NAME:MD SARFARAZ ALAM

EMAIL-ID:SAMALAMABD17@GMAIL.COM

**PROJECT-1**

**Data Lake Exploration and Optimization:**

**Use PySpark to explore and optimize data stored in a data lake. Leverage Azure Databricks for efficient querying and analysis of data lake contents.**

Project Overview:

Objective:

The main goal of this project is to explore and optimize data stored in a Data Lake using PySpark on Azure Databricks. This involves tasks such as data exploration, cleaning, transformation, and optimizing queries for efficient analysis.

Tools and Technologies:

**Azure Databricks:** Leveraging the Databricks workspace for easy collaboration and scalable data processing.

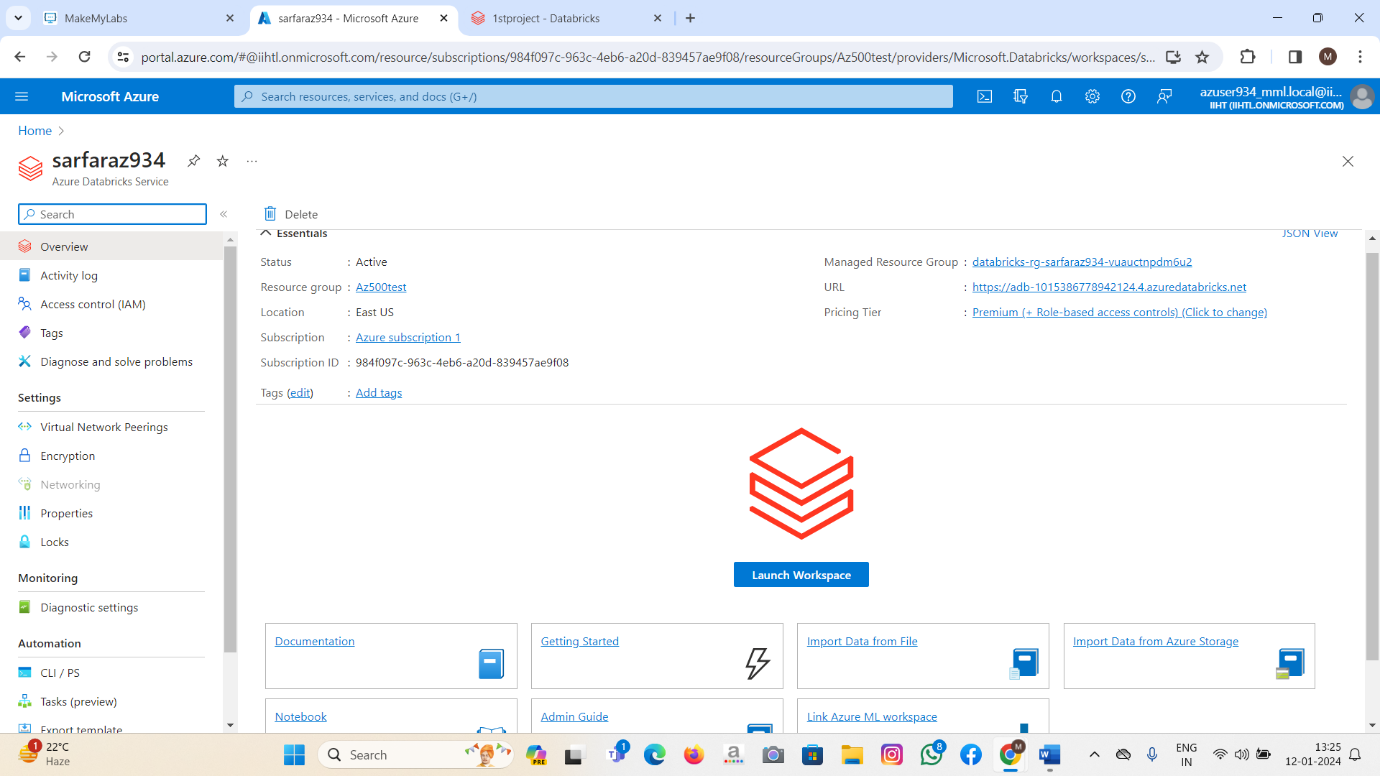
**PySpark:** Using PySpark, the Python API for Apache Spark, for data manipulation and analysis.

