**Alekhya Krishna Balivada Azure Databricks Assignment -7 (05-01-2024)**

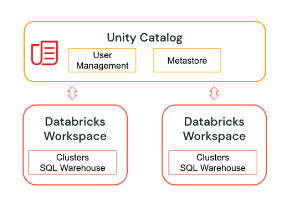
|  |  |
| --- | --- |
| Name | Alekhya Krishna Balivada |
| Date | 05-01-2024 |
| Type | Daily Assignment |
| Topic | Lakehouse, Unity Catalog, Azure Data Factory |

**Lakehouse:** New systems are beginning to emerge that address the limitations of data lakes. A lakehouse is a new, open architecture that combines the best elements of data lakes and data warehouses. Lakehouses are enabled by a new system design: implementing similar data structures and data management features to those in a data warehouse directly on top of low cost cloud storage in open formats. They are what you would get if you had to redesign data warehouses in the modern world, now that cheap and highly reliable storage (in the form of object stores) are available.

* Lake house has the following key features:

1. Transaction support
2. Schema governance
3. BI support
4. End-to-end streaming

**Unity Catalog:** Unity Catalog provides centralized access control, auditing, lineage, and data discovery capabilities across Azure Databricks workspaces.



* Key features of unity catalog include:

1. Define once and secure everywhere
2. **Built-in auditing and lineage**
3. **Data discovery and so on**

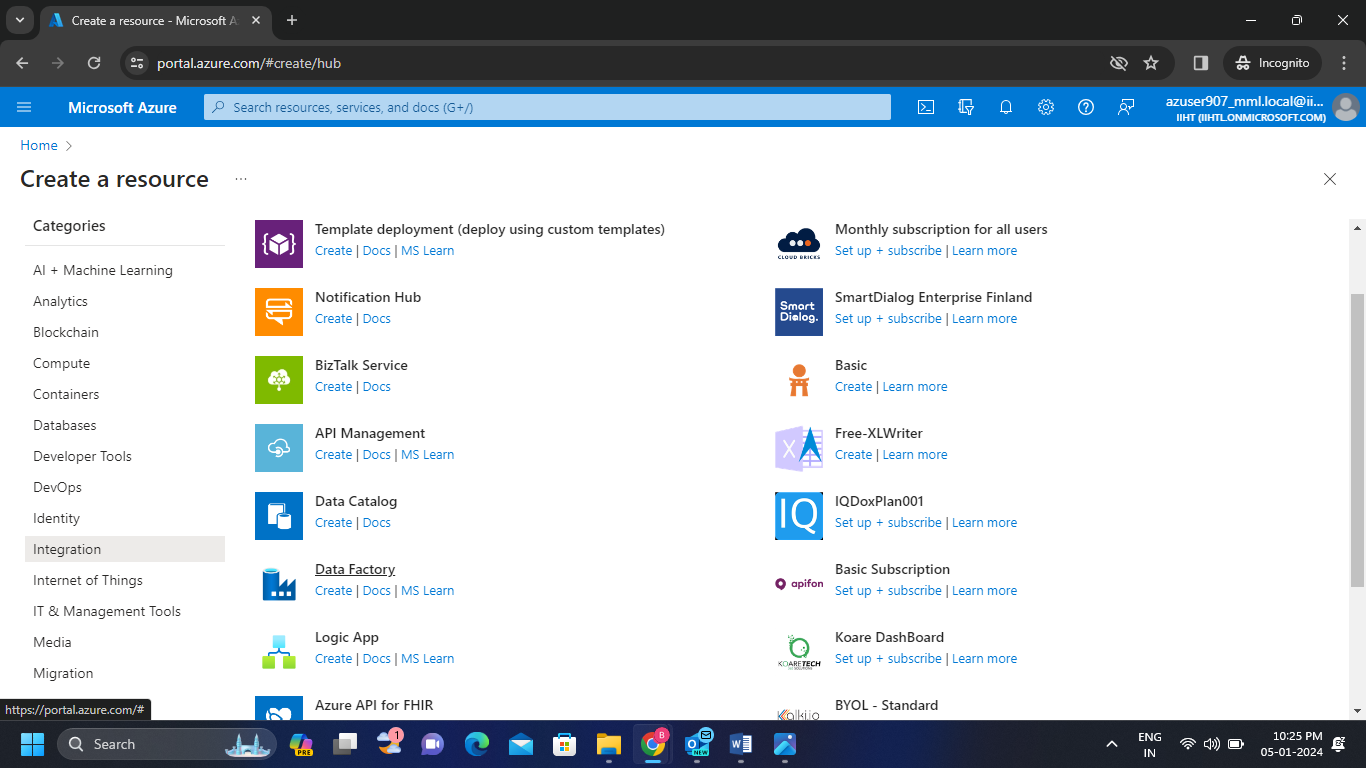
**Azure Data Factory:**

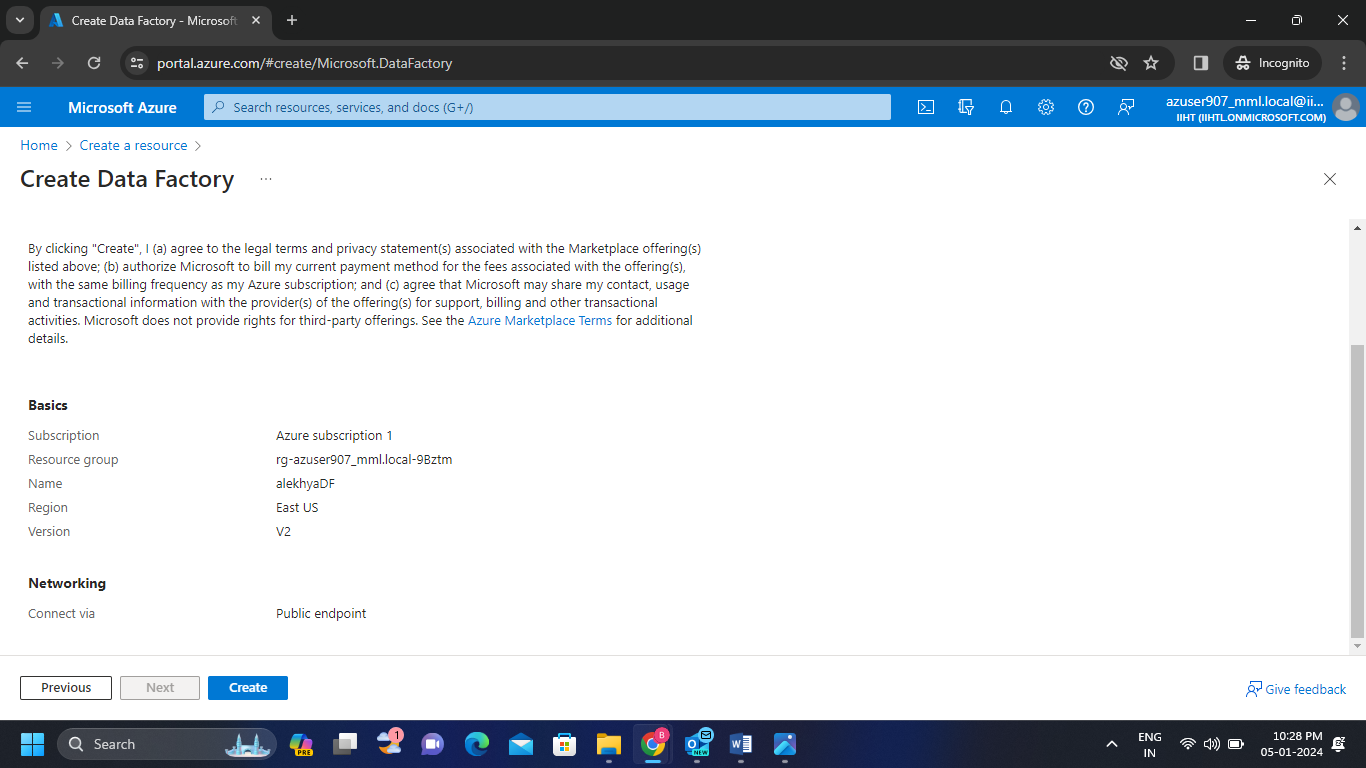
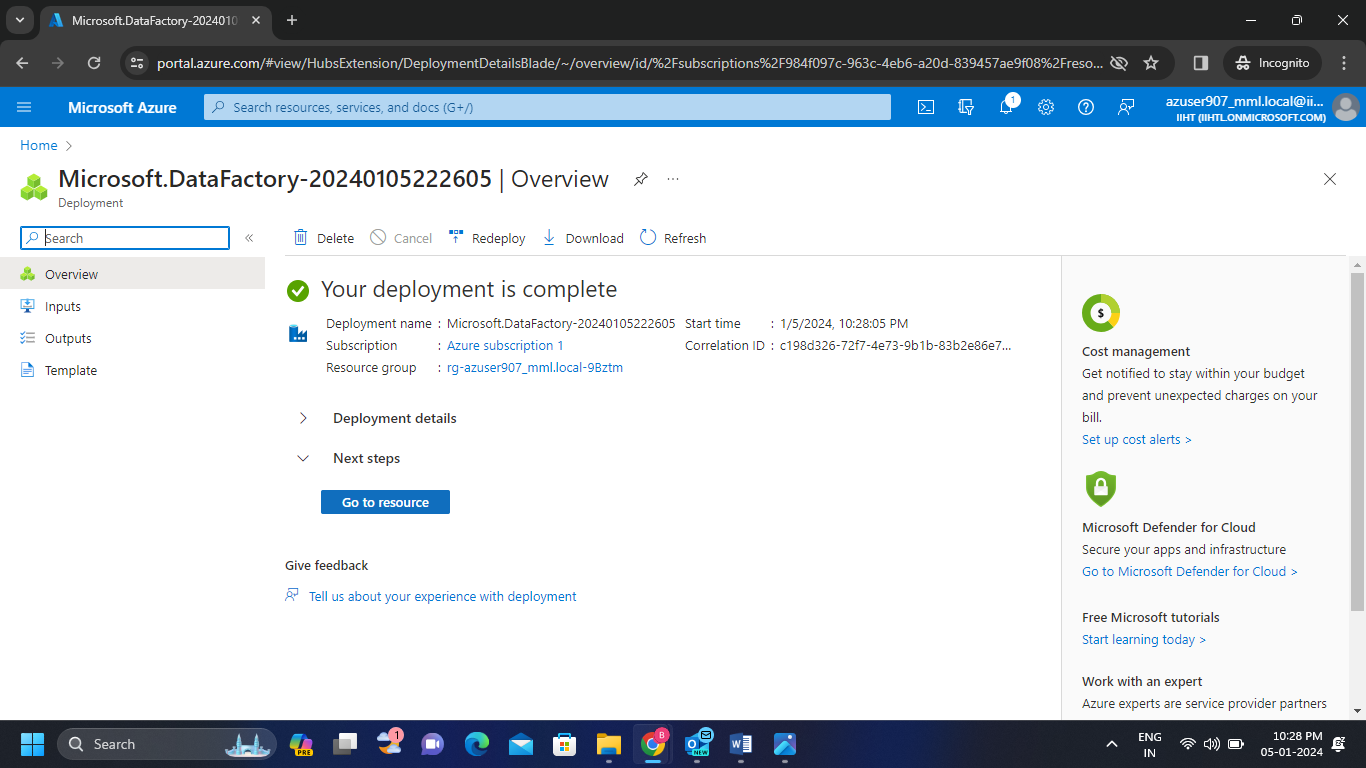
Azure Data Factory is a cloud-based, fully managed data integration service provided by Microsoft Azure. It enables users to create, orchestrate, and automate the process of collecting, transforming, and moving data from disparate sources to various destinations. This includes both on-premises and cloud-based data sources and targets.

