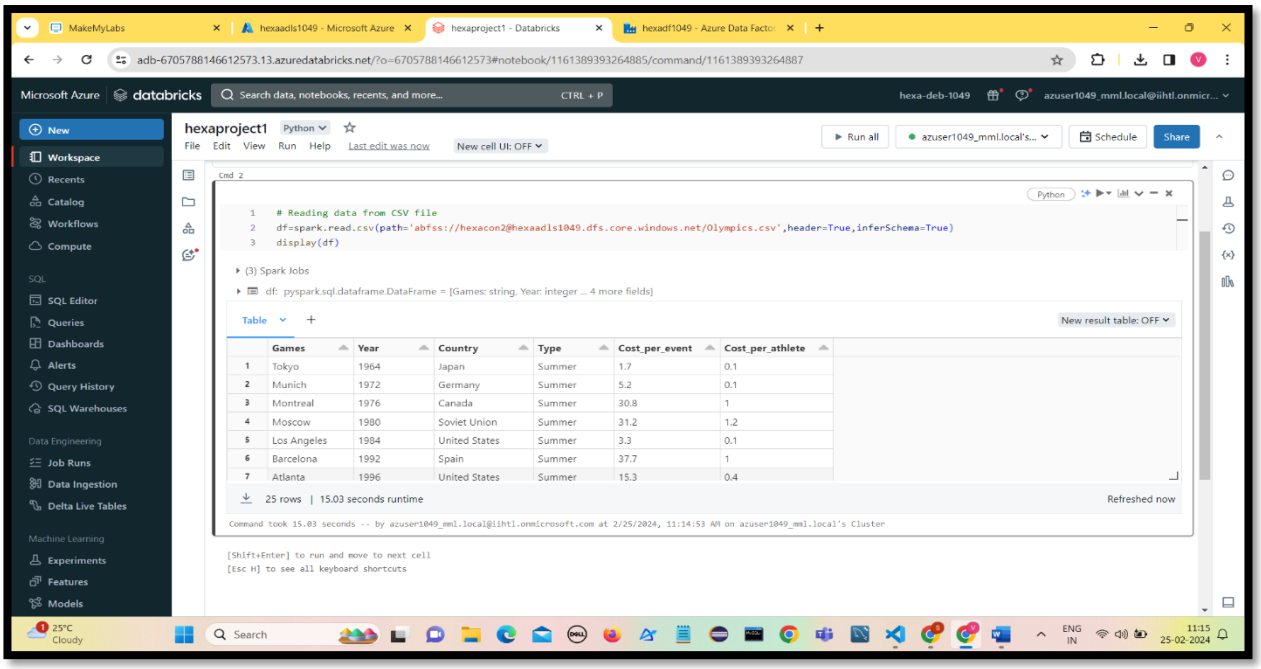
Now the compute is ready to execute Data Bricks notebook.

Applying transformations on Ingested Data into Azure Data Lake Storage Gen2.

Setting up the configuration with ADLS Gen2 account with path and key as parameters.



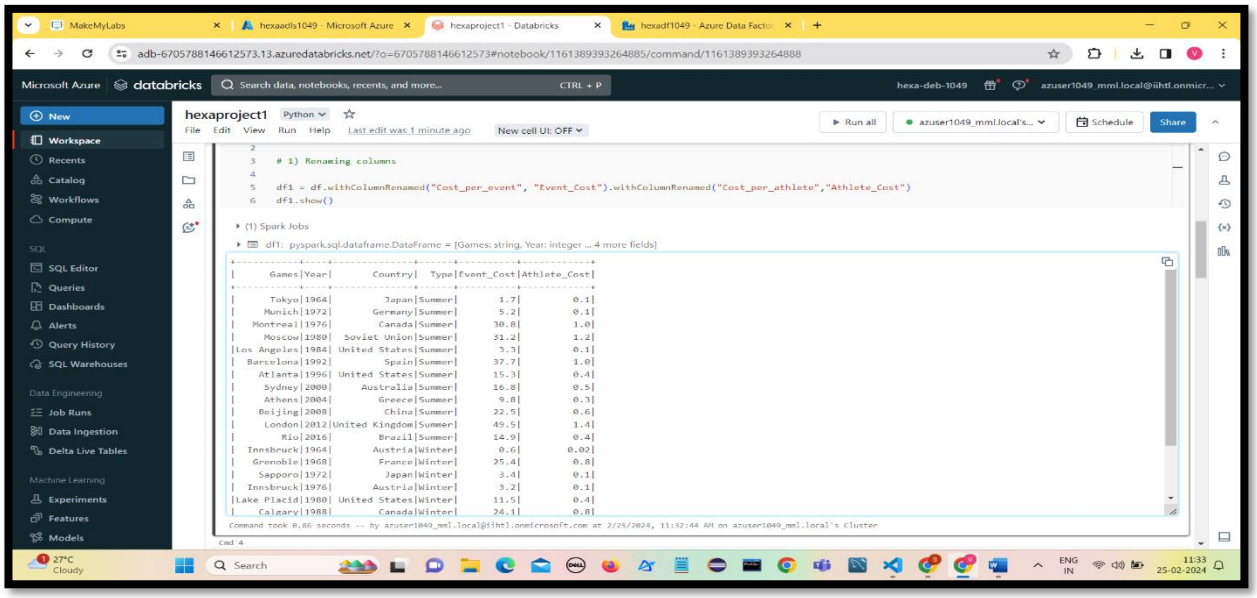
Reading data from Olympics.csv file using spark.read.csv where the data of Olympics is present.



Applying transformations on Ingested Data into Azure Data Lake Storage Gen2.

