

Daily Data Aggregation

~By

Alekhya Krishna Balivada

alekhya0830@gmail.com

Table of Contents

Sno	Topic	Pg.No.
1	Project Statement	1
2	Project Overview	1
3	Project Requirements	1
4	Architecture	2
5	Execution Overview	2
6	Project implementation	
	6.1 Azure account	3
	6.2 Storage Account	3
	6.3 Uploading CSV file in storage account container	7
	6.4 Azure Databricks	8
	6.5 Creating Databricks Cluster and Notebook	10
	6.6 Mounting ADLS with Databricks	12
	6.7 Azure Data Factory	14
	6.8 Integrating ADF with Azure Databricks	17
	6.9 Final Output	24
7	Copying the Aggregate data using Data Flow Activity	26
	Conclusion	37

1. Project Statement

Implement a daily data aggregation pipeline using Azure Data Factory to move raw data and Azure Databricks to aggregate and summarize the data based on daily intervals.

2. Project Overview

The project aims to use Azure Data Factory (ADF) and Azure Databricks to construct a daily data aggregation pipeline. With the help of this pipeline, raw data from source systems will be moved to a location where Azure Databricks will be used for daily data aggregation and summarization.

3. Project Requirements

- a. Azure Subscription
- b. Data Sources
- c. Azure Storage account
- d. Azure Databricks
- e. Azure Data Factory

3.1 Azure Subscription

- We need Azure Subscription plan to perform this project

3.2 Data Sources

- I have taken a CSV file as ‘project.csv’.
- I performed various operations on this ‘project.csv’ file.

3.3 Azure Storage account

- An Azure Storage Account is a Microsoft Azure service that provides highly scalable and durable cloud storage for various types of data. It serves as a central repository for storing and managing different types of data, including blobs, files, tables, and queues.
- The ‘project.csv’ file is uploaded inside the storage account container.

3.4 Azure Databricks

- A Databricks workspace is established to leverage Apache Spark for data processing.
- A Databricks cluster is configured with the necessary libraries and settings.
- A Databricks notebook is developed to perform data aggregation and summarization.
- The notebook is parameterized to accept a date range, enabling daily processing.

3.5 Azure Data Factory

- ADF is utilized for data orchestration and movement.

- Linked services are configured to establish connections with raw data sources and destination.
- Datasets are defined to represent raw data and the destination for aggregated data.
- An ADF pipeline is created to copy raw data from source to destination using Data Flow activity.

4. Architecture



5. Execution overview

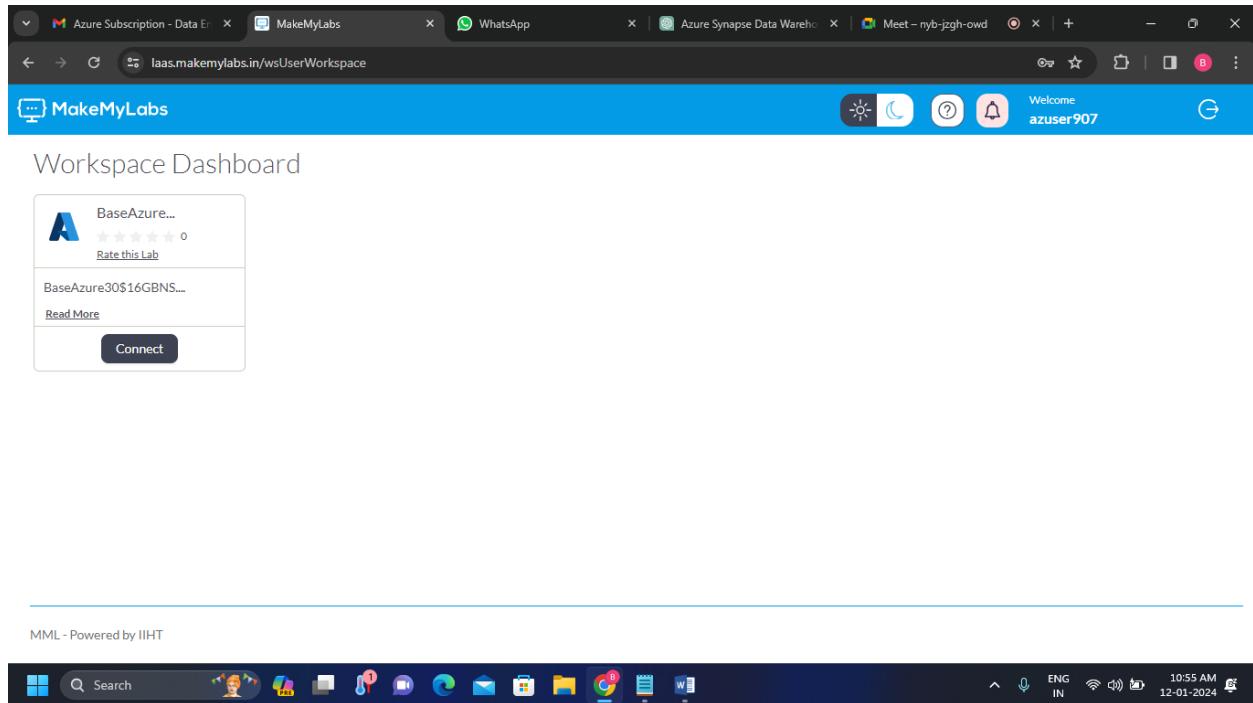
- a. Login to the Azure Subscription account.
- b. Create a storage account and upload a file inside a storage account container.
- c. Create an Azure Databricks workspace.
- d. Create a cluster and notebook in databricks workspace.
- e. Connect ADLS with ADF.
- f. Create Azure data factory resource.
- g. Create pipeline for databricks in ADF.
- h. Debug to get the final output.

6. Project implementation

- Project implementation involves the step by step procedure along with screenshots that we performed in the Azure account.

6.1 Azure account

- Log in to the Azure account and connect to the Workspace Dashboard.



6.2 Storage account

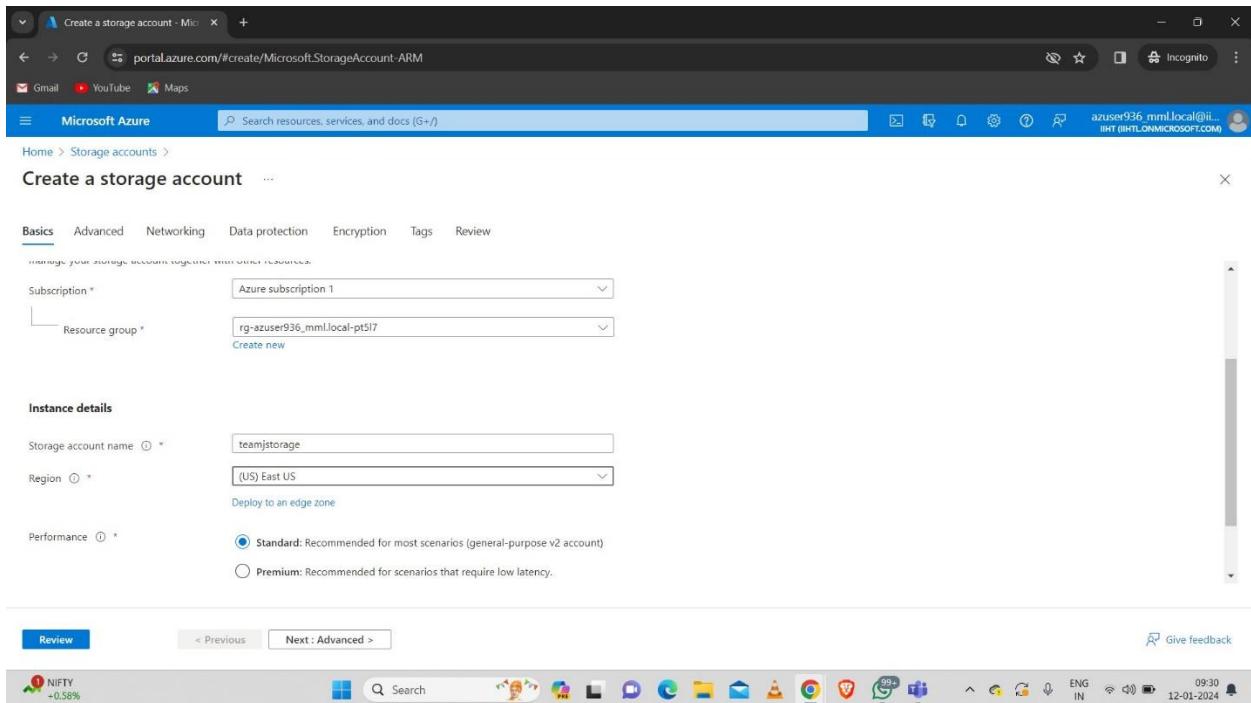
Step-1: Search for Storage account.