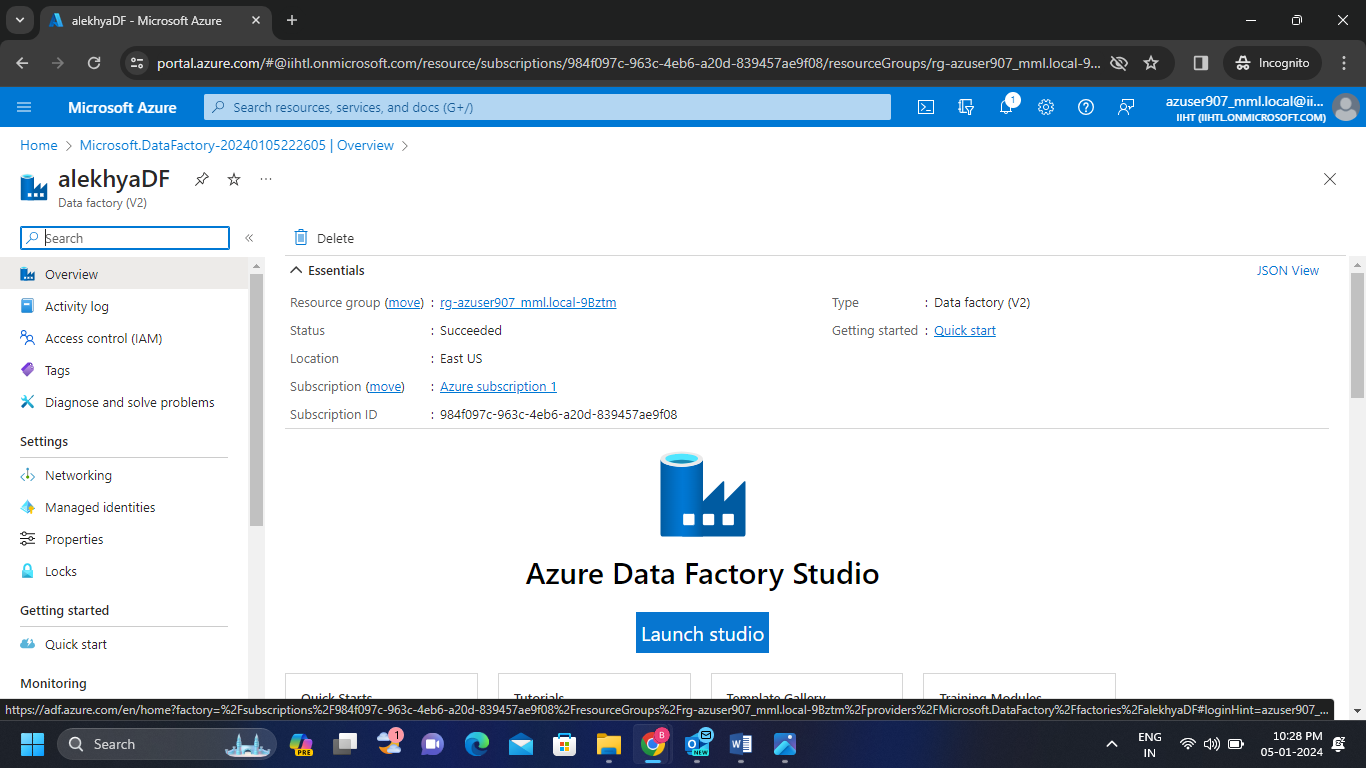
Key components and functionalities of Azure Data Factory include:

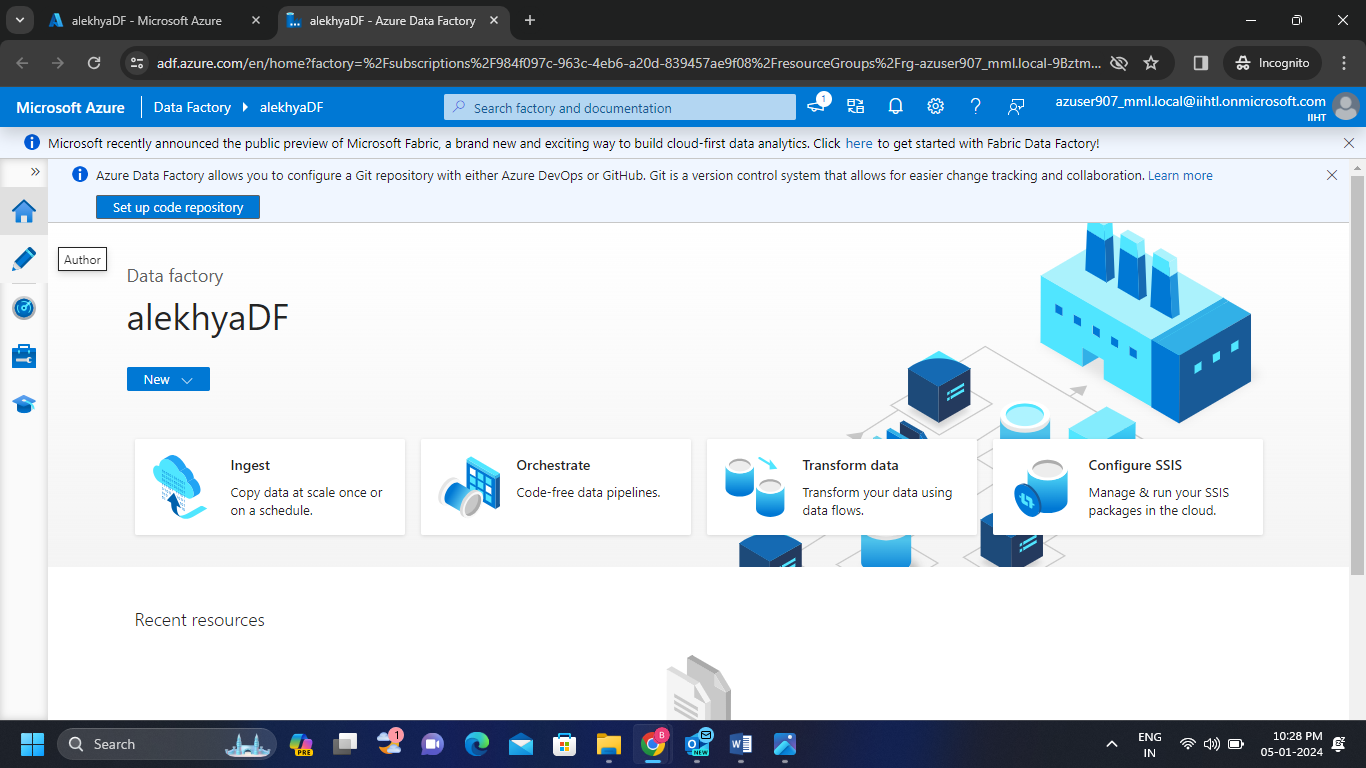
1. Data pipelines
2. Data connections
3. Data movement
4. Data transformation
5. Monitoring and Management
6. Integrating with other Azure services

**Copying Data in Azure Data Factory**

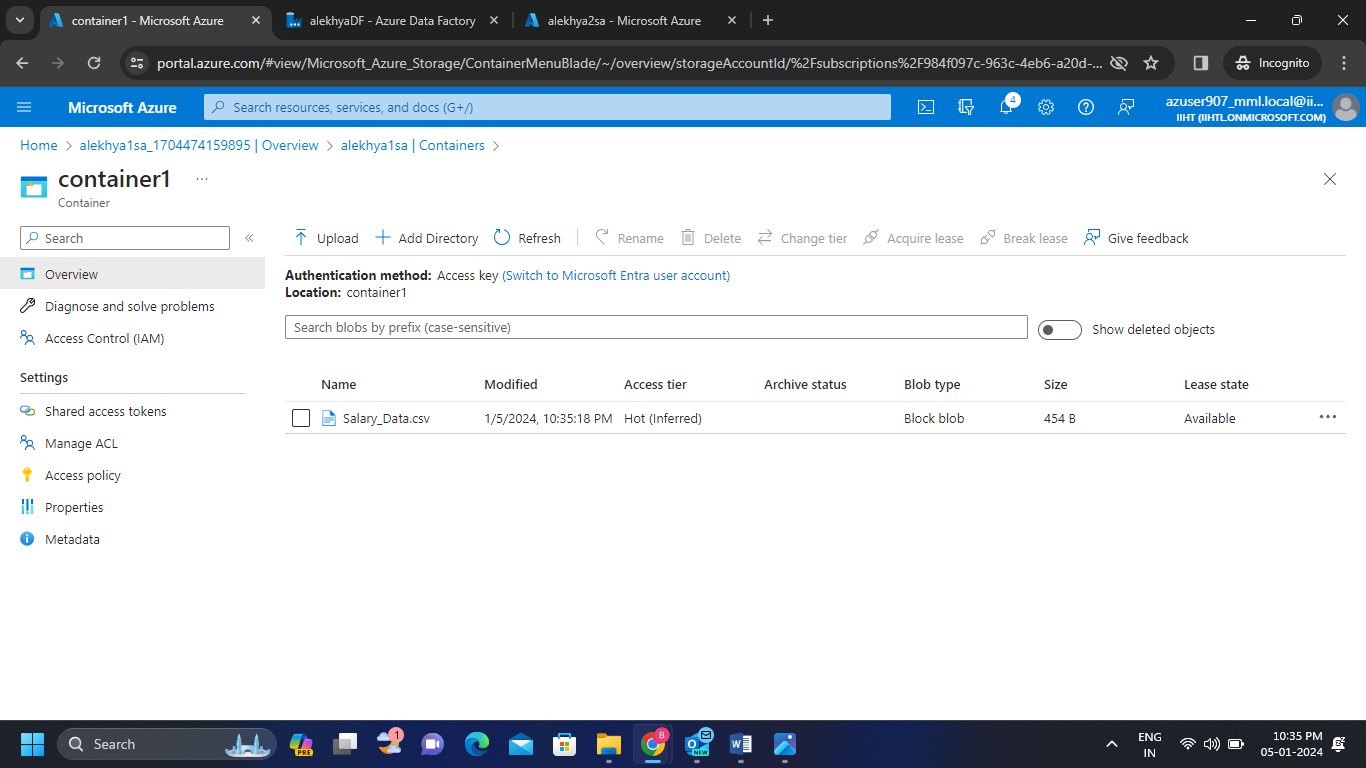
**Step-1:**  Create an Azure Data Factory by clicking on Data Factory.