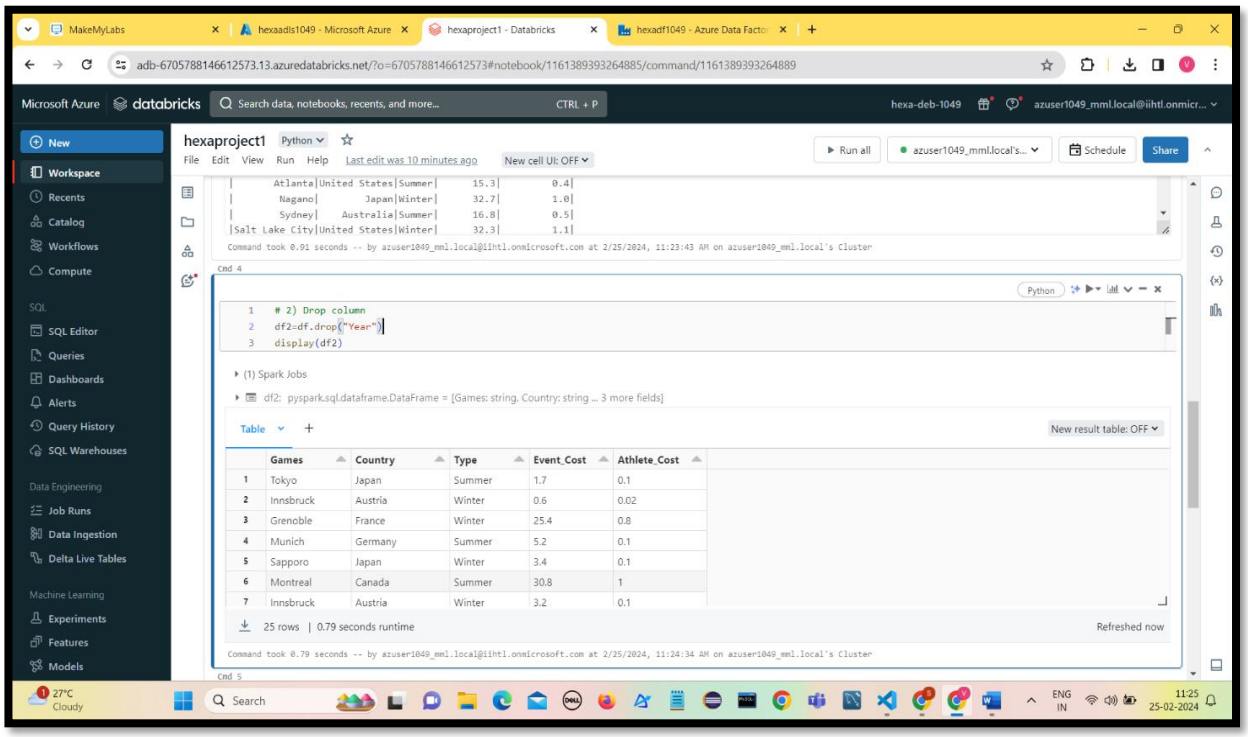
1) Renaming Columns

Rename the column Cost_per_event with Event_Cost and Cost_per_athlete with Athlete_Cost.



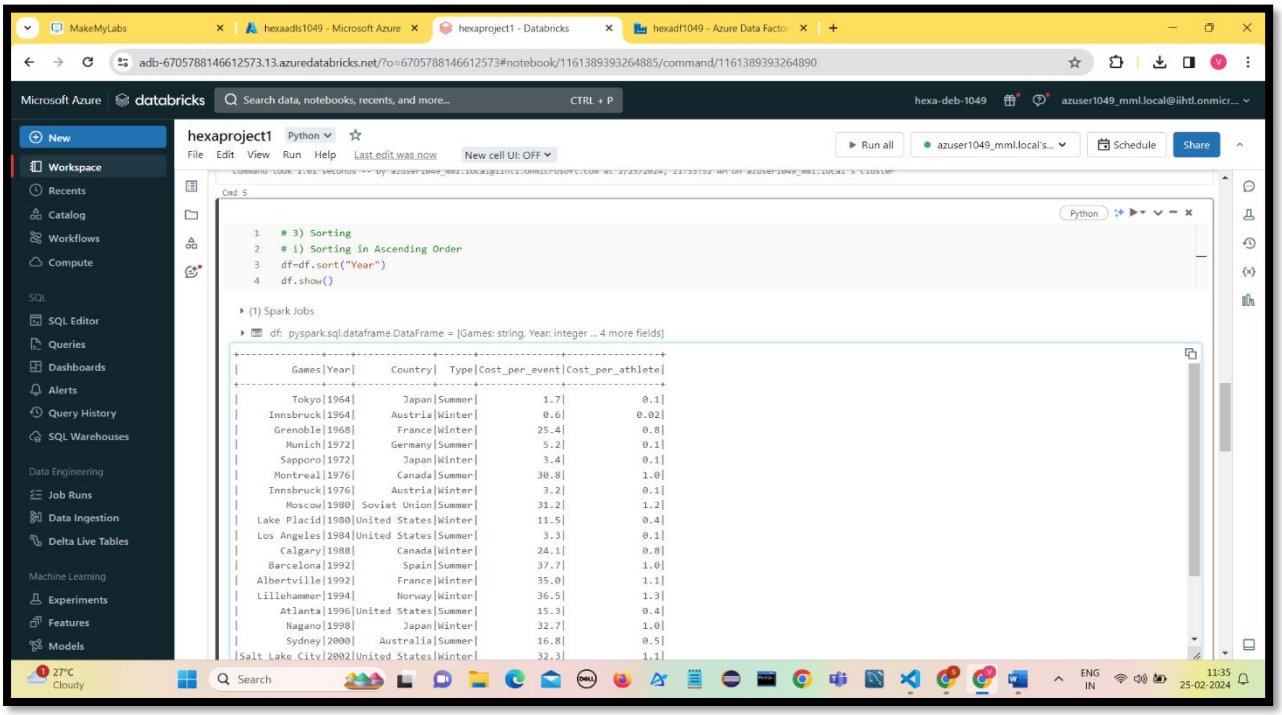
2) Dropping a column

Drop Year column from the data and display the result.