The screenshot shows the Microsoft Azure portal homepage. At the top, there's a search bar with the placeholder "Search resources, services, and docs (G+)" and a "Clear all" button. Below the search bar is a "Search history" section with items like "synapse workspace", "azure synapse", "devops", "storage", and "databr". To the right of the search bar are several icons: "Quickstart Center" (with a lightning bolt icon), "More services" (with a right-pointing arrow icon), and a user profile icon. The main content area has sections for "Recent services" (Storage accounts, Azure Synapse Analytics, Resource groups, Azure DevOps organizations, Azure Databricks, SQL databases, Virtual machines) and "Recent resources" (rg-azuser936_mml.local-pt517). A "Last Viewed" section shows "16 hours ago". At the bottom, there are navigation links for "Subscriptions", "Resource groups", "All resources", and "Dashboard". A "Tools" section at the bottom includes links for Microsoft Learn, Azure Monitor, Microsoft Defender for Cloud, and Cost Management. The taskbar at the bottom of the screen shows various pinned icons and the date/time as 12-01-2024.

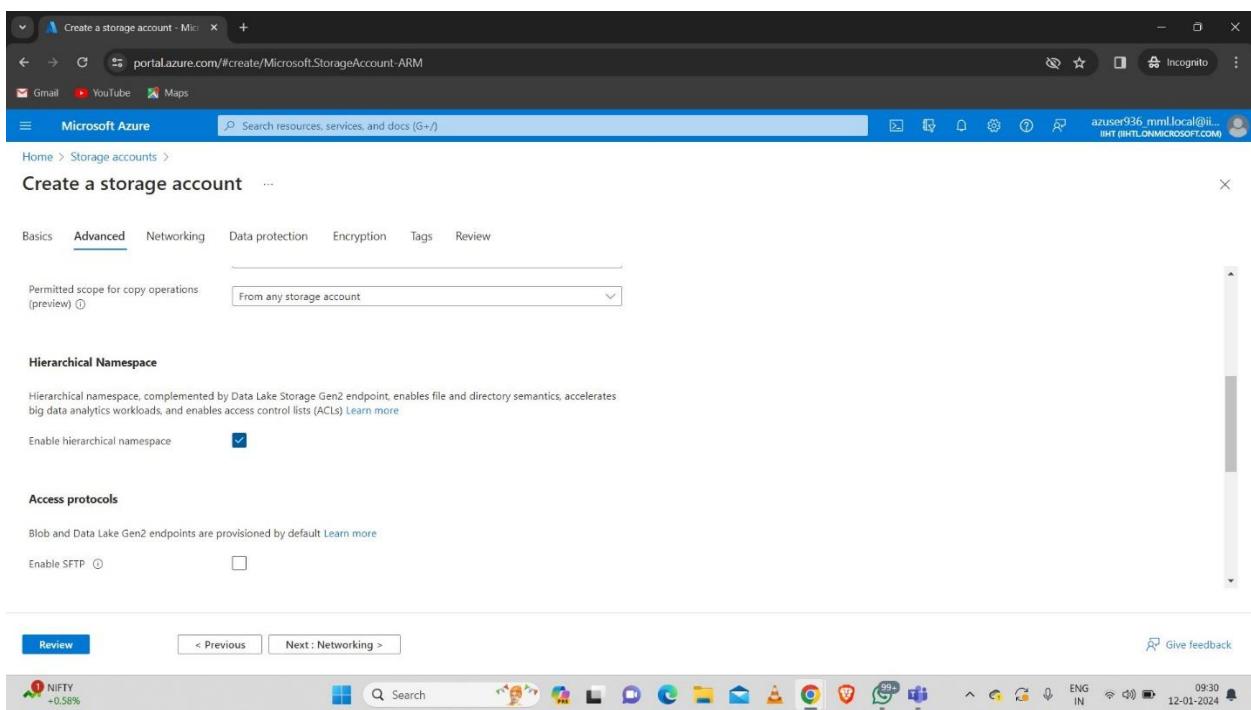
Step-2: Click on create to create the storage account.

The screenshot shows the "Storage accounts" page in the Microsoft Azure portal. The URL in the address bar is "https://portal.azure.com/#view/HubsExtension/BrowseResourceBlade/resourceType/Microsoft.Storage%2FStorageAccounts". The page title is "Storage accounts". There are buttons for "+ Create", "Restore", "Manage view", "Refresh", "Export to CSV", "Open query", "Assign tags", and "Delete". Below these are filter options: "Filter for any field...", "Subscription equals all", "Resource group equals all", "Location equals all", and "Add filter". A "No grouping" dropdown and a "List view" dropdown are also present. The main table lists 58 storage accounts, each with a checkbox, name, type, kind, resource group, location, and subscription information. The columns are: Name, Type, Kind, Resource group, Location, Subscription. The table includes rows such as "834mlworkspace934210684", "azuser8530044069014", "cs21003200311c2b722", "cs21003200311c2b730", "cs71003200308a34627", "cs71003200311c2b72c", "cs71003200311c2b72e", "csg1003200308a6b52b", "csg1003200308a6ba0", "csg10032003107a77c0", and "csg10032003114kh248". At the bottom, there are navigation buttons for "< Previous", "Page 1 of 1", and "Next >". The taskbar at the bottom of the screen shows various pinned icons and the date/time as 12-01-2024.

Step-3: Fill the necessary data. Here I have created a storage account with name “teamjstorage”.



The screenshot shows the Microsoft Azure portal with the URL portal.azure.com/#create/Microsoft.StorageAccount-ARM. The page title is "Create a storage account". The "Basics" tab is selected. The account name is "teamjstorage", the region is "(US) East US", and the performance level is "Standard". The browser taskbar at the bottom shows various pinned icons.



The screenshot shows the Microsoft Azure portal with the URL portal.azure.com/#create/Microsoft.StorageAccount-ARM. The page title is "Create a storage account". The "Advanced" tab is selected. It includes sections for "Permitted scope for copy operations" (set to "From any storage account"), "Hierarchical Namespace" (with a note about Data Lake Storage Gen2 endpoint), and "Access protocols" (blob and Data Lake Gen2 endpoints). The browser taskbar at the bottom shows various pinned icons.

The screenshot shows the Azure portal interface for creating a storage account. The 'Review' tab is active. The account details are as follows:

Setting	Value
Subscription	Azure subscription 1
Resource Group	rg-azuser936_mmlocal-pt517
Location	eastus
Storage account name	teamjstorage
Deployment model	Resource manager
Performance	Standard
Replication	Read-access geo-redundant storage (RA-GRS)

Below the main form, there are buttons for 'Create' (highlighted in blue), '< Previous' and 'Next >', and a link to 'Download a template for automation'. The status bar at the bottom shows the weather (23°C, mostly sunny) and system information (Windows 10, ENG IN, 09:31, 12-01-2024).

➤ Here we can see that the storage account is successfully created.

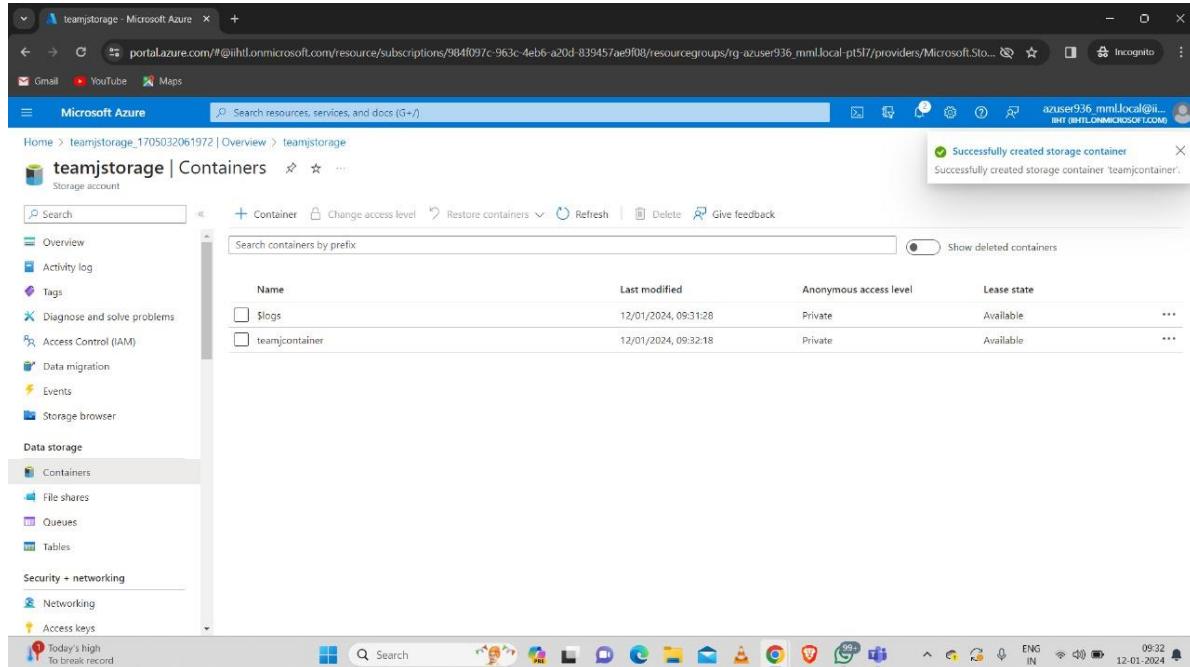
The screenshot shows the Azure portal deployment details for 'teamjstorage_1705032061972'. The deployment status is 'Deployment succeeded'. Deployment details include:

- Deployment name: teamjstorage_1705032061972
- Subscription: Azure subscription 1
- Correlation ID: e47e8246-b3e1-4757-9185-70b423ec35a3
- Resource group: rg-azuser936_mmlocal-pt517

The deployment is complete, as indicated by the green checkmark icon. A 'Go to resource' button is available. The status bar at the bottom shows the weather (23°C, mostly sunny) and system information (Windows 10, ENG IN, 09:31, 12-01-2024).

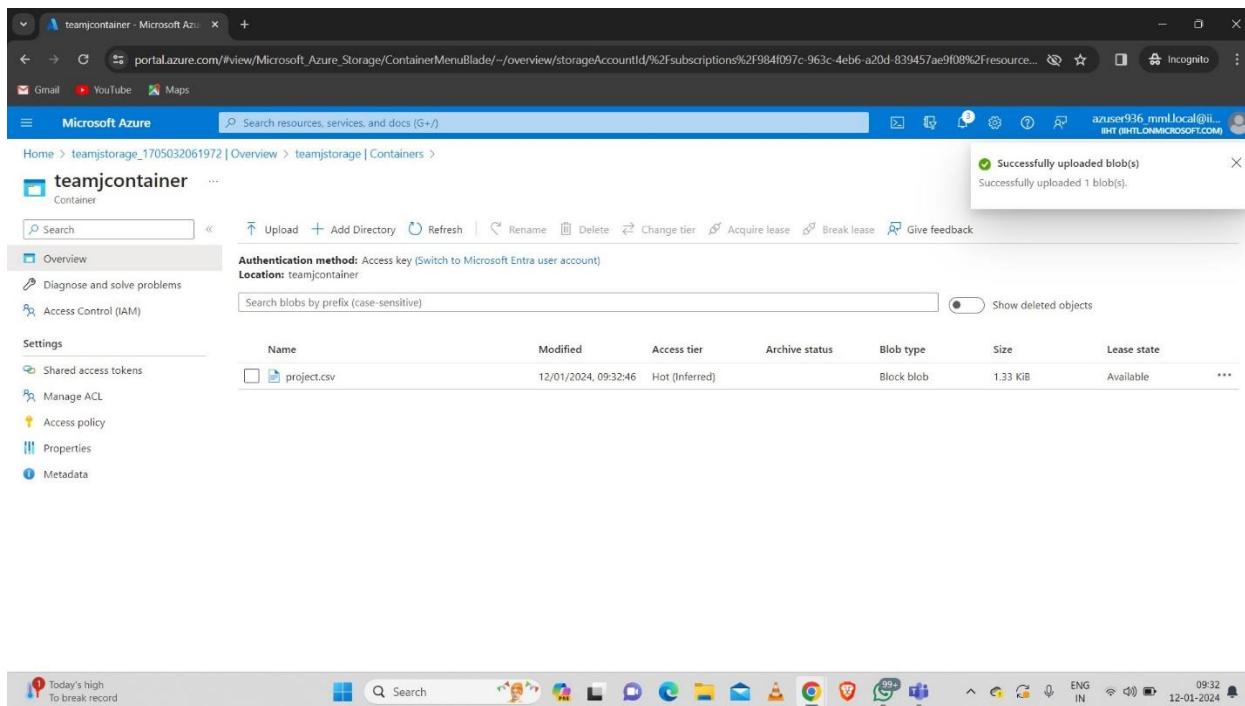
6.3 Uploading CSV file in storage account container

Step-1: Open the storage account and create a container. Here we created a container with name “teamjcontainer”.



The screenshot shows the Microsoft Azure Storage Container list for the 'teamjstorage' account. A success message at the top right states 'Successfully created storage container' and 'Successfully created storage container teamjcontainer'. The table lists two containers: '\$logs' and 'teamjcontainer'. The 'teamjcontainer' row shows 'Last modified' as 12/01/2024, 09:32:18, 'Anonymous access level' as Private, and 'Lease state' as Available. The left sidebar shows navigation options like Overview, Activity log, Tags, and Data storage (Containers, File shares, Queues, Tables). The bottom taskbar includes icons for various Windows applications.

Step-2: Open the container and upload the csv file inside the container. Here I uploaded “project.csv” file inside the container.



The screenshot shows the Microsoft Azure Storage Container blob list for the 'teamjcontainer' account. A success message at the top right states 'Successfully uploaded blob(s)' and 'Successfully uploaded 1 blob(s)'. The table lists one blob named 'project.csv' with details: Modified: 12/01/2024, 09:32:46, Access tier: Hot (Inferred), Archive status: Not yet archived, Blob type: Block blob, Size: 1.33 KiB, Lease state: Available. The left sidebar shows navigation options like Overview, Diagnose and solve problems, and Settings (Shared access tokens, Manage ACL, Access policy, Properties, Metadata). The bottom taskbar includes icons for various Windows applications.

6.4 Azure Databricks

Step-1: Search for Azure Databricks

The screenshot shows the Microsoft Azure portal homepage. In the top navigation bar, the URL is `portal.azure.com/#home`. The search bar at the top right contains the query `Azure Databricks`. Below the search bar, there's a section titled "Azure services" with icons for "Create a resource", "Storage accounts", "Azure Synapse Analytics", "Resource groups", and "Azure Databricks". The "Azure Databricks" icon is highlighted with a white border. To the right of this, there's a detailed card for "Azure Databricks" with a red cube icon, a "Description" section, and a "Free training from Microsoft" section. Below the services section, there's a "Resources" section showing recent resources like "teamstorage" and "rg-azuser936_mml.local-pt517", and a "Navigate" section with links for "Subscriptions", "Resource groups", and "All resources". At the bottom of the page, there's a "Tools" bar with various icons and a status bar showing the date and time.

Step-2: Click on create to create the databricks workspace.

The screenshot shows the Microsoft Azure portal with the URL `portal.azure.com/#view/HubsExtension/BrowseResource/resourceType/Microsoft.Databricks%2Fworkspaces`. The search bar at the top right still has the query `Azure Databricks`. The main content area displays a table of existing Databricks workspaces. The columns are "Name", "Type", "Resource group", "Location", and "Subscription". The table shows six entries:

Name	Type	Resource group	Location	Subscription
AzureDB-921	Azure Databricks Service	rg-azuser921_mml.local-yVpZ	Central India	Azure subscription 1
FINAL_PROJECT	Azure Databricks Service	rg-azuser924_mml.local-Khila	Central India	Azure subscription 1
Project_teamk	Azure Databricks Service	rg-azuser918_mml.local-4736r	Central India	Azure subscription 1
Project_workspace	Azure Databricks Service	rg-azuser909_mml.local-EQYSl	Central India	Azure subscription 1
sarfraz934	Azure Databricks Service	Az500test	East US	Azure subscription 1
TEAMG	Azure Databricks Service	rg-azuser926_mml.local-M0g0t	Central India	Azure subscription 1

At the bottom of the page, there are navigation links for "Previous", "Page 1 of 1", and "Next >". The status bar at the bottom shows the date and time as 12-01-2024 09:33.

Step-3: Fill the necessary fields. Here I gave the name for workspace as “teamjDBworkspace”.

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Resource group * Create new

Workspace name *

Region *

Pricing Tier *

We selected the recommended pricing tier for your workspace. You can change the tier based on your needs.

Managed Resource Group name

Review + create < Previous Next : Networking >

Step-4: Databricks workspace is successfully created.

rg-azuser936_mml.local-pt5l7_teamjDBworkspace | Overview

Your deployment is complete

Deployment name : rg-azuser936_mml.local-pt5l7_teamjDBworkspace

Subscription : Azure subscription 1

Resource group : rg-azuser936_mml.local-pt5l7

Start time : 12/01/2024, 09:34:34

Correlation ID : 5ac3e97d-6325-455f-a82f-54ea45ffa454

Deployment succeeded

Deployment 'rg-azuser936_mml.local-pt5l7_teamjDBworkspace' to resource group 'rg-azuser936_mml.local-pt5l7' was successful.

