## 

## Project Report:

## Data Quality Monitoring System

## By-

## Siddhant Mundhe

## 

|  |  |  |  |
| --- | --- | --- | --- |
| **Sno** |  | **Topic** | **Pg.No.** |
| 1 |  | ProjectStatement | 3 |
| 2 |  | ProjectOverview | 3 |
| 3 |  | ProjectRequirements | 3 |
| 4 |  | Architecture | 6 |
| 5 |  | Tools Used | 7 |
| 6 |  | Execution Overview | 8 |
| 7 |  | How it Works | 9 |
| 8 |  | Tasks Performed | 10 |
| 9 |  | Outputs | 11 |
|  | 9.1 | Setting up AzureDatabricks | 8 |
|  | 9.2 | Creating storage Account(ADLS Gen2) | 10 |
|  | 9.3 | Ingestion ADLS with Databricks | 12 |
|  | 9.4 | Data Profiling | 14 |
|  | 9.5 | Data Quality | 17 |
|  | 9.6 | Continuous Monitoring | 24 |
|  | 9.7 | Generating Alerts | 26 |
| 10 |  | Conclusion | 38 |

## Project Statement:-

## Develop a data quality monitoring system using PySparkSQL for data profiling and quality assessments & explain a developed process for Azure Databricks for continuous monitoring of data quality & generate alerts for anamolies.

## Project Overview:

The Data Quality Monitoring System aims to ensure the reliability and accuracy of data by continuously monitoring data quality metrics and detecting anomalies in real-time. This system leverages Azure Databricks for data processing and PySparkSQL for data profiling and anomaly detection.

* **Project Requirements**

### Functional Requirements:

1. **Data Ingestion:**
   * Ingest data from one or more sources into Azure Databricks. Supported sources may include Azure Data Lake Storage, Azure Blob Storage, or other compatible data sources.
   * Ensure that the ingested data is accessible and can be queried using PySparkSQL.
2. **Data Profiling:**
   * Develop PySparkSQL scripts to perform data profiling tasks.
   * Profile the data to identify patterns, outliers, missing values, data distributions, and other relevant statistics.
   * Generate summary statistics and insights about the data's structure, quality, and characteristics.
3. **Quality Assessment:**
   * Implement PySparkSQL scripts to assess data quality based on predefined metrics and rules.
   * Evaluate data quality aspects such as completeness, accuracy, consistency, and timeliness.
   * Identify and flag potential data quality issues or anomalies within the dataset.
4. **Continuous Monitoring:**
   * Schedule PySparkSQL scripts as Databricks Jobs to run at regular intervals for continuous monitoring.
   * Ensure that data quality assessments are performed on new data as it arrives or at predefined intervals.
5. **Alerting:**
   * Implement alerting mechanisms to notify stakeholders of any detected data quality issues or anomalies.
   * Configure alert thresholds and notification channels for timely response to critical data quality issues.
   * Integrate with external alerting services if necessary, such as Azure Monitor or email alerts.

### Non-Functional Requirements:

1. **Scalability:**
   * Ensure that the data quality monitoring system can scale to handle large volumes of data efficiently.
   * Design the system architecture to accommodate future growth and increased data complexity.
2. **Reliability:**
   * Build the system with robust error handling and fault tolerance mechanisms.
   * Minimize the risk of data loss or corruption during data processing and monitoring activities

.

1. **Performance:**
   * Optimize PySparkSQL scripts and data processing workflows for performance.
   * Aim to minimize processing latency and maximize throughput to meet monitoring frequency requirements.
2. **Security:**
   * Implement appropriate security measures to protect sensitive data and ensure compliance with regulatory requirements.
   * Utilize Azure Databricks' built-in security features, such as role-based access control (RBAC) and data encryption.
3. **Maintainability:**
   * Structure the codebase in a modular and maintainable manner to facilitate future updates and enhancements.
   * Document the system architecture, data flows, and codebase to aid in troubleshooting and knowledge transfer.
4. **Usability:-**
   * Design user-friendly interfaces for configuring, monitoring, and accessing data quality metrics and reports.
   * Provide documentation and training materials to assist users in understanding and using the system effectively.

* **Architecture Diagram:-**



* **TOOLS /TECHNOLOGIES USED**

In the project structure provided, the main tools and technologies used are:

1. **PySparkSQL**:

PySparkSQL is a Python API for Apache Spark that allows you to work with structured data using SQL queries. We can use PySparkSQL for data profiling, quality assessment, and data transformation tasks.

1. **Azure Databricks**:

Azure Databricks is a cloud-based big data platform built on Apache Spark. It provides an integrated environment for data engineering, data science, and analytics. We can use Azure Databricks for data processing, continuous monitoring of data quality, and for running PySparkSQL jobs.

1. **Azure Data Lake Storage**:

Azure Data Lake Storage is a scalable and secure data lake solution provided by Microsoft Azure. We can use it to store raw and processed data for your data quality monitoring system.

1. **Azure Monitor**:

Azure Monitor is a monitoring and logging service provided by Azure. We can use it to monitor the performance and health of your Azure Databricks clusters and jobs. Additionally, We can set up alerts in Azure Monitor to notify you of any anomalies detected in your data quality metrics.

* **EXECUTION OVERVIEW:-**

1. **Data Profiling**:
   * Data profiling is performed using PySparkSQL to analyze the structure, content, and quality of the datasets.
   * Various profiling techniques are applied to understand the characteristics of the data, such as summary statistics, data distributions, data types, and uniqueness constraints.
   * Key profiling metrics are computed for each dataset, including:
     + Count of rows
     + Count of distinct values
     + Minimum and maximum values
     + Data types
     + Null value counts
     + Data distribution histograms
2. **Quality Assessment**:
   * Quality assessment metrics are calculated to evaluate the data quality based on predefined criteria.
   * Metrics such as completeness, accuracy, consistency, timeliness, and uniqueness are computed for each dataset.
   * Quality rules and thresholds are defined to identify anomalies or deviations from expected data quality standards.
   * Data quality scores are generated for each metric, indicating the level of adherence to quality standards.
3. **Continuous Monitoring**:
   * Azure Databricks is utilized for continuous monitoring of data quality in near real-time.
   * Scheduled jobs or streaming pipelines are set up to monitor incoming data for anomalies or deviations from expected patterns
4. **Alert Generation:-**
   * Automated alerts or notifications are triggered when data quality issues are detected, based on predefined thresholds or rules.
   * Monitoring dashboards provide real-time visibility into data quality metrics and trends, enabling proactive intervention when necessary.

* **HOW IT WORKS ?**

1. **Data Collection**: The system retrieves data from various sources, such as databases, data lakes, or streaming platforms. This data could be structured, semi-structured, or unstructured.
2. **Data Processing**: Once the data is collected, it undergoes processing to clean, transform, and prepare it for analysis. This step may involve tasks like data normalization, handling missing values, and removing duplicates.
3. **Data Profiling**: Using PySparkSQL, the system performs data profiling to gain insights into the structure, content, and quality of the data. This involves analyzing data distributions, identifying patterns, and computing summary statistics.
4. **Data Quality Assessment**: Based on the insights gained from data profiling, the system assesses the quality of the data. It checks for anomalies, inconsistencies, and deviations from expected patterns. Common data quality checks include completeness, accuracy, consistency, and timeliness.
5. **Continuous Monitoring**: The data quality monitoring system operates continuously, periodically collecting, processing, profiling, and assessing data. This ensures that data quality issues are detected in a timely manner and can be addressed promptly.
6. **Alert Generation**: If anomalies or data quality issues are detected, the system generates alerts to notify stakeholders. These alerts can be sent via email, Slack, or integrated with other notification systems. The alerts provide details about the issues found and may include recommendations for corrective actions.
7. **Feedback Loop**: Feedback from stakeholders and the outcomes of corrective actions are incorporated back into the system. This feedback loop helps improve the effectiveness and accuracy of data quality assessments over time.

resources:

* **TASKS PERFORMED**

1. **Integration with Azure Services**: ADB seamlessly integrates with other Azure services like Azure Data Lake Storage, Azure Blob Storage, , allowing you to read/write data from/to these services as part of your data quality monitoring pipeline.
2. **Data Processing with PySpark**: ADB provides an environment where we can run PySpark jobs for data processing tasks, including data profiling, data cleaning, and transformation.
3. **Job Scheduling**: We schedule PySpark jobs to run at specific intervals for continuous monitoring of data quality. This ensures that your data quality checks are performed regularly and automatically.
4. **Monitoring and Logging**: Azure monitor provides built-in monitoring and logging capabilities, allowing you to monitor job performance, resource utilization, and errors. We can set up alerts based on predefined thresholds to get notified of any issues.

### FINAL OUTPUT STEPS :-

## We have started with setting up Azure Databricks

## C:\Users\welcome_\Pictures\Screenshots\Screenshot (1092).png

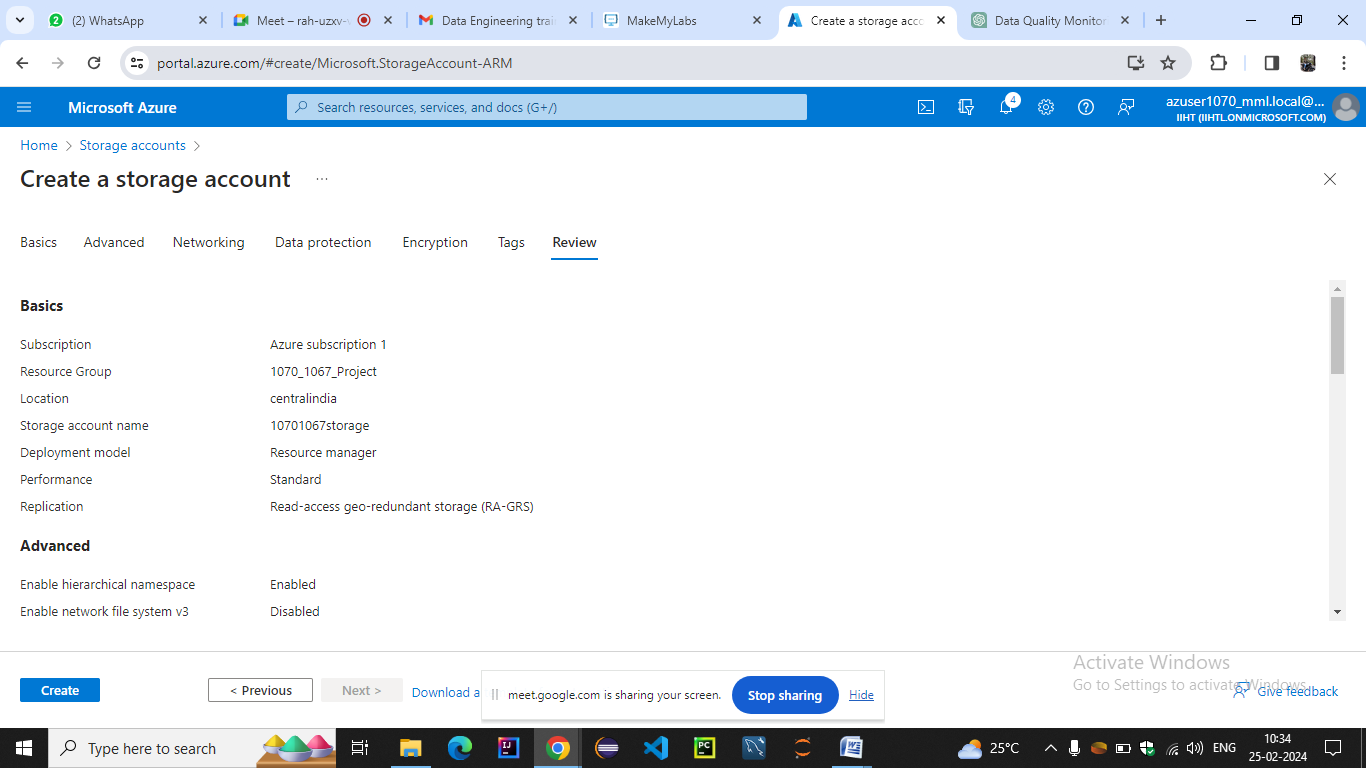
## After Launching Workspace we have created cluster