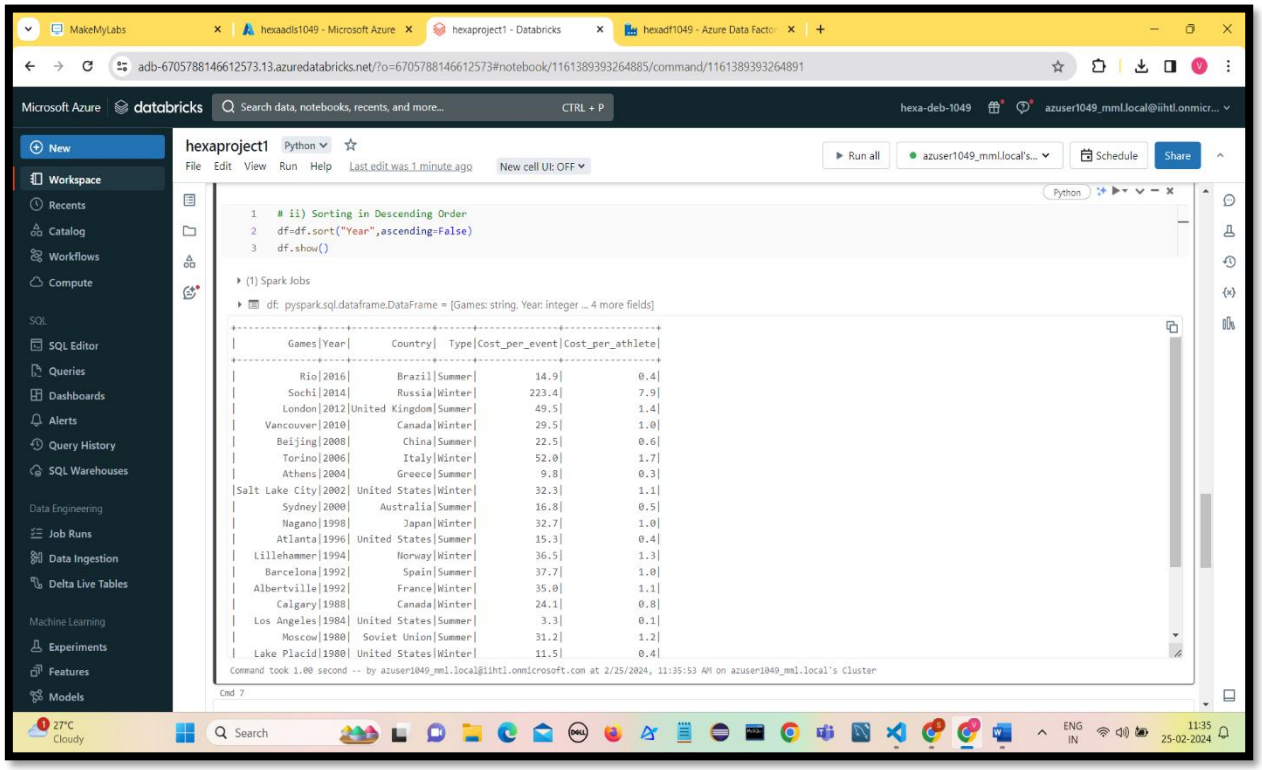


3) Sorting

Sort the Year column in Ascending Order.