Go to resource Pin to dashboard

Give feedback Tell us about your experience with deployment

Cost management Get notified to stay within your budget and prevent unexpected charges on your bill. Set up cost alerts >

Microsoft Defender for Cloud Secure your apps and infrastructure Go to Microsoft Defender for Cloud >

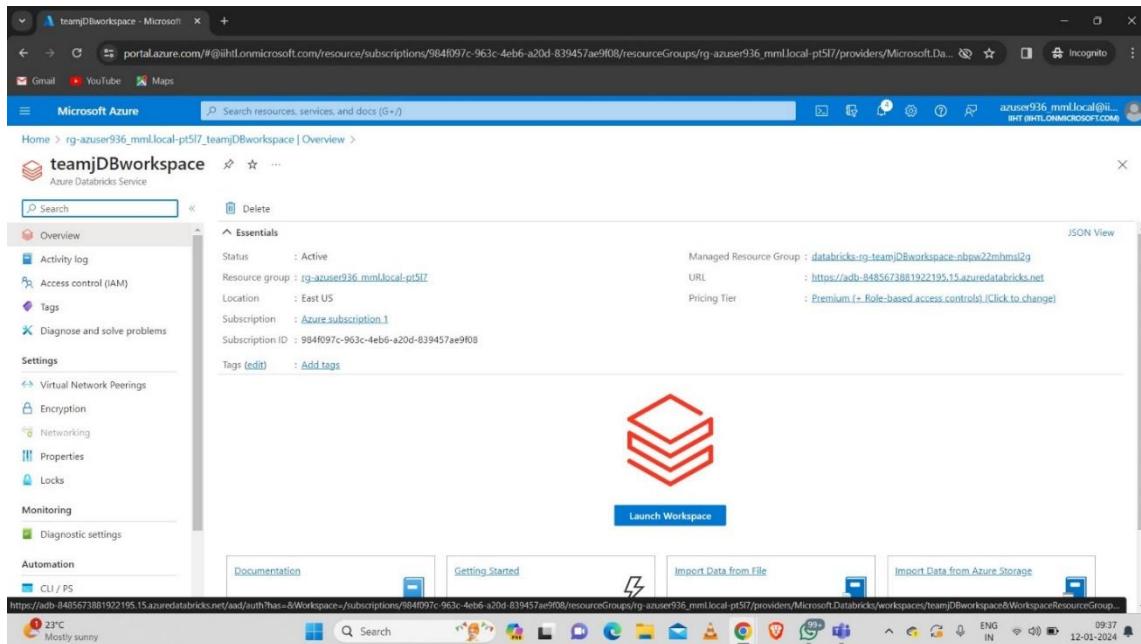
Free Microsoft tutorials Start learning today >

Work with an expert Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support. Find an Azure expert >

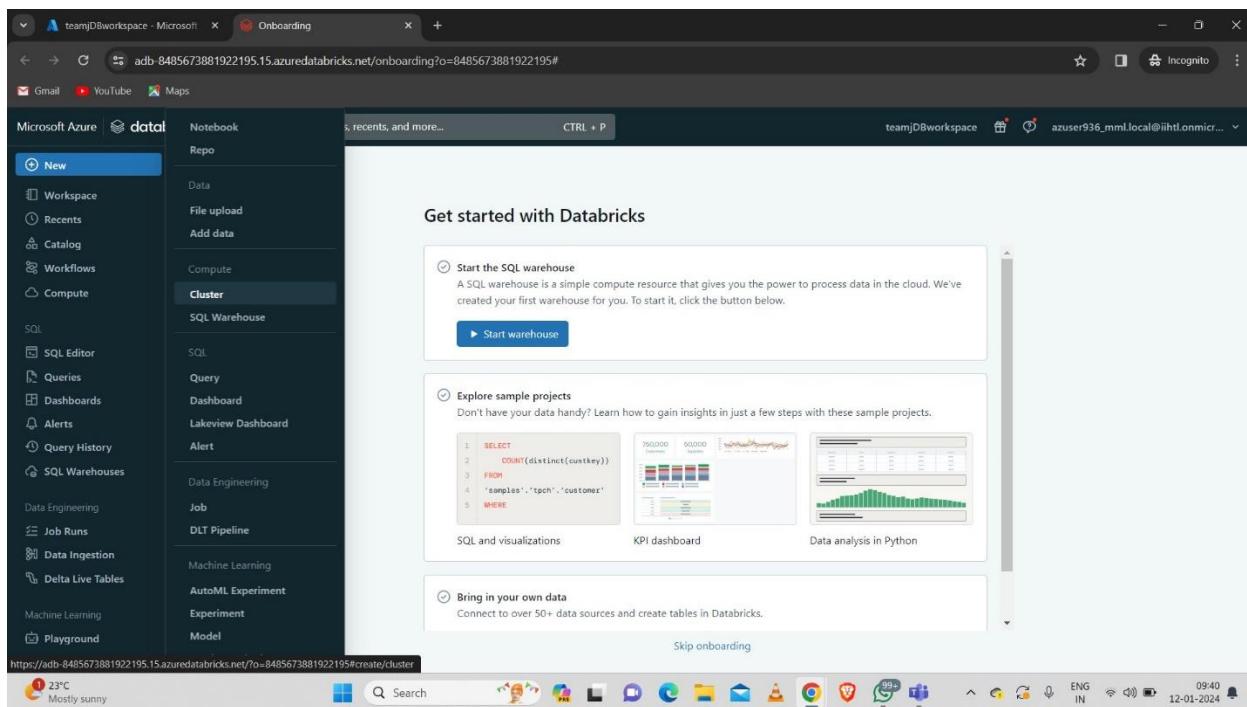
23°C Mostly sunny ENG IN 09:36 12-01-2024

6.5 Creating Databricks Cluster and Notebook

Step-1: Launch the databricks workspace.



Step-2: After launching the databricks workspace will be opened. Here click on New then click on cluster to create the new cluster.



Step-3: Fill the necessary fields in the cluster then click on create. Here I gave the name of the cluster as “teamjprojectCluster”

teamjDBworkspace - Microsoft | Create Cluster - Databricks | +

adb-8485673881922195.15.azuredatabricks.net/?o=8485673881922195#cluster

Microsoft Azure | databricks | Search data, notebooks, recents, and more... | CTRL + P | teamjDBworkspace | azuser936_mml.local@ihtl.onmicrosoft.com

teamjprojectCluster

Compute

Policy: Personal Compute

Single user access: azuser936_mml.local@ihtl.onmicrosoft.com

Performance

Databricks runtime version: Runtime: 13.3 LTS (Scala 2.12, Spark 3.4.1)

Node type: Standard_DS3_v2

Terminate after: 4320 minutes of inactivity

Tags

Add tags

Create compute | Cancel

23°C Mostly sunny | 09:41 12-01-2024

Step-4: Click on new notebook and create the new notebook.

teamjDBworkspace - Microsoft | Onboarding | +

adb-8485673881922195.15.azuredatabricks.net/onboarding?o=8485673881922195#onboarding

Microsoft Azure | databricks | Search data, notebooks, recents, and more... | CTRL + P | teamjDBworkspace | azuser936_mml.local@ihtl.onmicrosoft.com

Get started with Databricks

Start the SQL warehouse
A SQL warehouse is a simple compute resource that gives you the power to process data in the cloud. We've created your first warehouse for you. To start it, click the button below.

Explore sample projects
Don't have your data handy? Learn how to gain insights in just a few steps with these sample projects.