## C:\Users\welcome_\Pictures\Screenshots\Screenshot (1096).png

## We have taken our data source from kaggle and working on titanic boat dataset

## C:\Users\welcome_\Downloads\WhatsApp Image 2024-02-25 at 10.44.18 AM.jpeg

## We have Created Storage Account (ADLS GEN 2) for storing file



## We have launched account and created container and uploaded file in it .And we can also do it with azure storage explorer

## C:\Users\welcome_\Pictures\Screenshots\Screenshot (1095).png

## We have launched workspace and created new notebook

## C:\Users\welcome_\Pictures\Screenshots\Screenshot (1097).png

## We have ingested the data from adls gen2 into workspace

## C:\Users\welcome_\Pictures\Screenshots\Screenshot (1098).png

## 8.PERFORMING DATA PROFILING

## Print the schema

## C:\Users\welcome_\Pictures\Screenshots\Screenshot (1113).png

## Display the data

## C:\Users\welcome_\Pictures\Screenshots\Screenshot (1114).png

## Summary statistics (count,mean,stddev,min,max)

## C:\Users\welcome_\Pictures\Screenshots\Screenshot (1115).png

## Count null values for each column

## C:\Users\welcome_\Pictures\Screenshots\Screenshot (1116).png

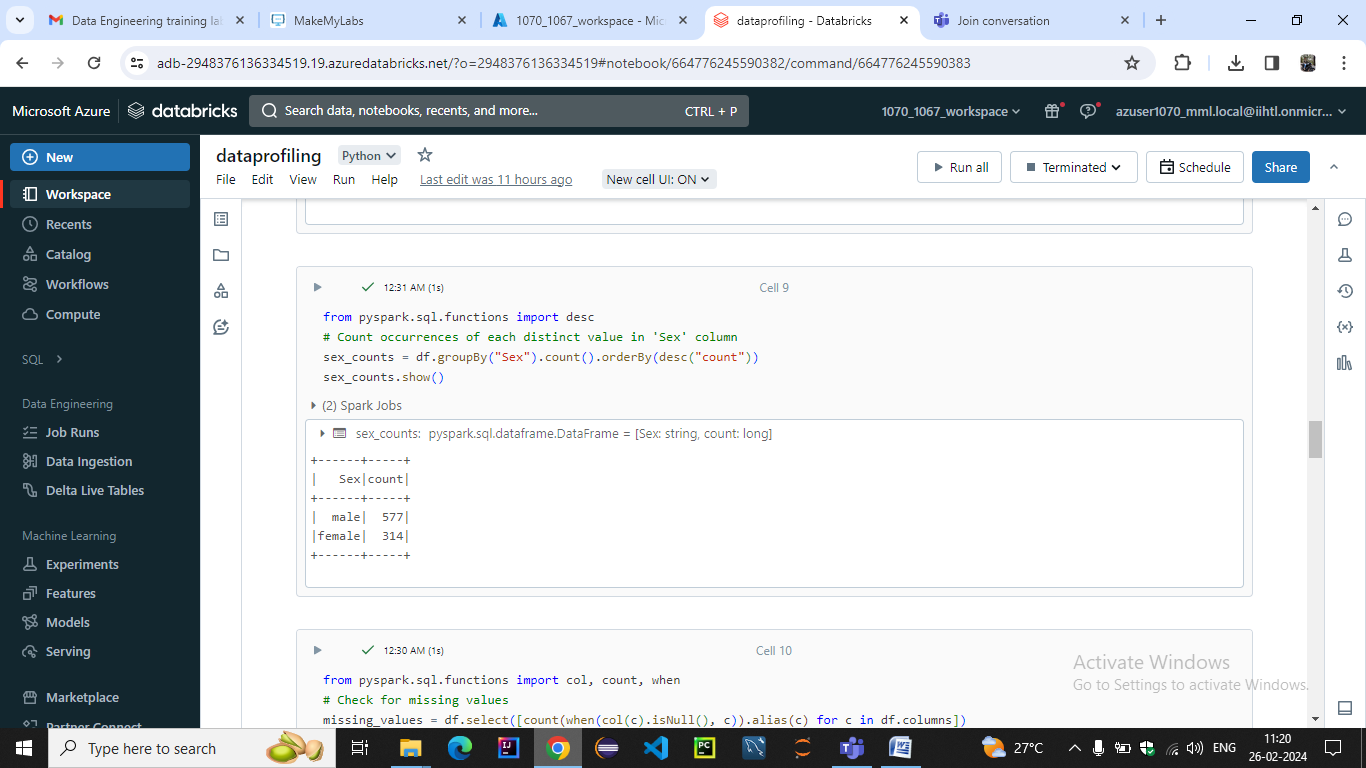
## DISTINCT VALUES

## C:\Users\welcome_\Pictures\Screenshots\Screenshot (1117).png

## COLUMN PROFILING (DISTINCTVALUES COUNT,AVG VALUE LENGTH)

## C:\Users\welcome_\Pictures\Screenshots\Screenshot (1118).png

## COUNT OCCURENCES OF EACH DISTINCT COLUMN LIKE(SEX )



## CHECK FOR MISSING VALUES

## C:\Users\welcome_\Pictures\Screenshots\Screenshot (1120).png

## COUNT OF DUPLICATE VALUES WITH CONDITION

## C:\Users\welcome_\Pictures\Screenshots\Screenshot (1121).png

## FREQUENCY COUNT

## C:\Users\welcome_\Pictures\Screenshots\Screenshot (1122).png

## OUTLIER DETECTION

## C:\Users\welcome_\Pictures\Screenshots\Screenshot (1123).png

## C:\Users\welcome_\Pictures\Screenshots\Screenshot (1124).png

## DATA HISTOGRAMS

## C:\Users\welcome_\Pictures\Screenshots\Screenshot (1125).png

## C:\Users\welcome_\Pictures\Screenshots\Screenshot (1126).png

## QUALITY

## COMPLETENESS SCORE

## C:\Users\welcome_\Pictures\Screenshots\Screenshot (1127).png

## COMPLETENESS

## C:\Users\welcome_\Pictures\Screenshots\Screenshot (1128).png

## ACCURACY

## C:\Users\welcome_\Pictures\Screenshots\Screenshot (1129).png

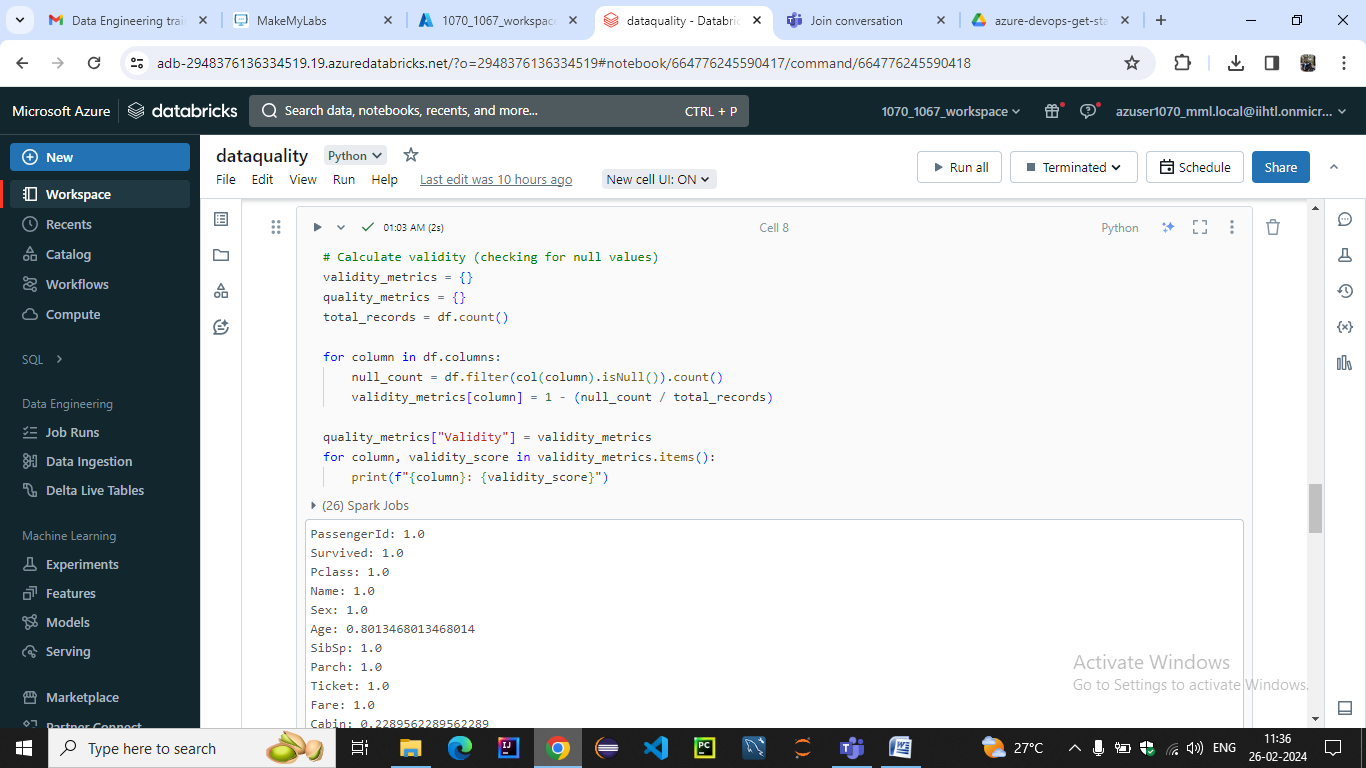
## CONSISTENCY

## C:\Users\welcome_\Pictures\Screenshots\Screenshot (1130).png

## UNIQUENESS

## C:\Users\welcome_\Pictures\Screenshots\Screenshot (1131).png

## VALIDITY



## RELIABILITY

## C:\Users\welcome_\Pictures\Screenshots\Screenshot (1133).png

## INTEGRITY

## C:\Users\welcome_\Pictures\Screenshots\Screenshot (1134).png

## PRECISION

## C:\Users\welcome_\Pictures\Screenshots\Screenshot (1135).png

## RELEVANCE

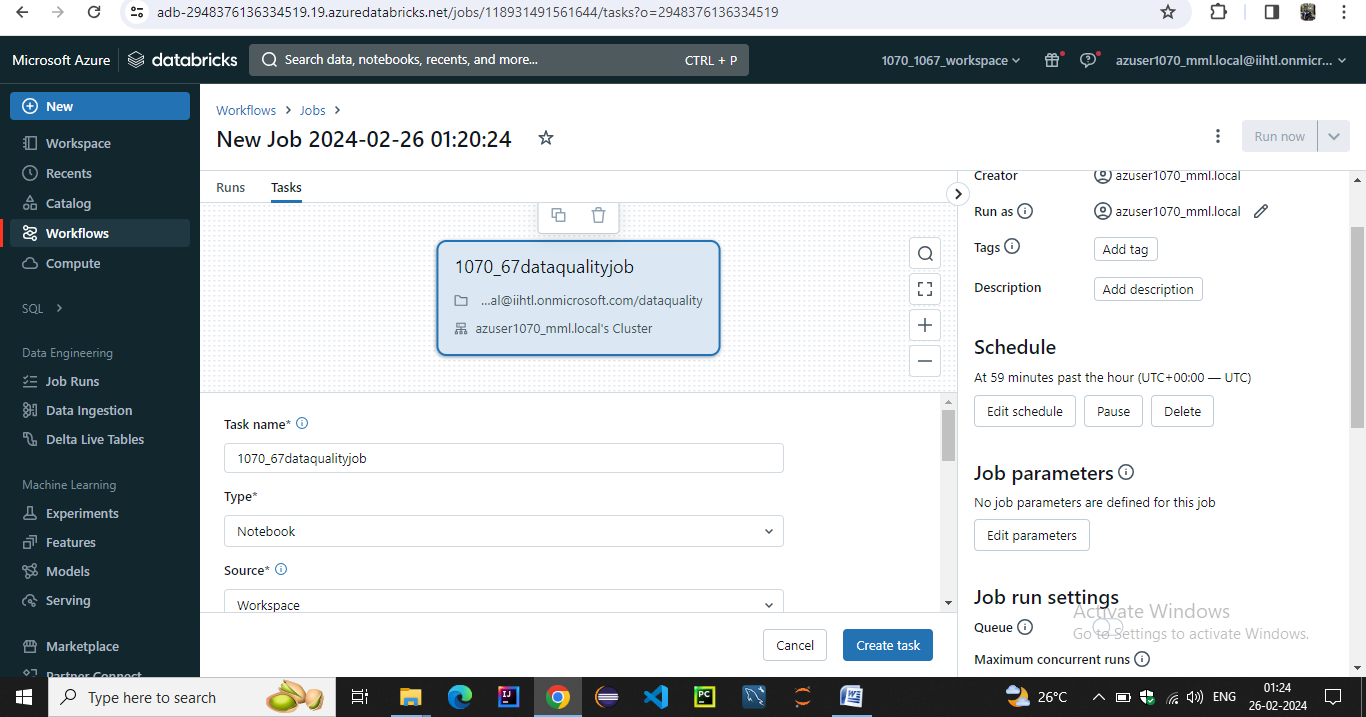
## C:\Users\welcome_\Pictures\Screenshots\Screenshot (1136).png