**Azure Data Lake Storage:** Storing and managing the data lake in Azure Data Lake Storage.

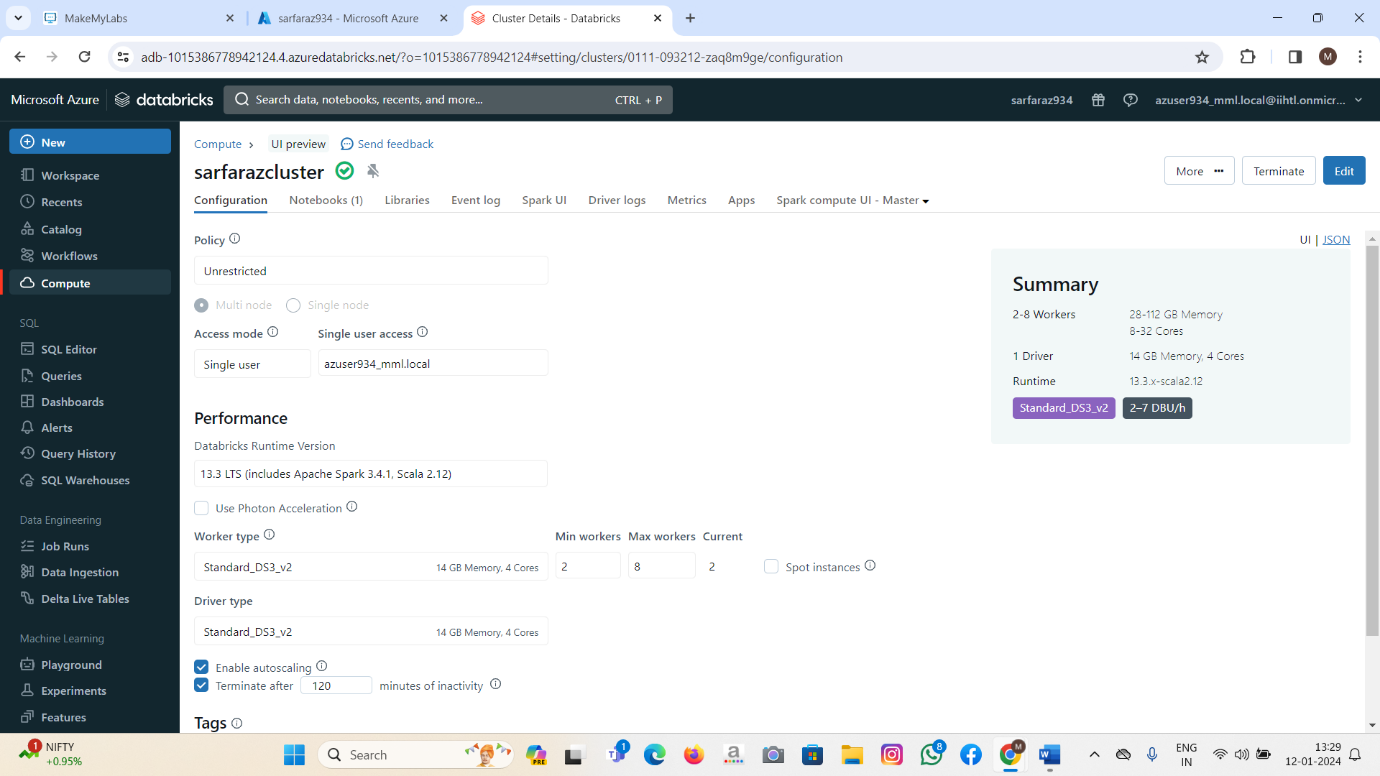
**Project Steps:**

**Step 1: Setup Azure Databricks Workspace**

* 1. Create Databricks Workspace:



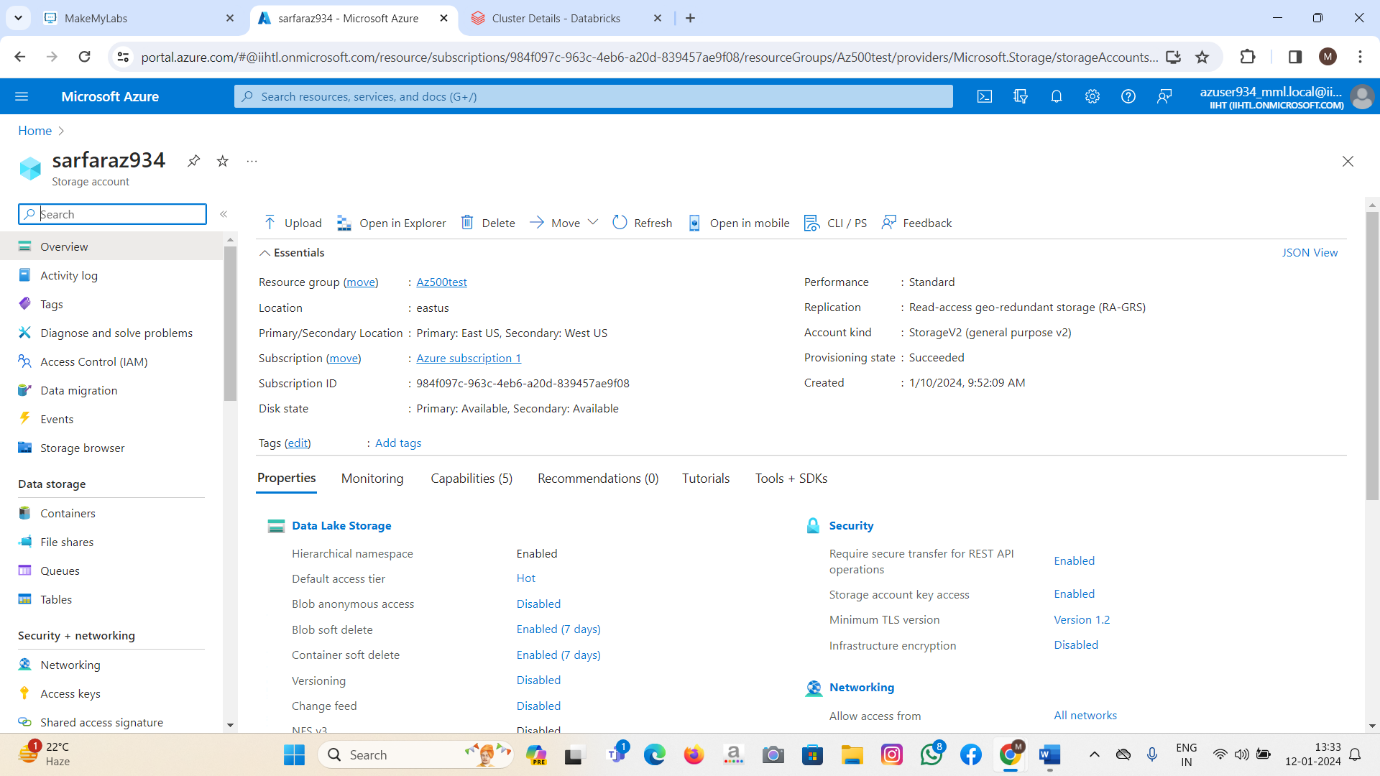
* Created an Azure Databricks workspace with appropriate configurations.
  1. Configure Cluster:
* Set up a Databricks cluster with required specifications.