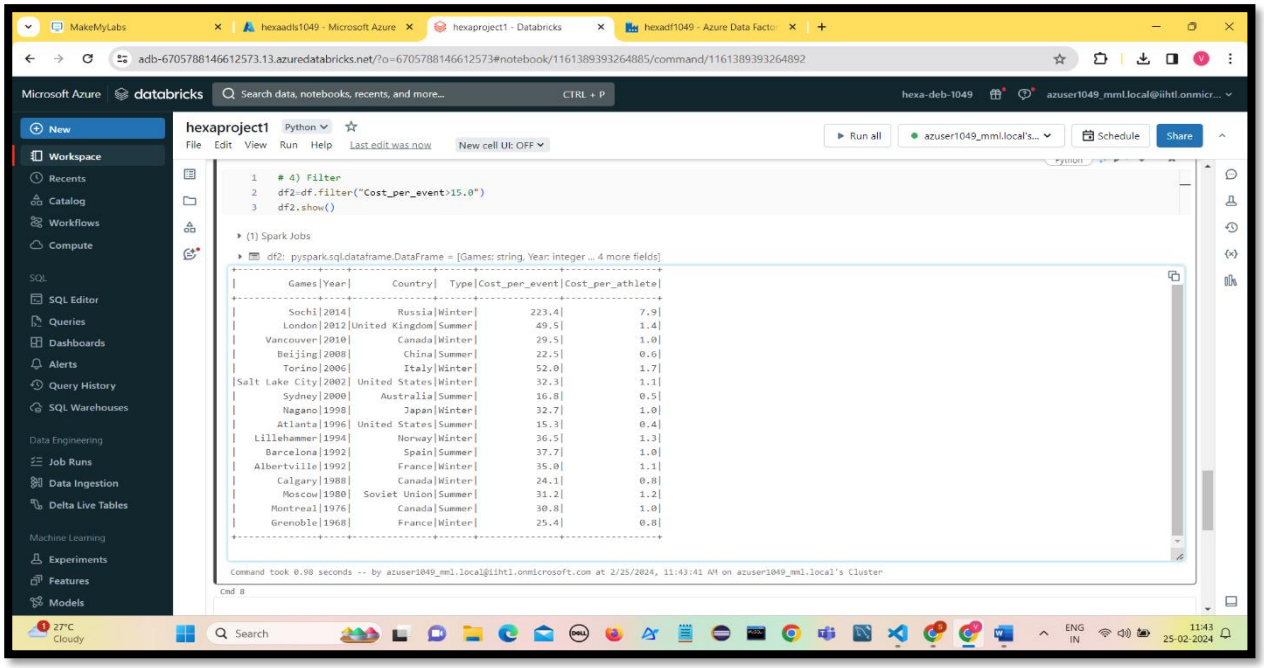


Sort the year in Descending Order using ascending=False



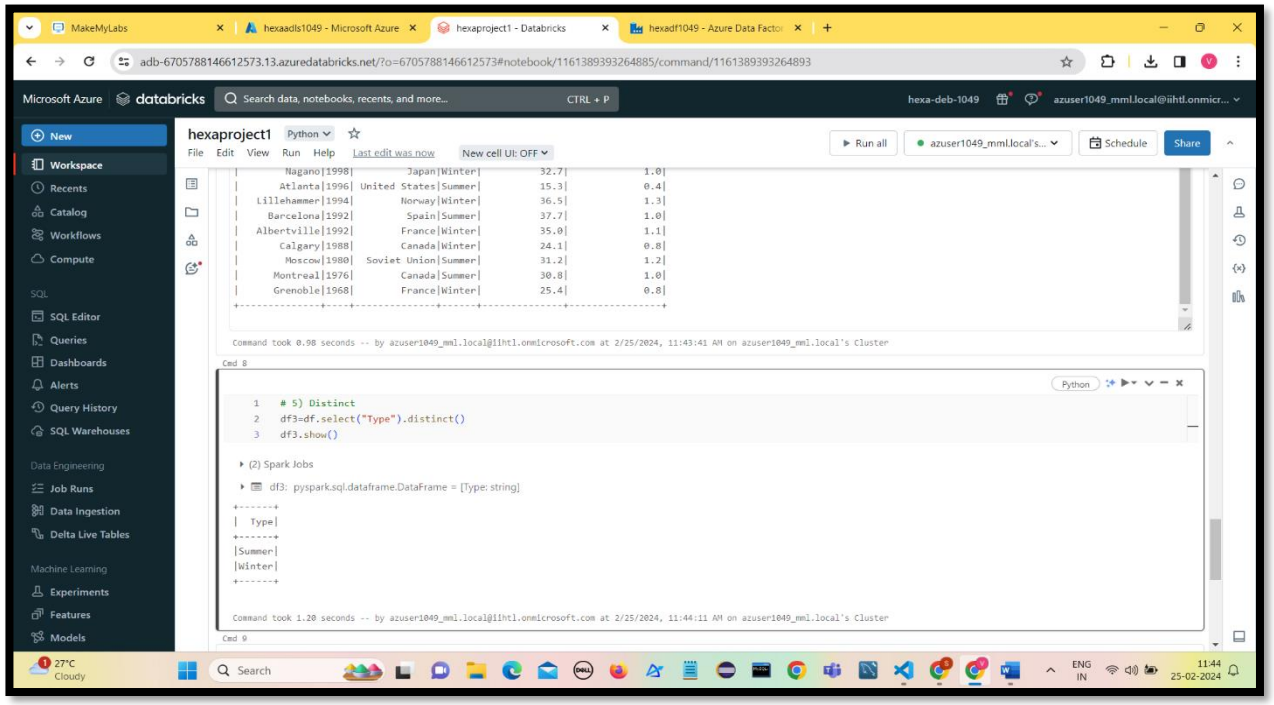
4) Filtering

Filter the Olympics Data where Cost_per_event>15.0



5) Distinct

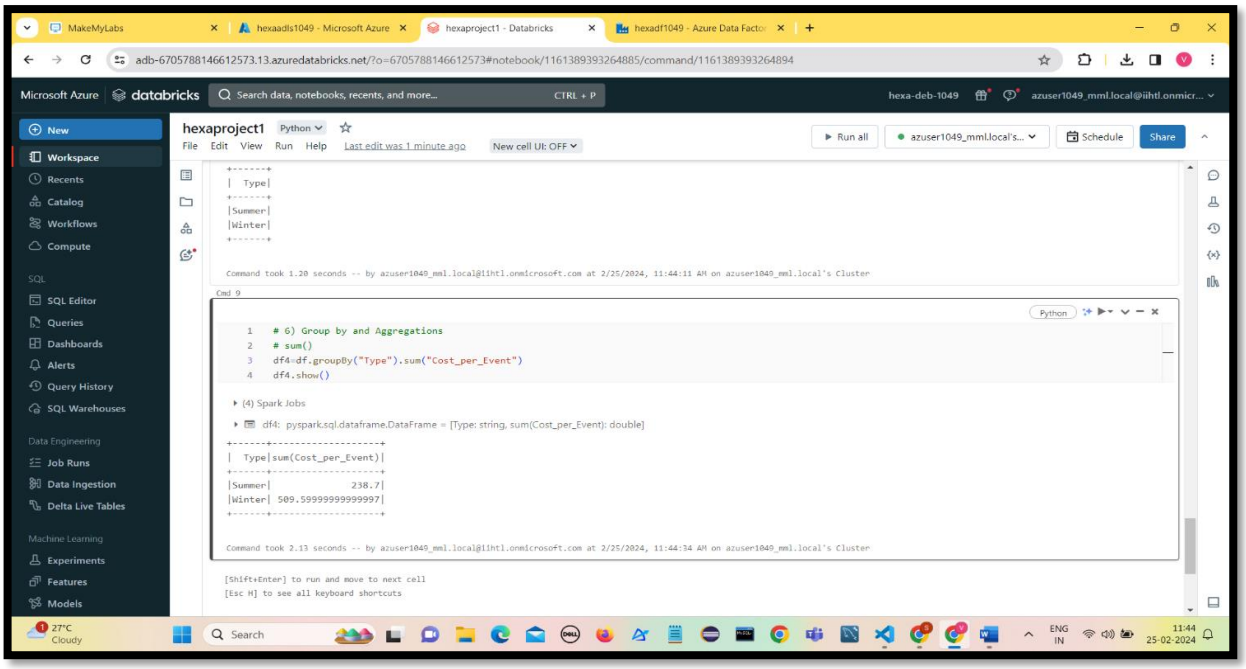
Selecting the distinct values of a Type column



