**Project Overview:** Data Transformation Pipeline with Azure Data Factory and Azure Databricks

**Introduction:**

In today's data-driven world, organizations face the challenge of efficiently transforming and processing vast amounts of data stored in various formats. This project aims to address this challenge by leveraging the capabilities of Azure Data Factory (ADF) and Azure Databricks to create a robust data transformation pipeline.

**Objective:**

The primary objective of this project is to develop a data pipeline that extracts data from a CSV file stored in Azure Storage, transforms it into the Parquet file format, and stores the transformed data in another location. Additionally, Azure Databricks will be used to explore and optimize the conversion process to ensure efficiency and scalability.

**Key Components:**

**1. Azure Data Factory (ADF):**

* ADF serves as the orchestration hub, managing workflow execution and data movement tasks.
* It facilitates the definition and execution of pipelines, coordinating activities involved in data extraction, transformation, and loading (ETL).

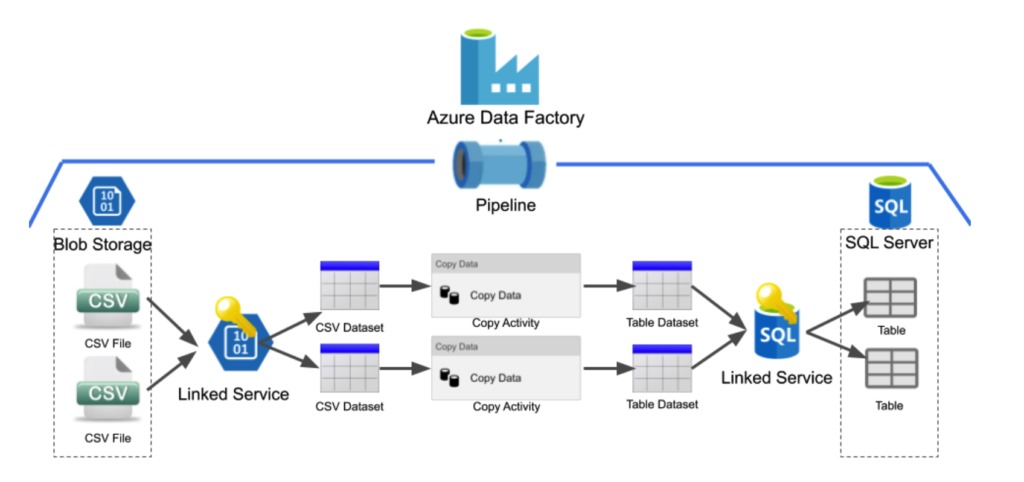
**2. Azure Databricks:**

* Azure Databricks acts as the analytical engine, enabling data exploration, transformation, and optimization.
* Through interactive notebooks, it provides a collaborative environment for data scientists and engineers to analyze, preprocess, and enhance data quality.

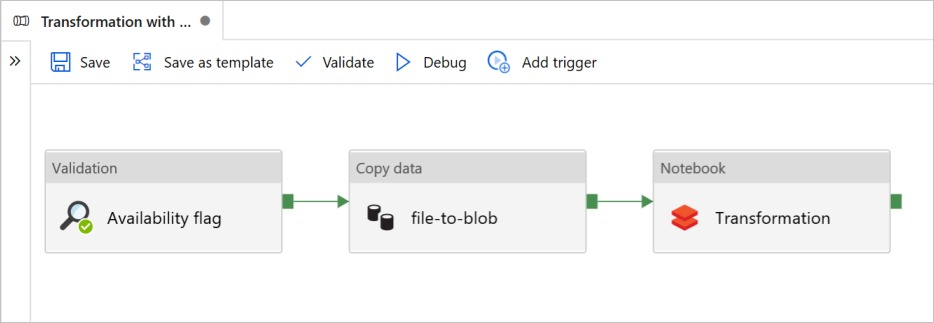
**3. Azure Storage:**

* Azure Storage serves as the repository for both source and transformed data.
* It offers scalable and secure storage solutions, accommodating large datasets while ensuring data integrity and accessibility.

**Architecture diagram:**



*Fig 1 shows the architectural digram*



*Fig 2 shows the copying data and transformation mechanism*

**Workflow:**

**1. Data Extraction:**

* ADF triggers the extraction process, retrieving CSV files from Azure Storage.
* It ensures seamless connectivity and data retrieval, handling authentication and authorization securely.

**2. Data Transformation:**

* Azure Databricks notebooks are utilized to delve into the data structure and content, identifying patterns and anomalies.
* Advanced data processing techniques are applied to cleanse, transform, and optimize the data for subsequent analysis and storage.

**3. Data Loading:**

* ADF orchestrates the loading of transformed data into Azure Storage, converting it into Parquet format.
* It manages data transfer efficiently, ensuring data consistency and integrity throughout the process.

**4.Monitoring and Optimization:**

* Continuous monitoring of pipeline performance is conducted through ADF and Databricks monitoring tools.
* Insights gathered from monitoring data are used to optimize pipeline efficiency, enhancing processing speed and resource utilization.

**Azure Resources Used for this Project:**

**1. Azure Data Factory (ADF):**

* Azure Data Factory serves as the core orchestration service for the data transformation pipeline.
* ADF pipelines are defined to automate the workflow for extracting data from CSV files, transforming it, and loading it into Parquet files.
* Integration Runtimes within ADF are configured to establish connections to Azure Storage and Databricks clusters, facilitating seamless data movement and transformation**.**

**2. Azure Databricks:**

* Azure Databricks is utilized for advanced data exploration, transformation, and optimization tasks.
* Databricks Notebooks provide an interactive environment for data scientists and engineers to analyze the structure and content of the data, apply data cleansing and transformation techniques, and optimize processing performance.
* Databricks clusters are provisioned to execute Spark-based data processing tasks efficiently, leveraging scalable compute resources available in the Azure cloud environment.

**3. Azure Storage:**

* Azure Blob Storage is used to store both the source CSV files and the destination Parquet files.
* Blob storage containers are created to organize and manage the data files securely, providing scalable storage solutions for large datasets.
* Azure Storage features such as access control, encryption, and redundancy options are leveraged to ensure data integrity, availability, and compliance with security requirements.

**4. Azure Key Vault:**

* Azure Key Vault is employed to securely store and manage sensitive information such as connection strings, credentials, and encryption keys used by ADF and Databricks.
* Key Vault provides centralized management of secrets, enabling secure access to critical resources without exposing sensitive information in plain text.

**5. Azure Monitor:**

* Azure Monitor is utilized to monitor the performance and health of the data transformation pipeline.
* Metrics and logs generated by ADF pipelines and Databricks clusters are collected and analyzed using Azure Monitor, providing insights into pipeline execution, resource utilization, and potential bottlenecks.
* Monitoring alerts and notifications are configured to proactively identify and address issues, ensuring the reliability and efficiency of the data processing workflow.

**Project Requirements:**

1. Extract data from CSV files in Azure Storage.

2. Transform data into Parquet format for optimized storage.

3. Ensure scalability and performance for handling large datasets.

4. Integrate with Azure Data Factory for orchestration.

5. Utilize Azure Databricks for advanced processing and optimization.

6. Implement security measures and ensure compliance.

7. Provide monitoring and alerting functionalities.

8. Offer comprehensive documentation and training.

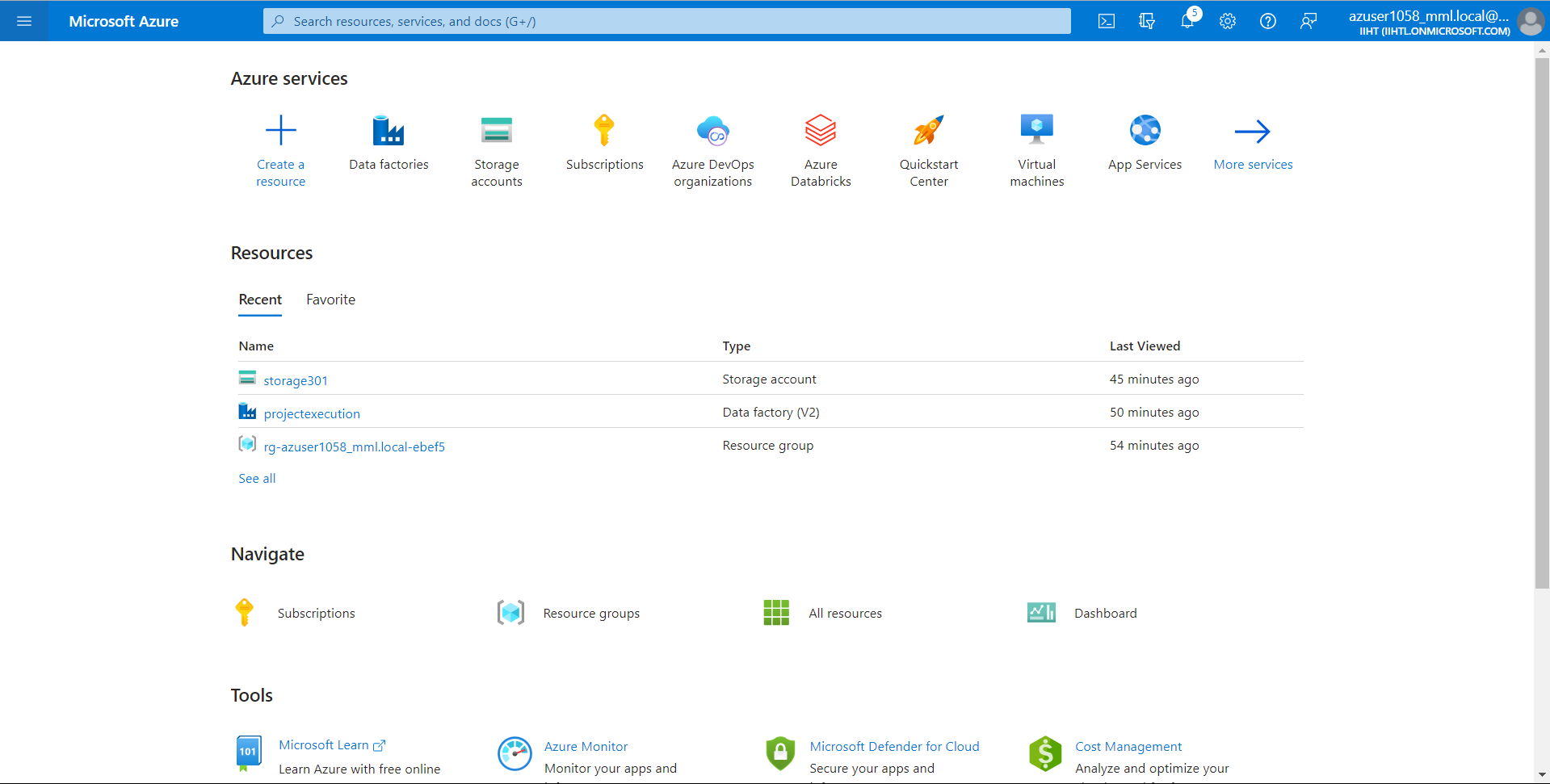
9. Optimize costs while utilizing Azure resources.

10. Ensure high availability and disaster recovery.

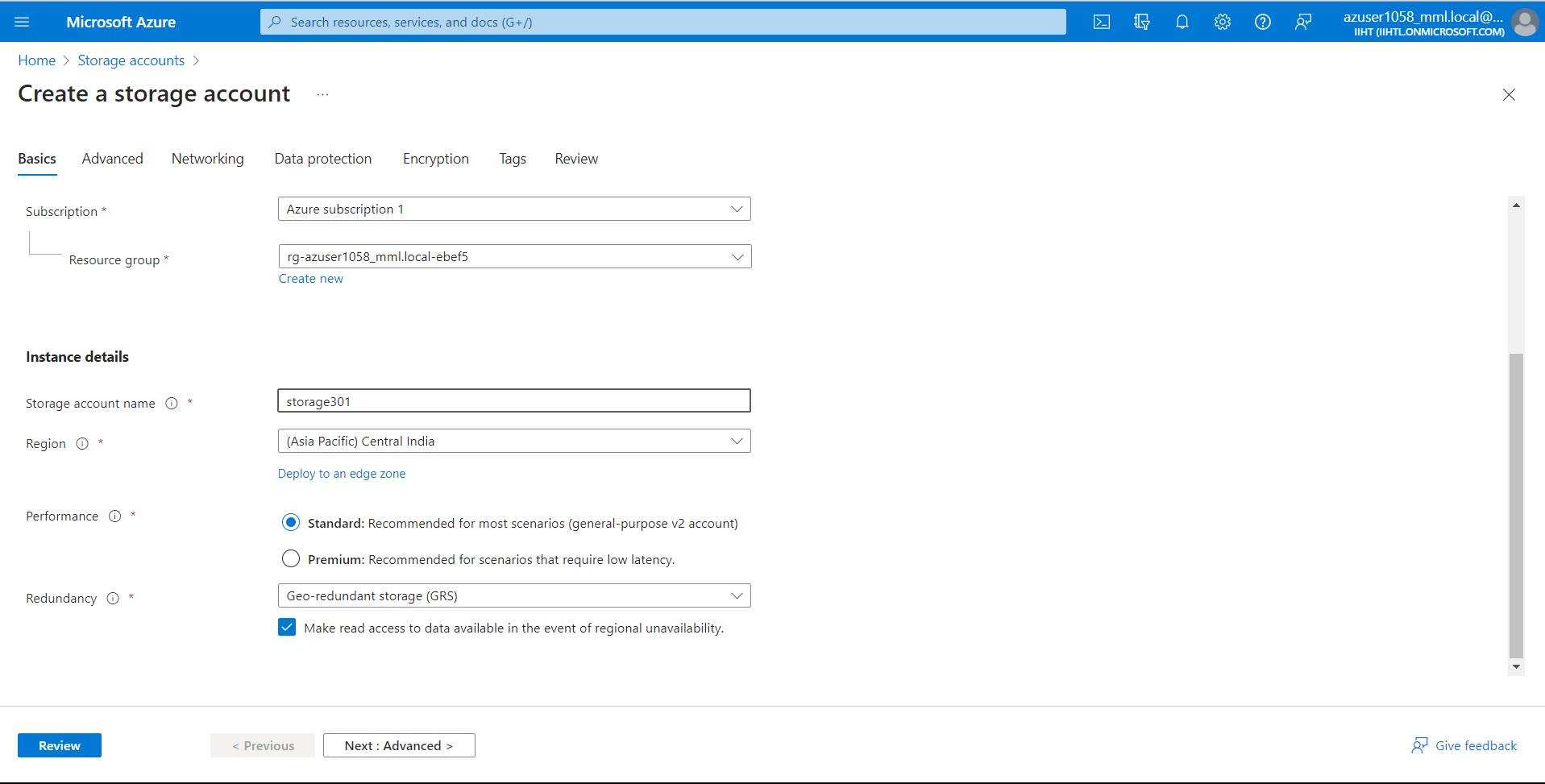
11. Conduct thorough testing for functionality and reliability.