**Step 2: Connect to Azure Data Lake Storage**

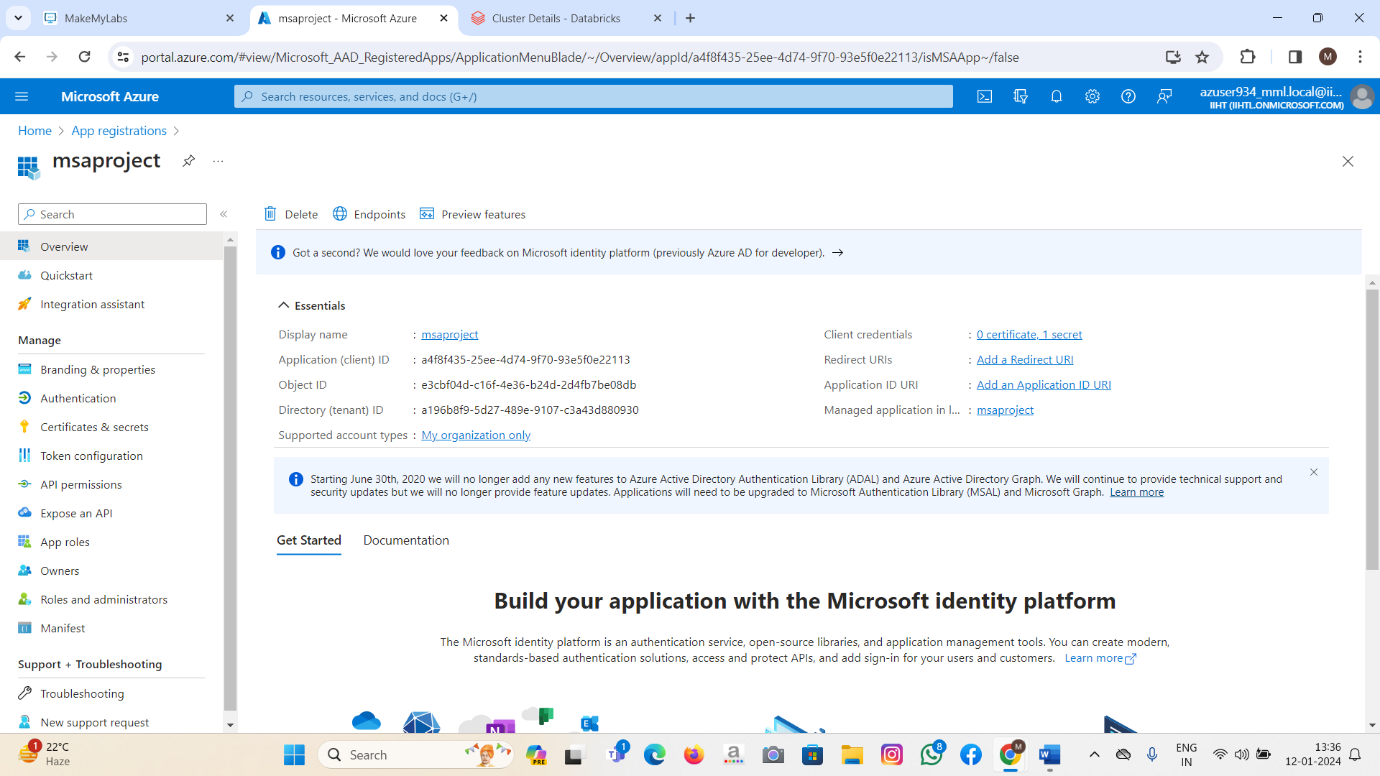