6) Group By and Aggregations

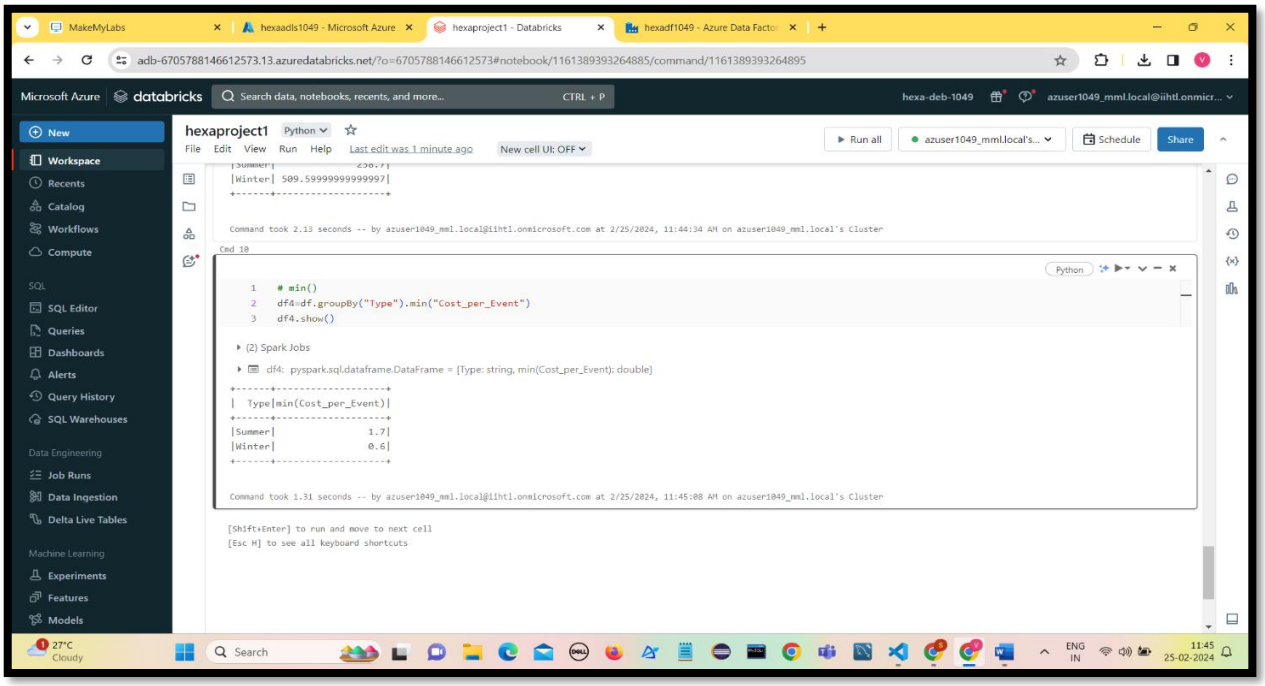
1) sum()

Find the total amount of Cost_per_Event for each Type.



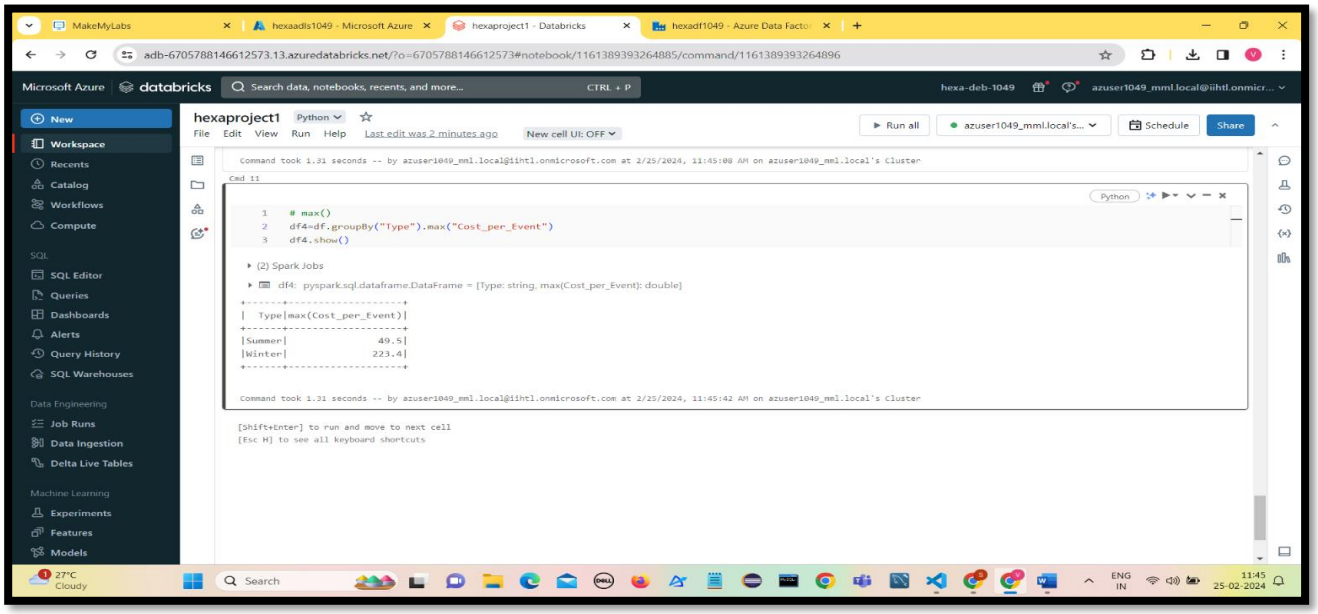
2) min()

Find the minimum of Cost_per_Event for each Type.