**Results:**

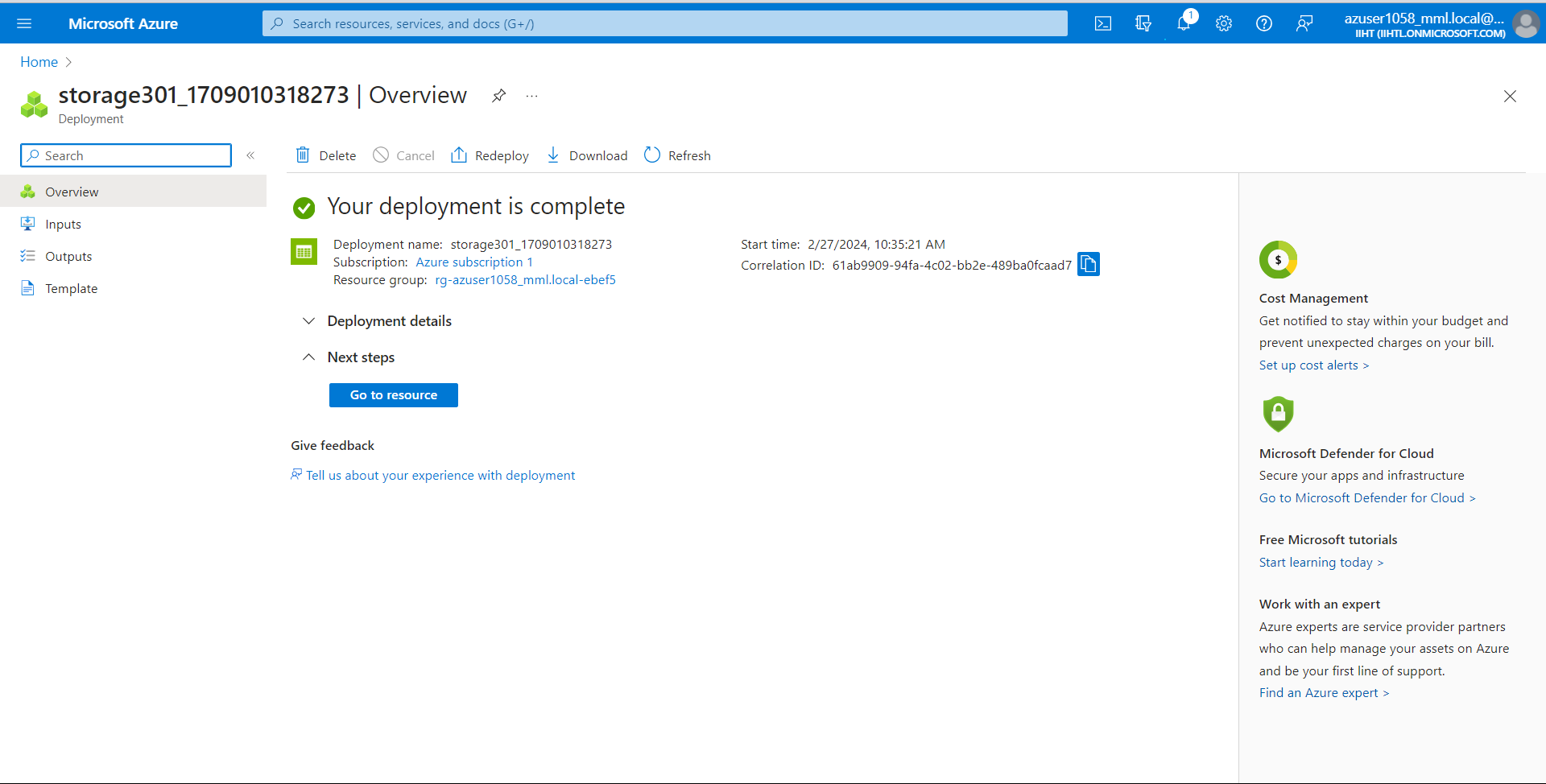
**Firstly, Go to Microsoft Azure Portal.**

****

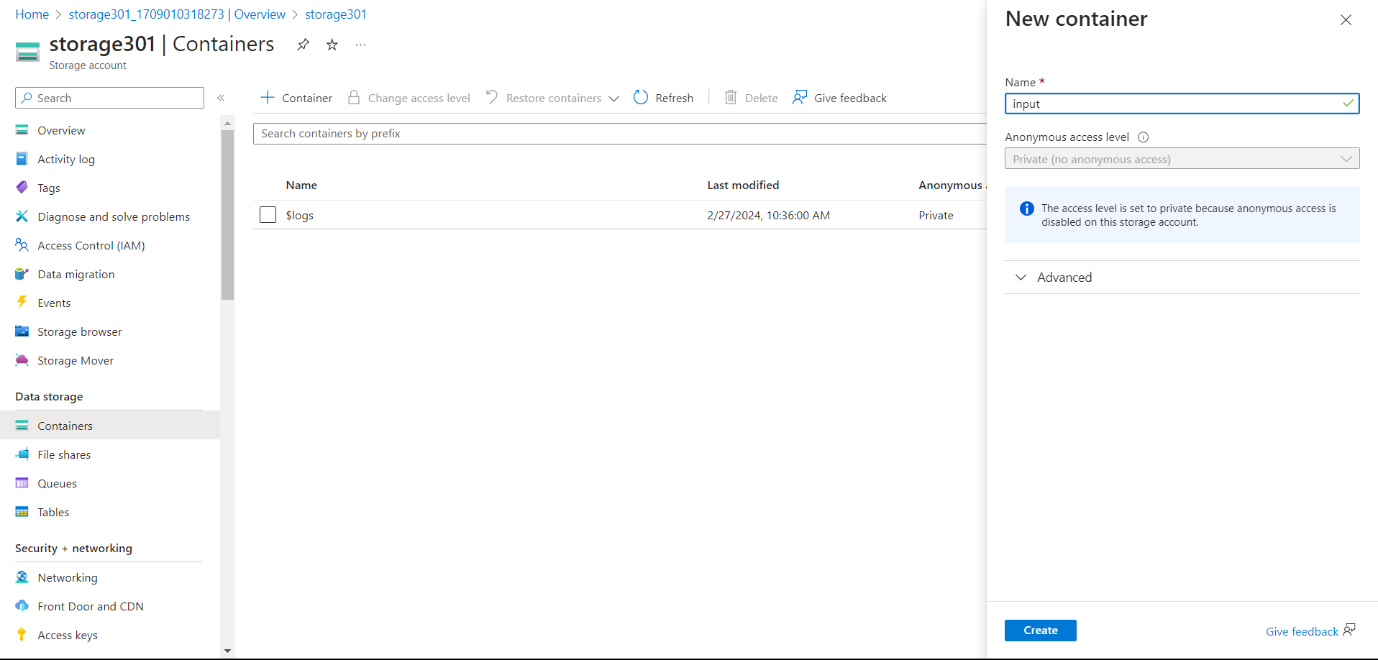
**Now lets create a storage account from portal, enter your resource group name and select any region, and click on Review + Create**.

****

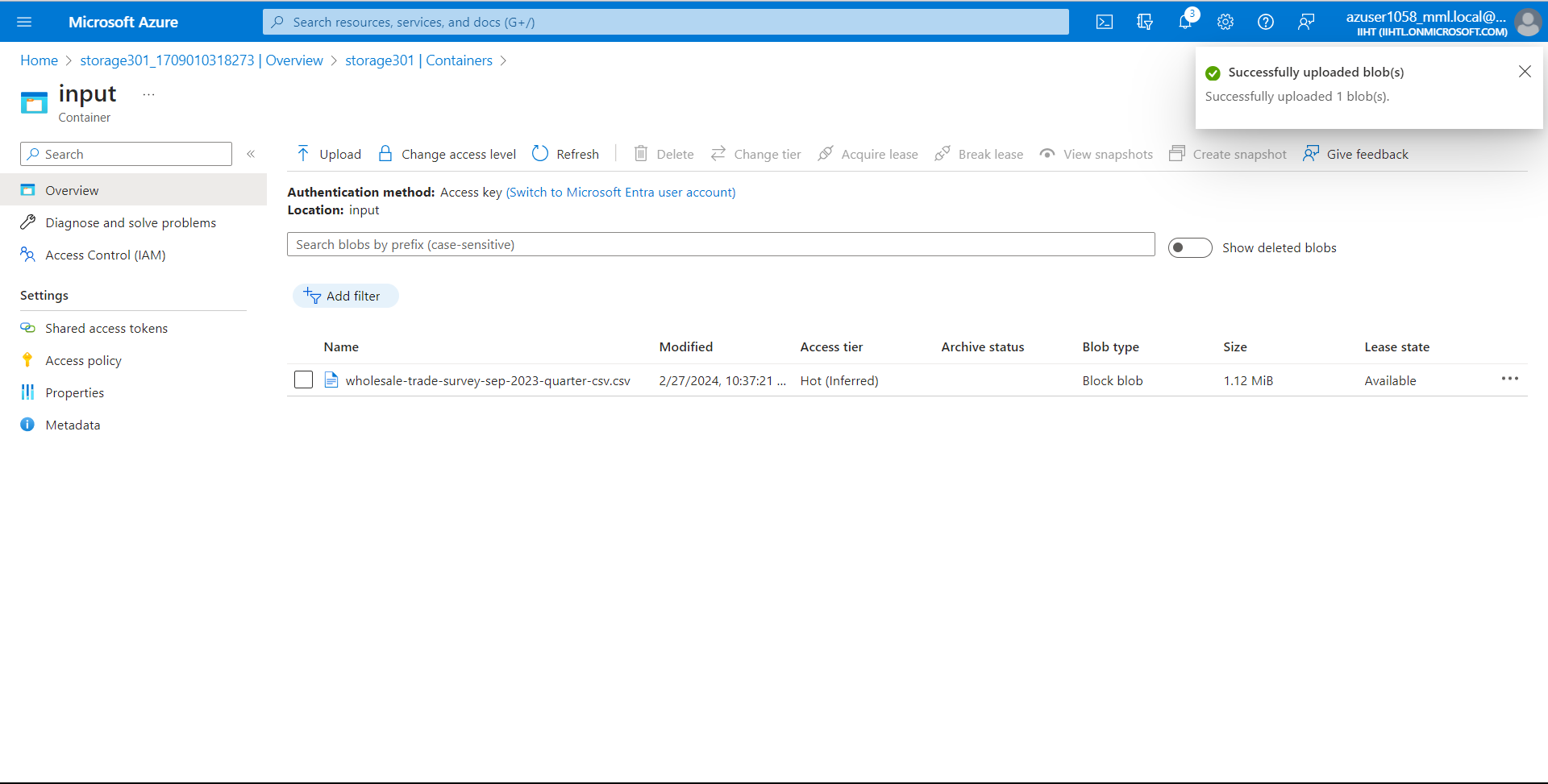
**Now we have successfully created storage account named “storage301” and click on “Go to resource”.**