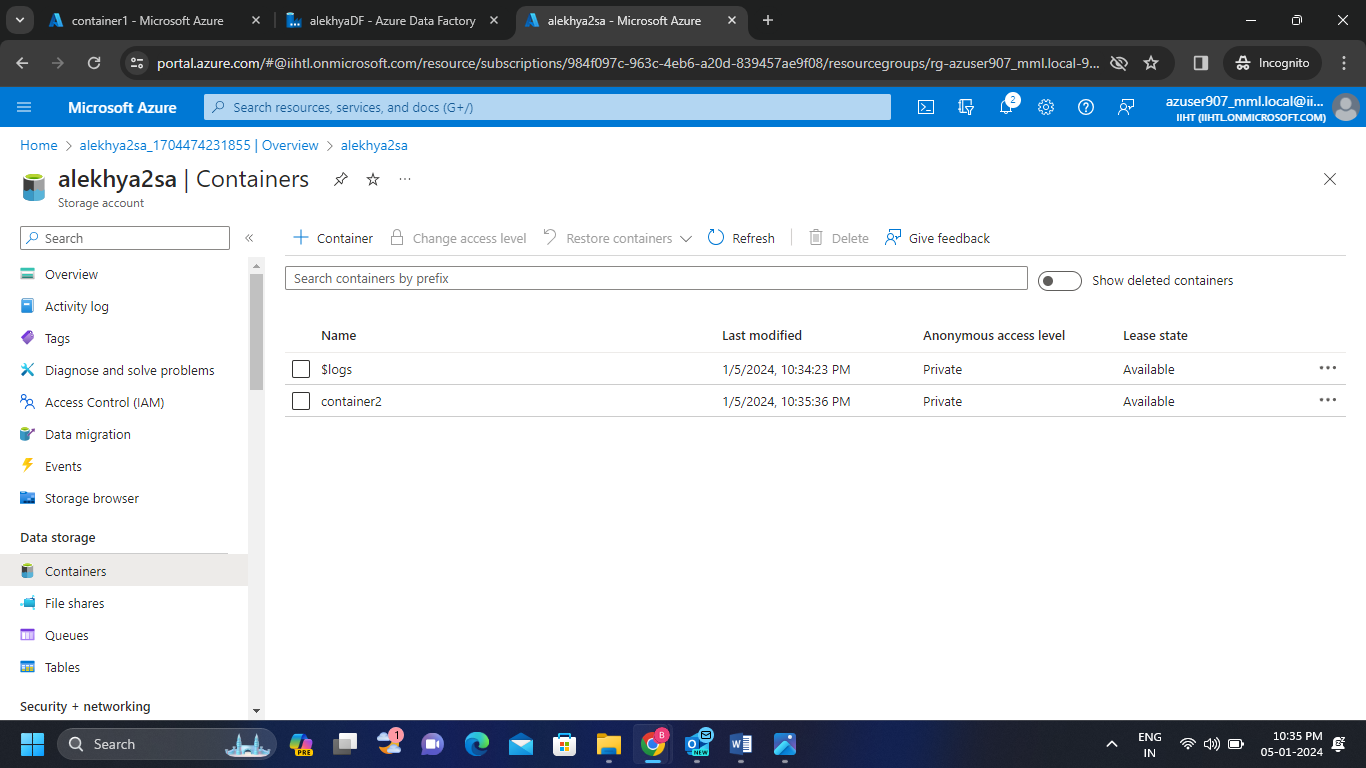
****

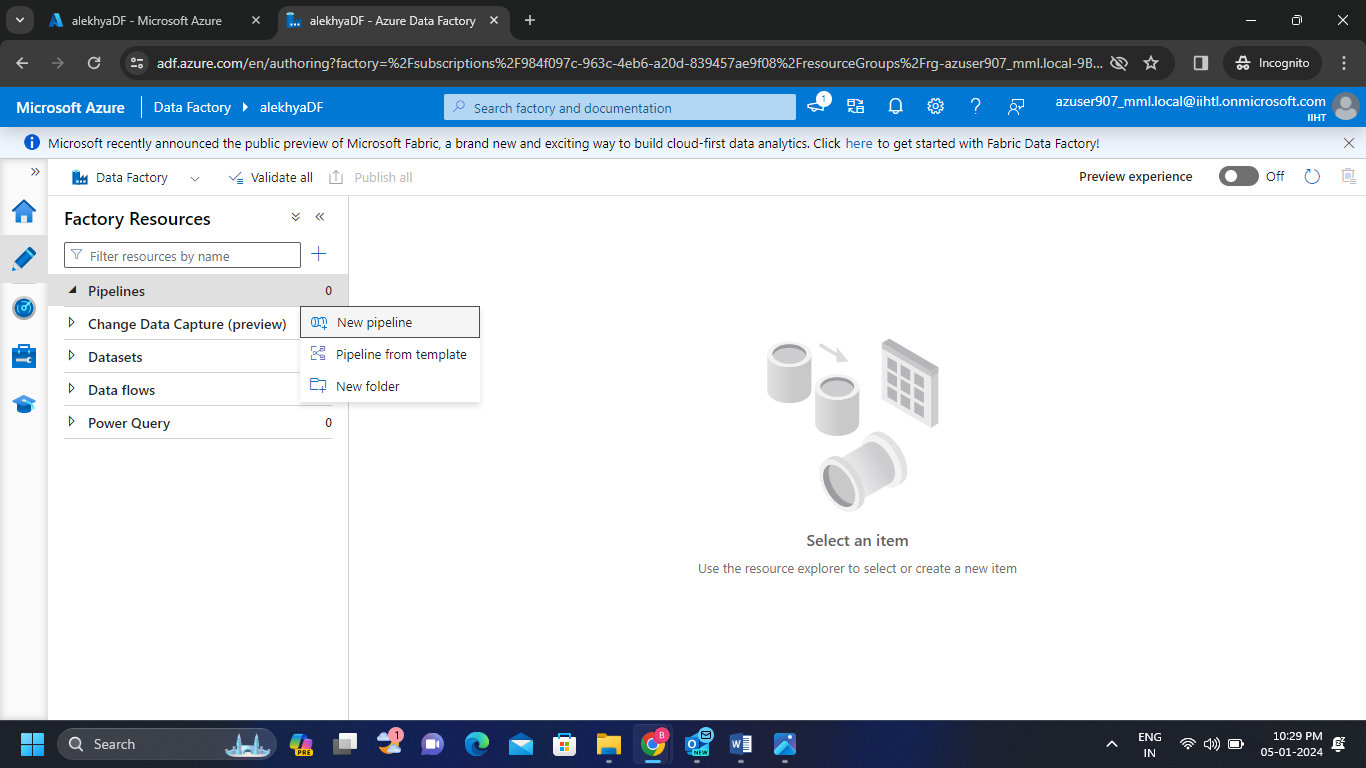
**Step-2:**  Launch the Azure Data Factory Studio then we get the following page.

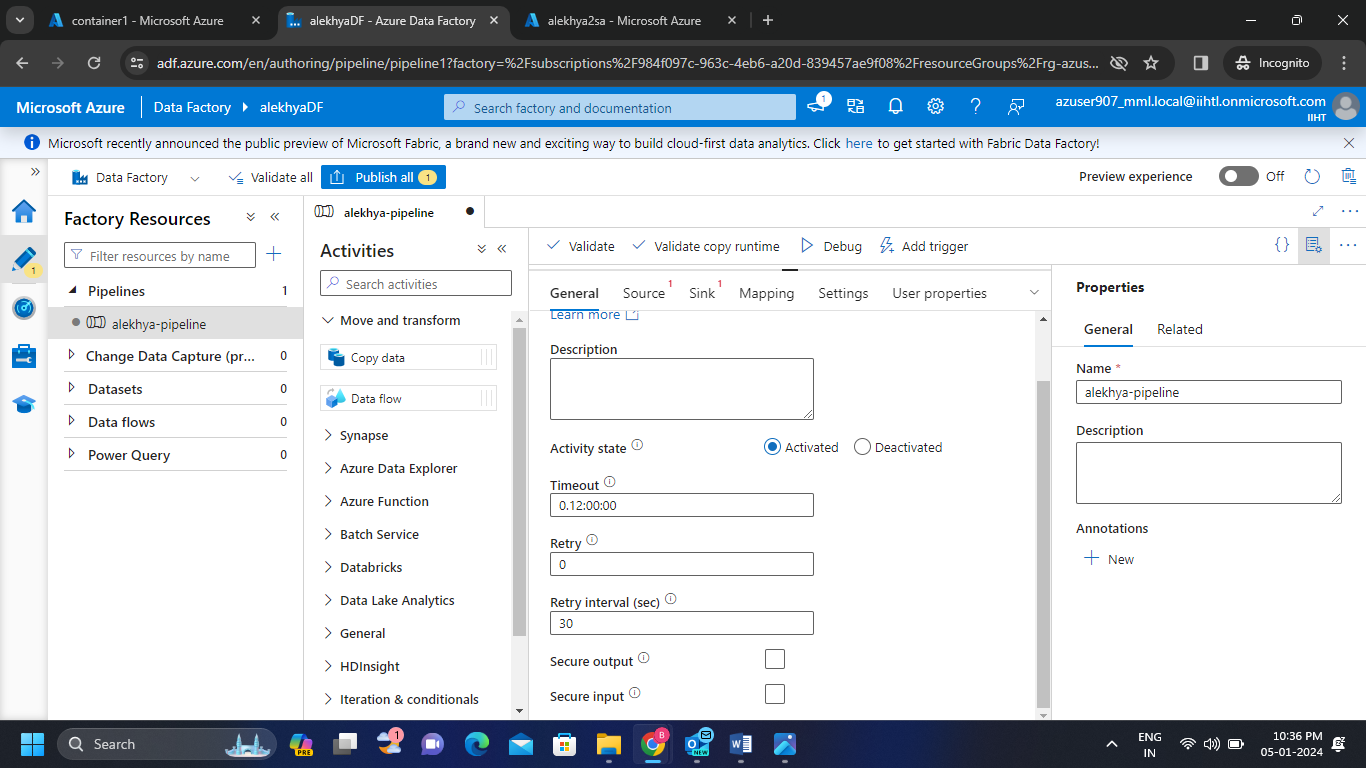
****

**Step-3:** Create 2 storage accounts and the create 2 new containers in each storage account. Now upload a csv file in one container and keep the another container empty.