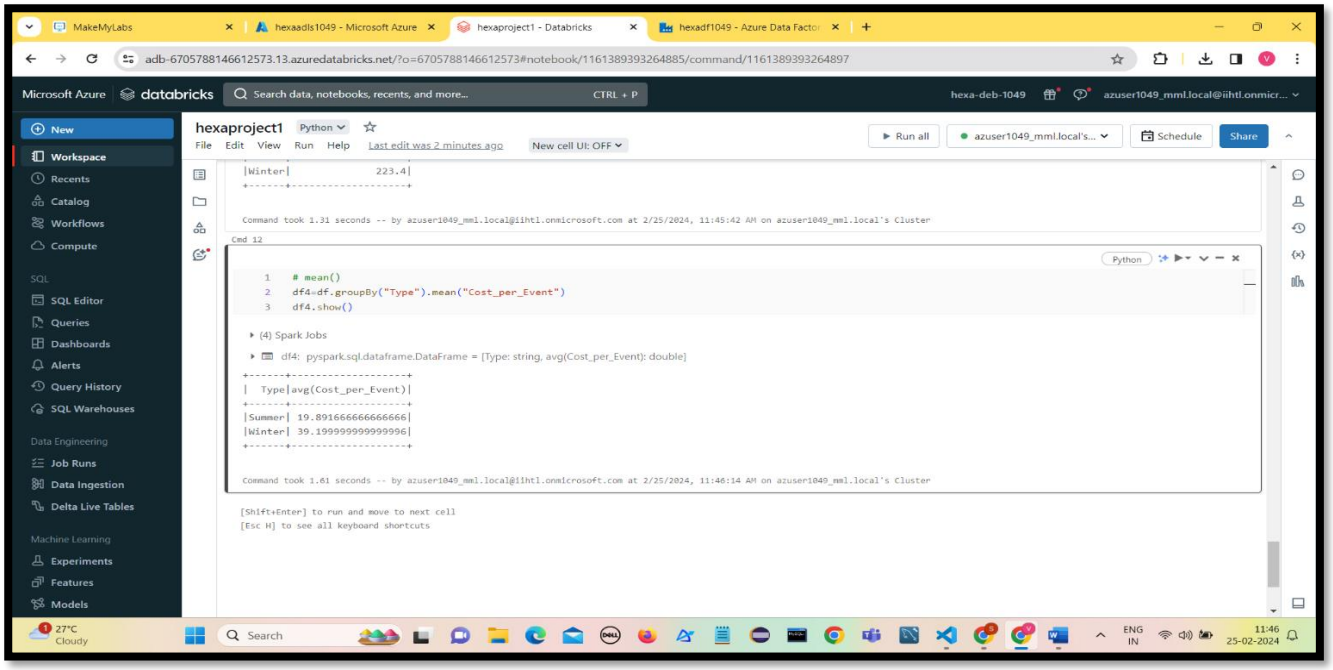
3) max()

Find the Maximum of Cost_per_Event for each Type.



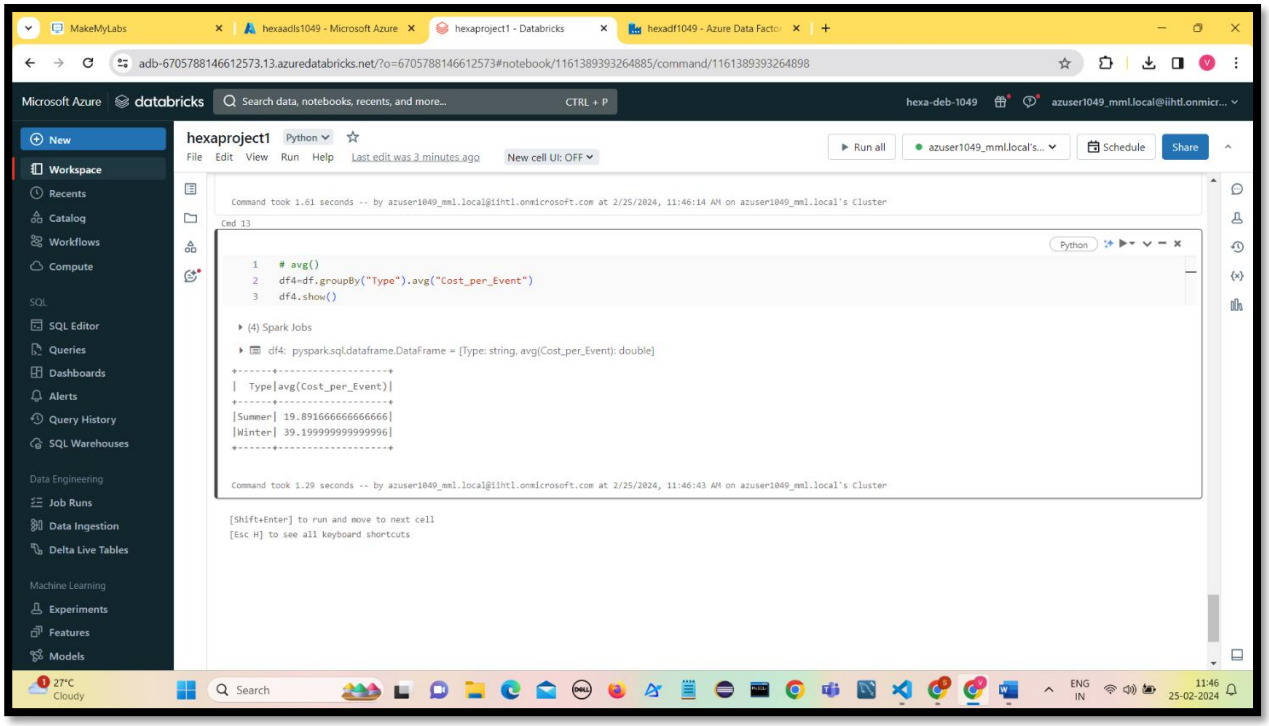
4) mean()