****

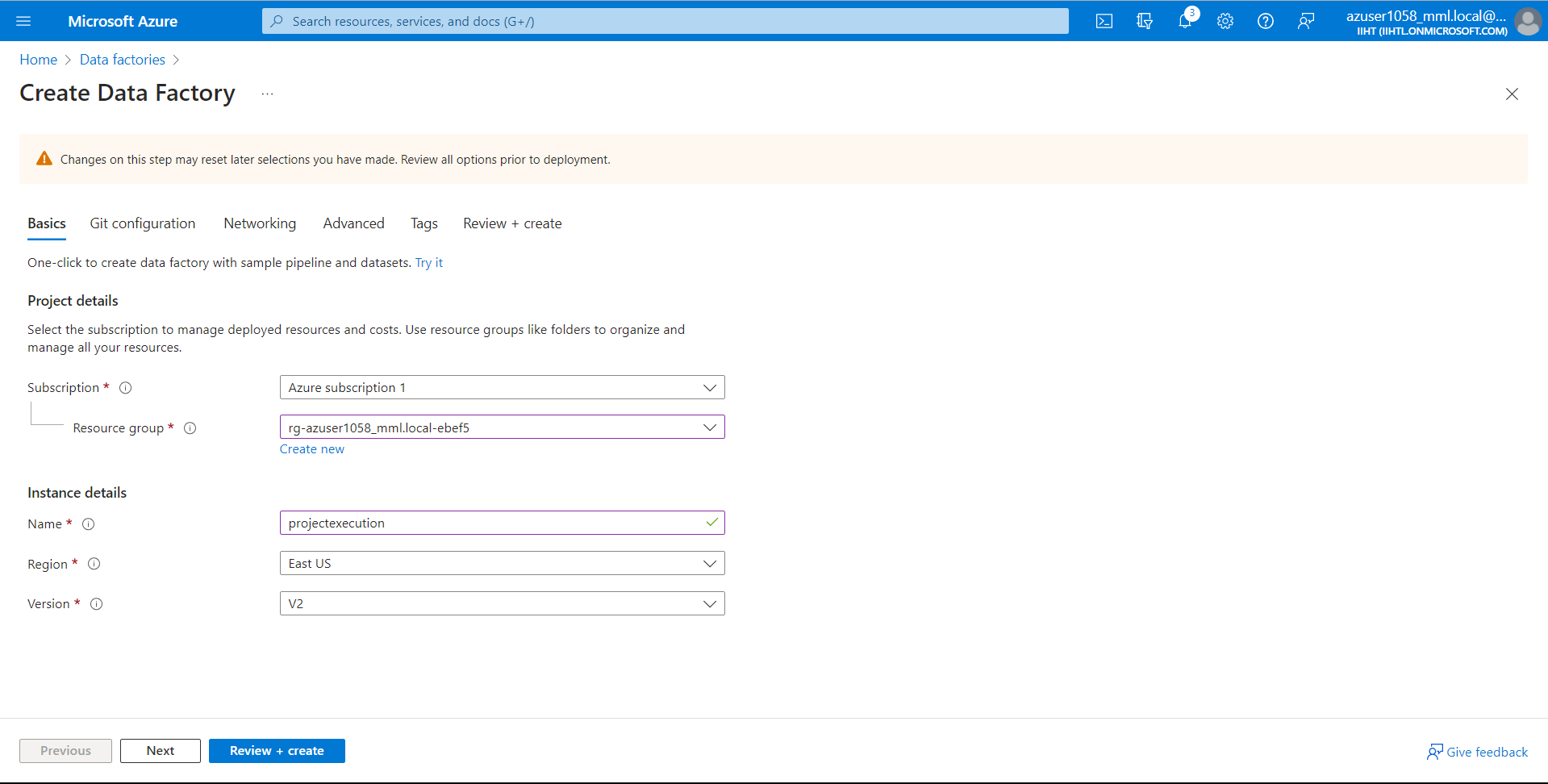
**Now select “Containers” and create a new container of required name. here I have given name as “input”.**

****

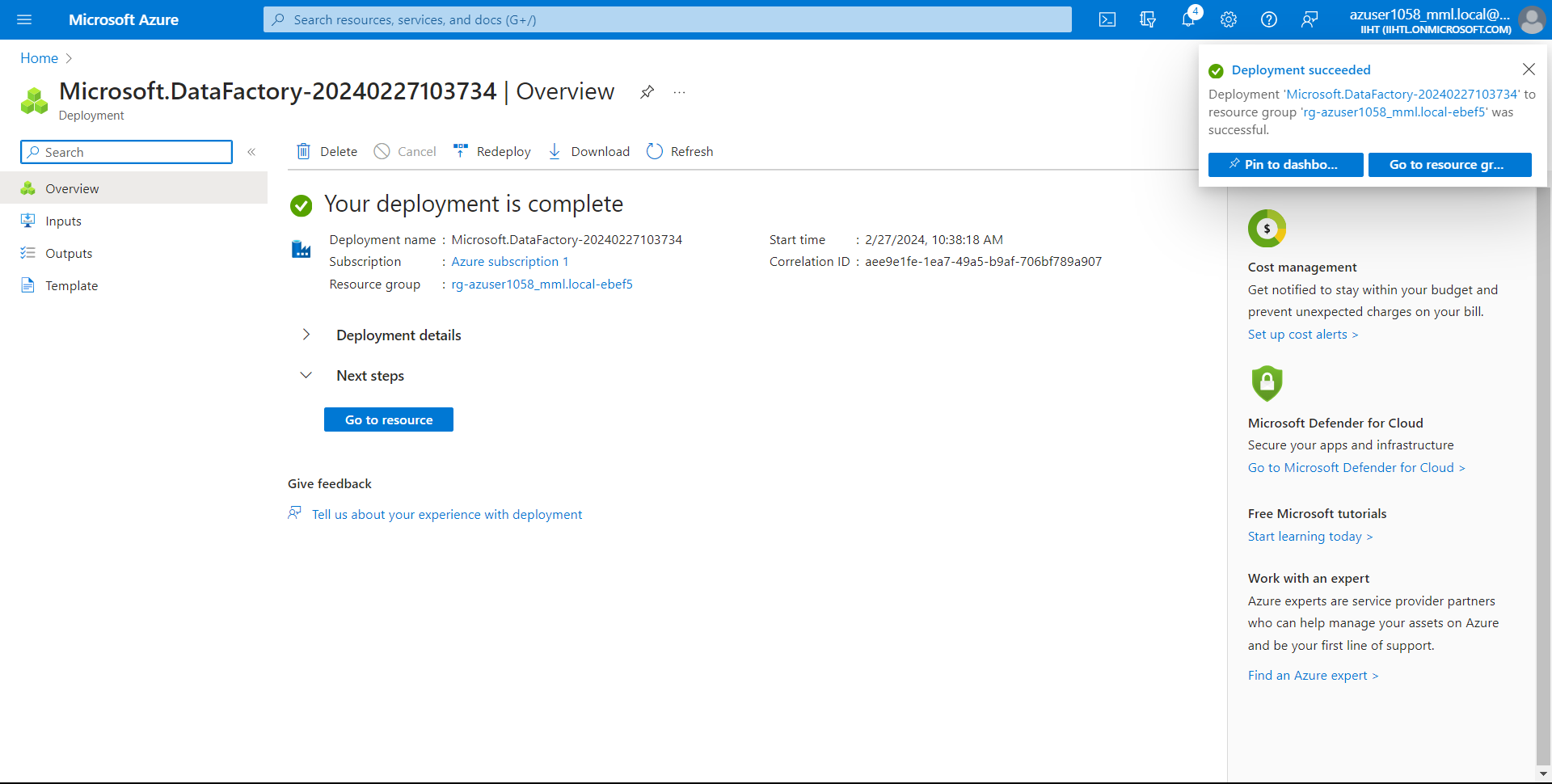
**Now upload a CSV File into input location and click on Upload.**

****

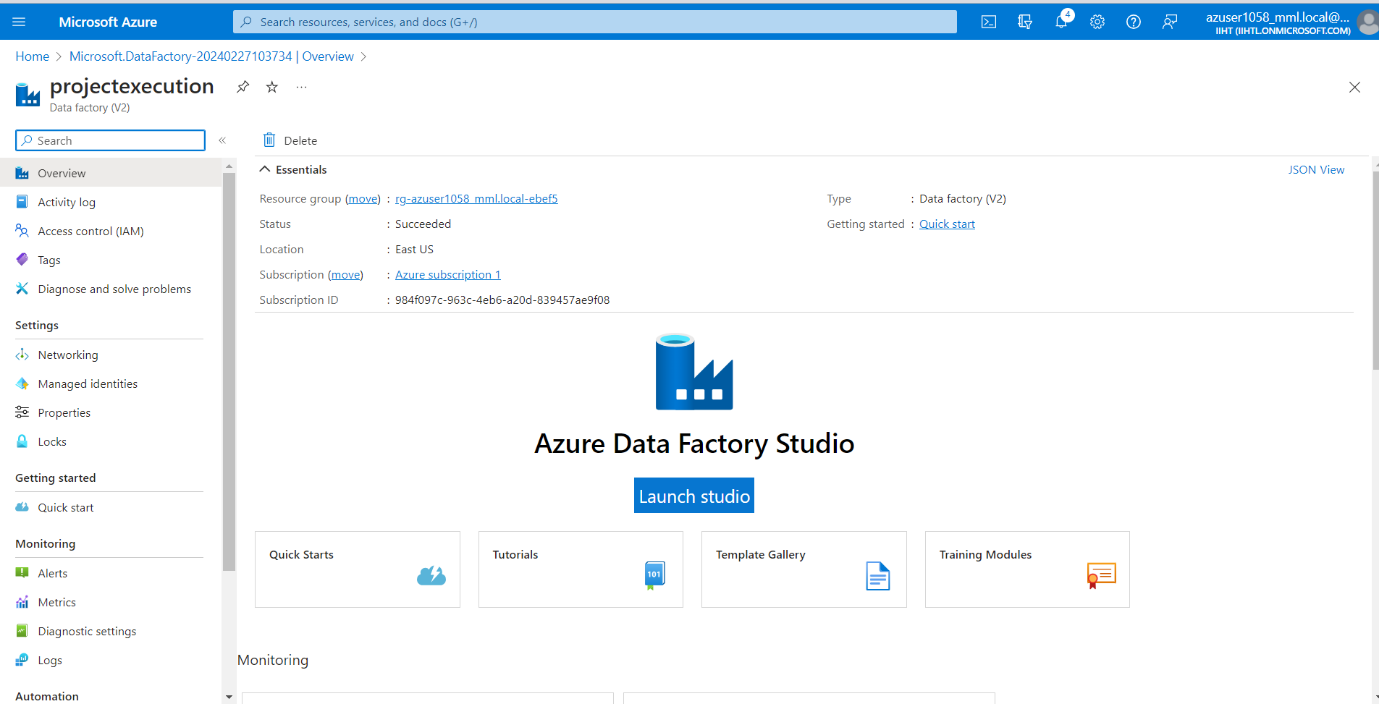
**Now lets create a Data Factory for further process. Create a Data Factory from your Azure portal.**