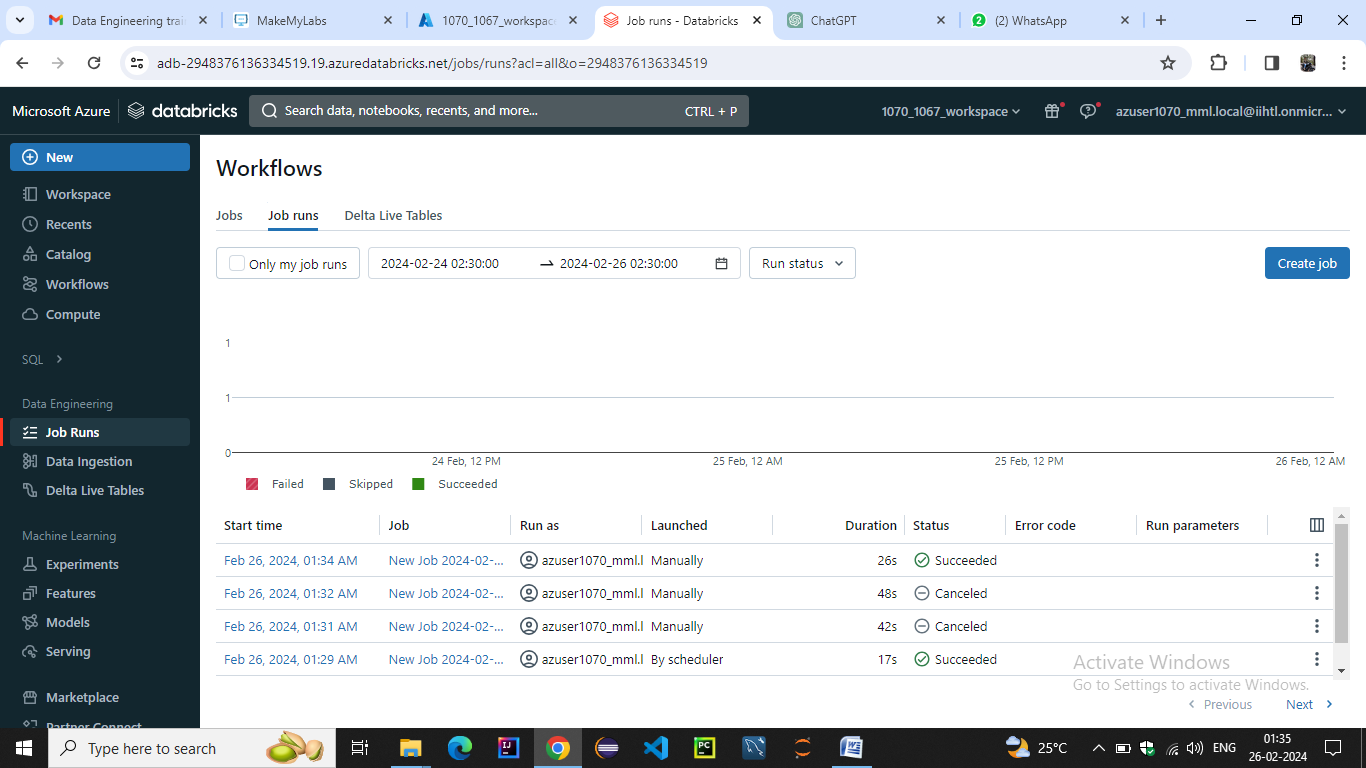
## CONFORMITY

## C:\Users\welcome_\Pictures\Screenshots\Screenshot (1137).png

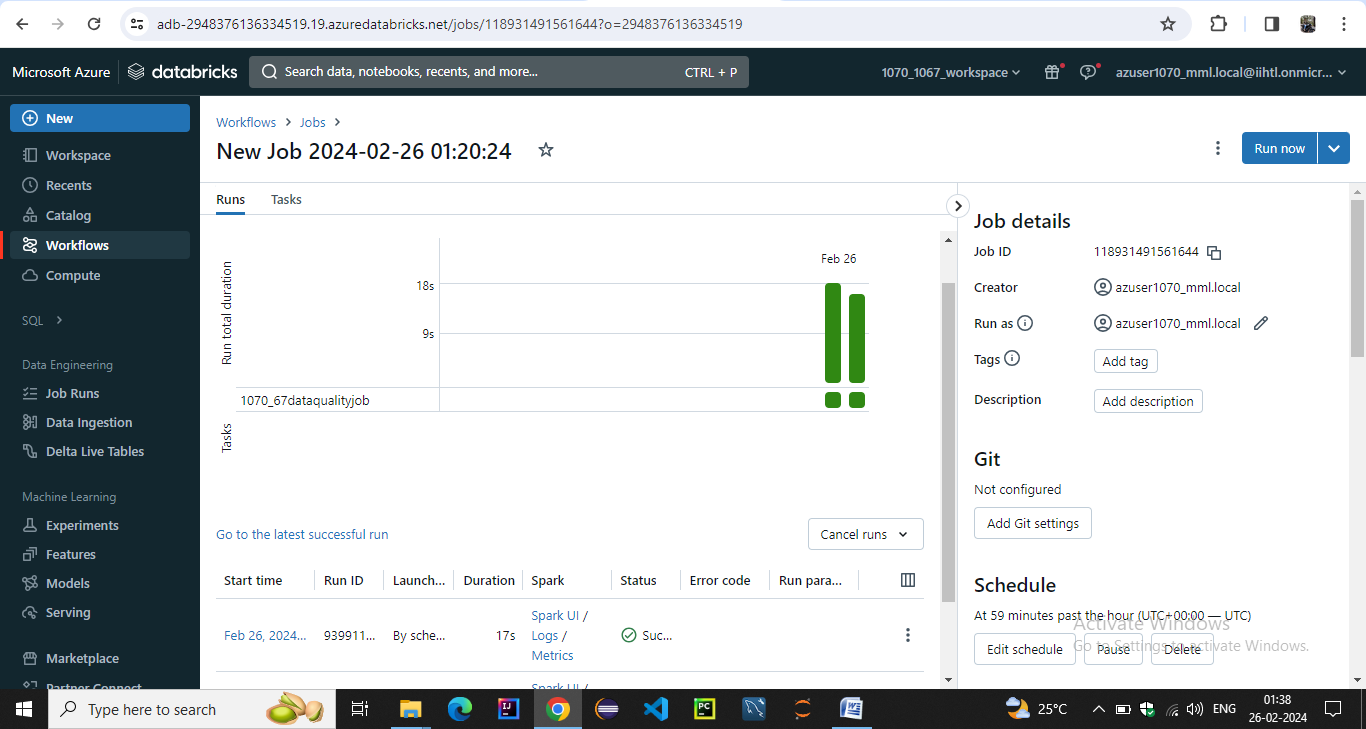
**9.PERFORMING CONTINUOUS MONITORING**

**We have scheduled job for continuous monitoring of data quality and data profile notebooks**

****

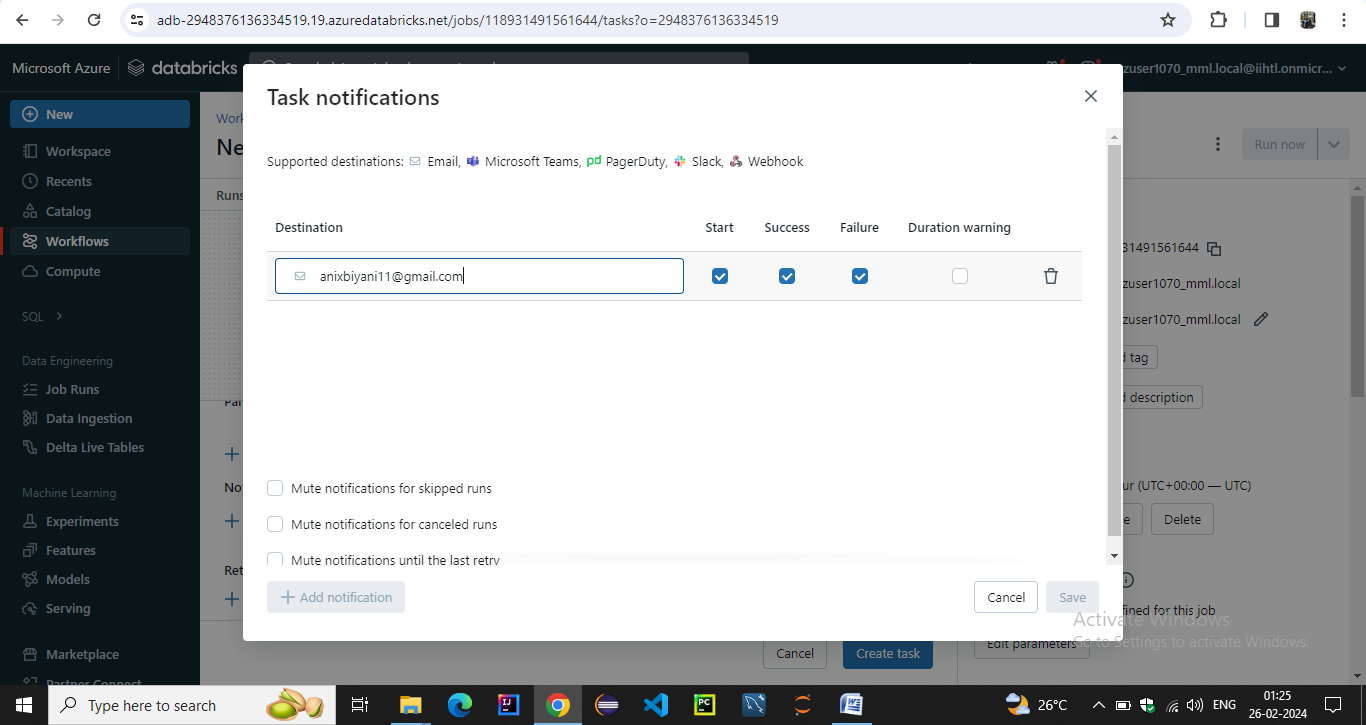
**Workflow of the jobs after scheduled interval **