SQL and visualizations
SELECT COUNT(DISTINCT(custkey))
FROM "samples"."tpcn"."customer"
WHERE

KPI dashboard
Data analysis in Python

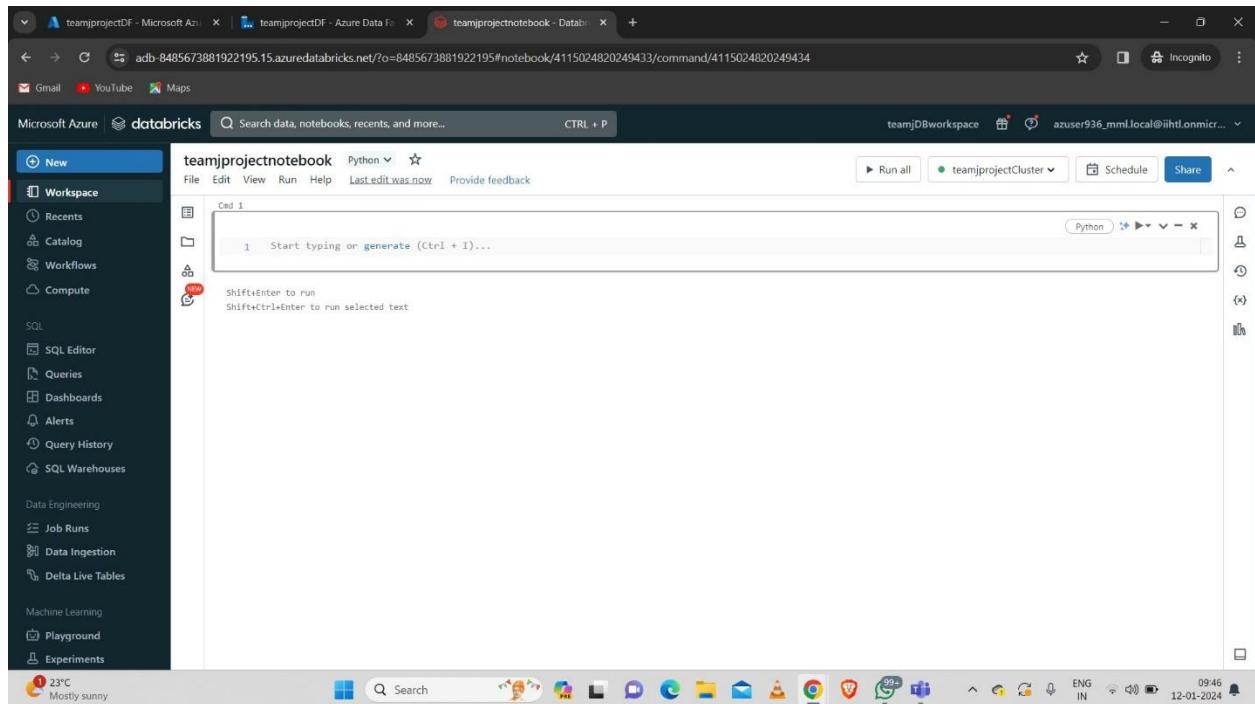
Bring in your own data
Connect to over 50+ data sources and create tables in Databricks.

Skip onboarding

23°C Mostly sunny | 09:40 12-01-2024

Step-5: Here I gave the notebook name as “teamjprojectnotebook”.

- Here we can see that the cluster is connected to the notebook.

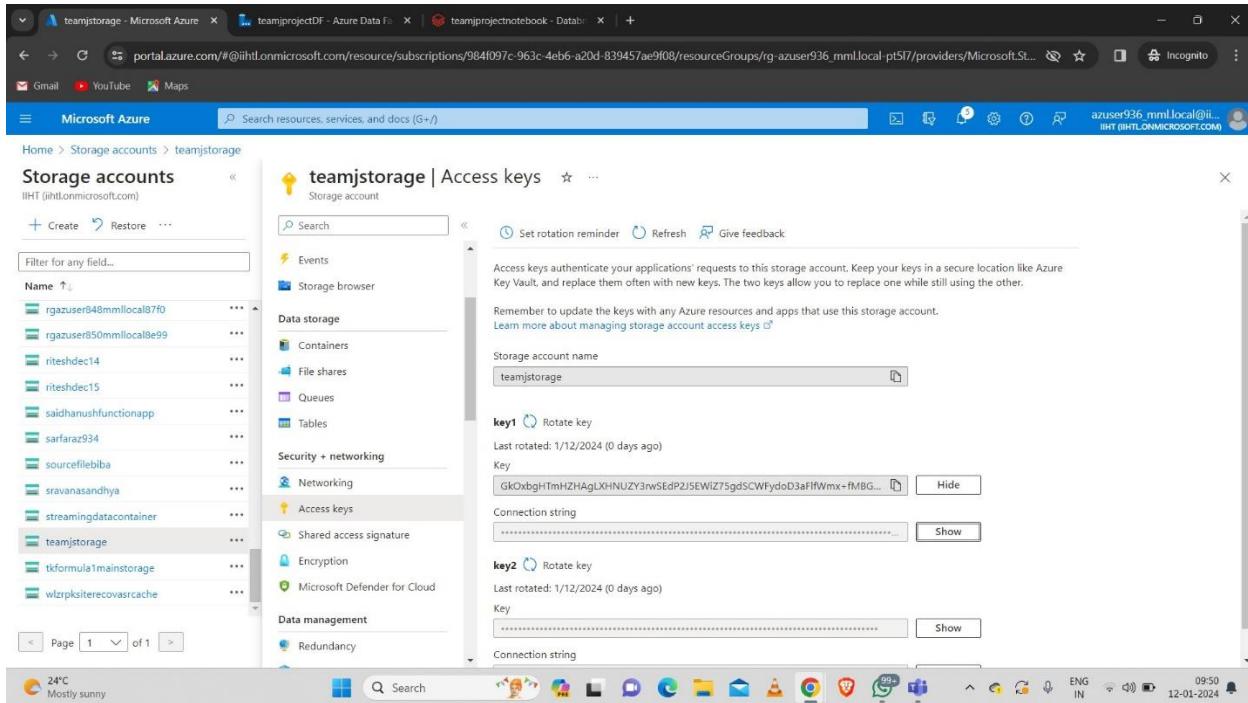


6.6 Mounting ADLS with Databricks

Step-1: Mount (Connect) the blob storage container with databricks with following command.

```
dbutils.fs.mount(source      =      'wasbs://<container-name>@<storage-account-name>.blob.core.windows.net',
                  mount_point = '/mnt/<mount-name>',
                  extra_configs = {'fs.azure.account.key.' <storage-account-name>.blob.core.windows.net': 'Accountkey'})
```

Step-2: To get the Account key go to the storage account then click on Access keys there we can see a key value copy that and paste at the account key value

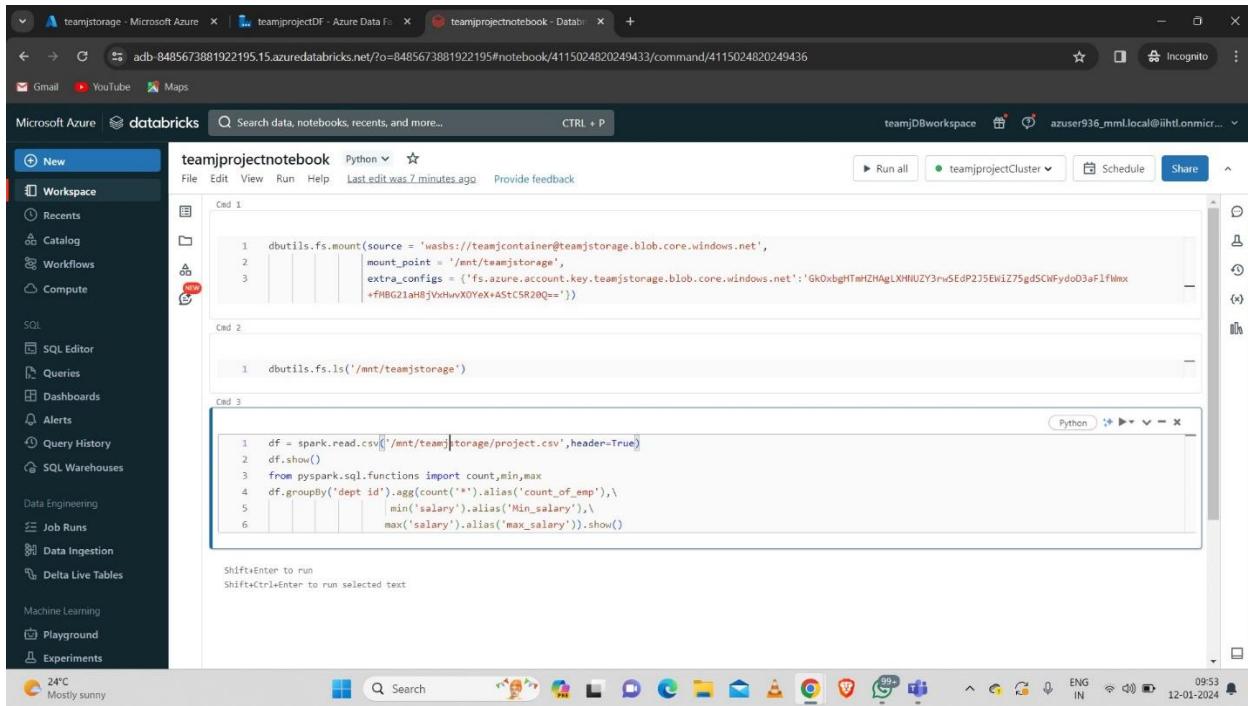


Step-3: Write the code as follows inside the notebook do not run the code as we need to integrate it with ADF.

```
dbutils.fs.mount(source = 'wasbs://teamjcontainer@teamjstorage.blob.core.windows.net',
mount_point = '/mnt/ teamjstorage ',
extra_configs=
{'fs.azure.account.key.teamjstorage.blob.core.windows.net':'DU/jzGolq9vQx9iWfpIxlhgnzhpJJ
UpCQ0I+6s43LQvDtawIMbs094pMDJxuLOJENsYW1u6gTs0H+AStqlfEwA=='})
```