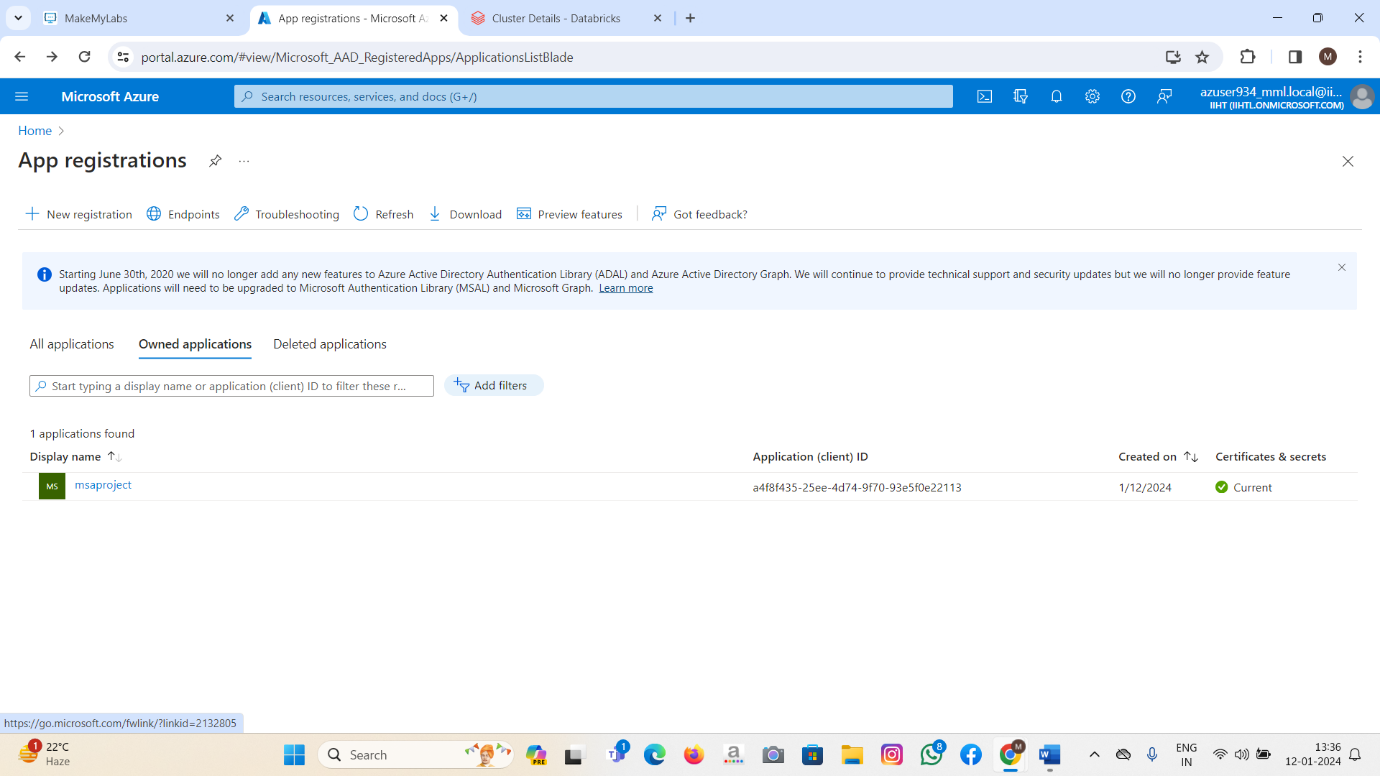
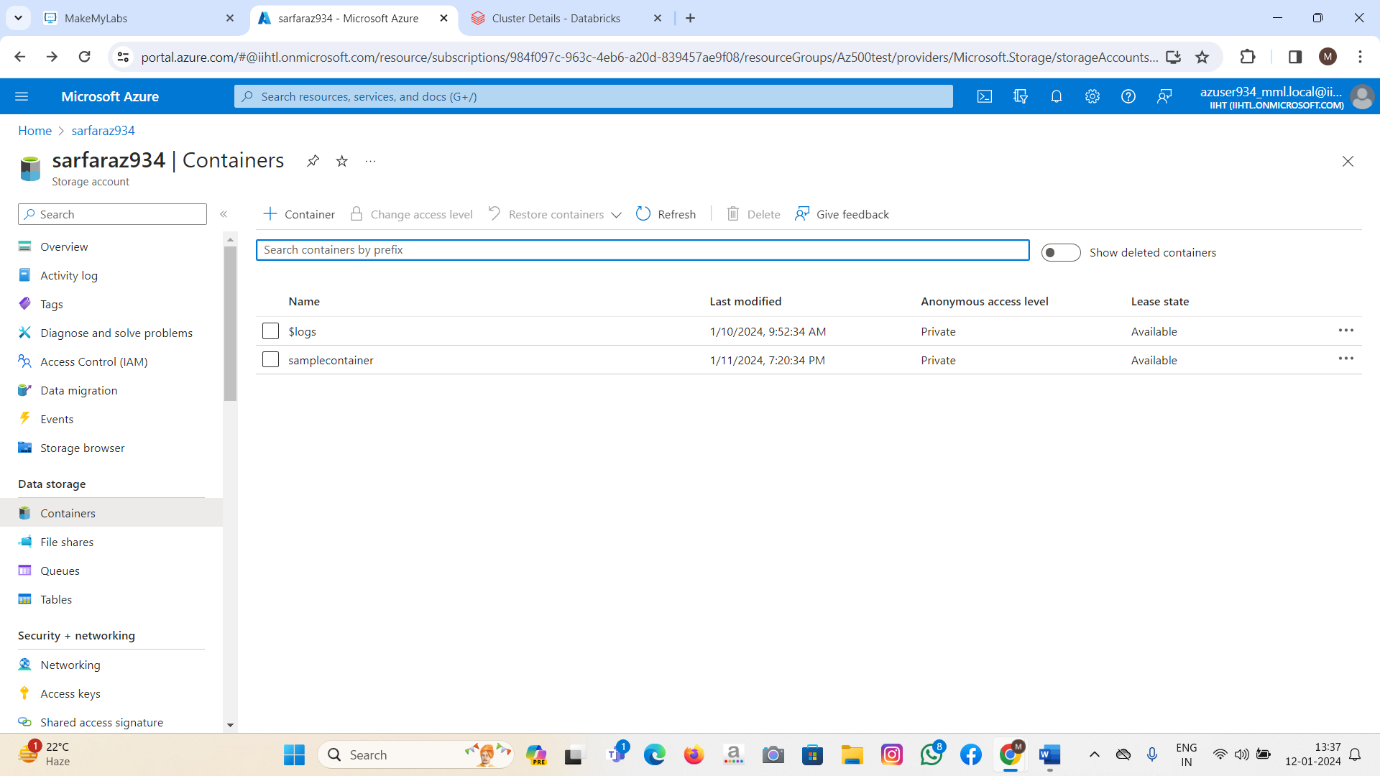
2.1. Connect Databricks to Data Lake Storage:

* Established a connection between the Databricks workspace and Azure Data Lake Storage.



2.2. Mount Data Lake Storage:

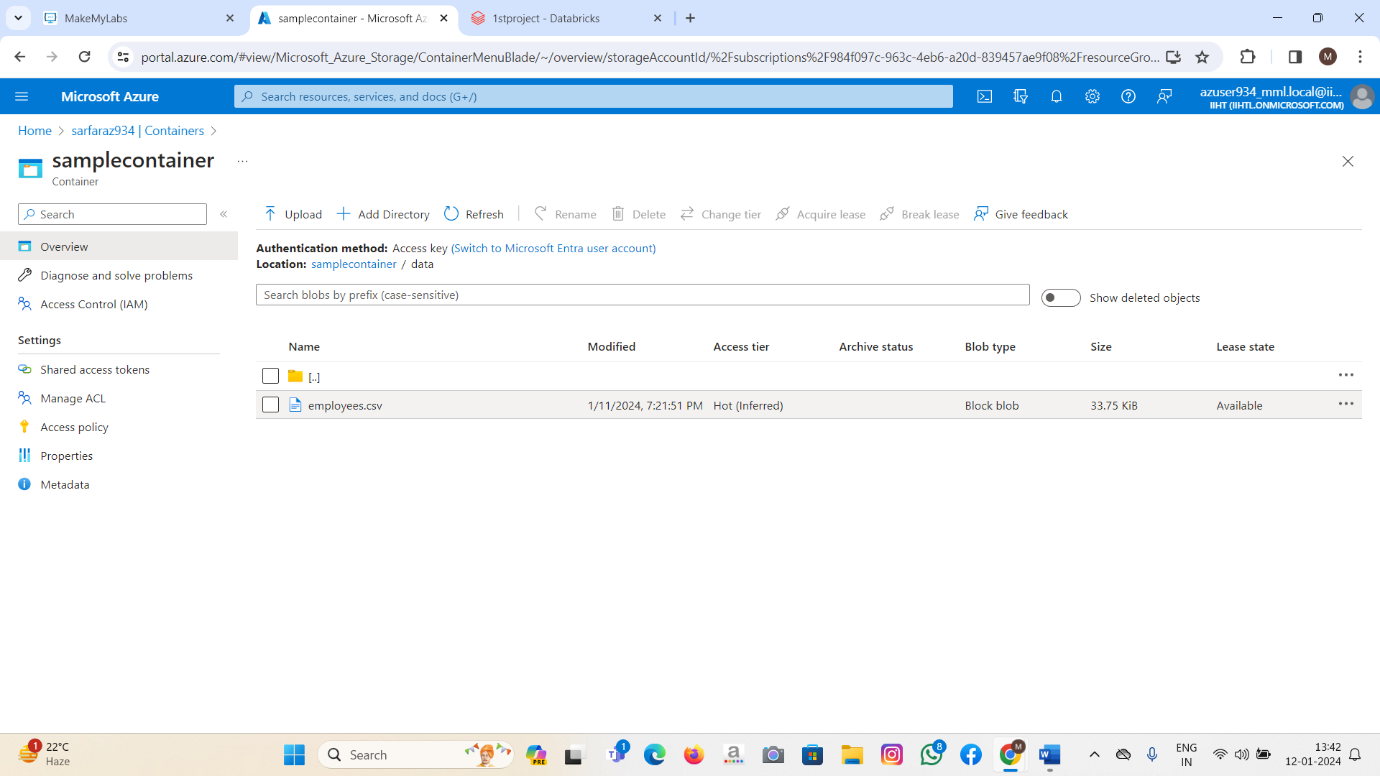
* Mounted the Data Lake Storage to make it accessible within the Databricks workspace.



**Step 3: Load Data into Databricks**

3.1. Load Sample Data:

* Loaded sample data from the Data Lake Storage into the Databricks workspace.



3.2. Explore Data:

* Used PySpark to read and explore the structure of the loaded data.