**Job details**

****

**We have setup job task notifications on email for start success failure**

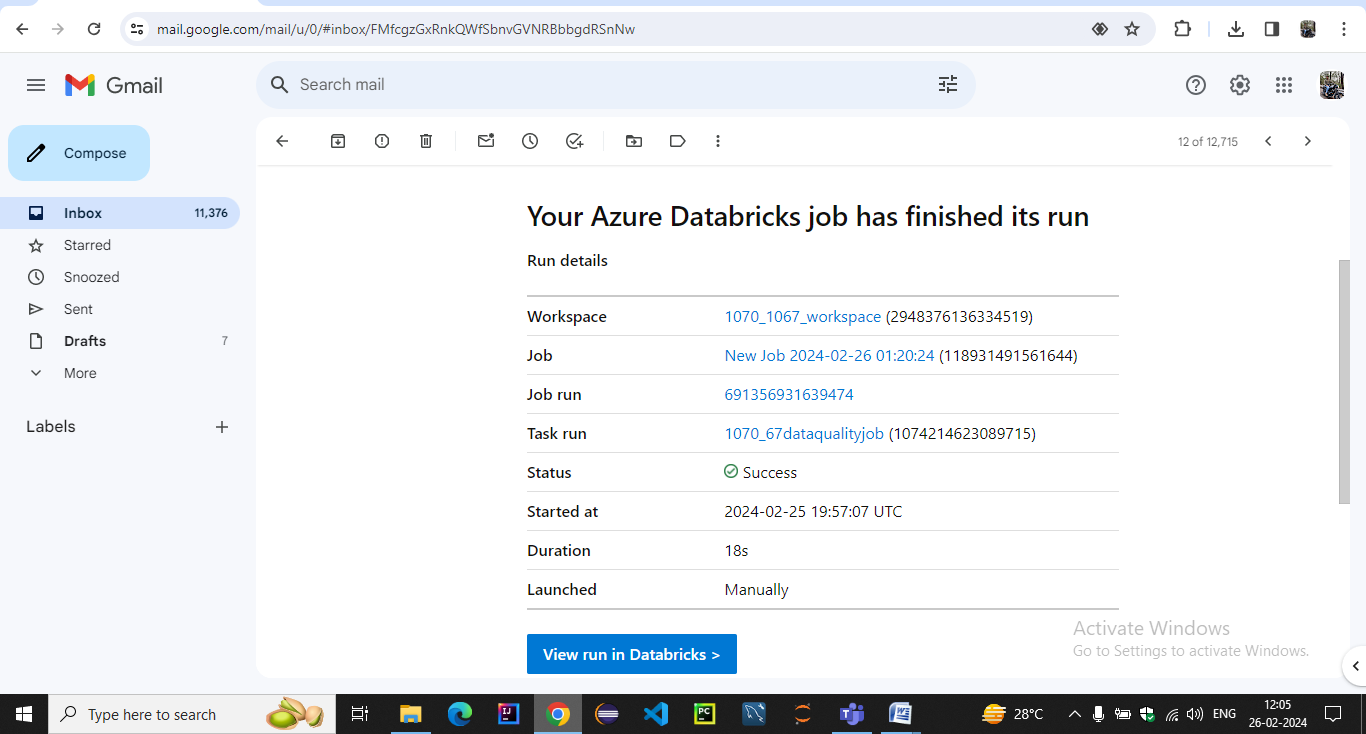
**After each interval**

****

**We have alerted tasks notifications for success start and failure notifications on the email which we provided**

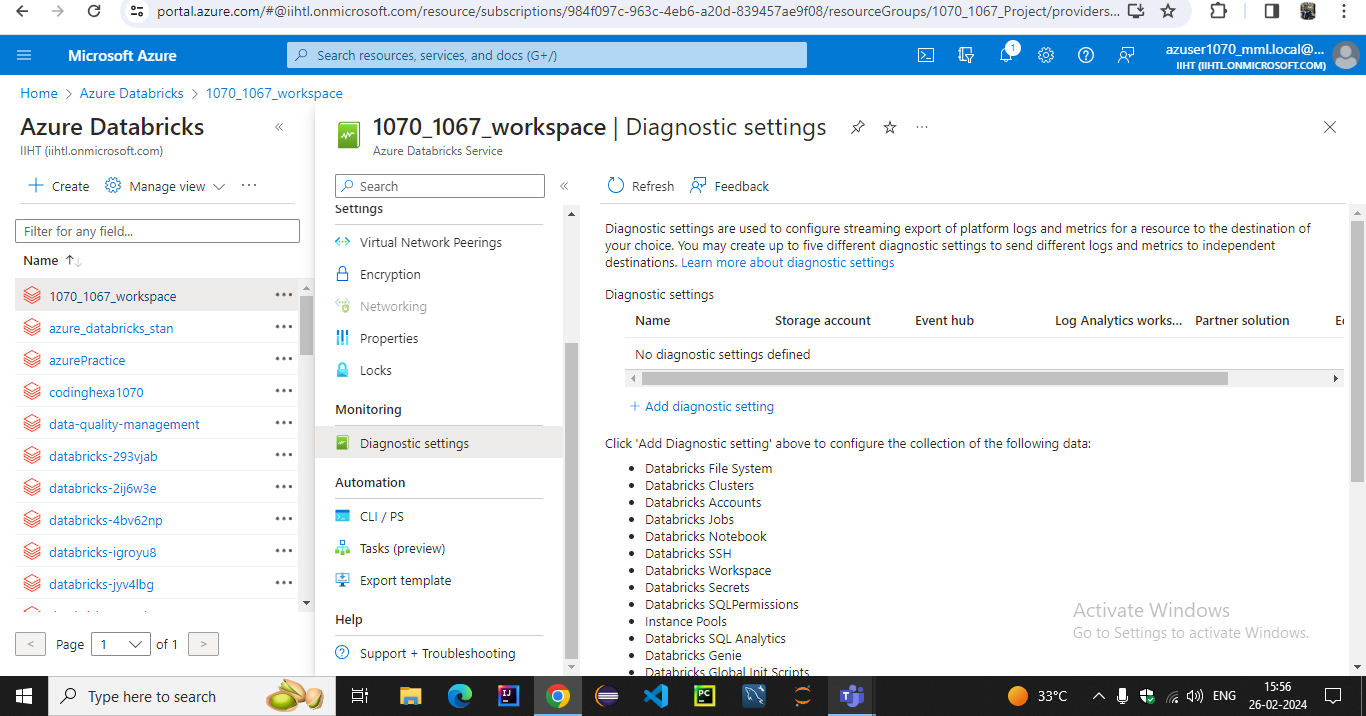
****

**After running at scheduled interval it also reverted success back**

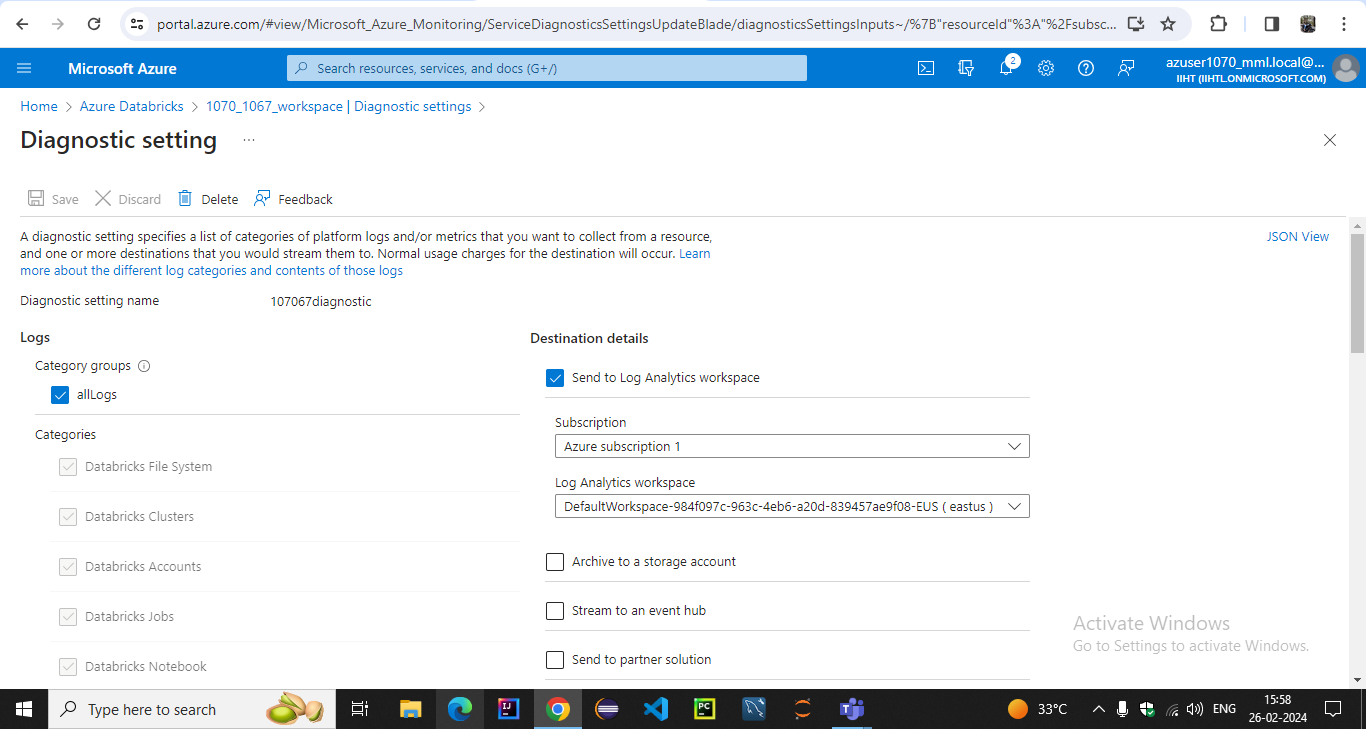