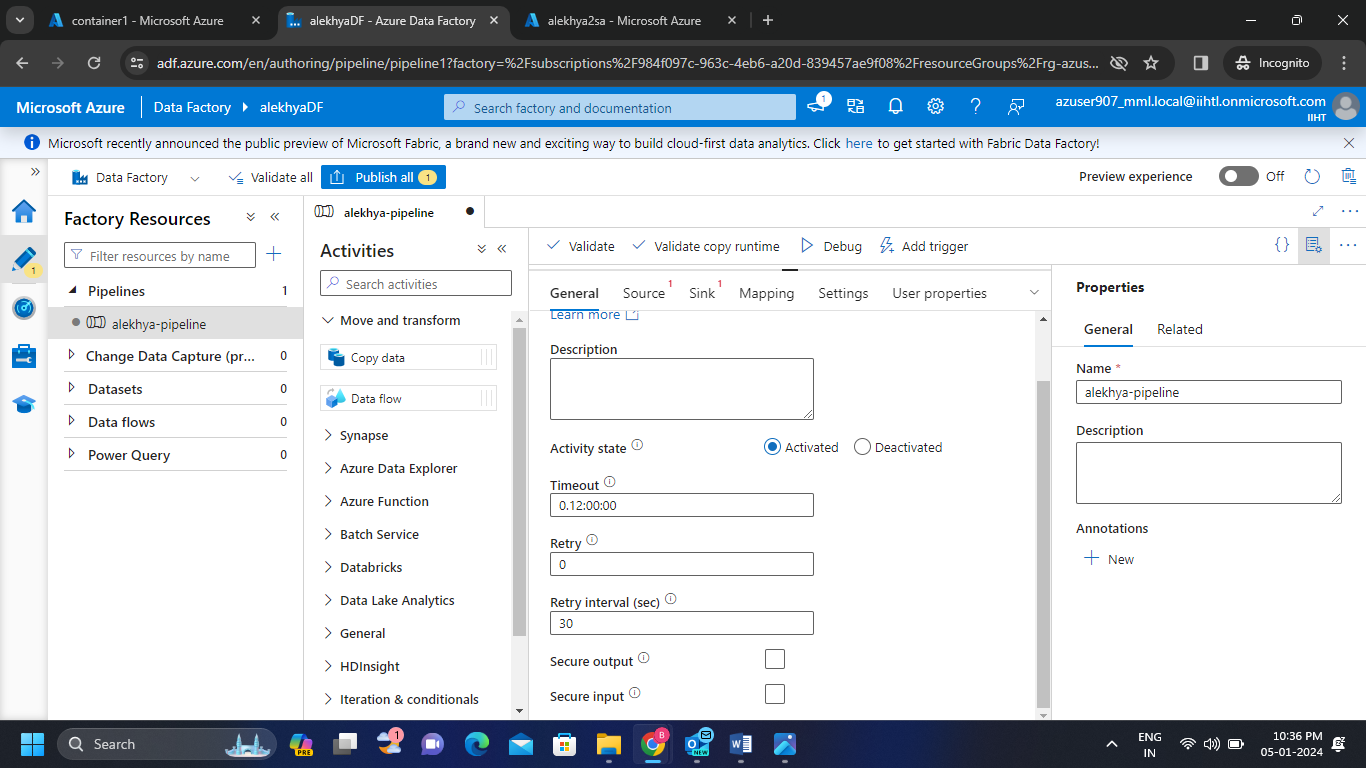
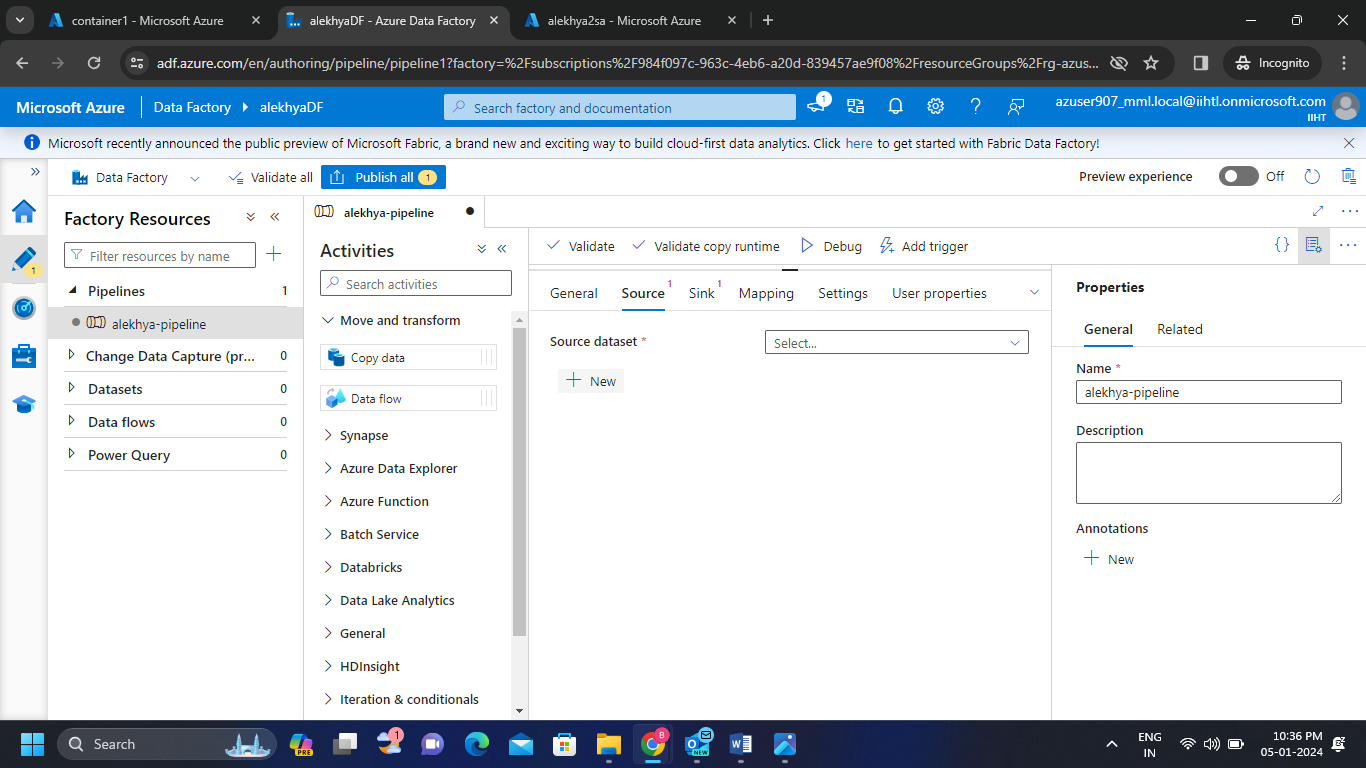
****