Find the Mean of Cost_per_Event for each Type.



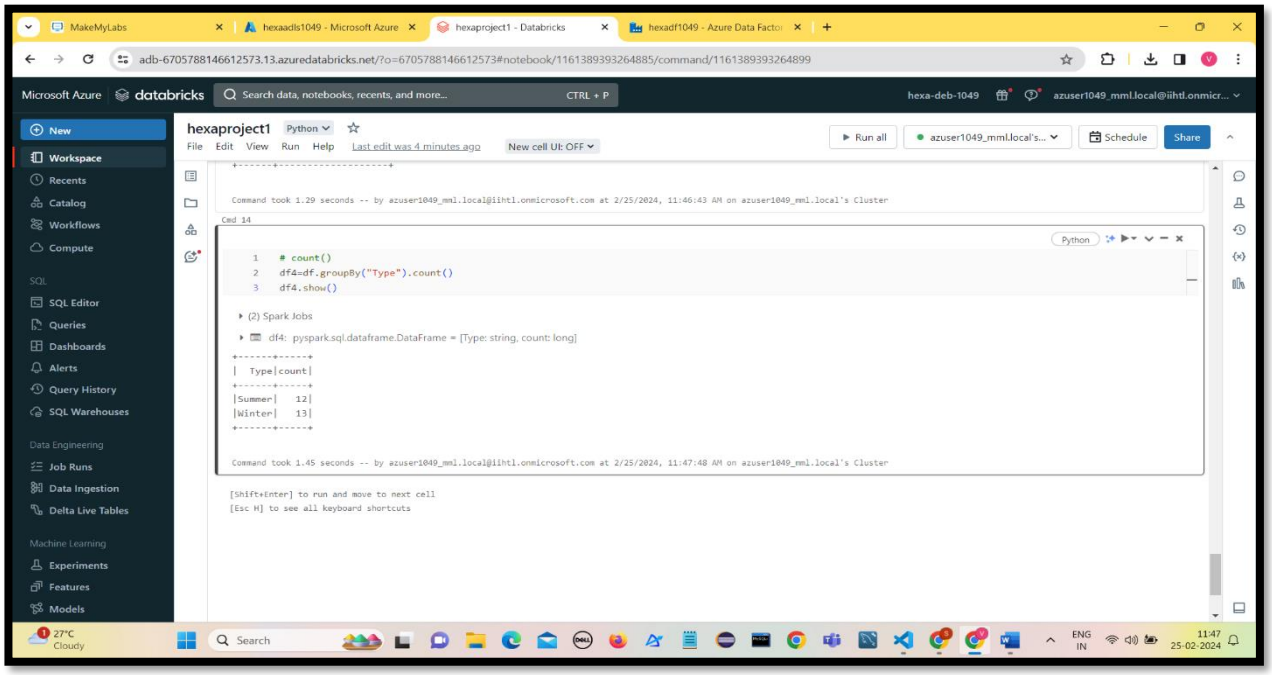
5) avg()

Find the Average of Cost_per_Event for each Type.



6) count()

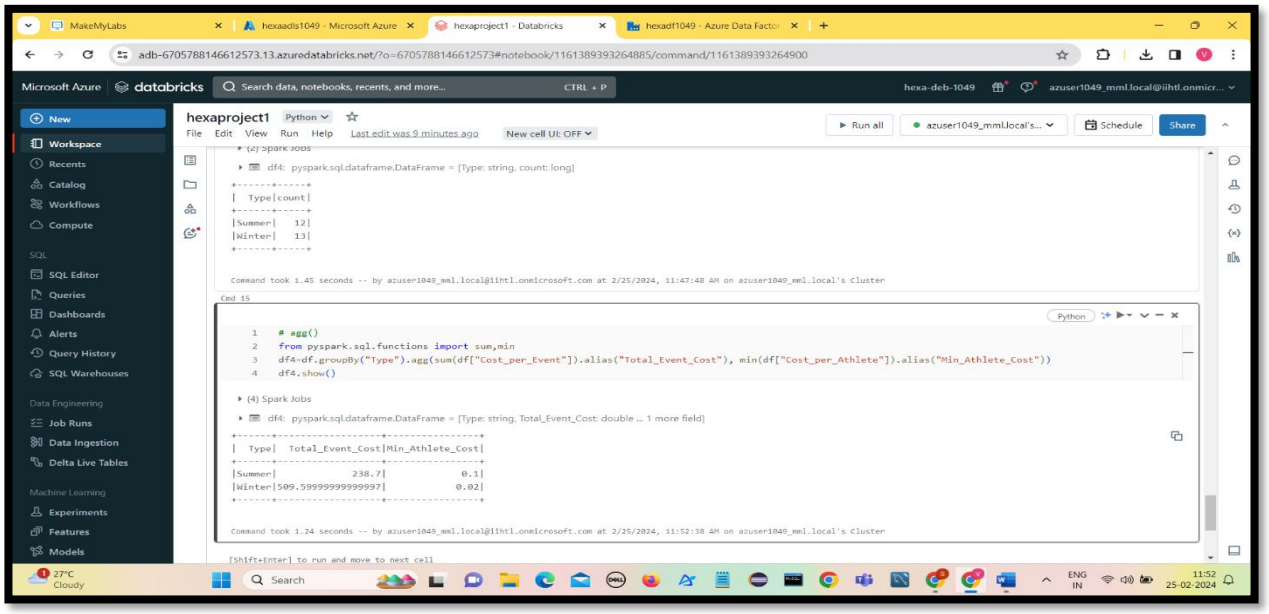
Find the count (Number of rows) for each Type.