****

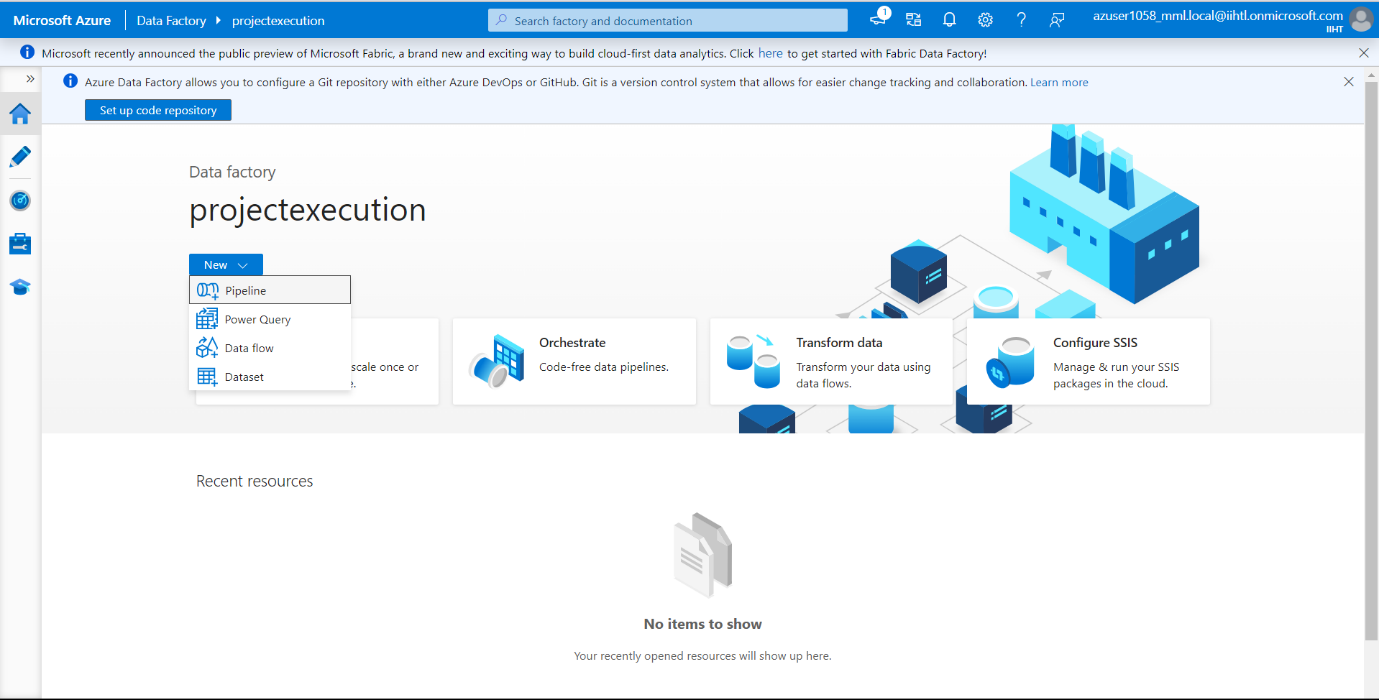
**Click on Review + create.**

****

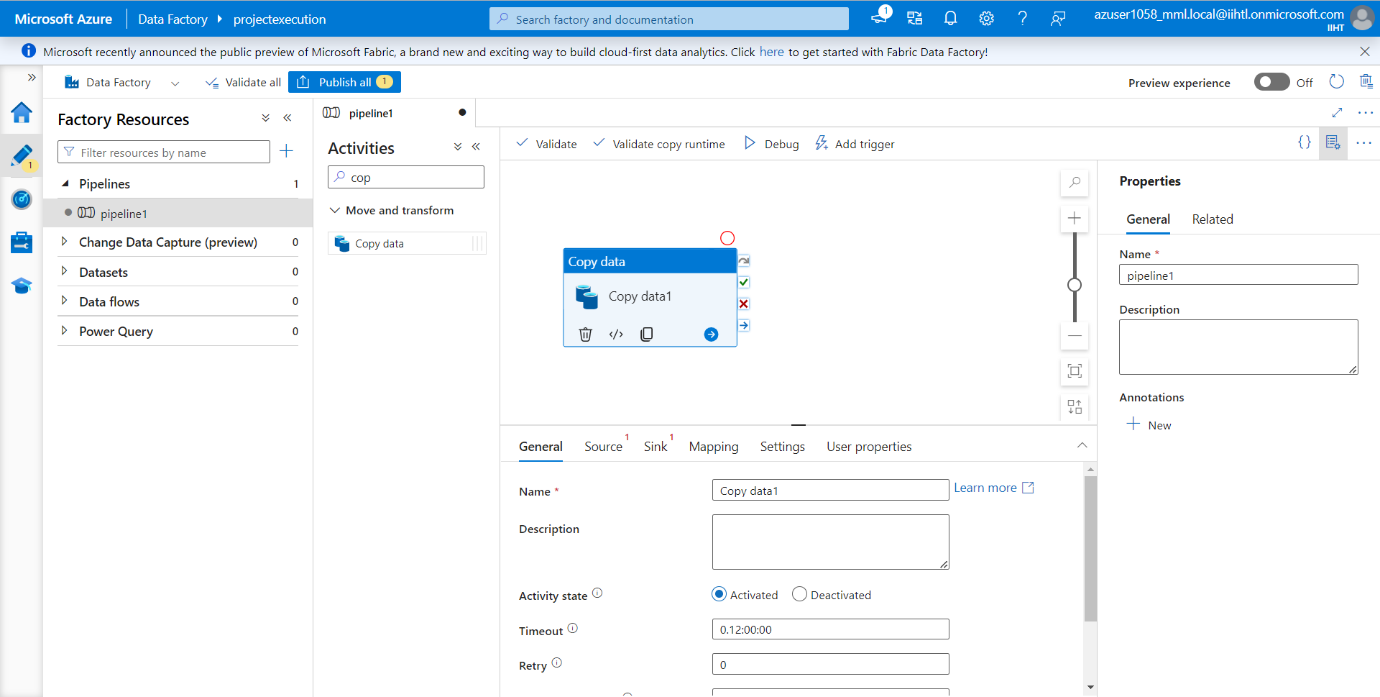
**Now Go to resource and launch the studio.**

****

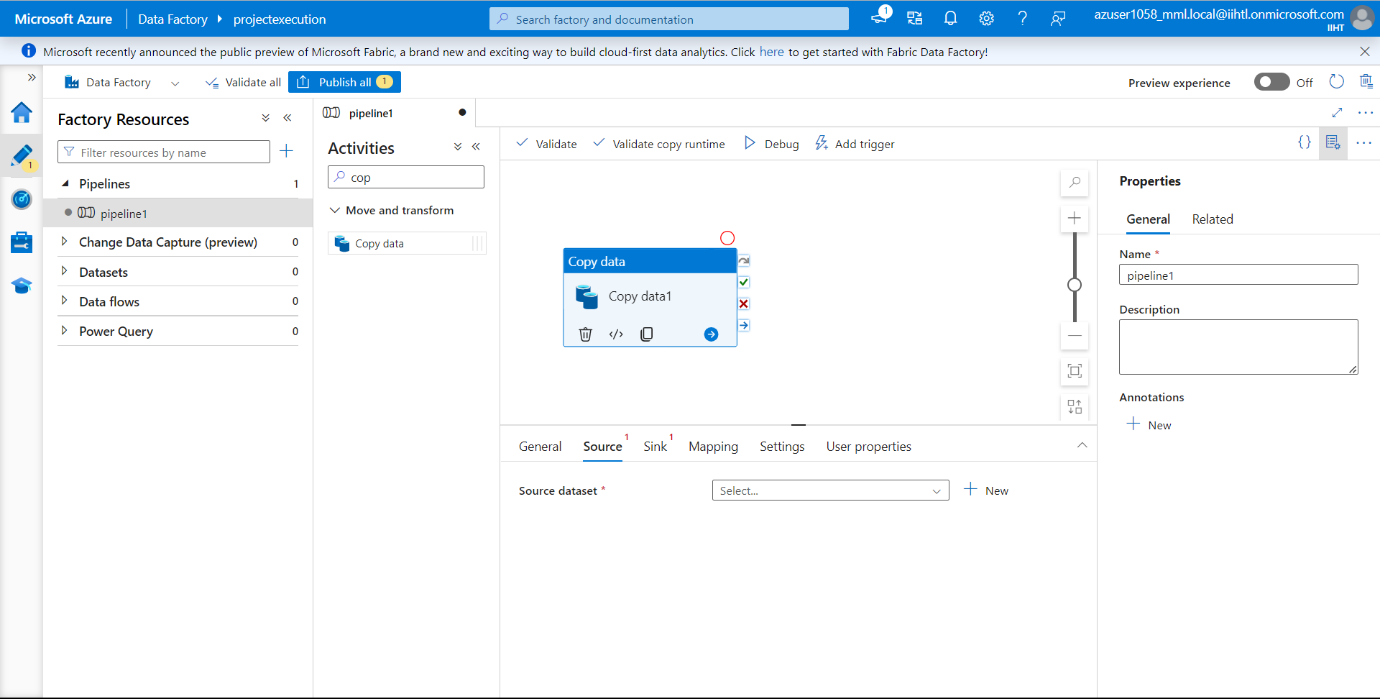
**Now lets create a pipeline from Data Factory. Click on new and select Pipeline.**

****

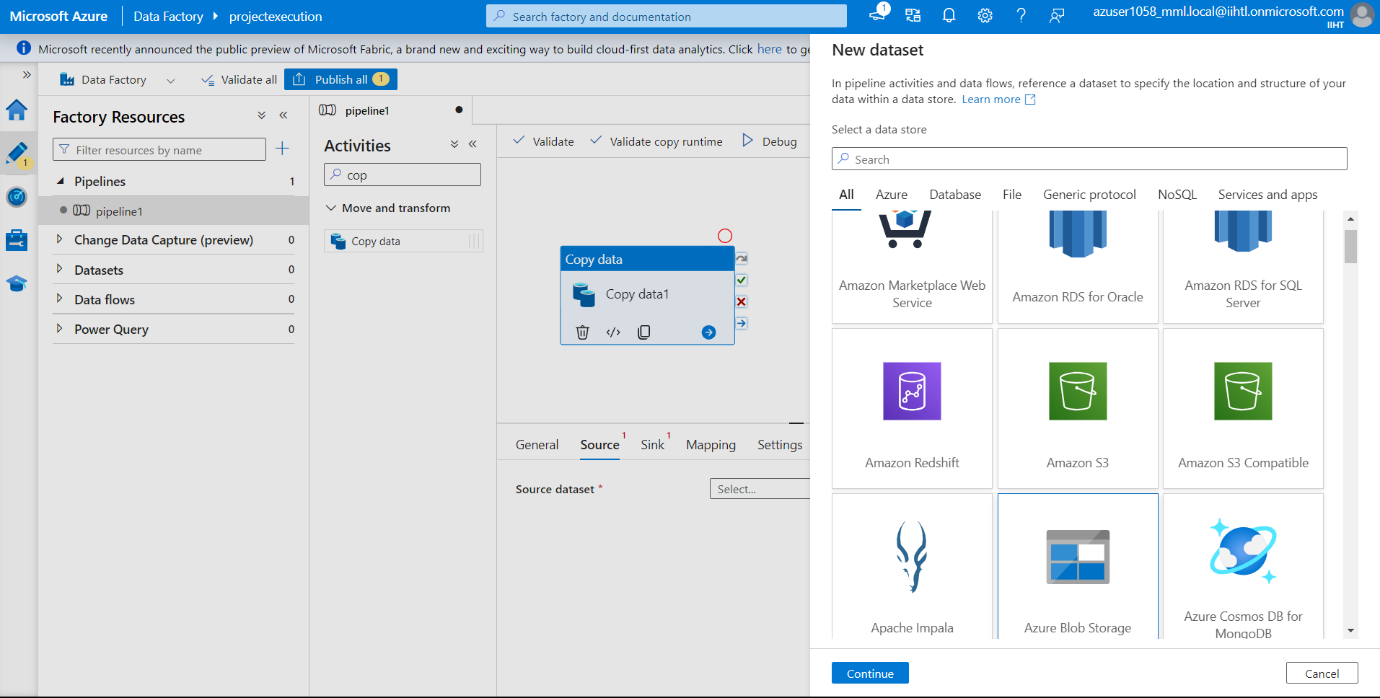
**Search for “copy data” in Activities and drag the option to the space allocated there.**

****

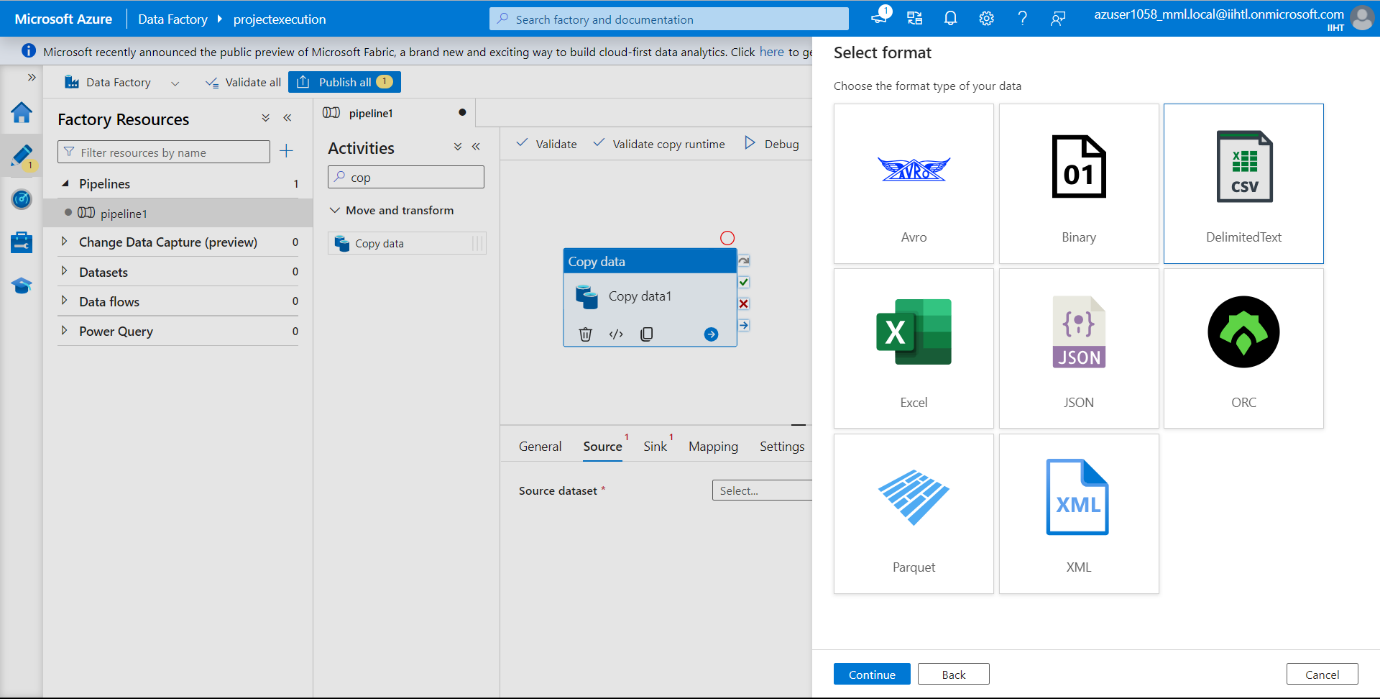
**Now Go to Source and click + New.**

****

**Now select Azure Blob Storage as New Dataset and click on continue.**

****

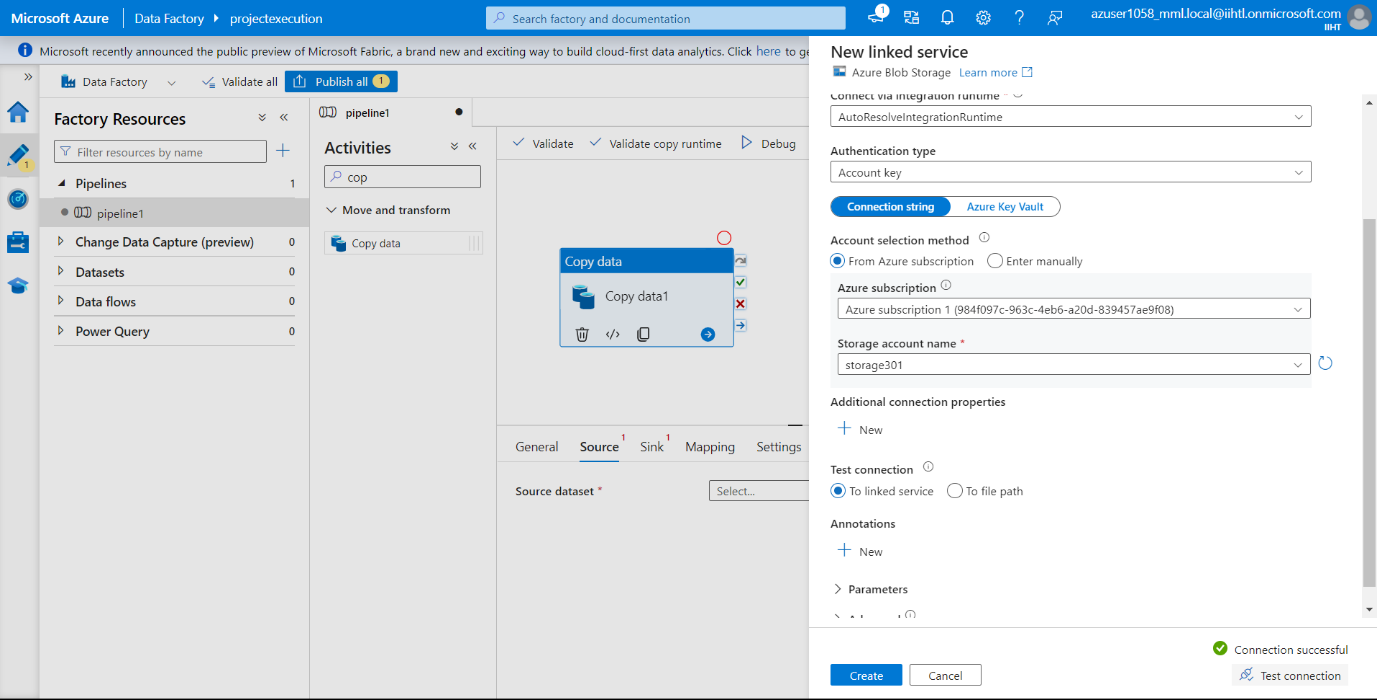
**Now we need to select the format. I have selected CSV file and click on continue.**