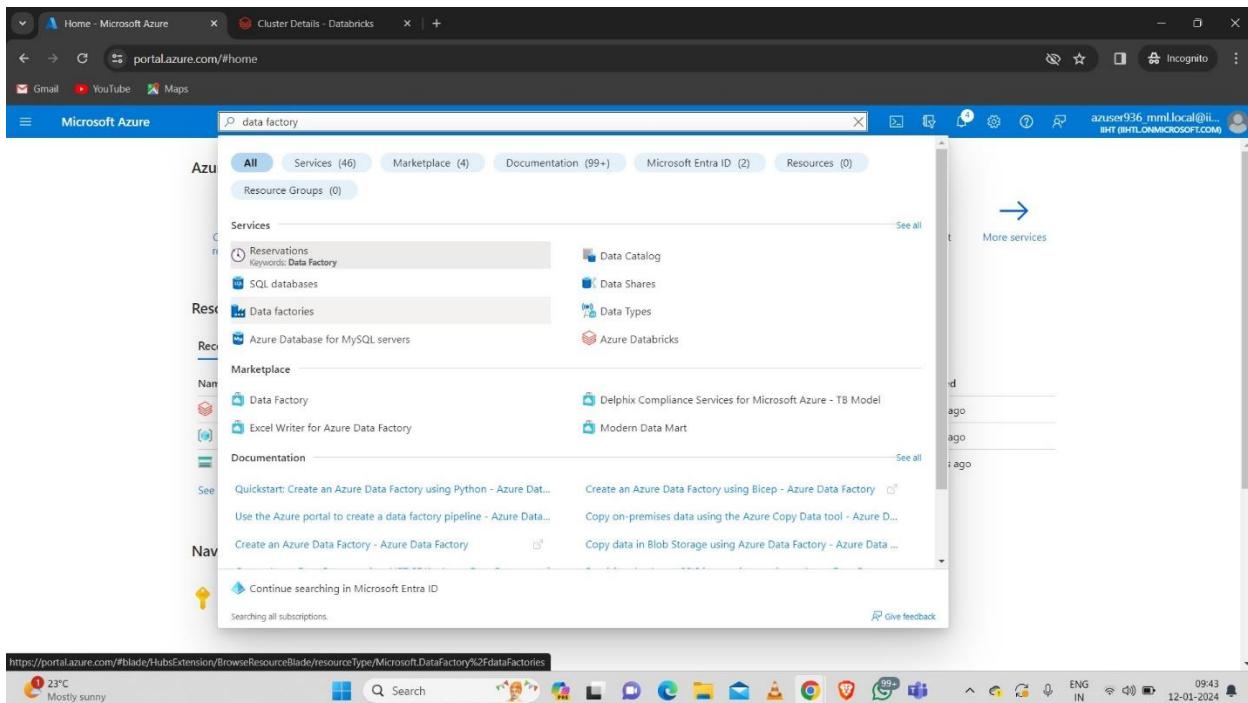
```
dbutils.fs.ls('/mnt/ teamjstorage ')
```

```
df = spark.read.csv('/mnt/ teamjstorage /project.csv',header=True)
df.show()
from pyspark.sql.functions import count,min,max
df.groupBy('dept id').agg(count('*').alias('count_of_emp'),\
    min('salary').alias('Min_salary'),\
    max('salary').alias('max_salary')).show()
```



6.7 Azure Data Factory

Step-1: Go to search bar and search for Data Factories



Step-2: Fill the necessary data and create the data factory. Here I gave the data factory name as “teamjprojectDF”.

TERMS

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. See the Azure Marketplace Terms for additional details.

Basics

Subscription	Azure subscription 1
Resource group	rg-azuser936_mml.local-pt5l7
Name	teamjprojectDF
Region	East US
Version	V2

Networking

Connect via	Public endpoint
-------------	-----------------

Previous Next Create

Step-3: Here we can see that the data factory is successfully created.

Microsoft.DataFactory-20240112094336 | Overview

Your deployment is complete

Deployment name : Microsoft.DataFactory-20240112094336
Subscription : Azure subscription 1
Resource group : rg-azuser936_mml.local-pt5l7

Start time : 12/01/2024, 09:44:43
Correlation ID : 5013e871-b684-4800-abd4-3b4be99488f0

Deployment succeeded

Deployment 'Microsoft.DataFactory-20240112094336' to resource group 'rg-azuser936_mml.local-pt5l7' was successful.

Pin to dashboard Go to resource group

Cost management

Get notified to stay within your budget and prevent unexpected charges on your bill.
Set up cost alert >