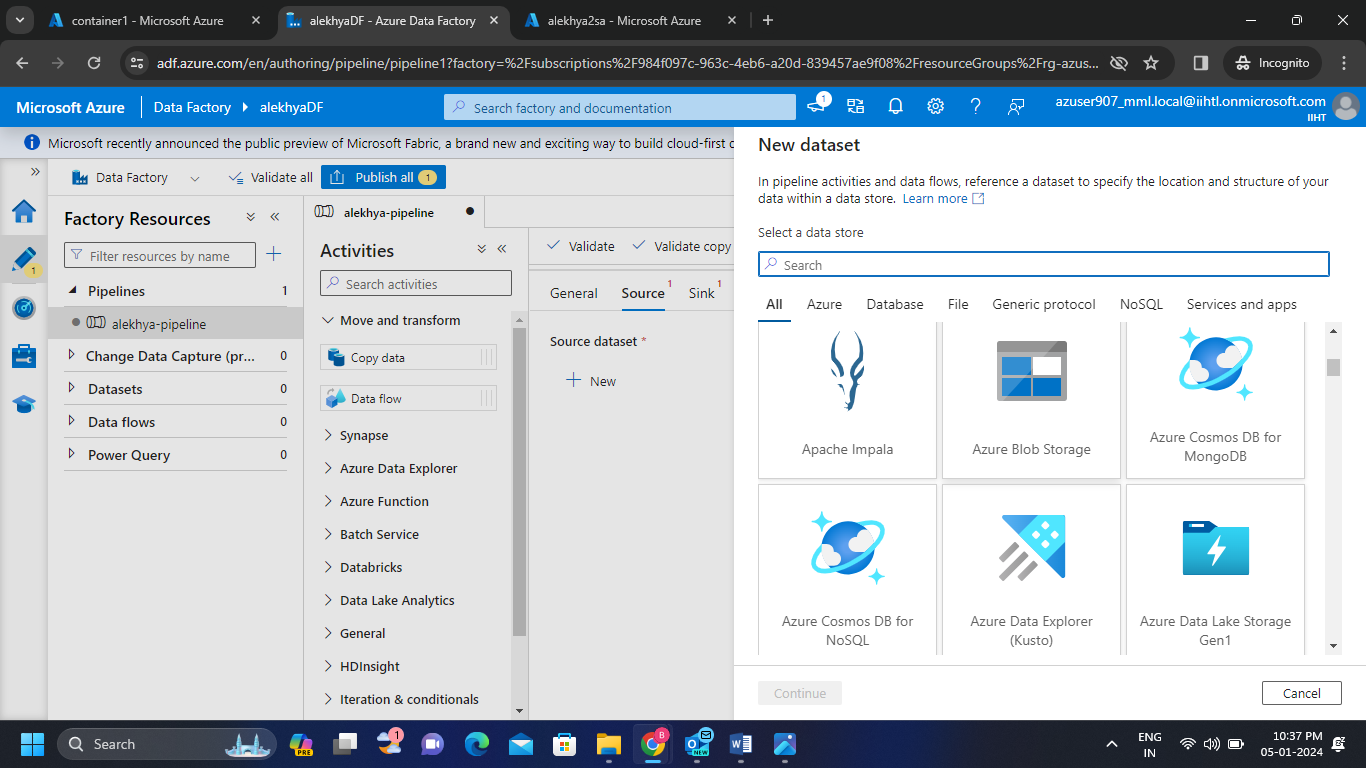
****

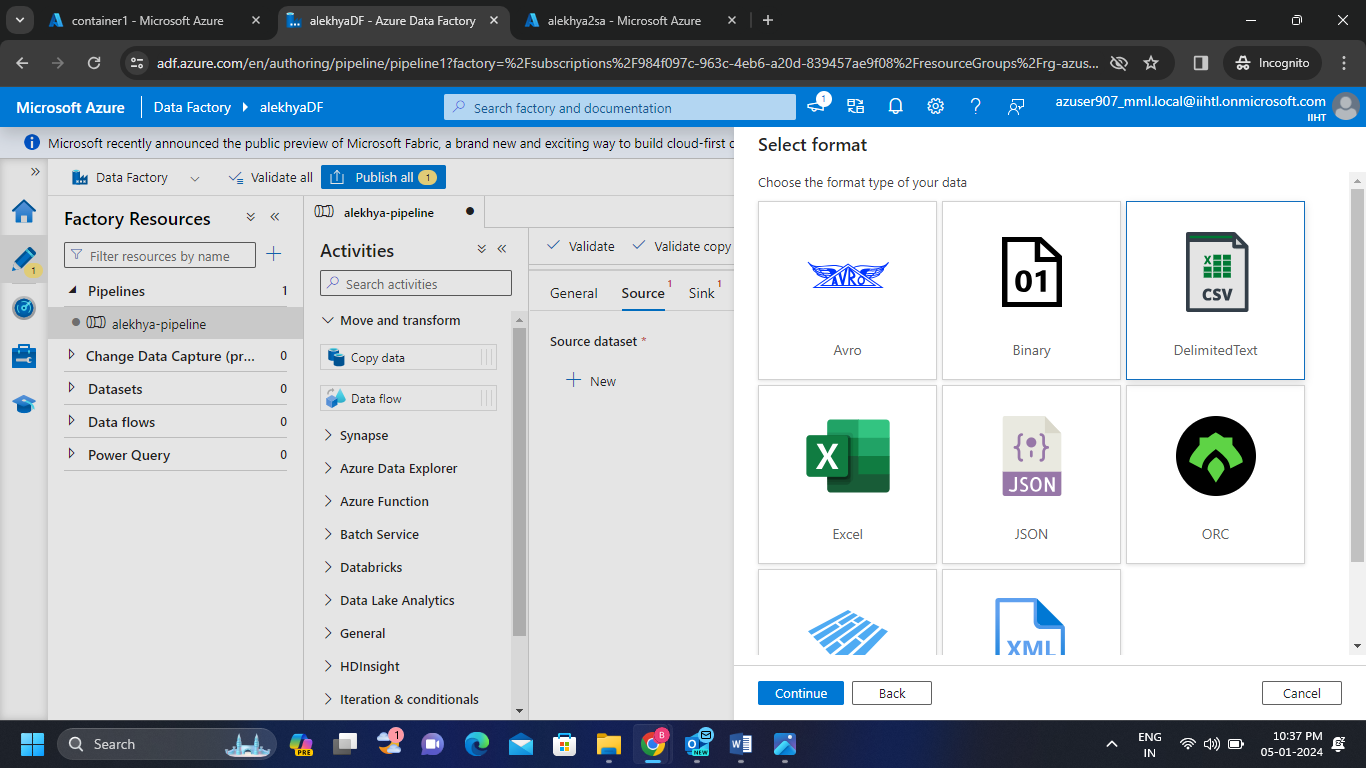
**Step-4:** Now in the data factory create a new pipeline and click on copy data.



**Step-5:**

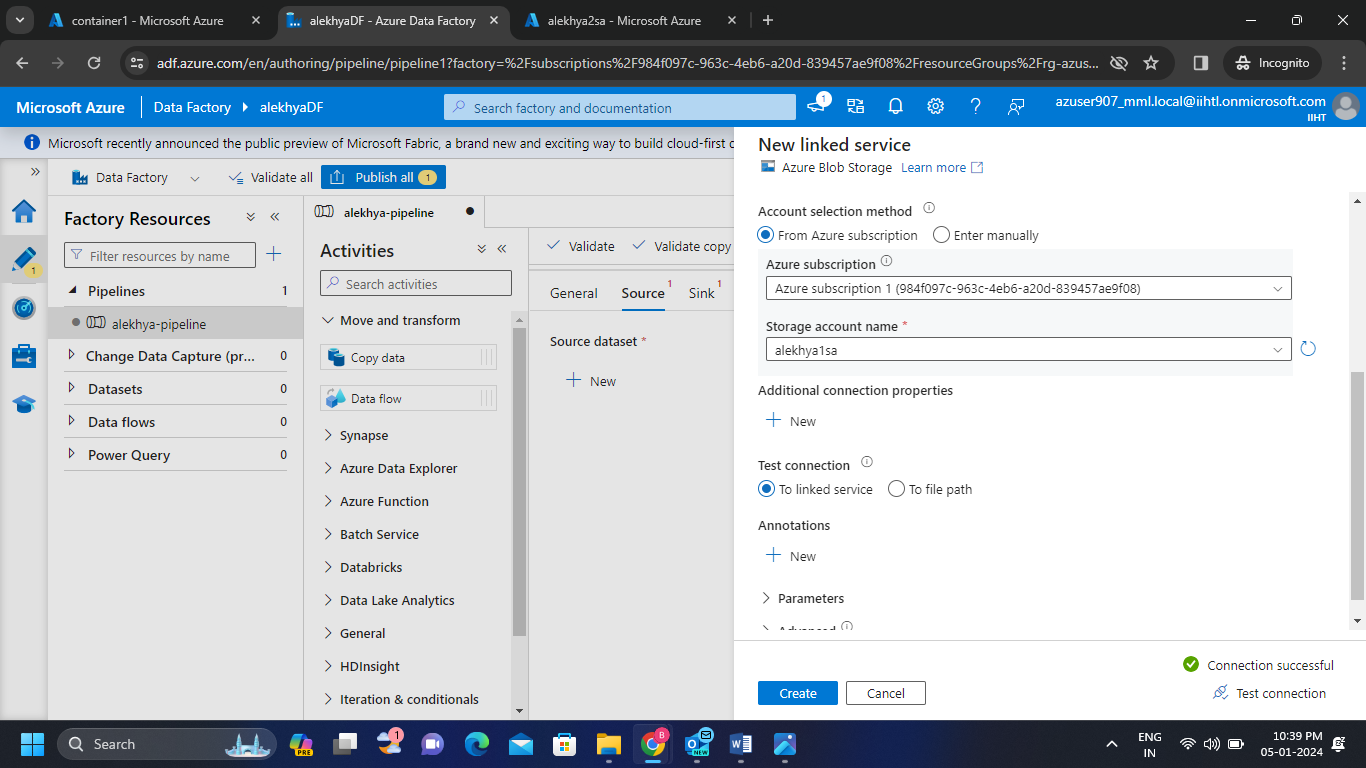
* Fill the necessary data in general block
* In the source block create new dataset(dataset-1).
* As we uploaded the csv file in container click on Azure Blob Storage and select csv format.



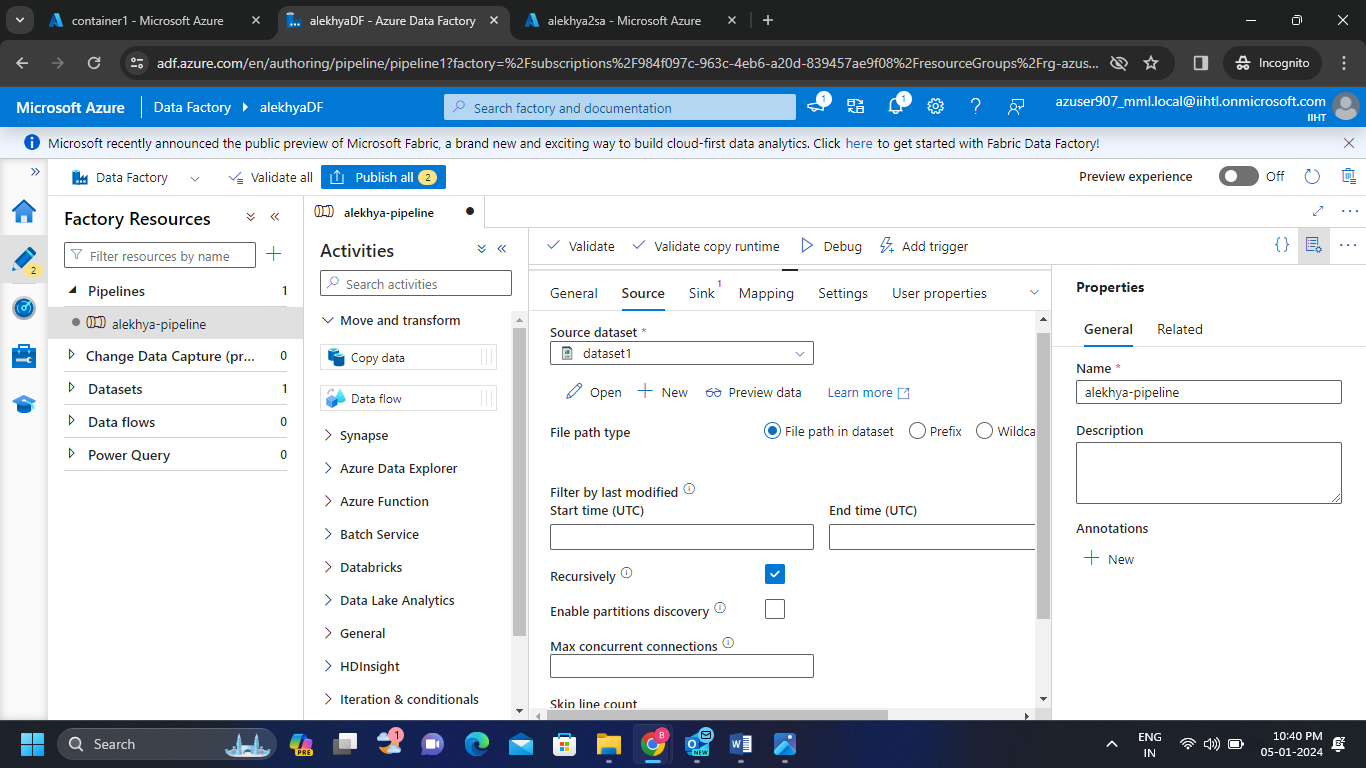


* Now click on new link service then link the first storage account where we uploaded the csv file and also give the path of the container 1 then click on test connection and create the dataset.