****

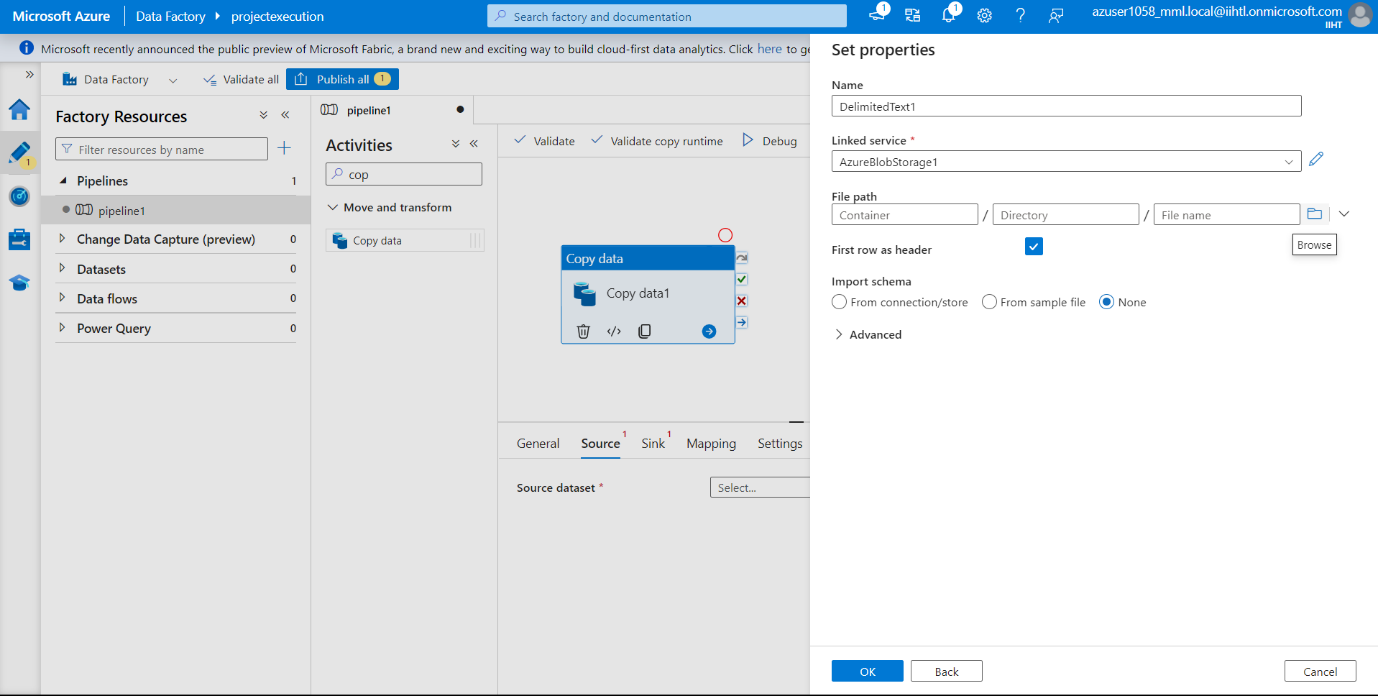
**Now lets do the Linked Service. Enter your required name and for storage account name I have selected the created storage account before and lets test the connection.**

****

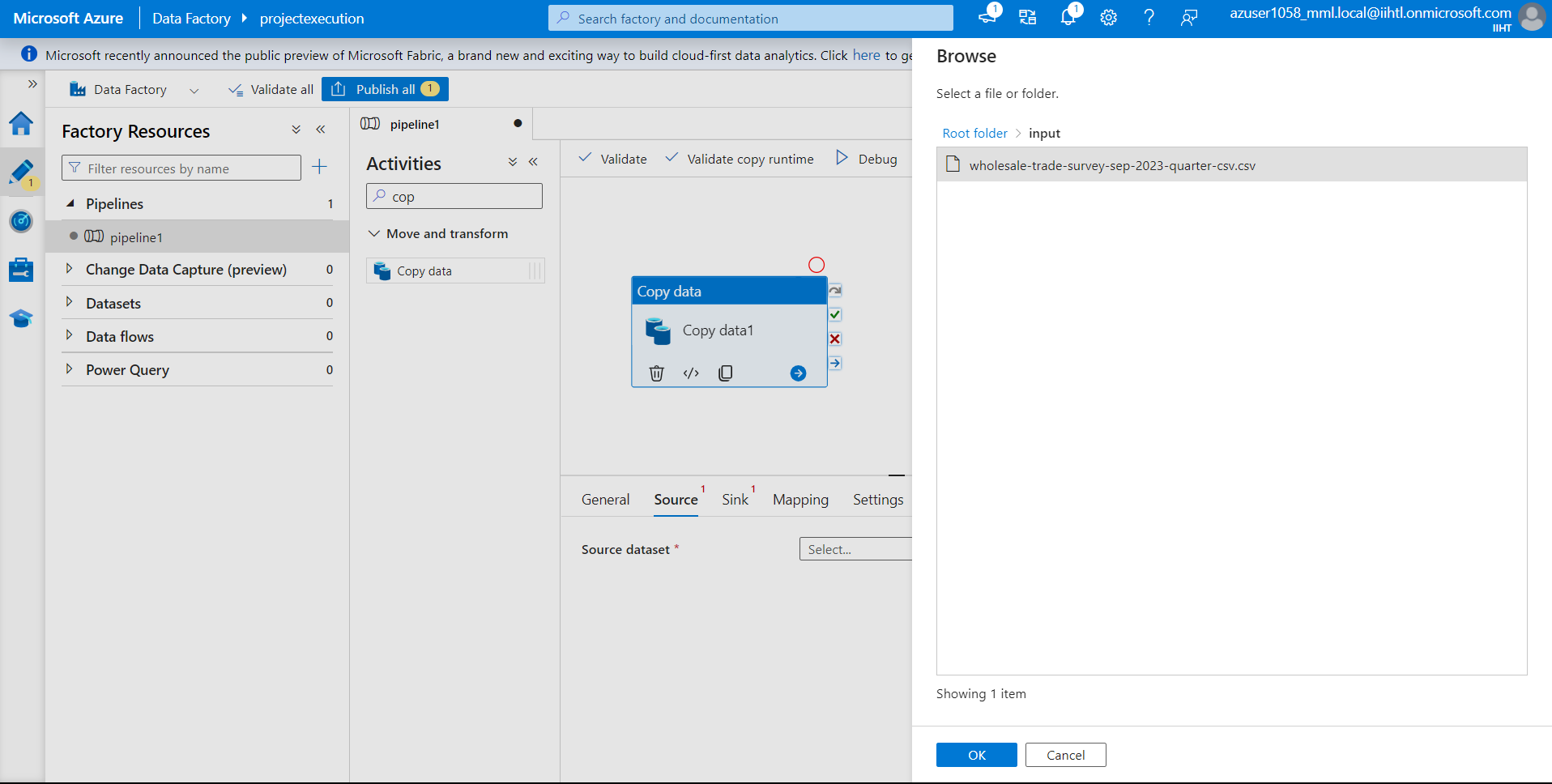
**Test connection is successful. Now click on Create.**

****

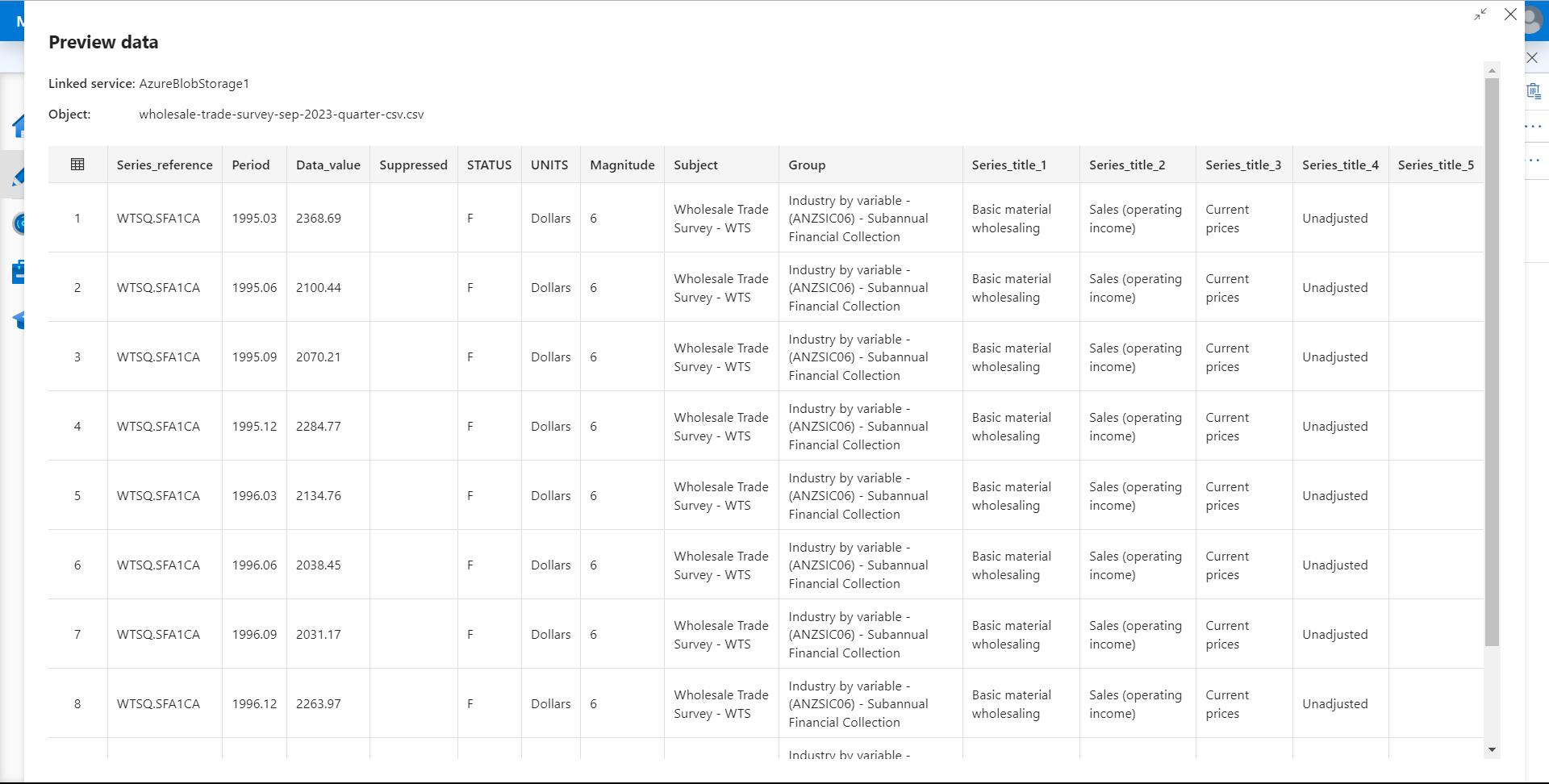
**Now we need to set properties. Now go to file path and click on browse.**