****

**10.Generating Alerts**

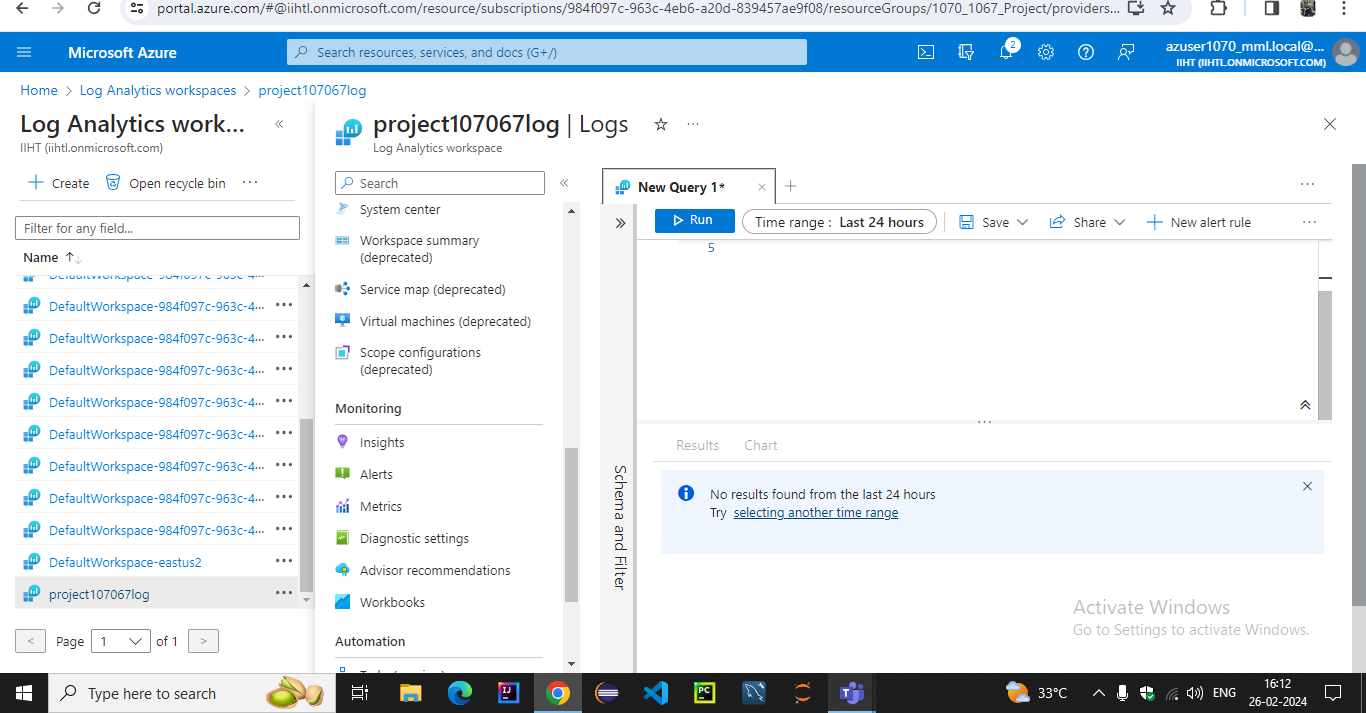
**For creating alerts we have to integrate azure databricks workspace to log analytics workspace**

****

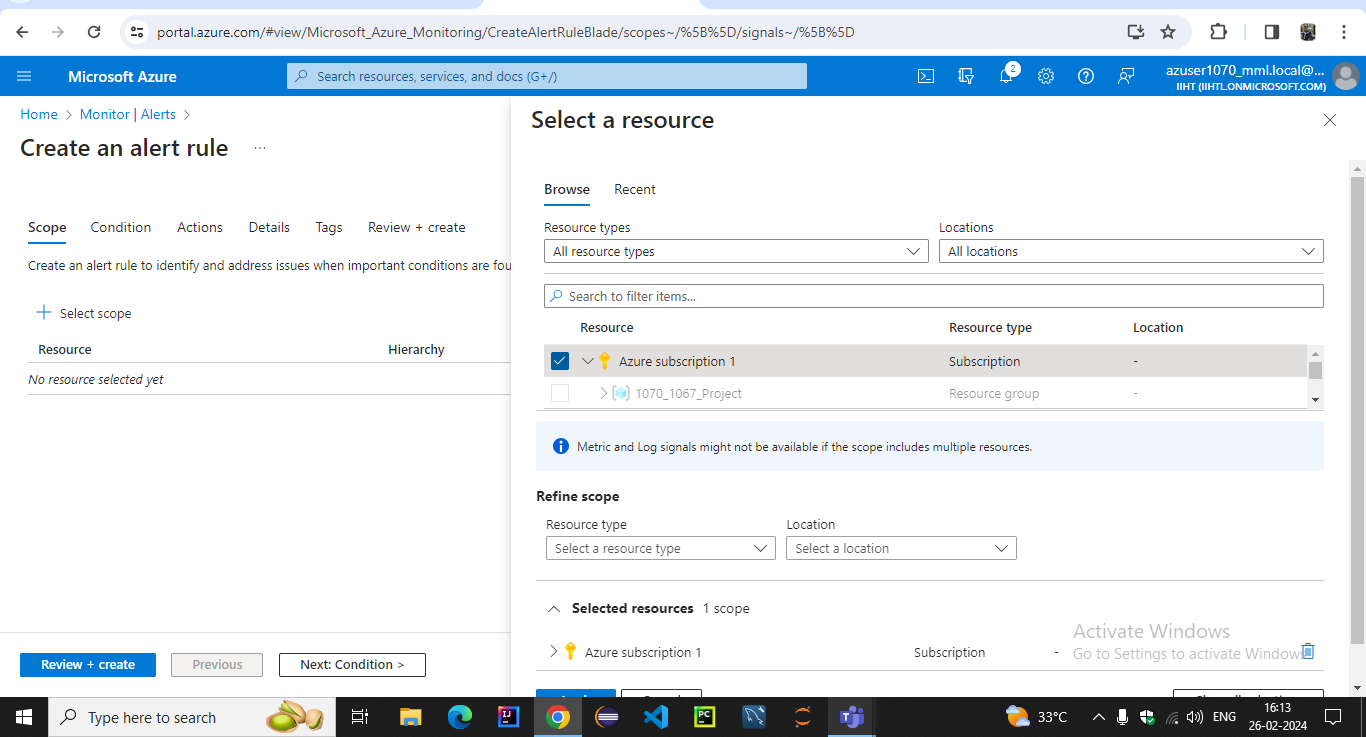
**we have diagnosed workspace to long analytics workspace**

****

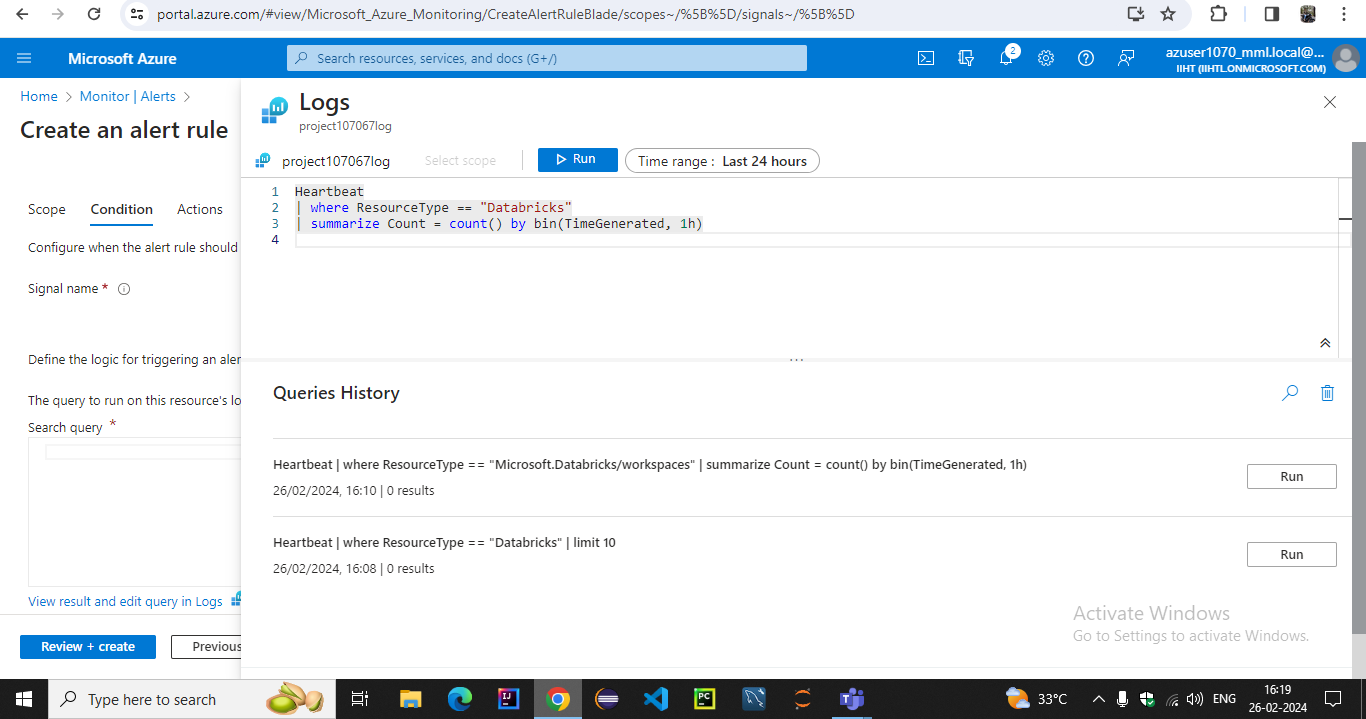
**In the logs workspace we checked our query and it runned successfully**