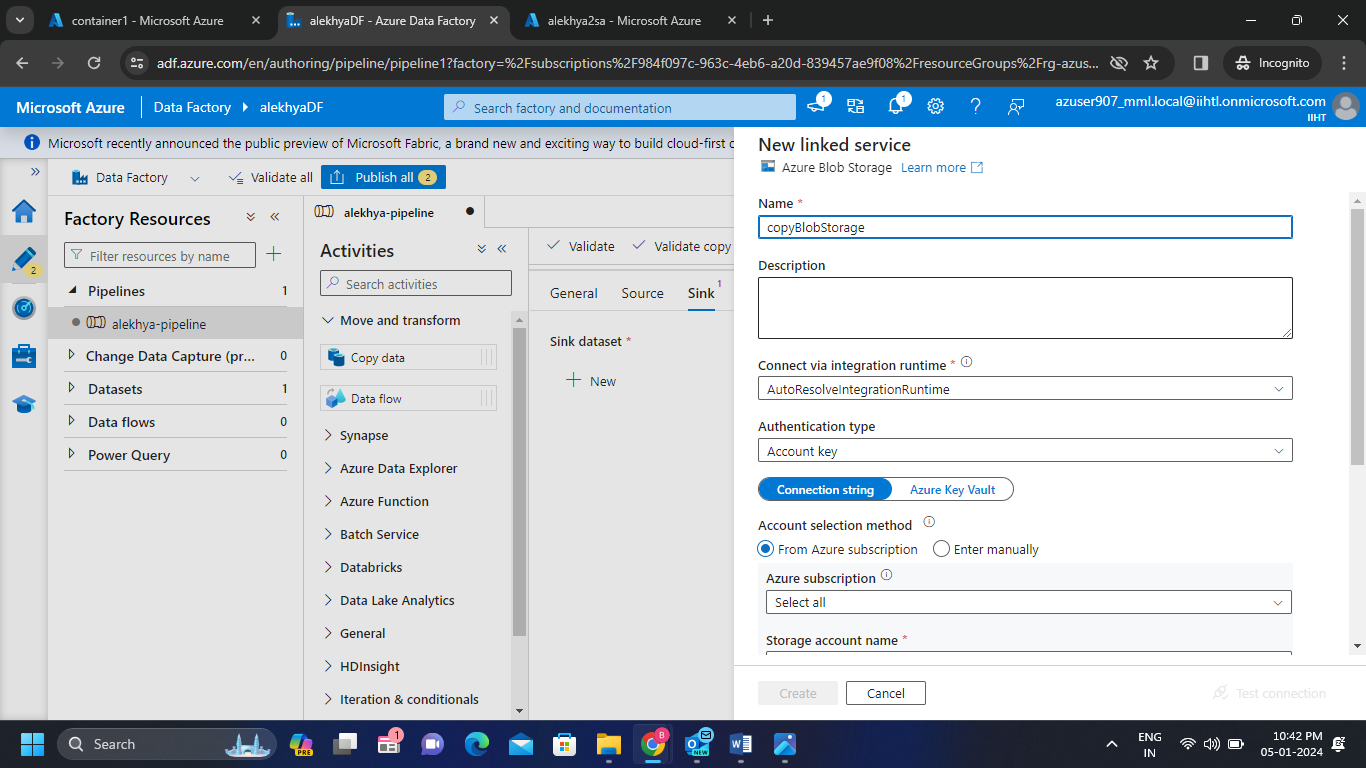


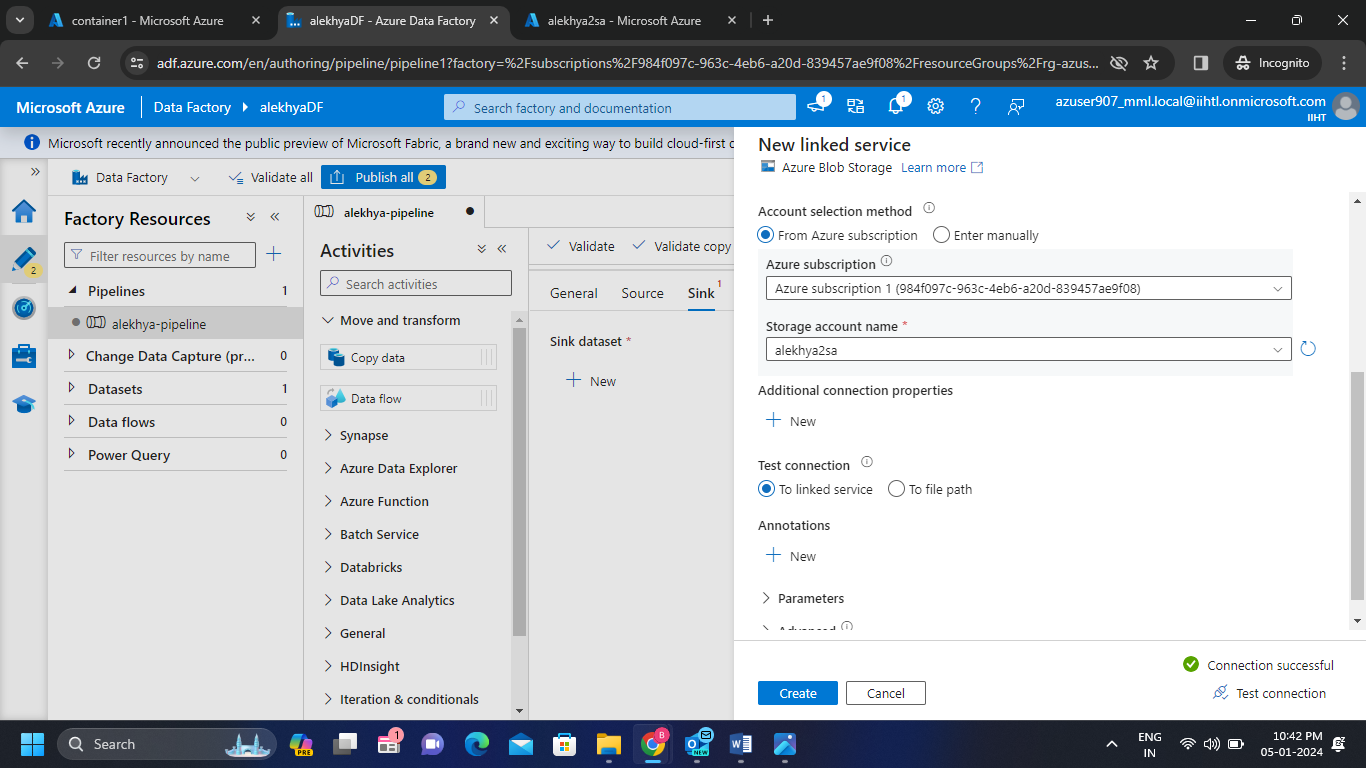


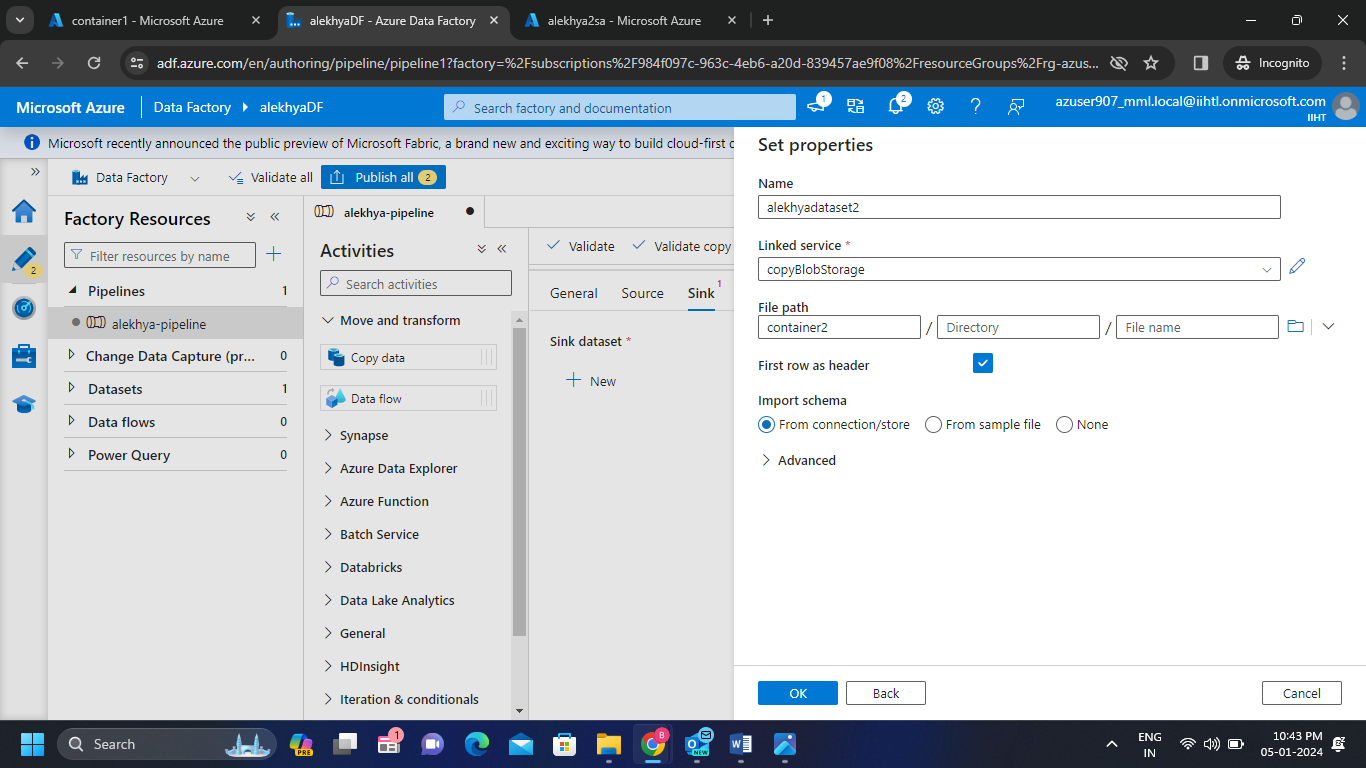
**Step-6:** In the sink block create a new dataset(dataset-2) to get the copied file which we uploaded in container1.