**Step 4: Data Exploration and Cleaning**

4.1. Explore Data Characteristics:

* Utilized PySpark functions to explore the structure and characteristics of the data.

4.2. Handle Missing Values and Outliers:

* Identified and handled missing values, outliers, and other data quality issues.

**Step 5: Data Transformation and Optimization**

5.1. Apply Transformations:

* Applied PySpark transformations to clean and structure the data.

5.2. Optimize Data Types and Columns:

* Optimized data types, column names, and other factors to improve query performance.

**Step 6: Query Optimization**

6.1. Use Spark for Querying:

* Leveraged Spark SQL for efficient querying of the data.

6.2. Optimize Queries:

* Implemented query optimization techniques such as caching, partitioning, and indexing.

**Step 7: Advanced Analysis and Visualization**

7.1. Perform Advanced Analytics:

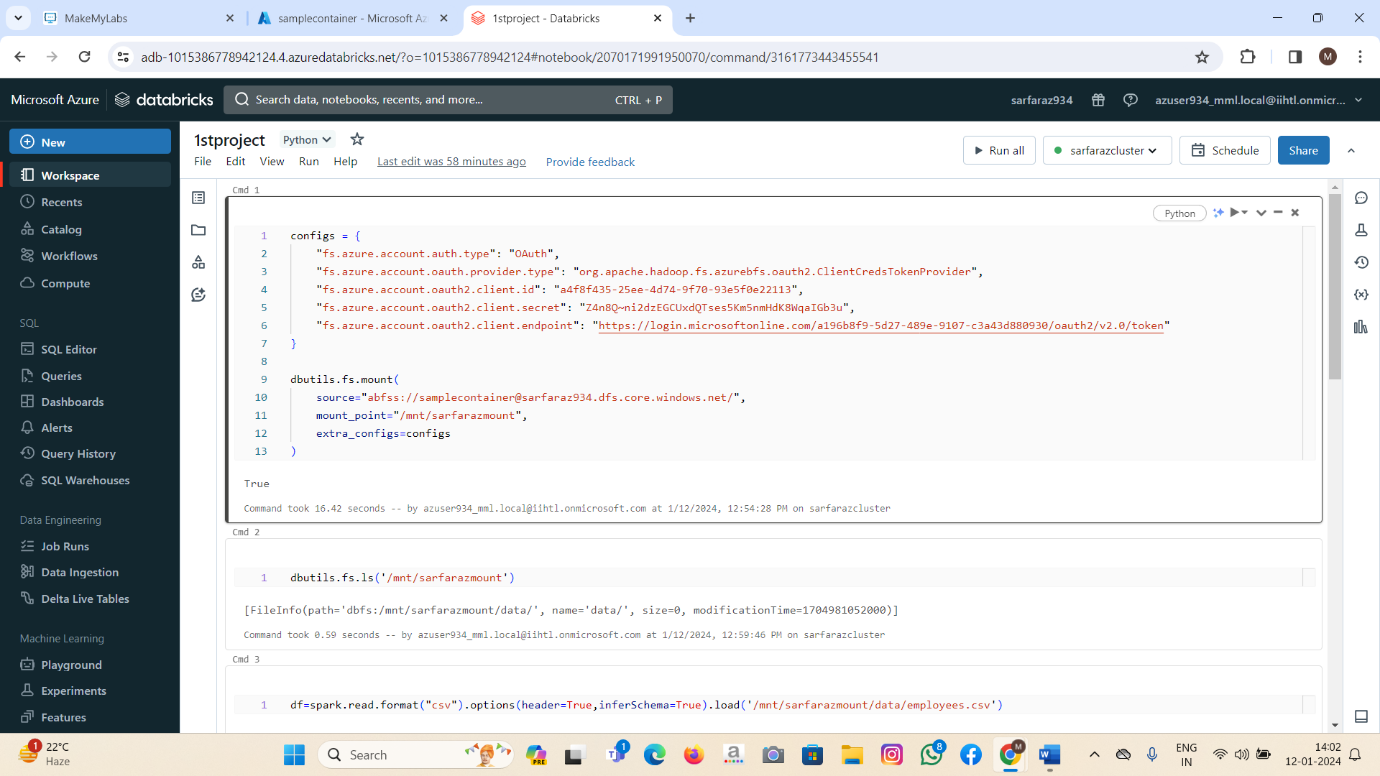
* Utilized PySpark for advanced analytics on the optimized data.