****

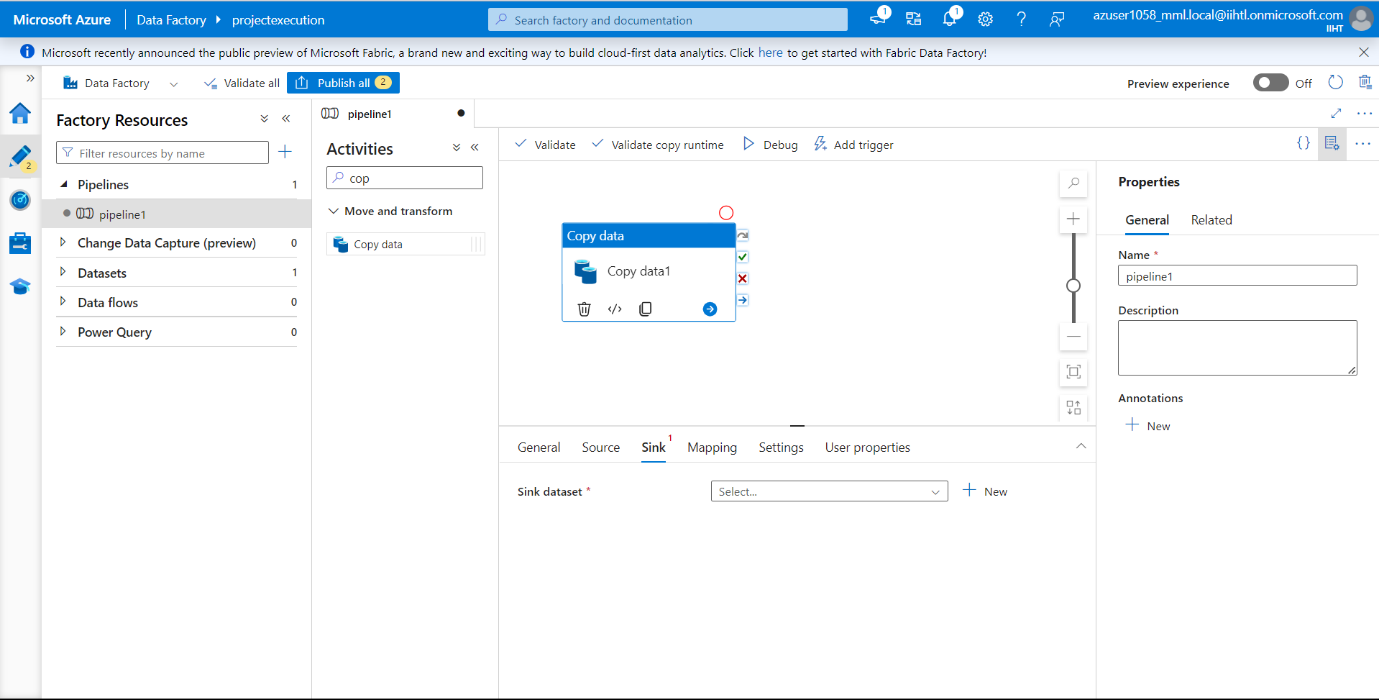
**Select the Source file. I have CSV file in input File and I have selected that click on Ok.**

****

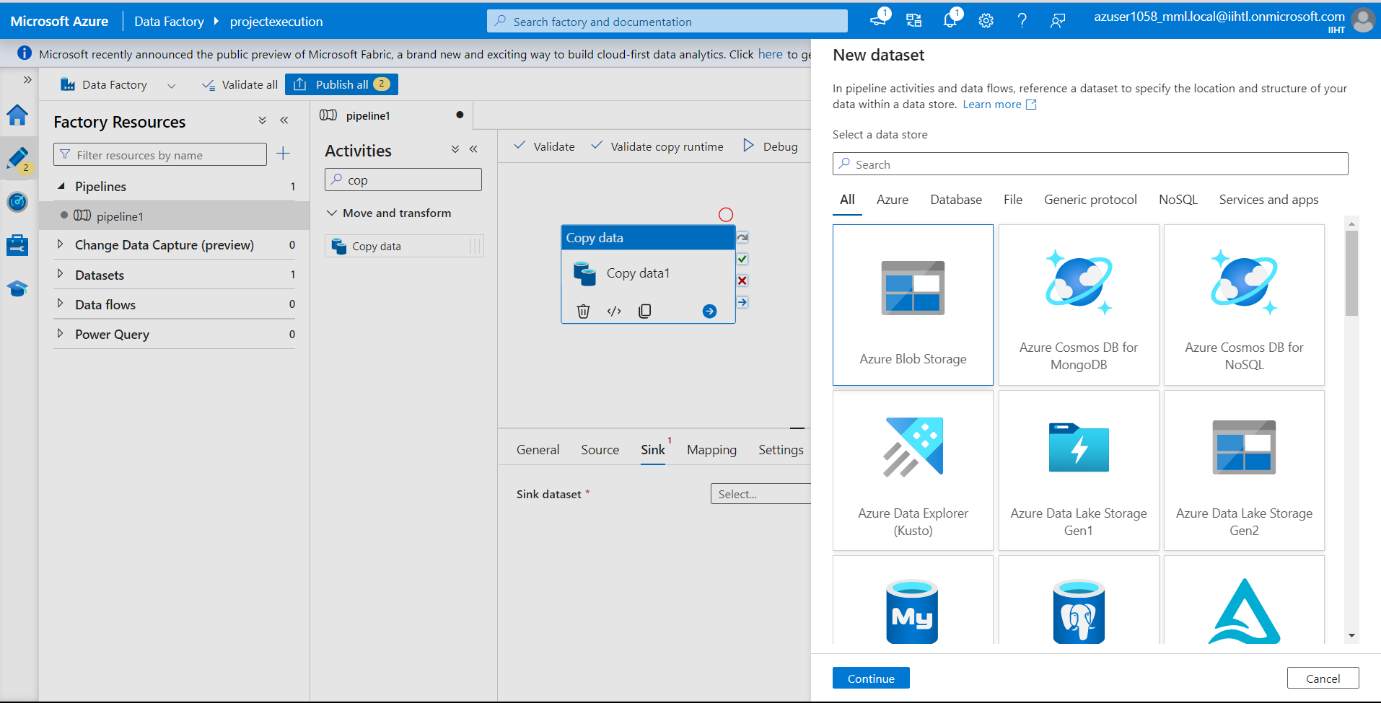
**Now the file is uploaded in the dataset. Let me Preview the data.**

****

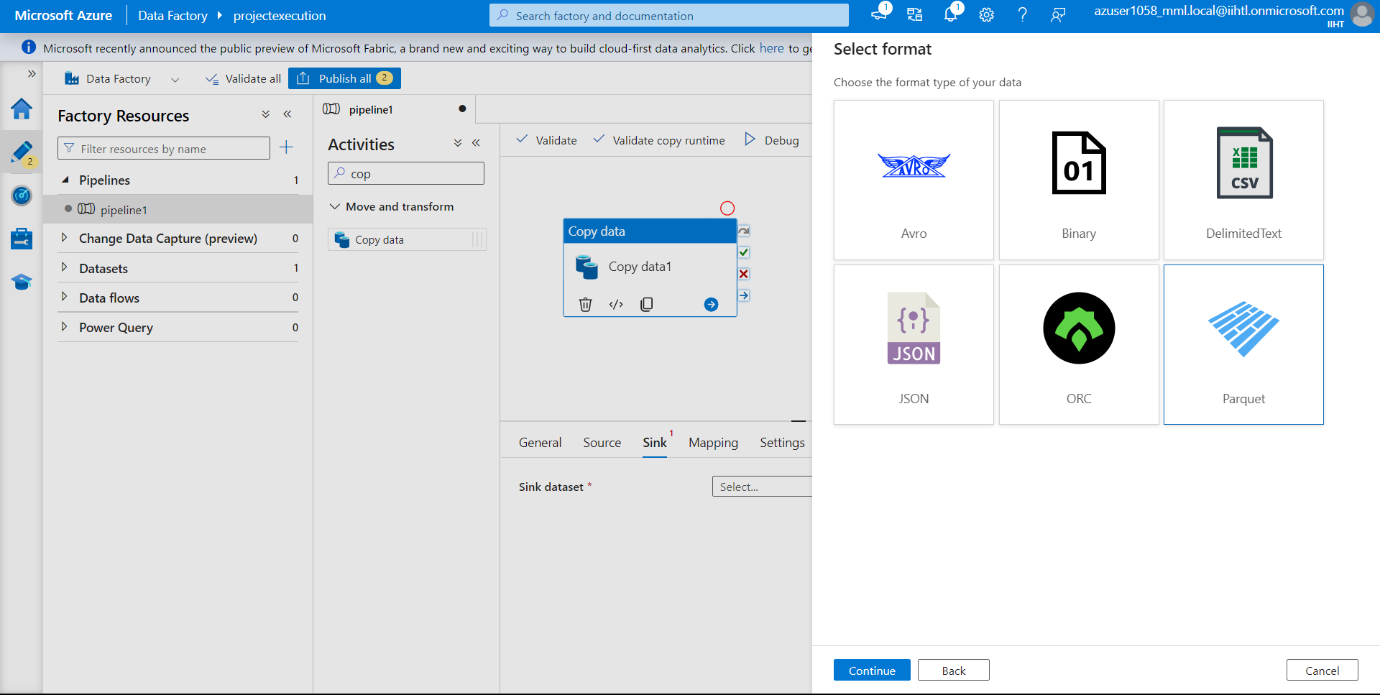
**Now go to Sink and click on + New.**

****

**Now create another dataset as Azure blob storage and click on continue.**

****

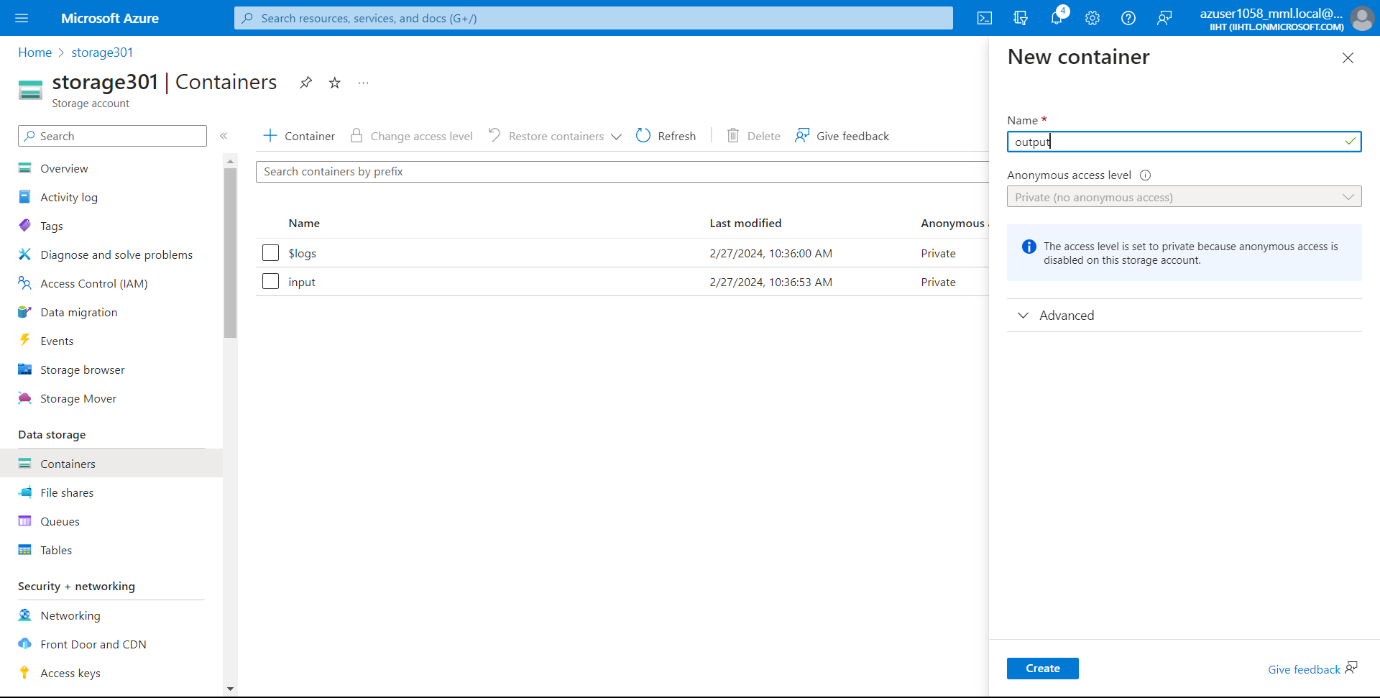
**Next Select Format as Parquet and click on Continue.**

****

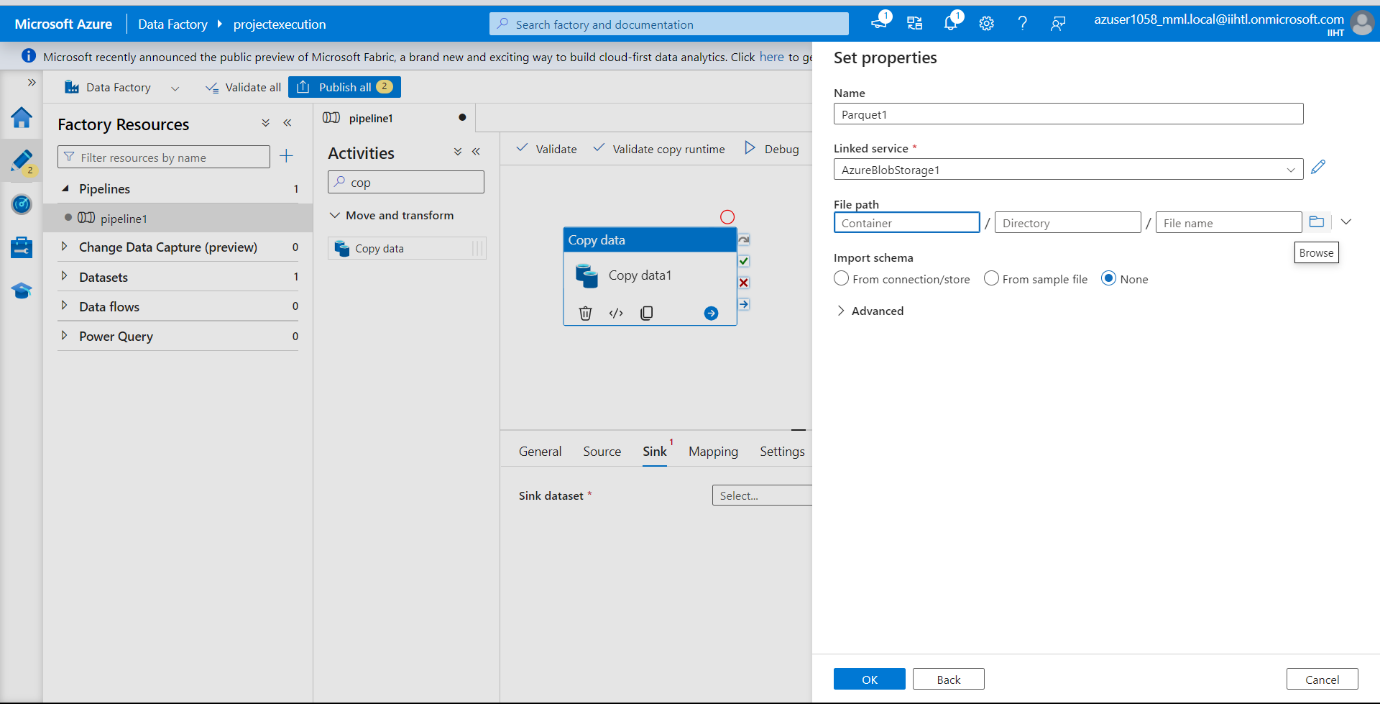
**Now select the default Linked service or you can create a new linked service.**