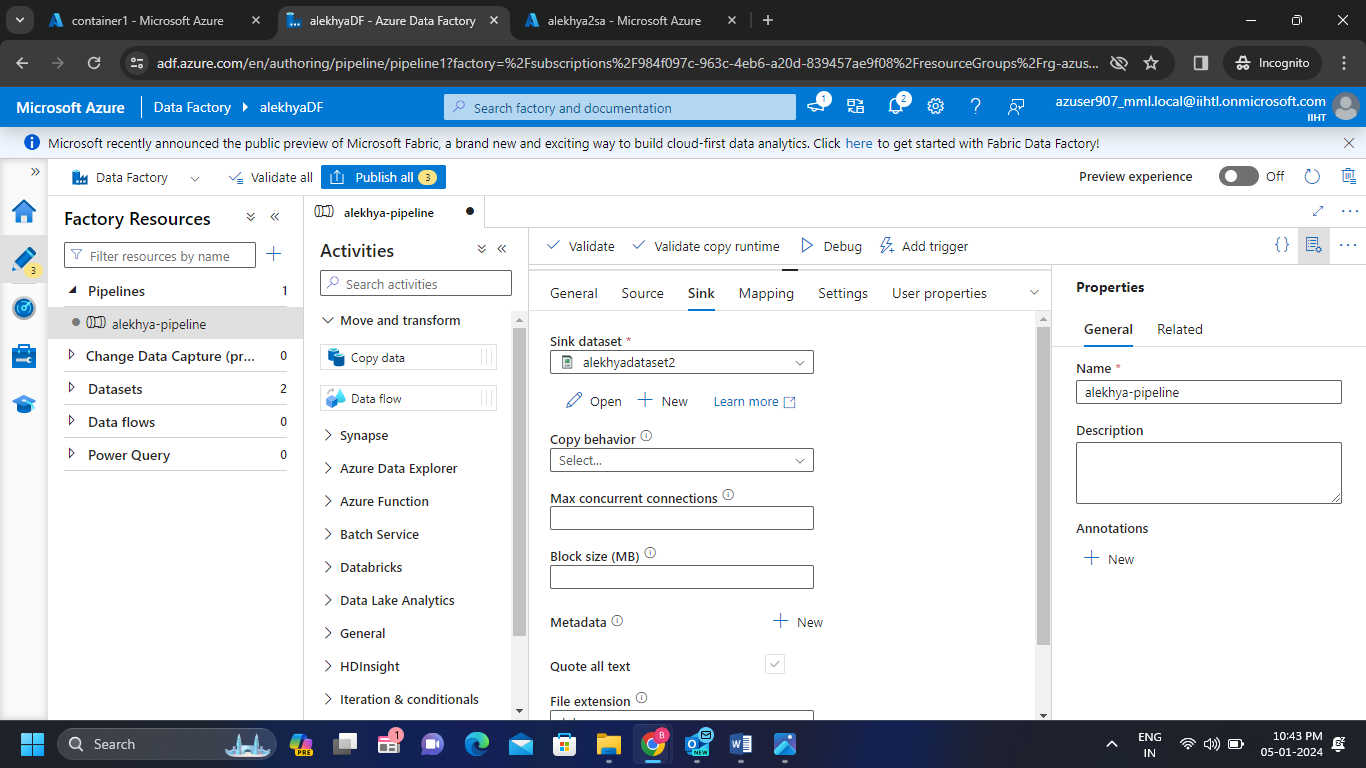
****

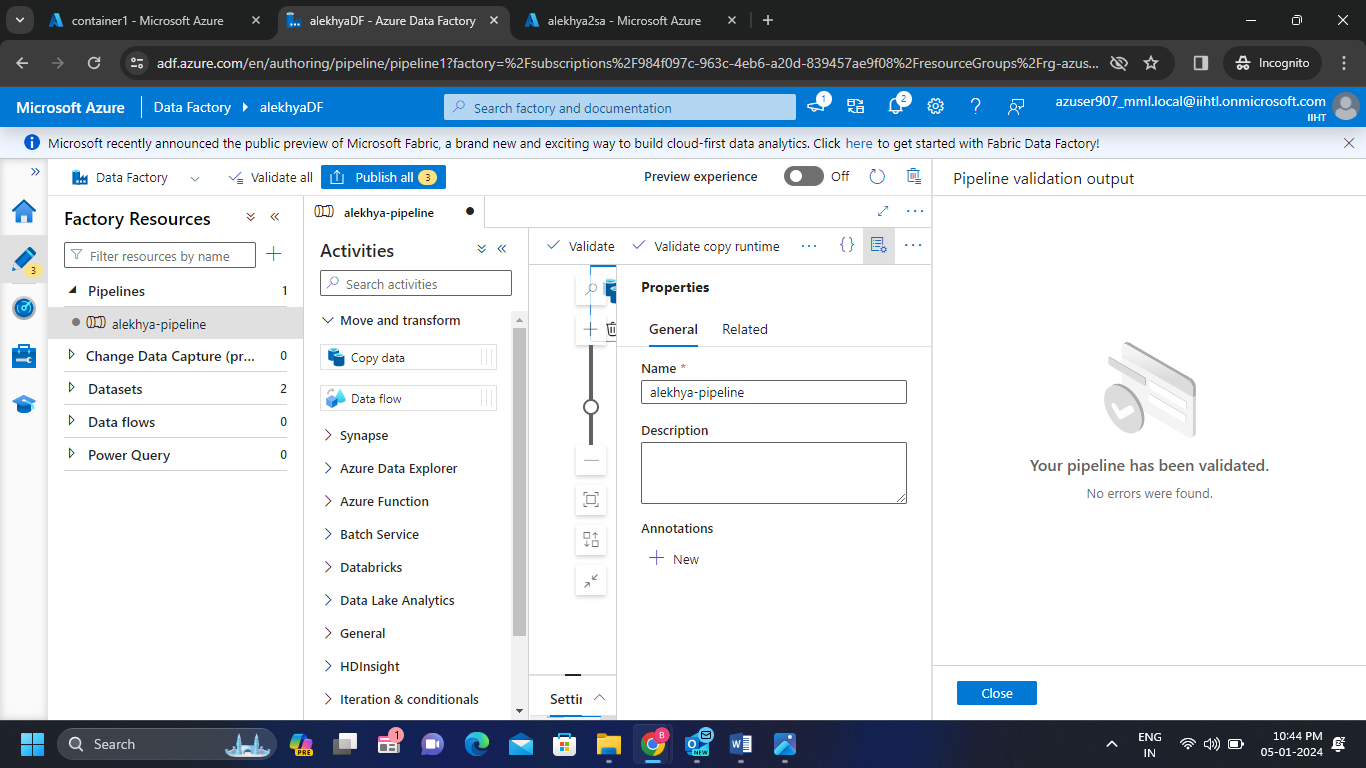
* Now click on new link service then link the storage account 2 where the copy of csv file will be pasted and also give the path of the container 2 which consists of no files then click on test connection and create the dataset.

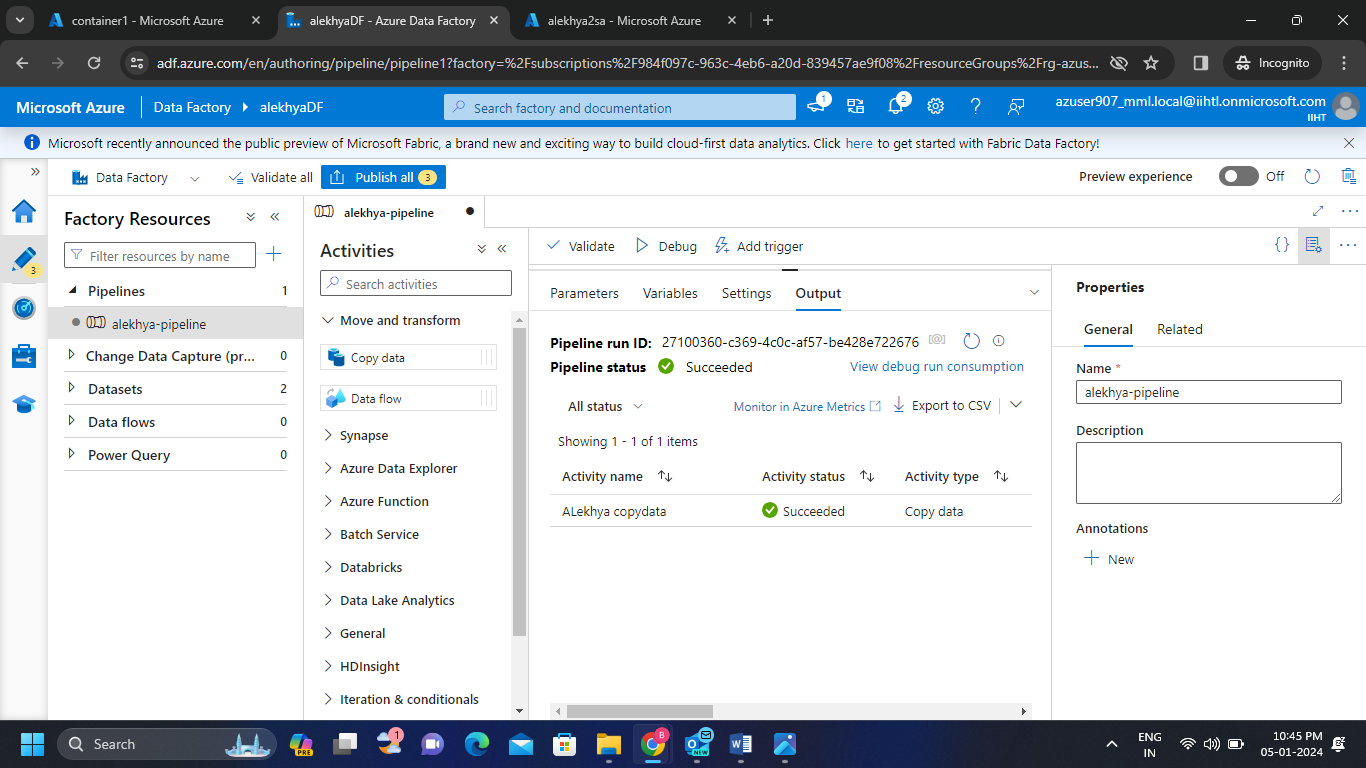
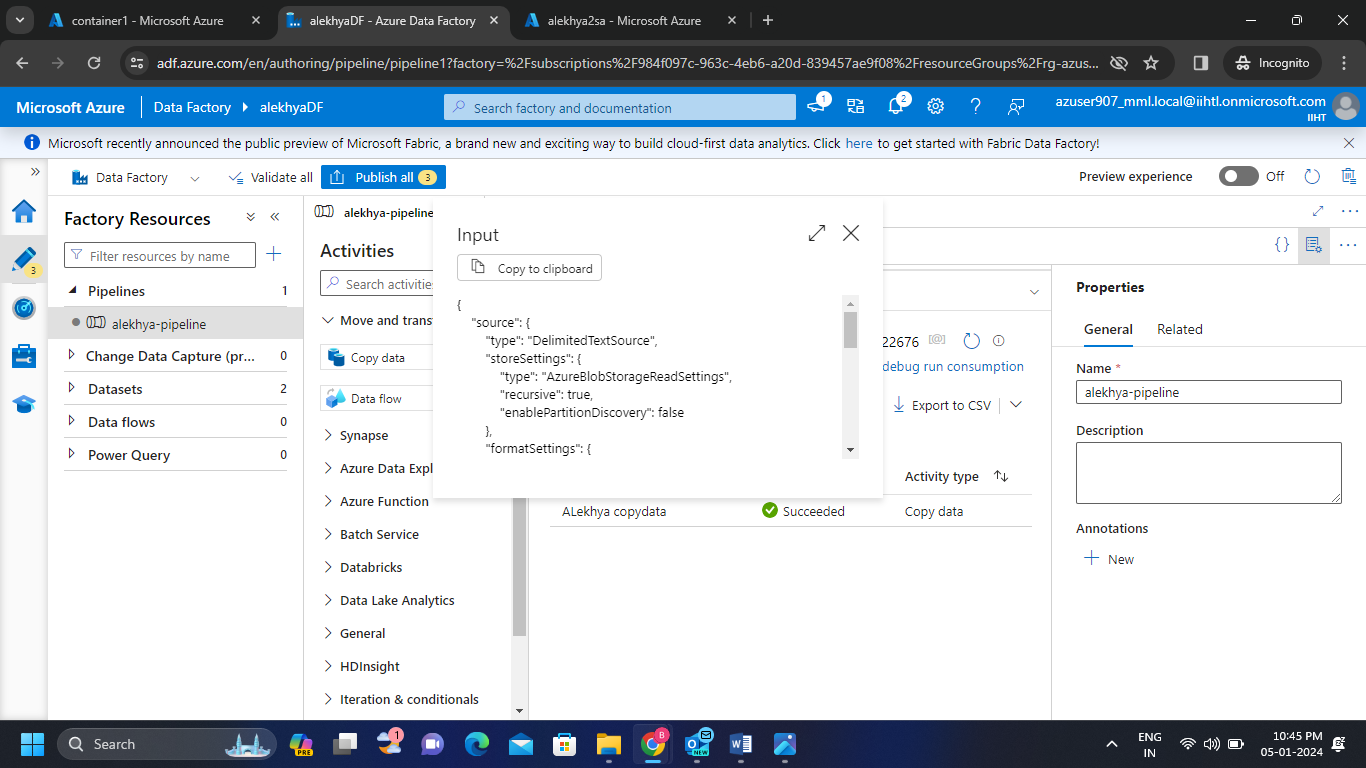
****