7.2. Visualize Results:

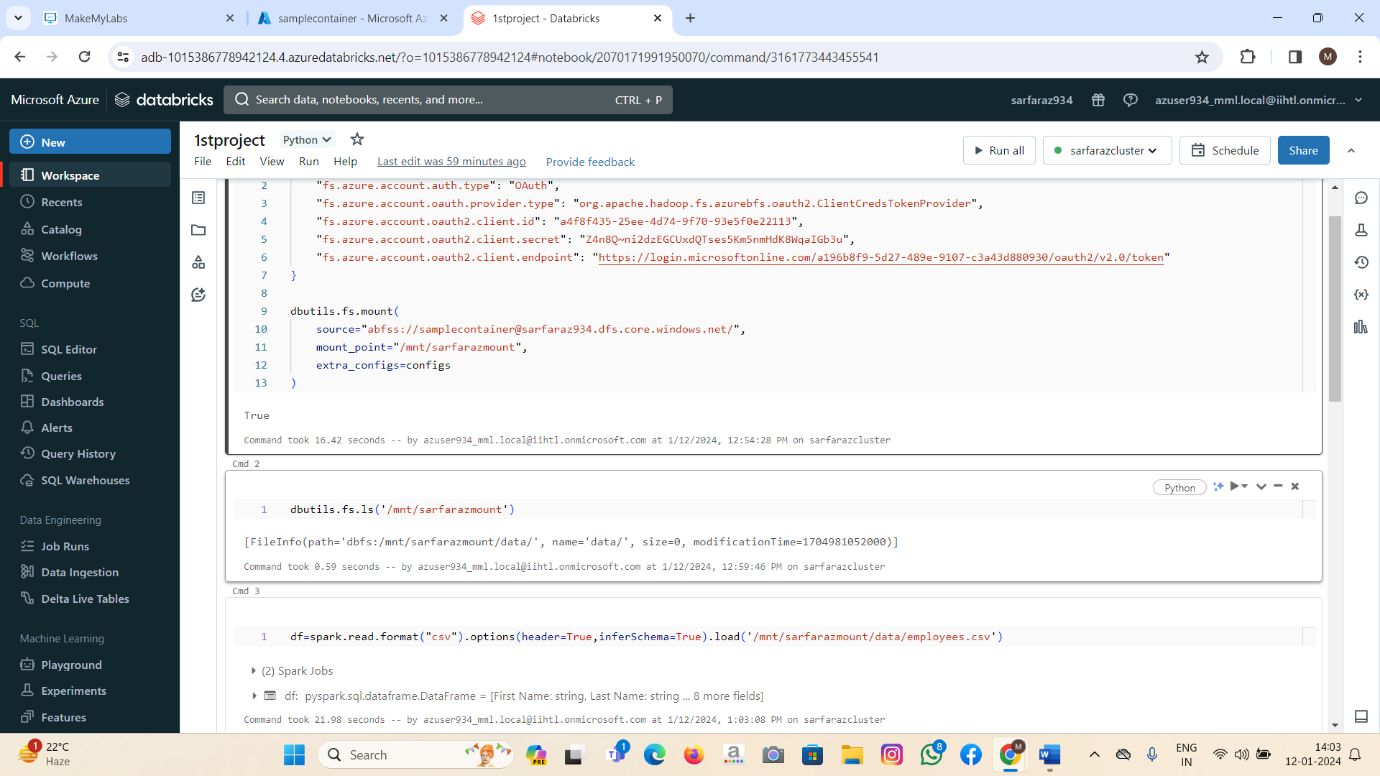
* Created visualizations using Databricks notebooks and libraries.