****

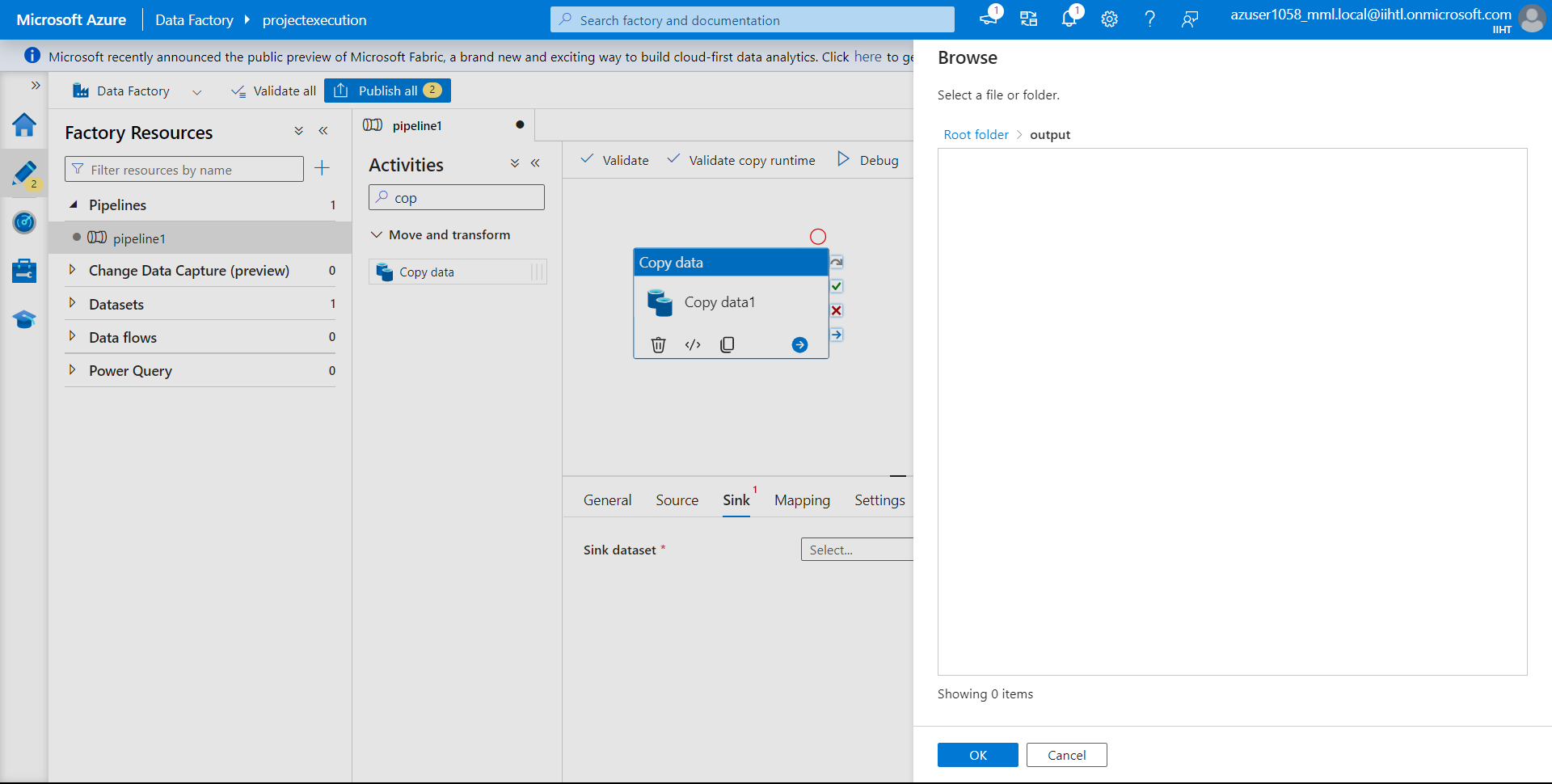
**Now I have created another new container in my storage account to store the output file.**

****

**Next set properties. Go to file path and click on browse and select the file name where we need to store the output.**

****

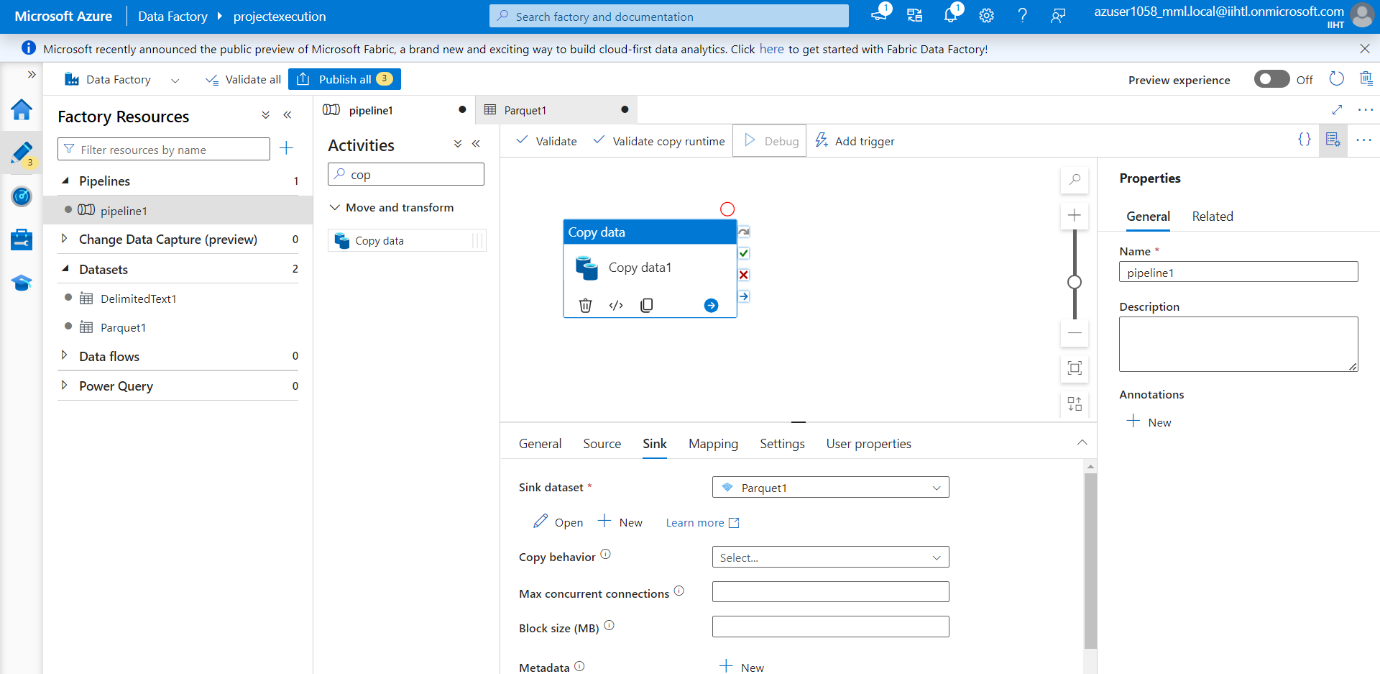
**Here I wanted to store the output in the new container which i named it as output and click on ok.**

****

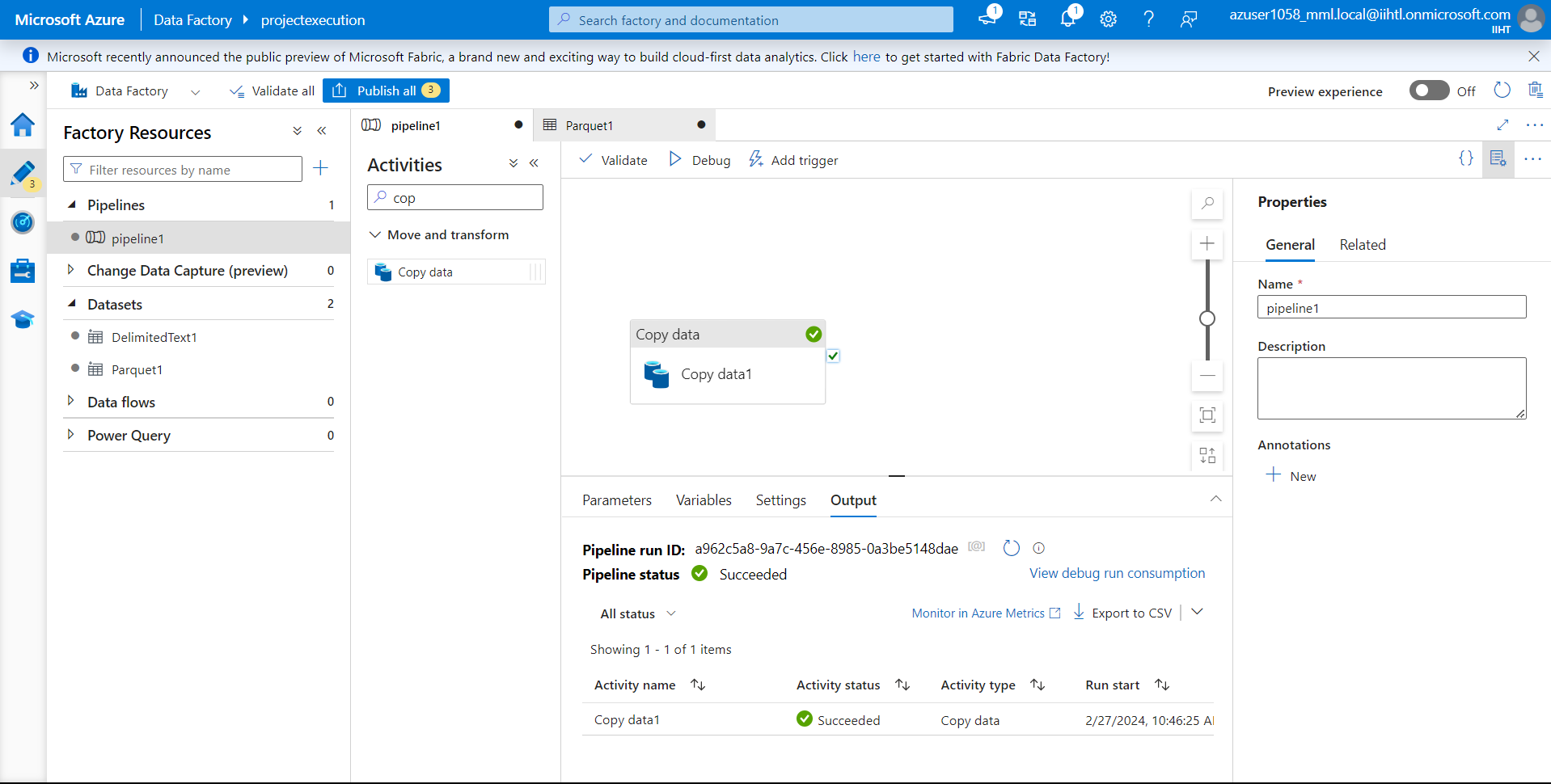
**Go to Sink connection and select open and save the output file as output.parquet in Output container.**