****

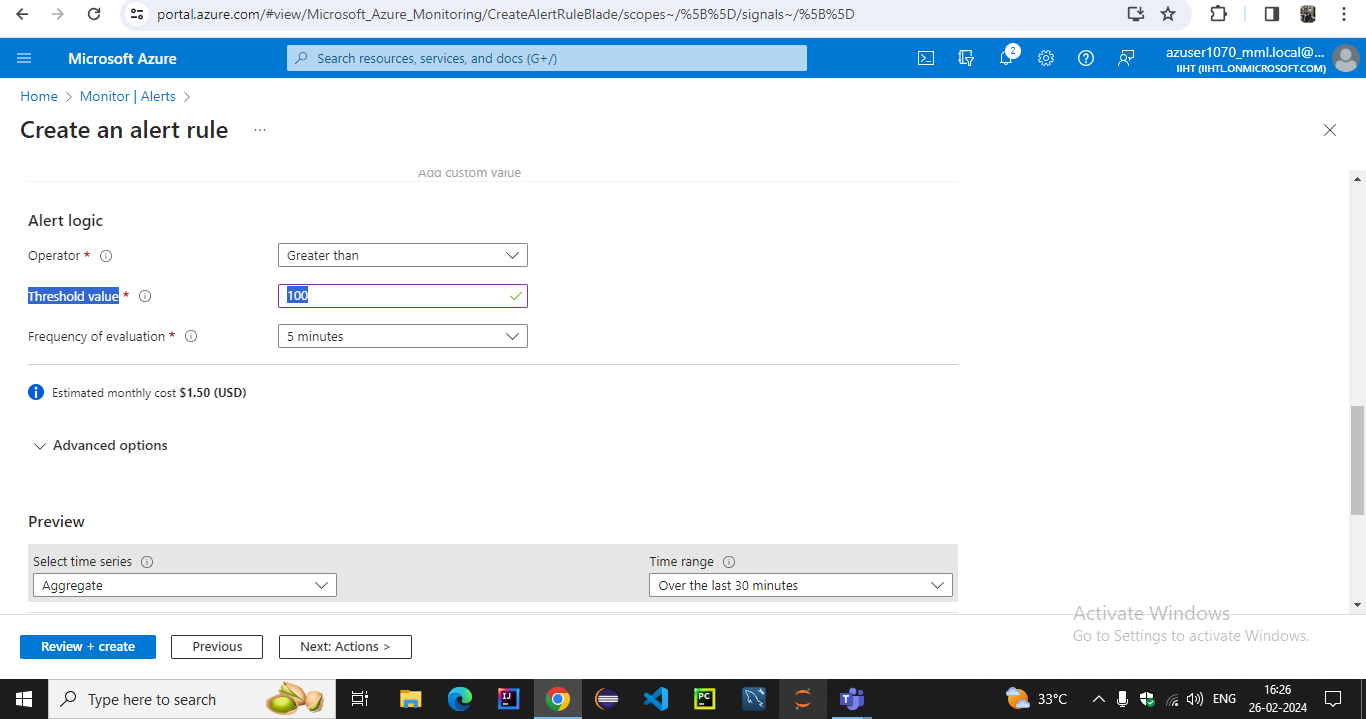
**Now going to azure monitor we have created alert with log metrics and configured scope with log analytics workspace**

****

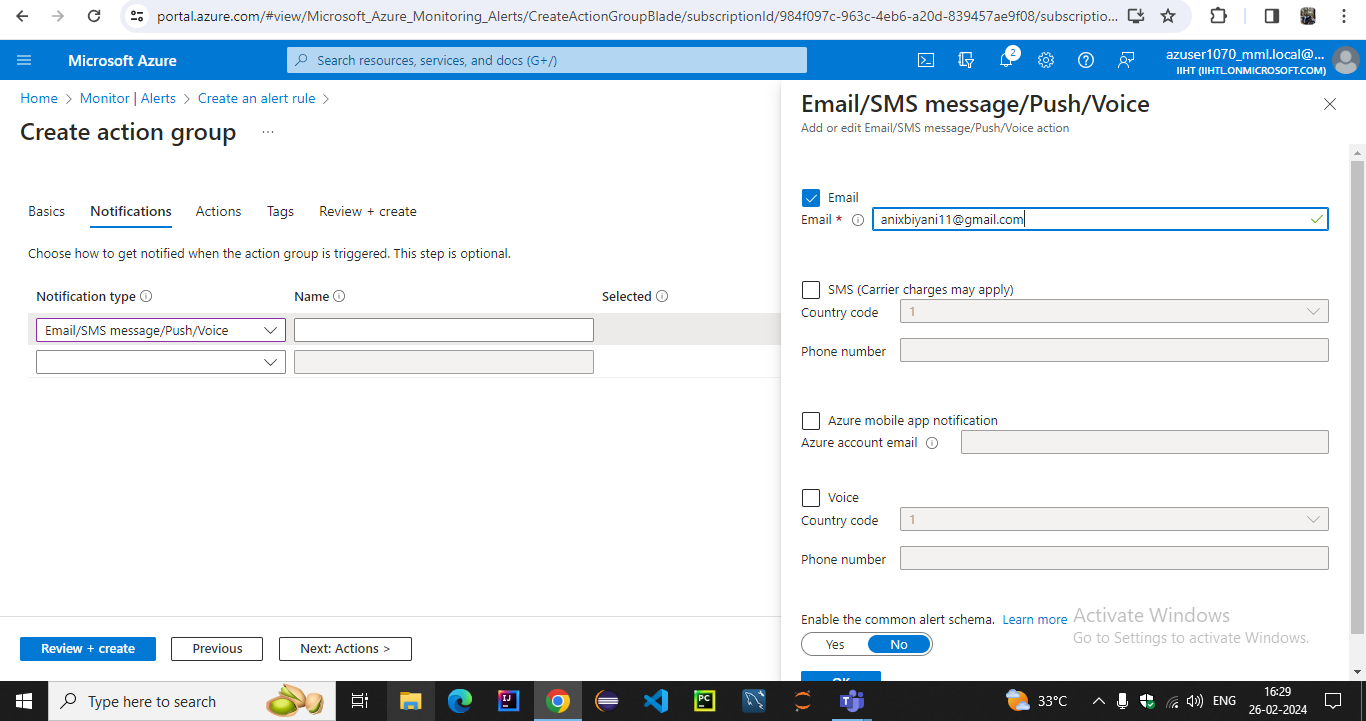
**We have specified the condition in the log query space with signal name as log analytics**

****

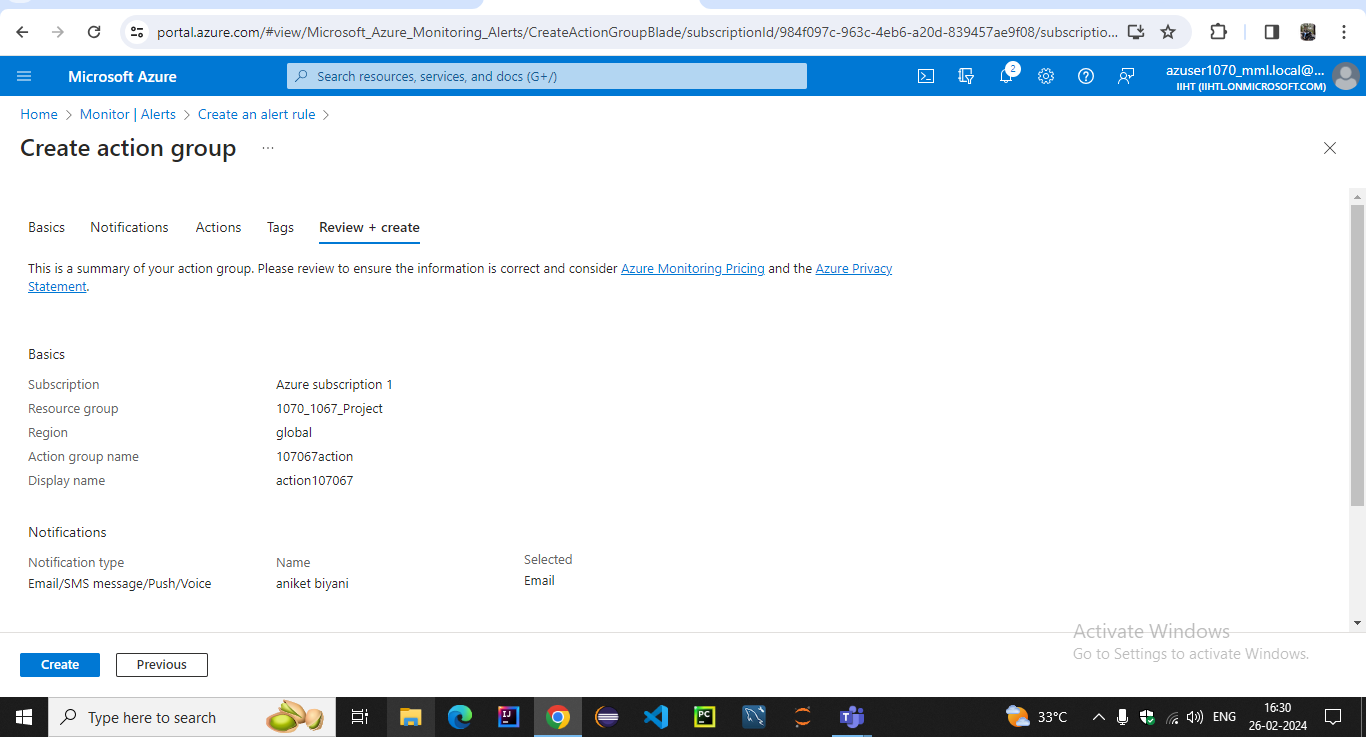
**In the alert logic we specified the threshold value operator and frequency of evaluation**