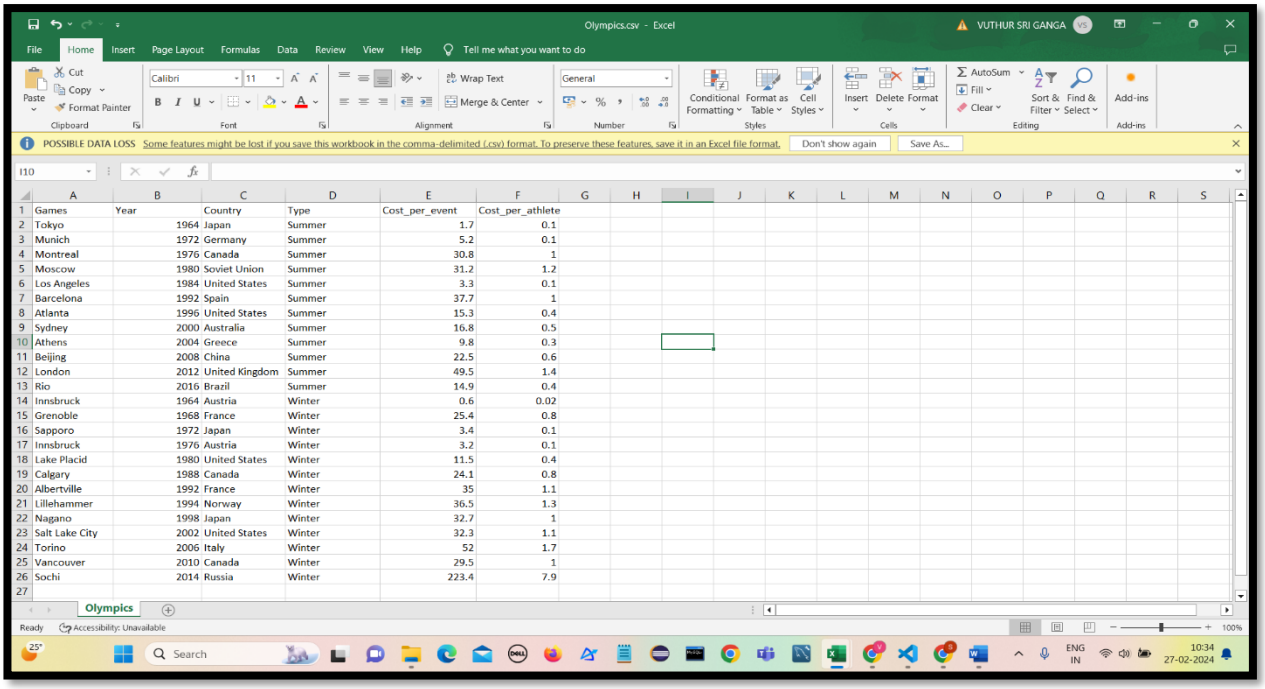
7) agg()

Using more than one aggregates at a time

Find the sum of Cost_per_Event and minimum of Cost_per_Athlete



Dataset used:



Games	Year	Country	Type	Cost_per_event	Cost_per_athlete
Tokyo	1964	Japan	Summer	1.7	0.1
Munich	1972	Germany	Summer	5.2	0.1
Montreal	1976	Canada	Summer	30.8	1
Moscow	1980	Soviet Union	Summer	31.2	1.2
Los Angeles	1984	United States	Summer	3.3	0.1
Barcelona	1992	Spain	Summer	37.7	1
Atlanta	1996	United States	Summer	15.3	0.4
Sydney	2000	Australia	Summer	16.8	0.5
Athens	2004	Greece	Summer	9.8	0.3
Beijing	2008	China	Summer	22.5	0.6
London	2012	United Kingdom	Summer	49.5	1.4
Rio	2016	Brazil	Summer	14.9	0.4
Innsbruck	1964	Austria	Winter	0.6	0.02
Grenoble	1968	France	Winter	25.4	0.8
Sapporo	1972	Japan	Winter	3.4	0.1
Innsbruck	1976	Austria	Winter	3.2	0.1
Lake Placid	1980	United States	Winter	11.5	0.4
Calgary	1988	Canada	Winter	24.1	0.8
Albertville	1992	France	Winter	35	1.1
Lillehammer	1994	Norway	Winter	36.5	1.3
Nagano	1998	Japan	Winter	32.7	1
Salt Lake City	2002	United States	Winter	32.3	1.1
Torino	2006	Italy	Winter	52	1.7
Vancouver	2010	Canada	Winter	29.5	1
Sochi	2014	Russia	Winter	223.4	7.9

Conclusion:

This project successfully demonstrated the implementation of a robust data pipeline for ingesting Olympics data into Azure Data Lake Storage Gen2 using Azure Data Factory (ADF) pipelines and performing data transformation tasks using PySpark notebooks in Azure Databricks. The project enables stakeholders to gain valuable insights and analysis from the processed data, contributing to data-driven decision-making and business intelligence efforts.

Vuthur Sriganga

Data Engineering Batch-1

vuthursriganga@gmail.com