Microsoft Defender for Cloud

Secure your apps and infrastructure
Go to Microsoft Defender for Cloud >

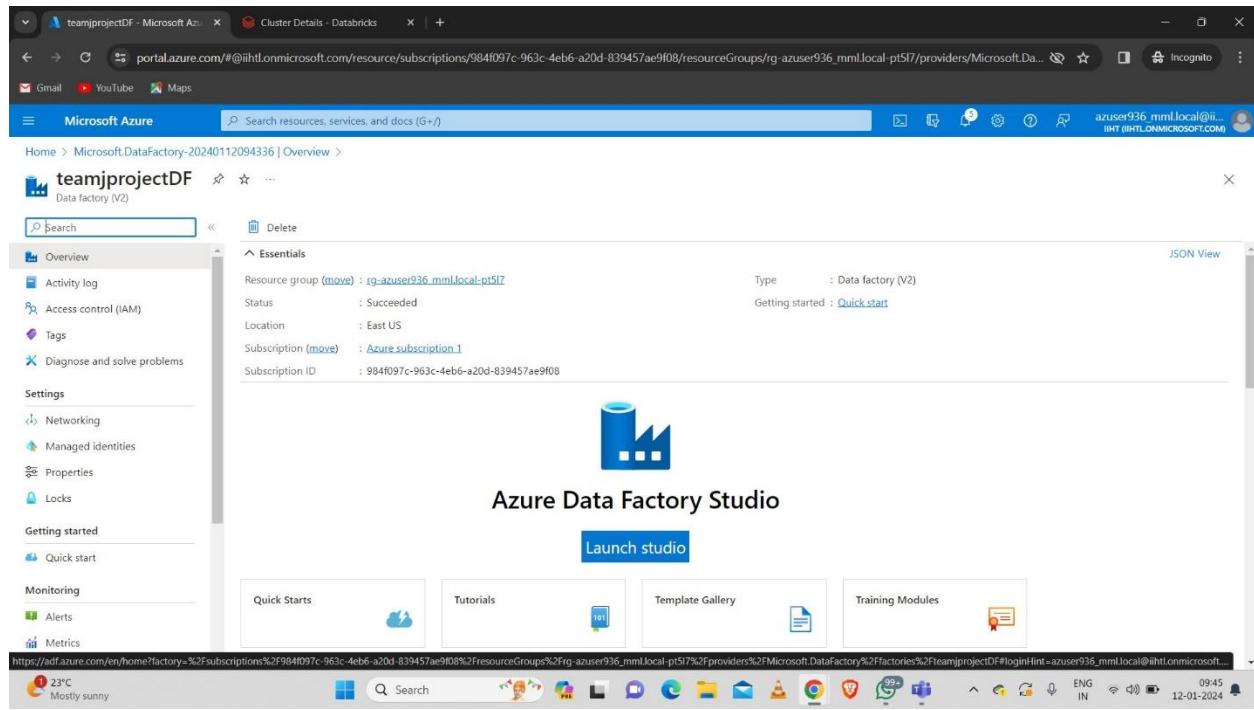
Free Microsoft tutorials

Start learning today >

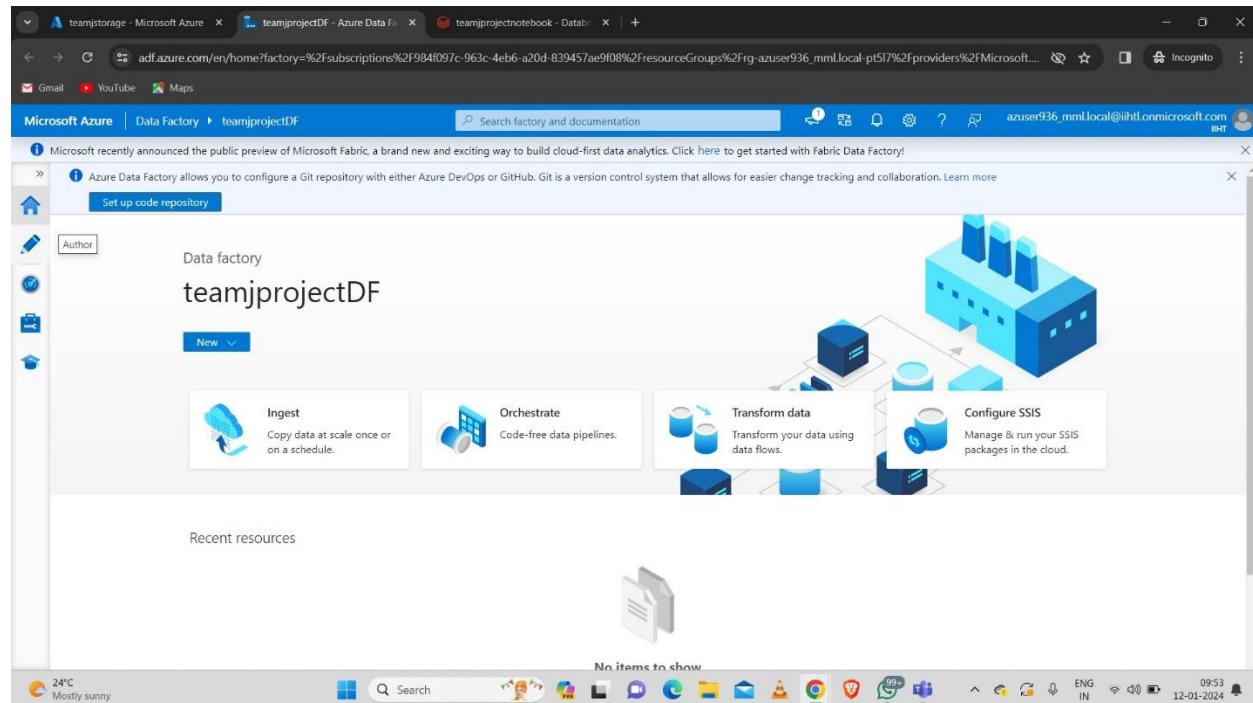
Work with an expert

Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.
Find an Azure expert >

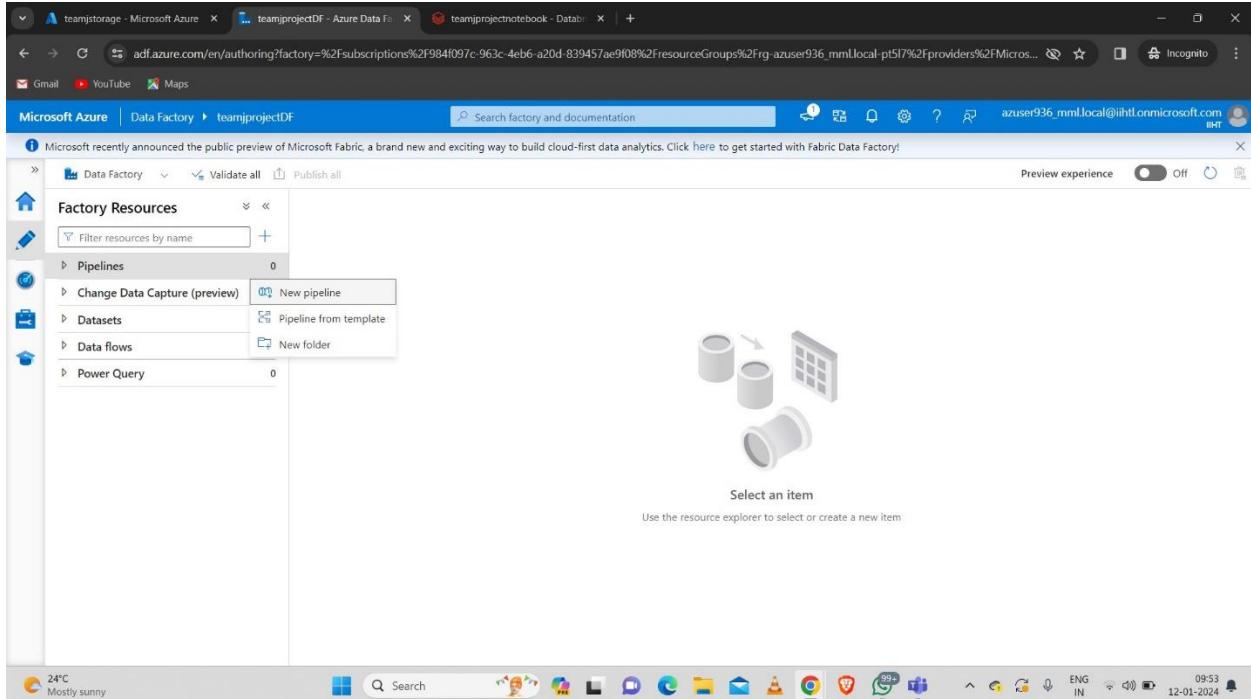
Step-4: Launch the Azure Data Factory studio.



Step-5: After launching we get the following page.



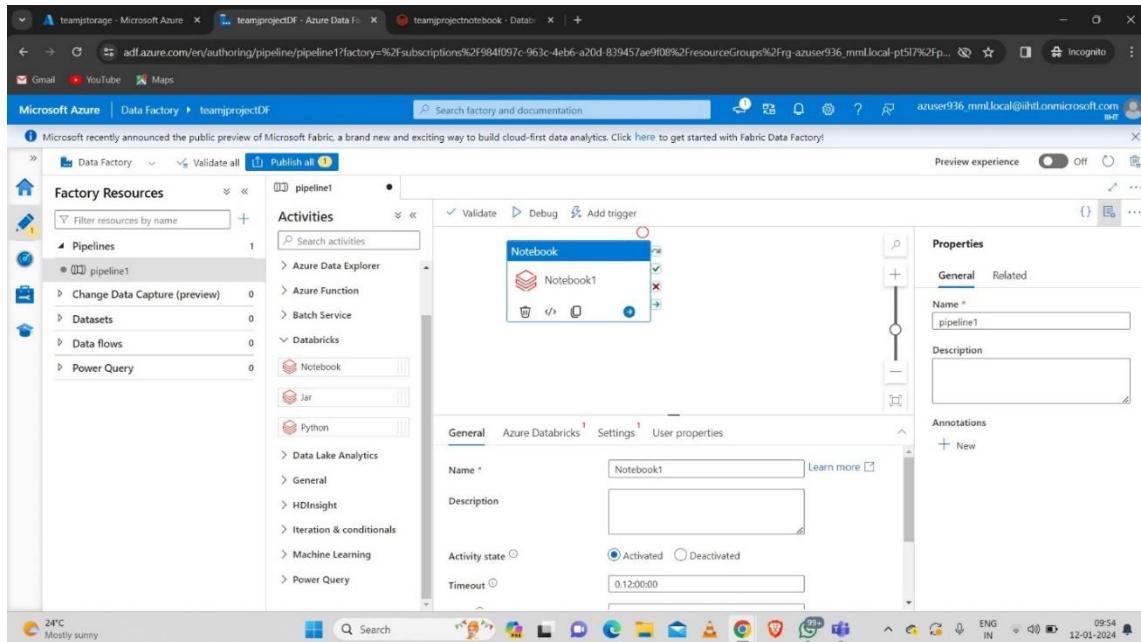
- Now click on author to get the following page
- Click on new pipeline to create one.