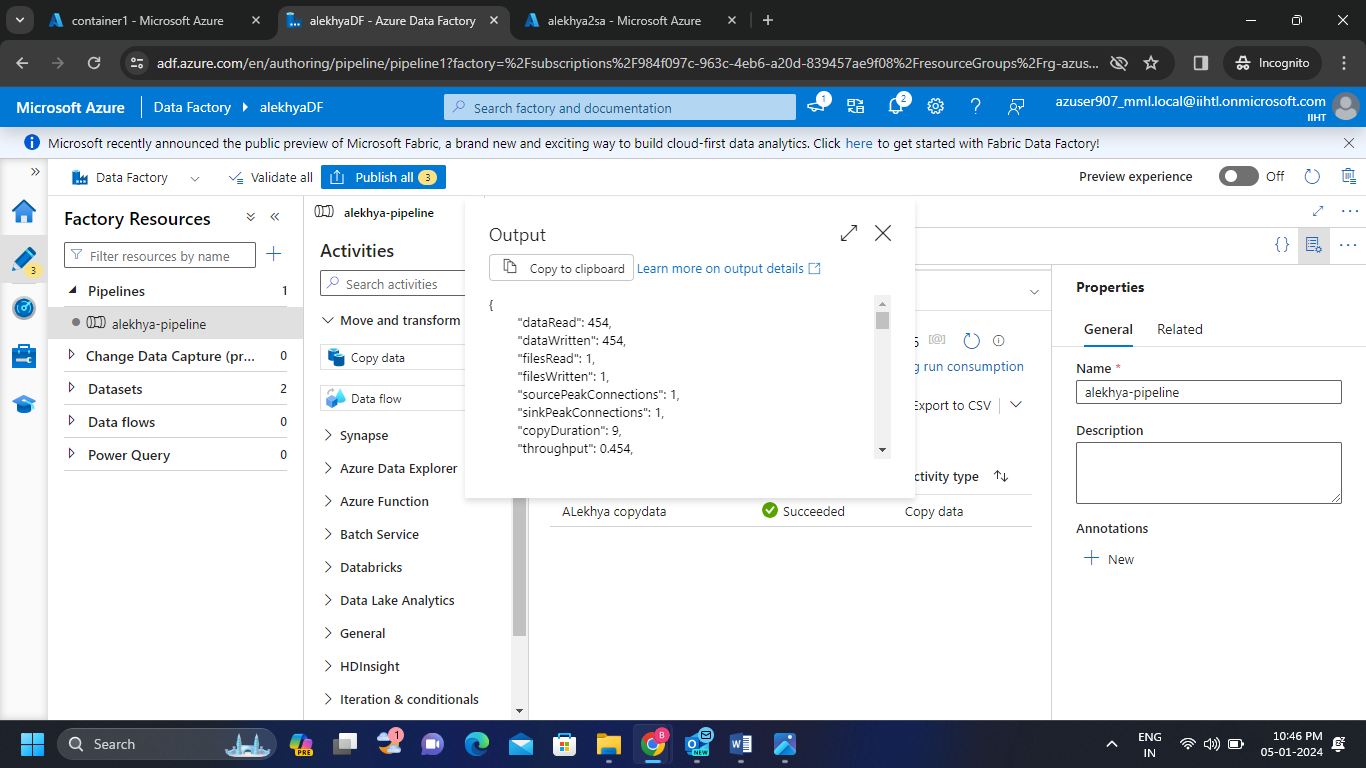
****

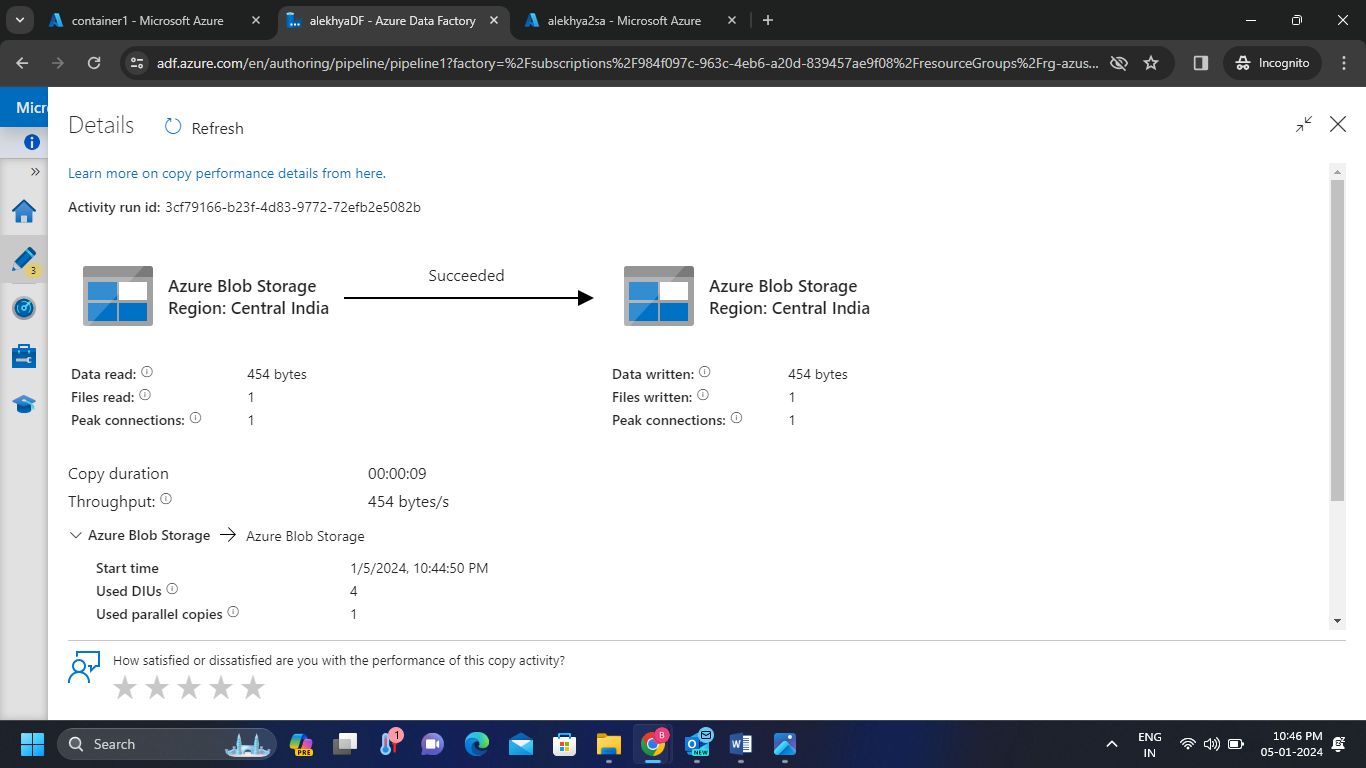
****

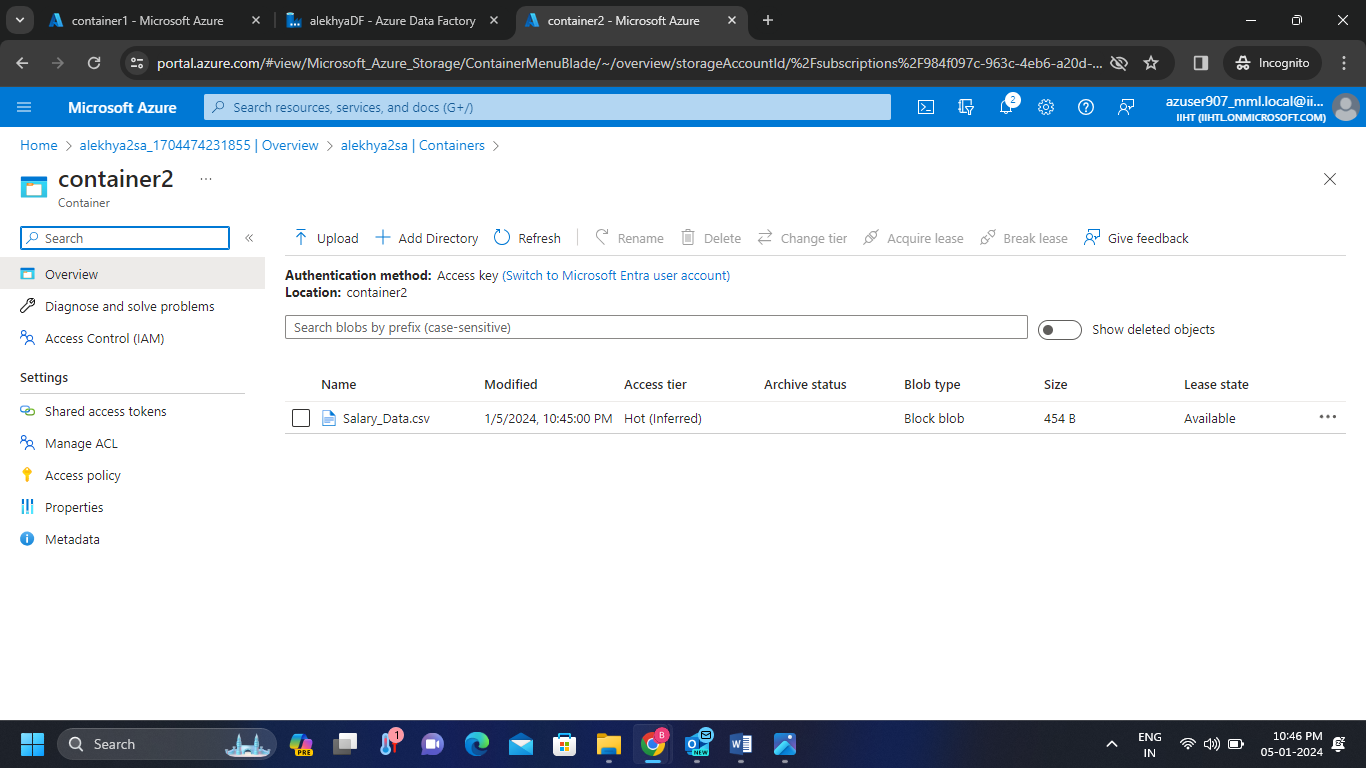
****

**Step-7:** Now click on validate and we can see the pipeline validation output.

* Now debug the copied file and we can see the activity status of the copied data.
* We can see the input code and output code of the copied data.



* We also can see the details the operations we have done above.

**Step-8 :** Hence we successfully copied the csv file from one storage account to another storage account using Azure Data Factory Pipelining.