**Step 8: Working On Notebooks**

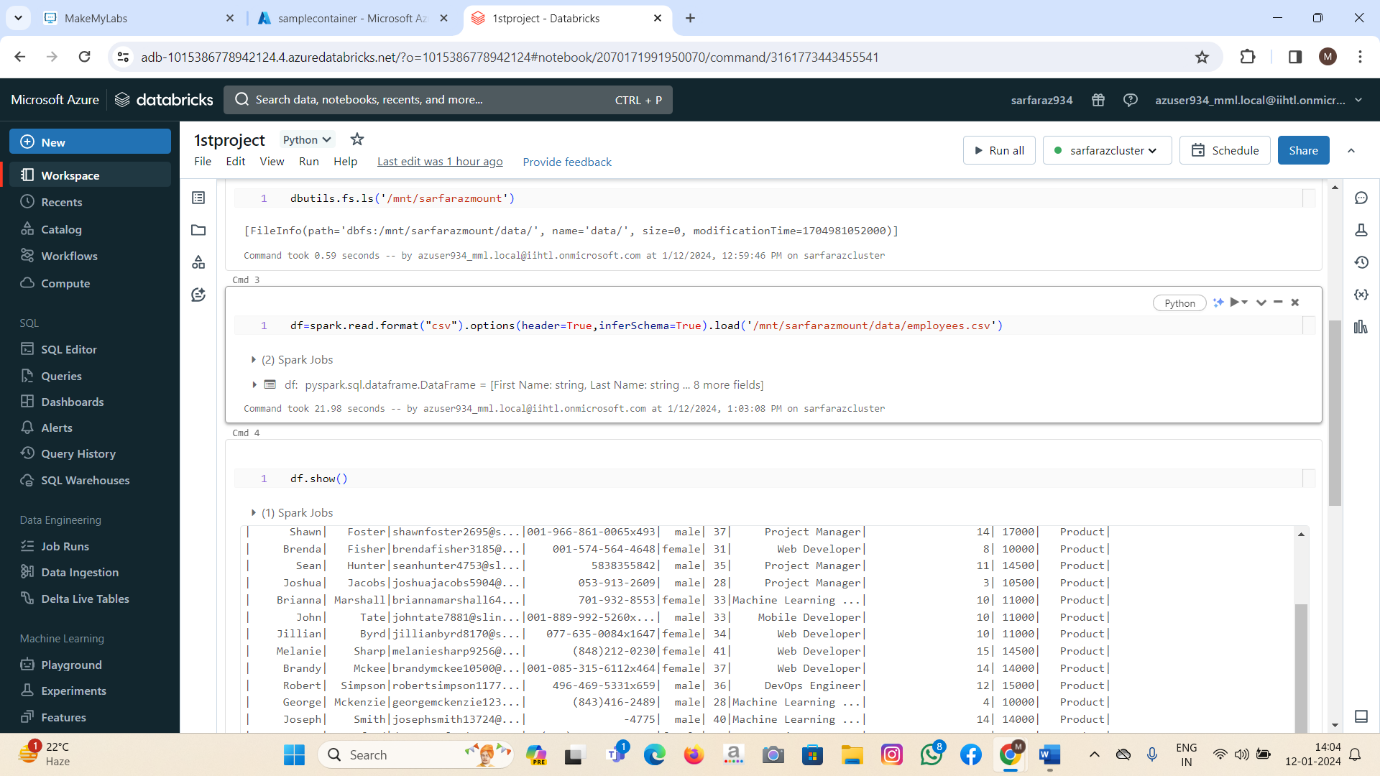
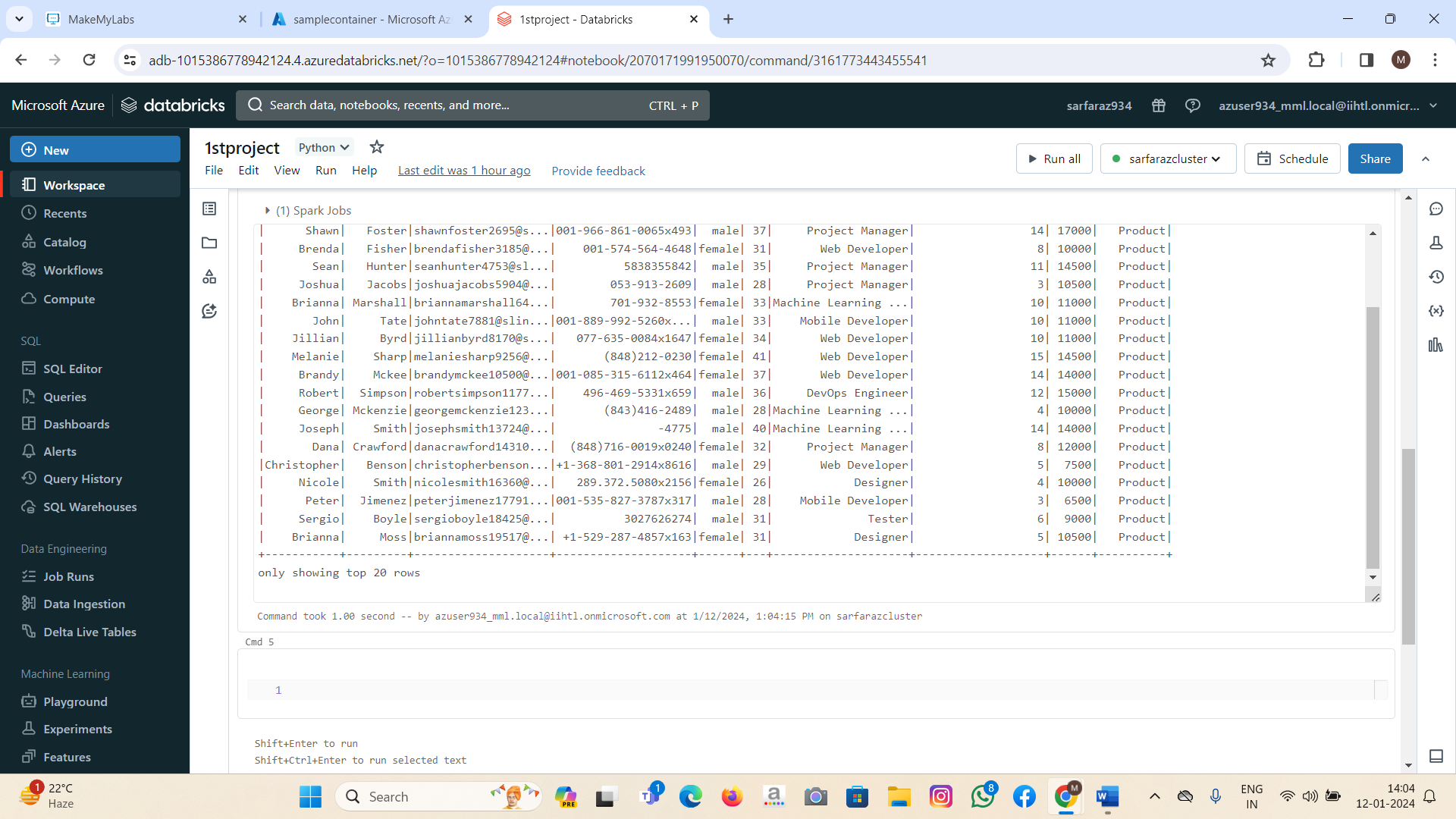
8.1. Implementing coding queries on the notebook



8.2. After running the Queries:



**Output:**

****

**Conclusion:**

This project provided a comprehensive approach to exploring and optimizing data in a Data Lake using PySpark on Azure Databricks. The steps covered data cleaning, transformation, query optimization, and advanced analysis, ensuring that the data is well-prepared for efficient and insightful analysis.