****

**We have set alert notifications email in the action group**



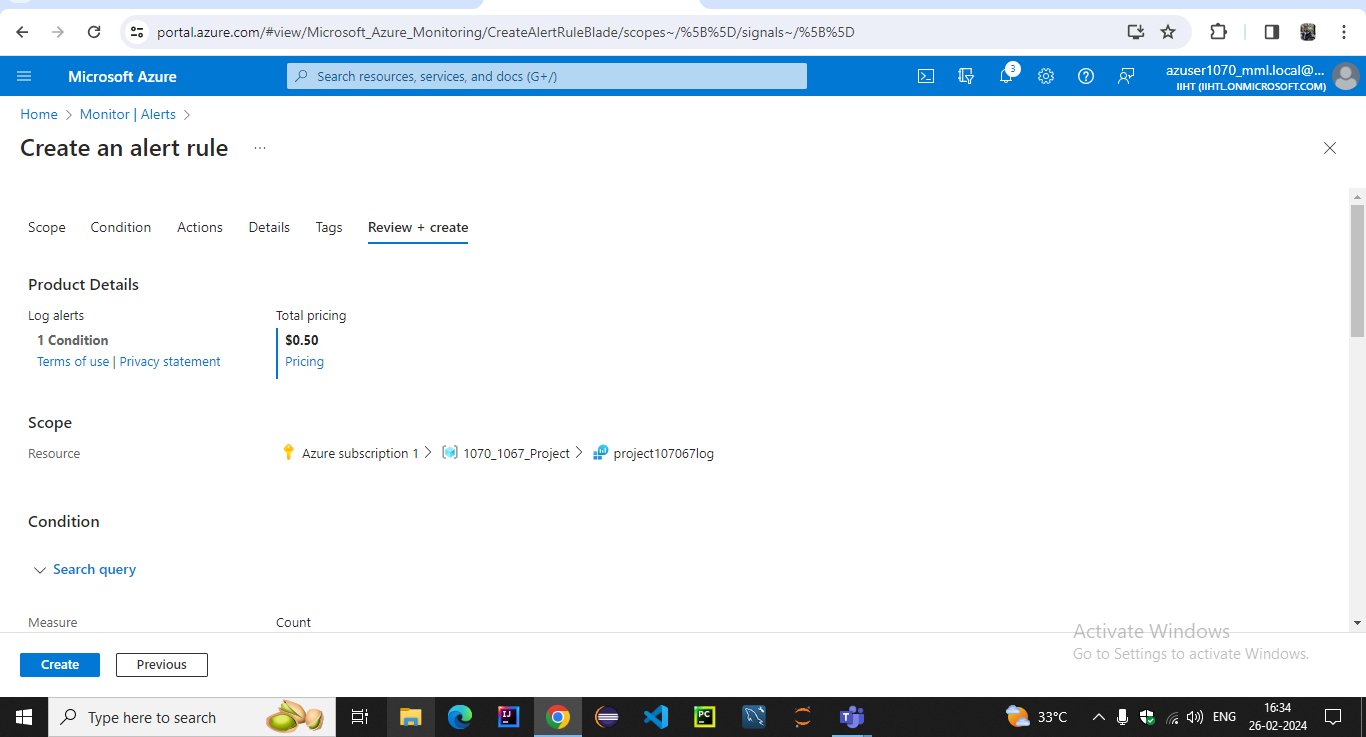
**We have created action group in the alerts**



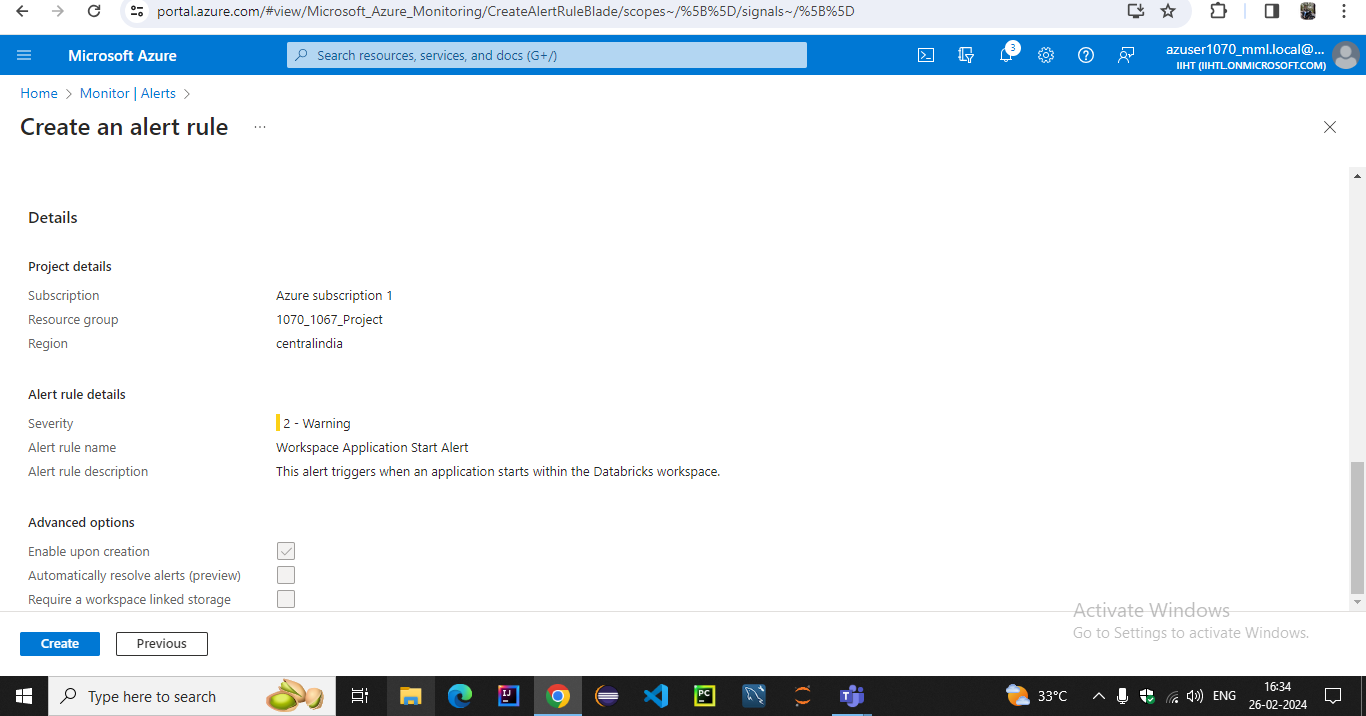
**We have created alert rule by providing severity name description**



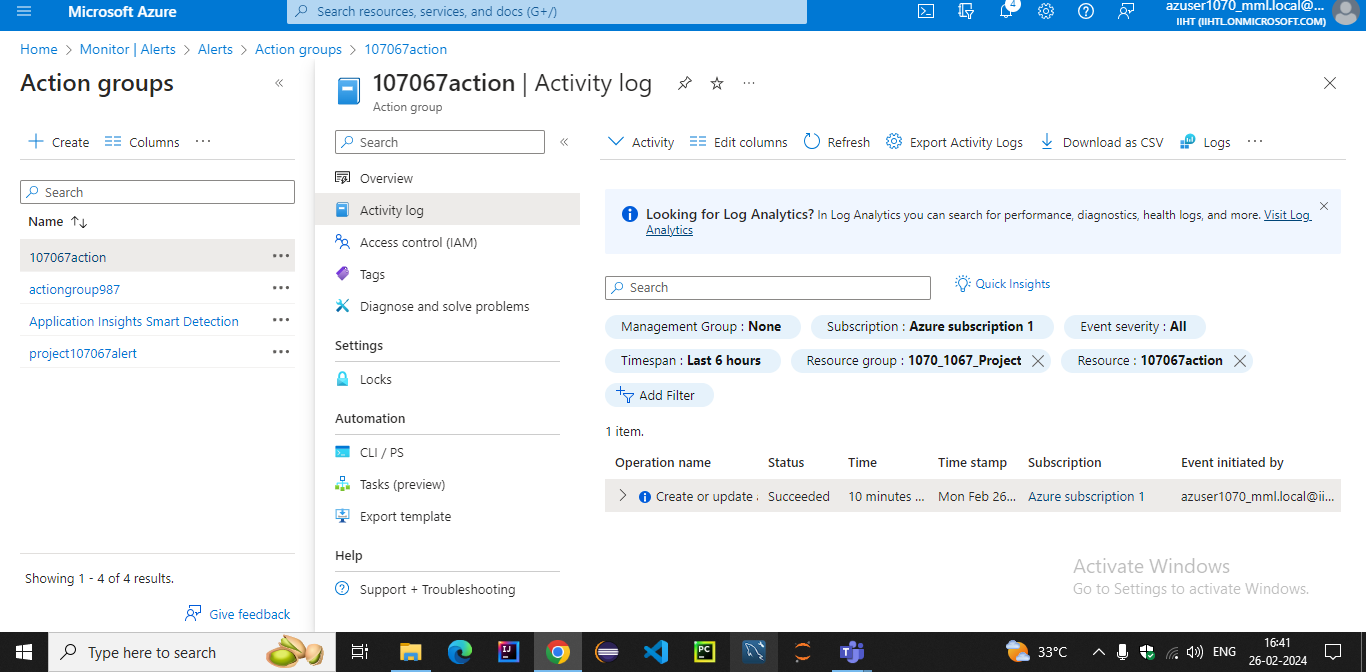
**Now reviving and creating with pricing of 0.5 dollar with scope and condition**

****

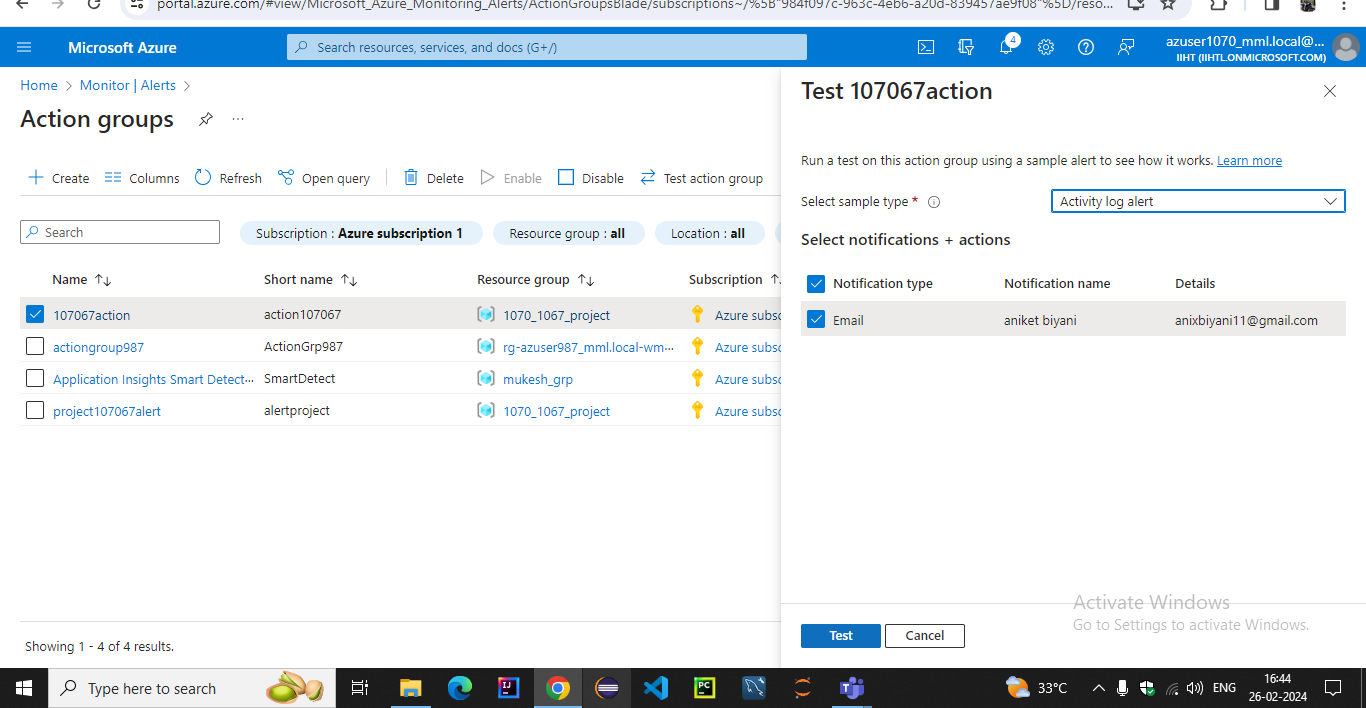
**Finally we created alert rule successfully**