****

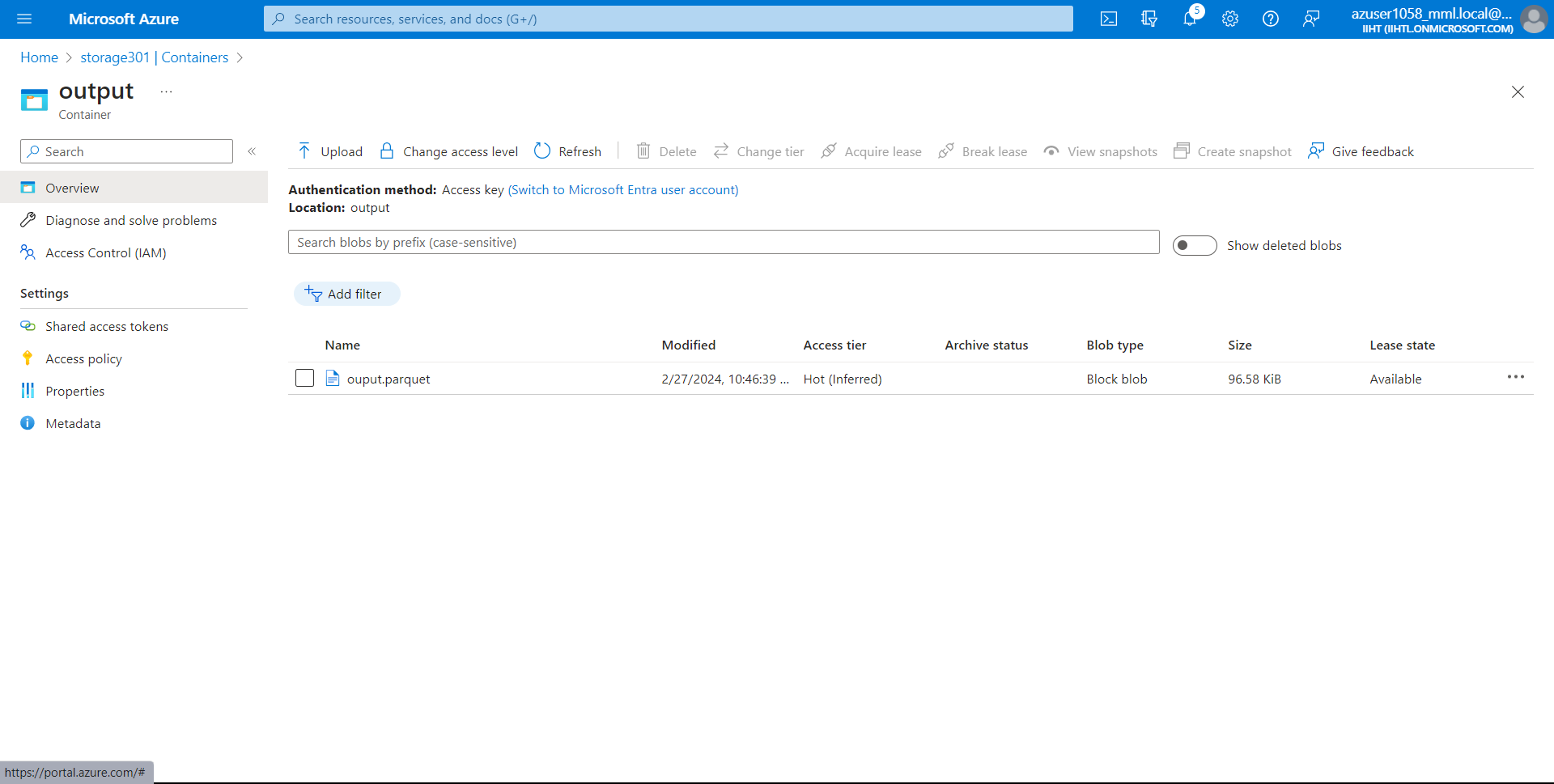
**Now go to pipeline and click on “Debug”.**

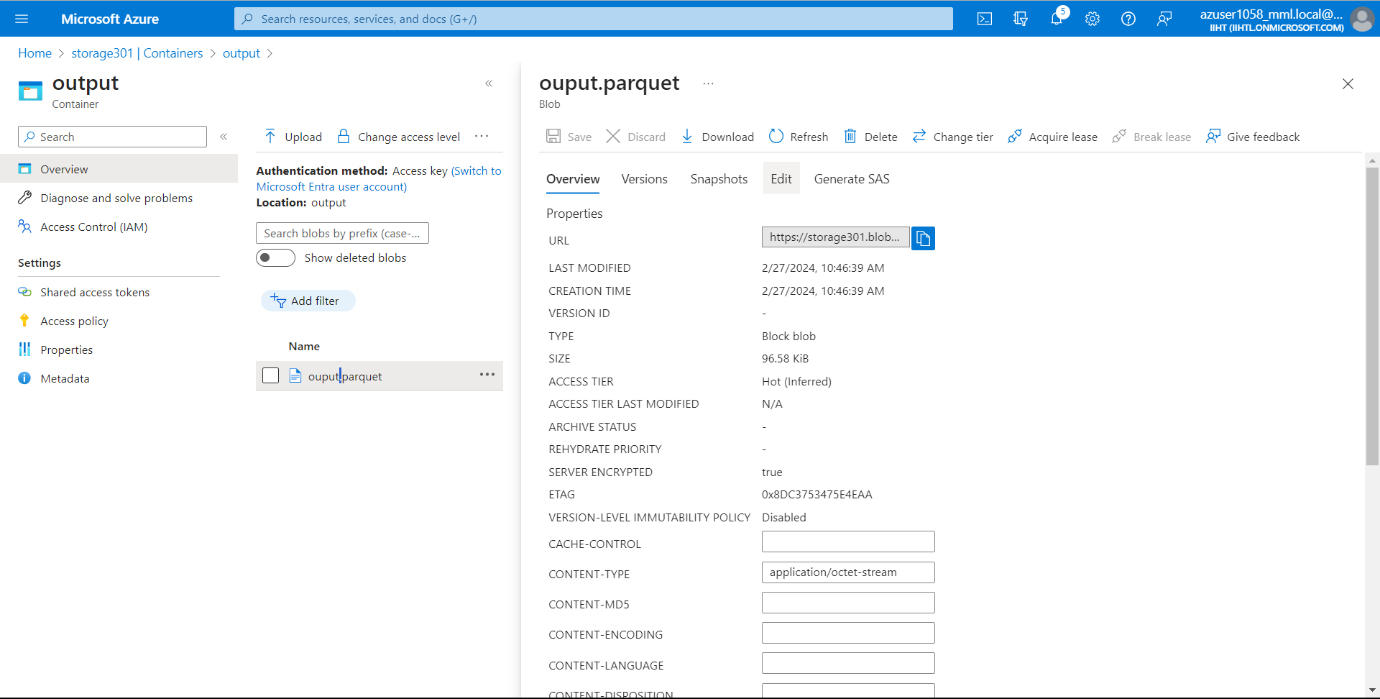
****

**Now we have successfully debugged.**

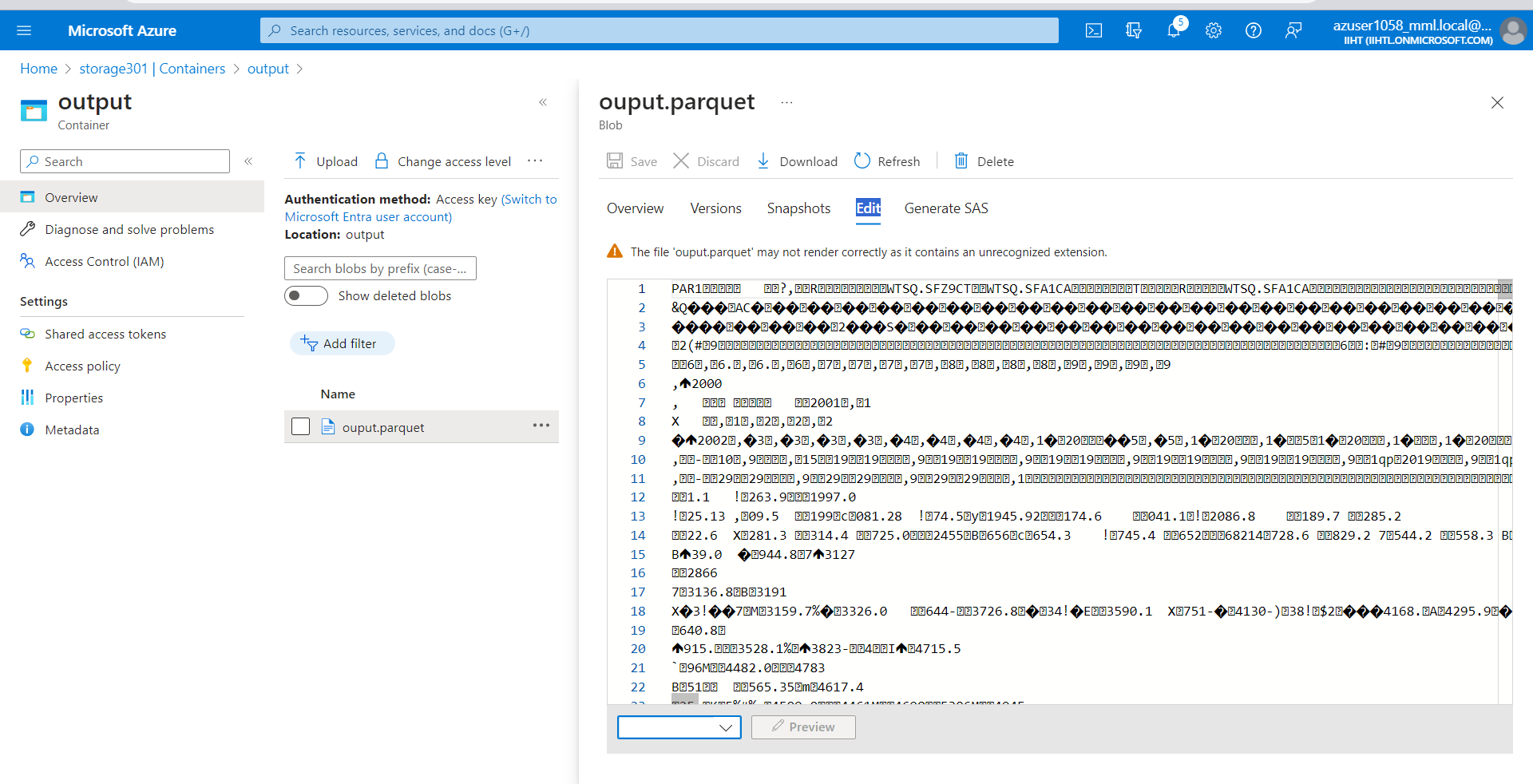
****

**Now check in the Output container which we have created for resulted output.**

**Open the output.parquet file and select edit.**

****

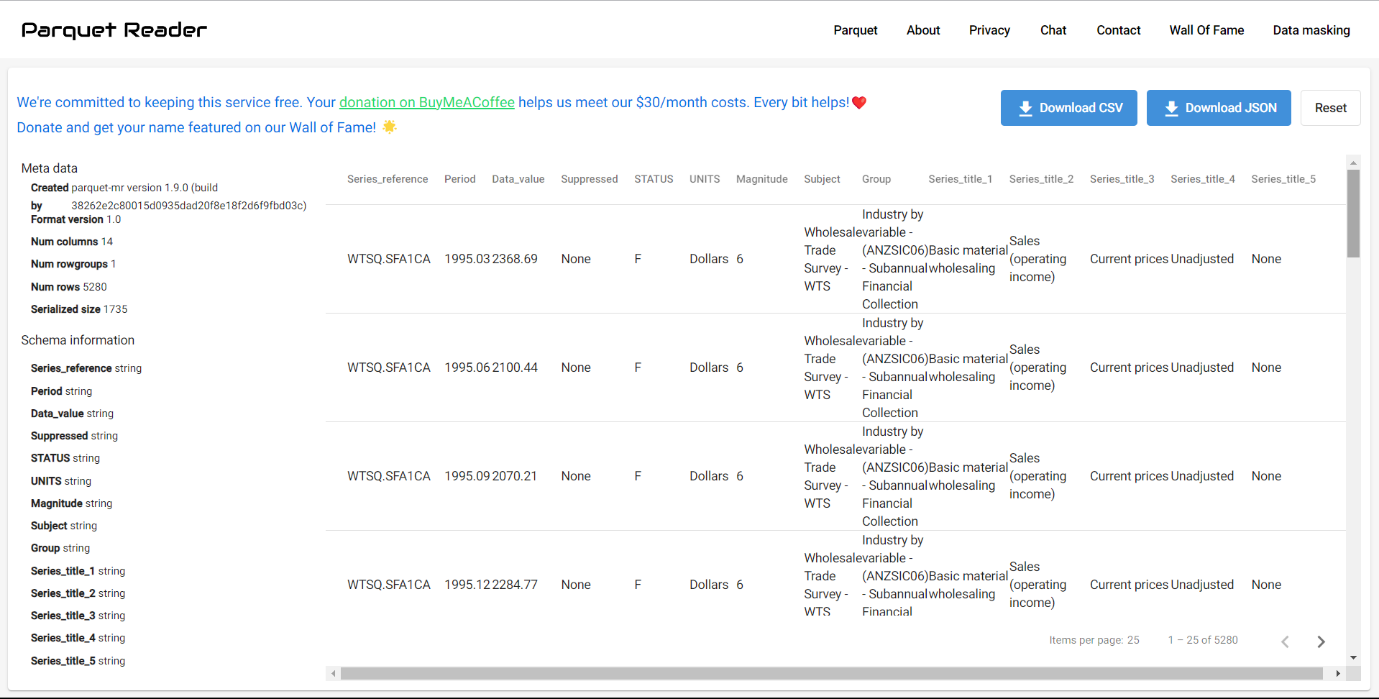
**Now we can see that the CSV File Format is Transformed into Parquet Format. And we can download the output. Parquet by selecting on download option.**

**Now let’s use a online platform to open the Parquet file.**

**I have uploaded the downloaded output. Parquet file.**

****

**FINAL OUTPUT:**

****

**Conclusion:**

In conclusion, the implementation of Azure Data Factory for copying data from a CSV file in Azure Storage to a Parquet file in another location offers a robust and scalable solution for data ingestion and transformation within the Azure ecosystem. By leveraging Azure Data Factory's capabilities for orchestrating data workflows, we have effectively bridged the gap between disparate data formats, ensuring seamless data integration while maintaining efficiency and reliability.

Through the creation of linked services, datasets, and pipelines, we have established a structured framework for data movement, enabling us to easily configure and manage data pipelines from source to destination. Additionally, the flexibility provided by Azure Data Factory allows for customization of data transformation activities, ensuring that the migrated data meets the specific requirements of the target environment.

Overall, this project demonstrates the power of Azure Data Factory in facilitating data-driven decision-making by enabling organizations to extract, transform, and load data from diverse sources into a unified and analytics-ready format. With Azure Data Factory, organizations can streamline their data workflows, accelerate time-to-insight, and unlock the full potential of their data assets to drive business growth and innovation.