****

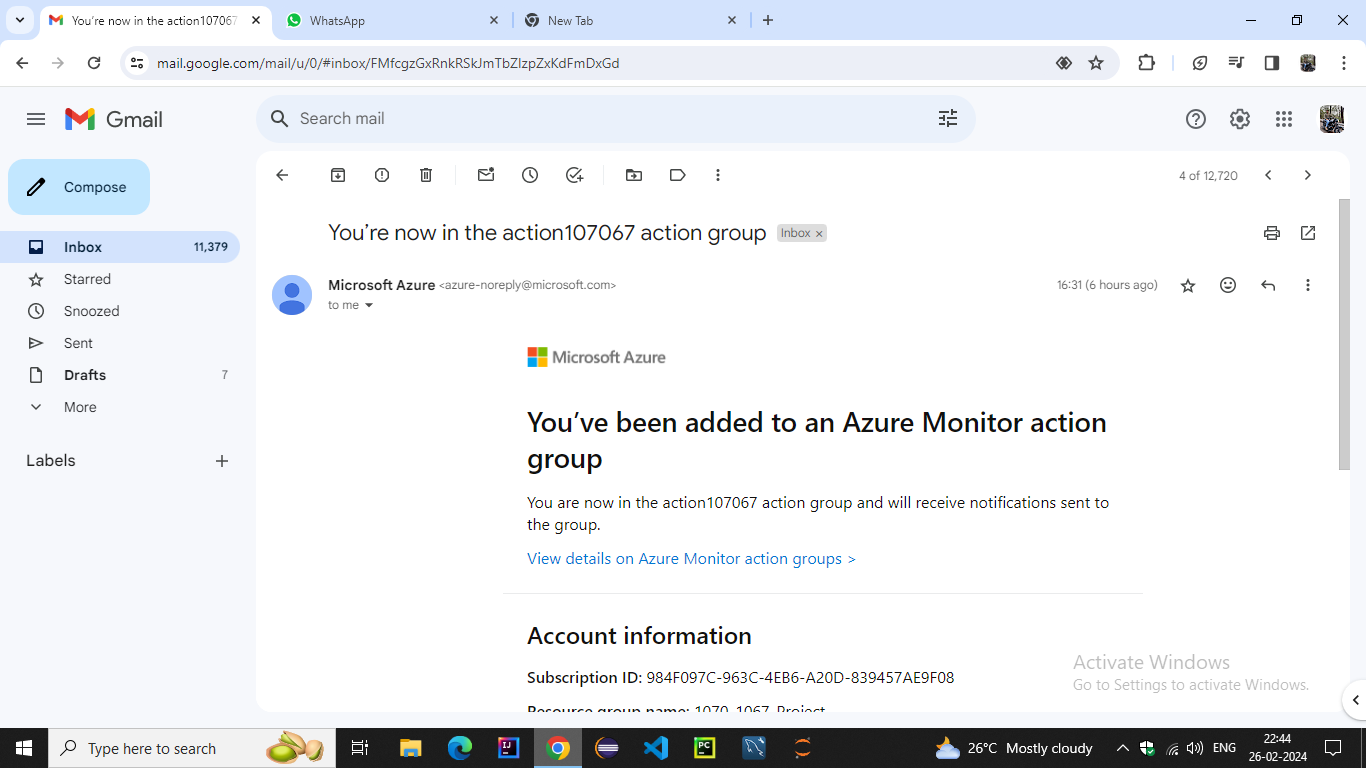
**We can see our actions in activity log**

****

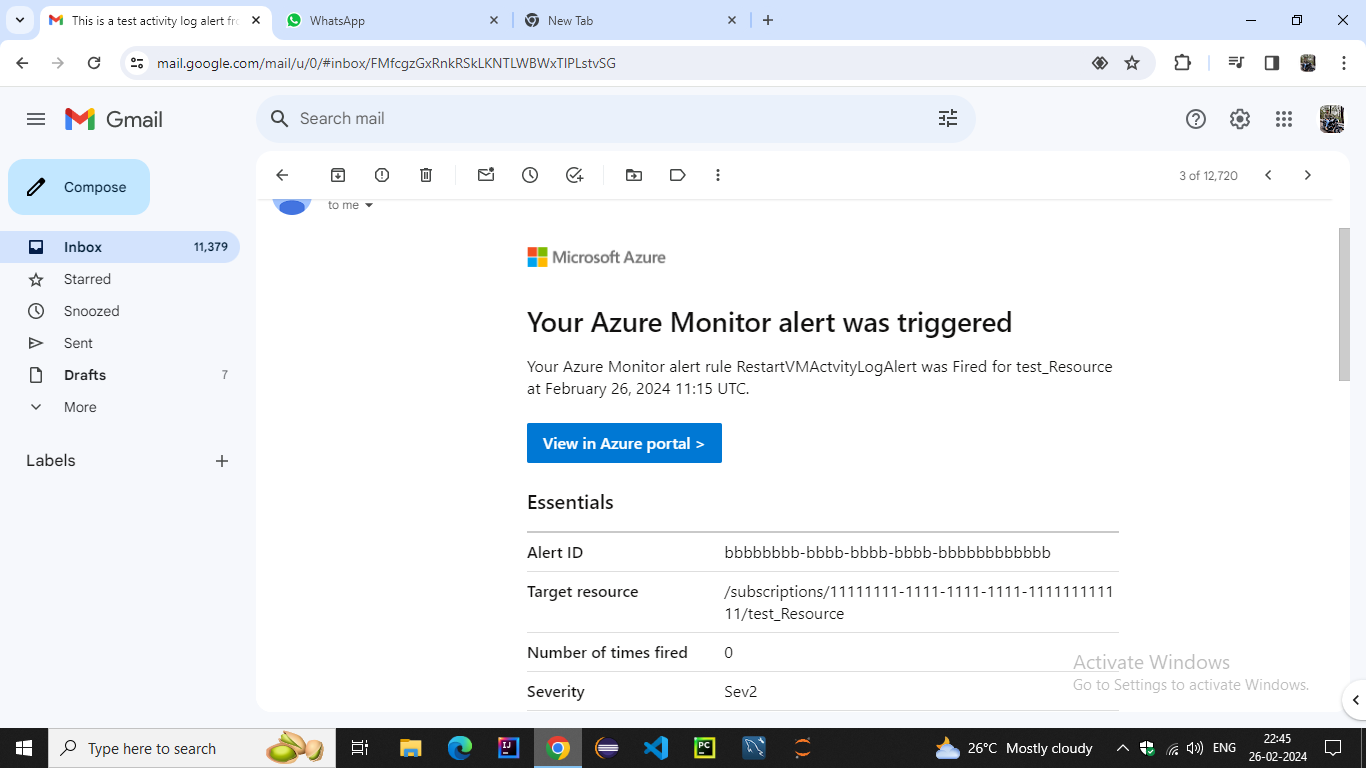
**Next step is to test our action we created for alert**

****

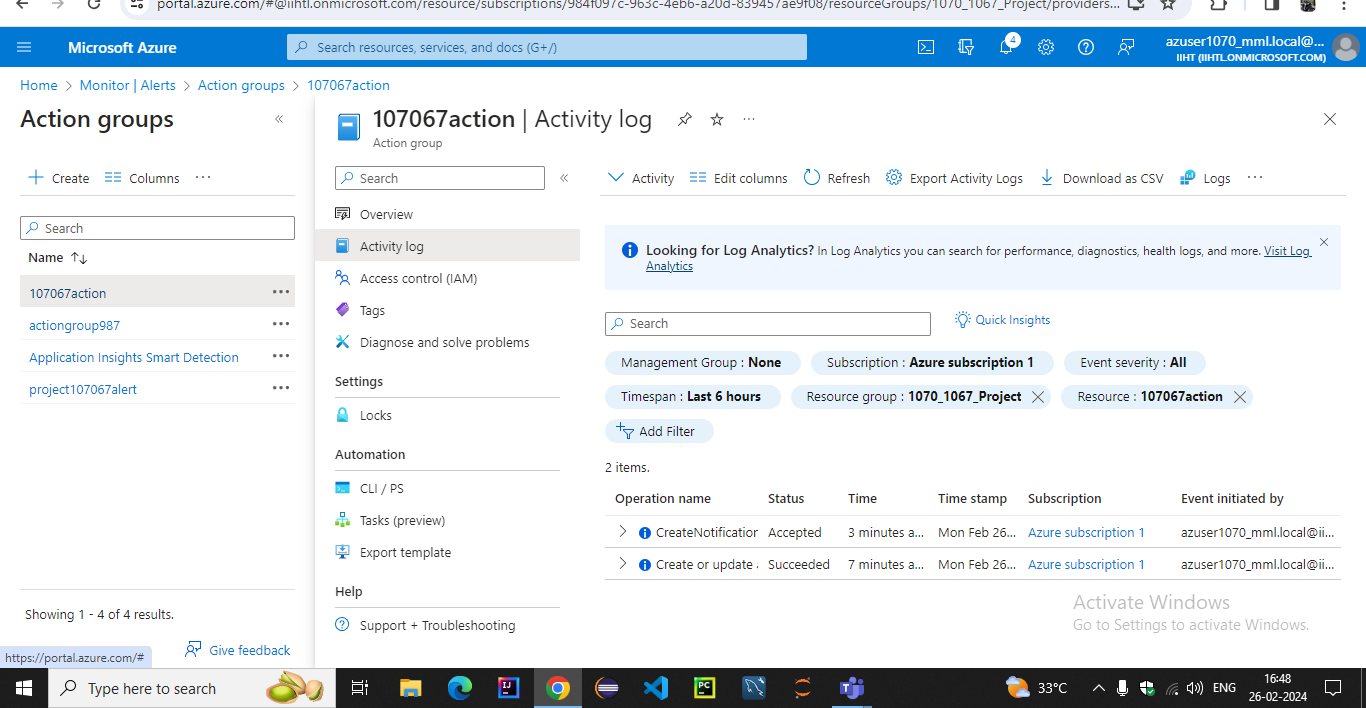
**After testing our alert has been added into azure monitor action group**

****

**Our azure monitor alert was triggered successfully**

****

**Now we can see our action triggered in the log with log status whenever alert get triggered**

****

### CONCLUSION:-

In conclusion, we have developed a robust data quality monitoring system leveraging PySparkSQL for data profiling and quality assessments, and Azure Databricks for continuous monitoring.

By defining quality metrics, implementing quality checks, and automating the process with Azure Databricks, we ensure that data quality issues are promptly detected and addressed.

The integration of alerting mechanisms allows stakeholders to be notified of anomalies in real-time, enabling timely corrective actions.

Overall, this system enhances data reliability, integrity, and trustworthiness, thereby improving decision-making and business outcomes