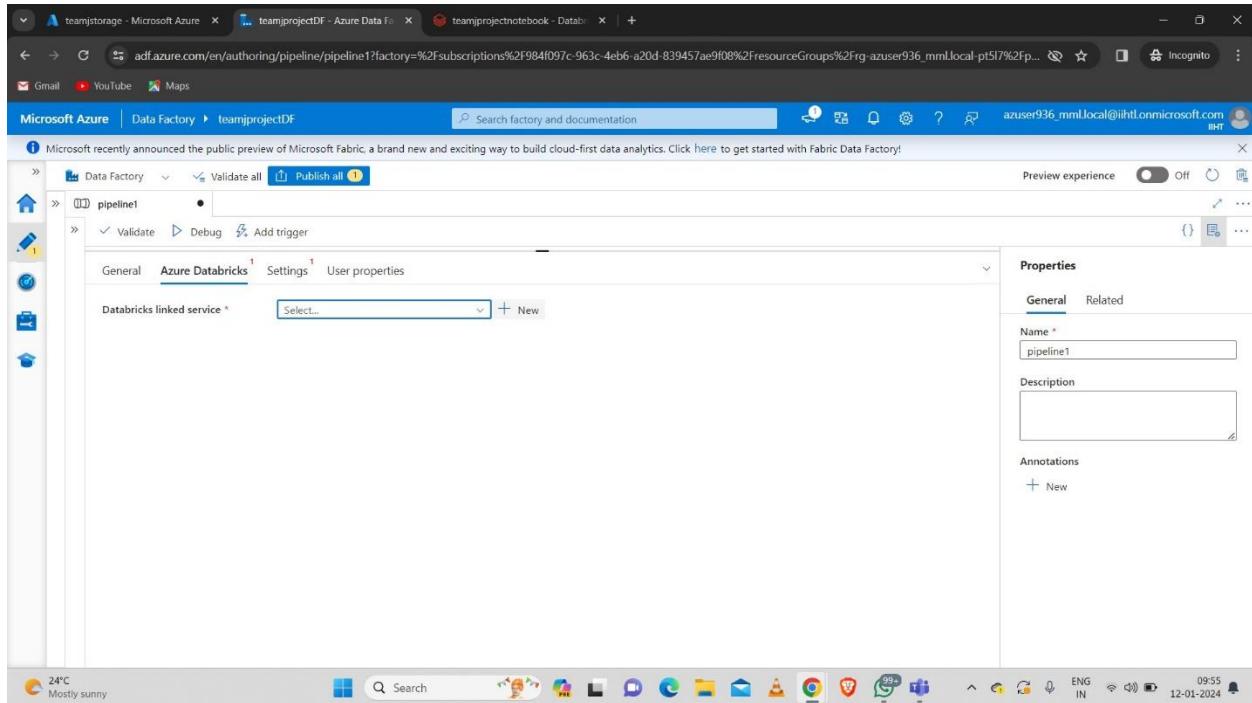
6.8 Integrating ADF with Azure Databricks

Step-1: Create a new pipeline.

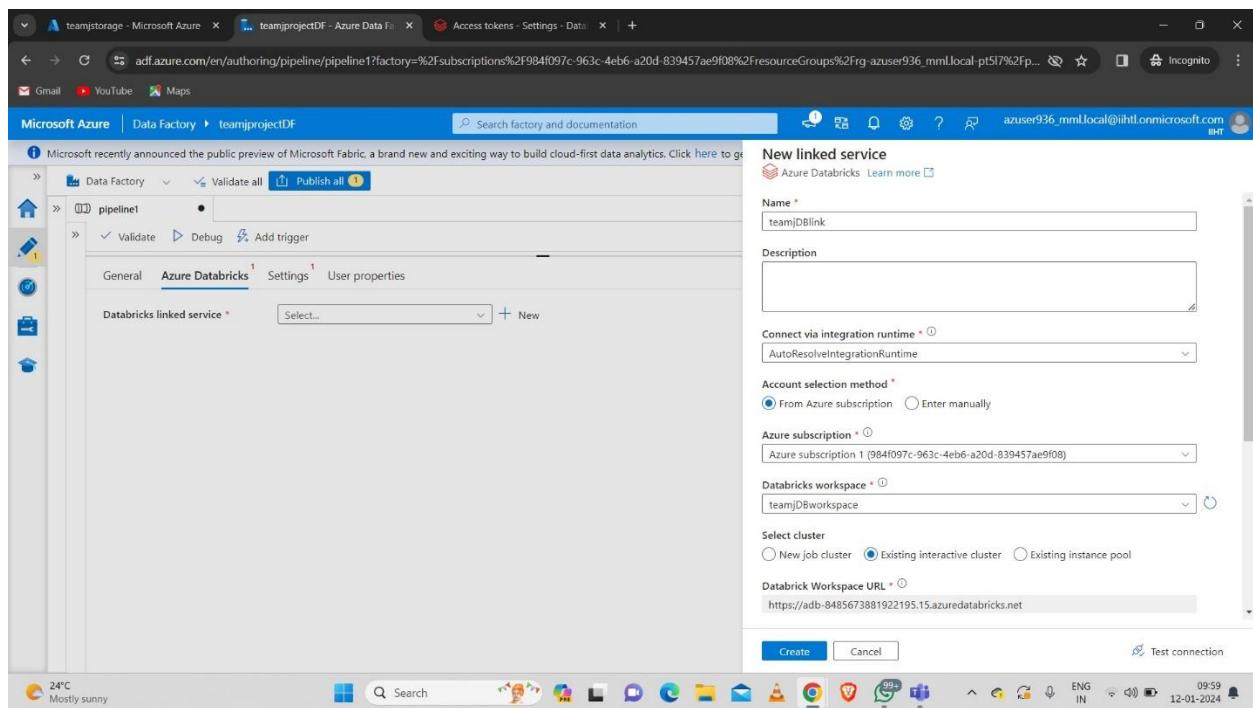
- Now go to Databricks in Activities bar and drag the Notebook.



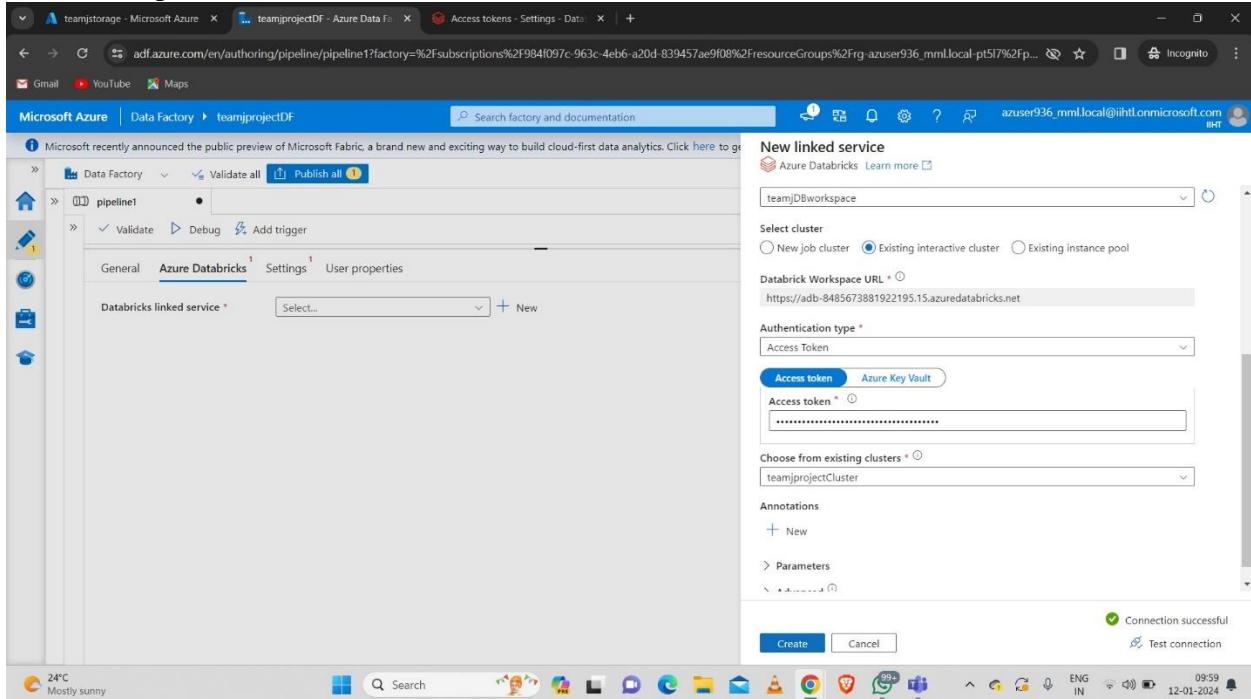
Step-2: Now click on new to link the databricks with this pipeline.



- Fill the necessary data. Select the Existing interactive cluster that we already created to run the notebook



- Choose the Authentication type as Access token and copy the access token from databricks and paste it.



Access token:

1. Go to Azure Databricks workspace.
2. Go to User Settings.
3. Go to Developer path.
4. Click on Manage Access token.
5. Generate the new Access token and copy it and paste in the ADF.

The screenshot shows the Azure Databricks Notebook interface. On the left, the sidebar includes options like New, Workspace, Recents, Catalog, Workflows, Compute, SQL, SQL Editor, Queries, Dashboards, Alerts, Query History, SQL Warehouses, Data Engineering, Job Runs, Data Ingestion, Delta Live Tables, Machine Learning, and Playground. The main area displays a Python notebook titled "teamjprojectnotebook". It contains three cells:

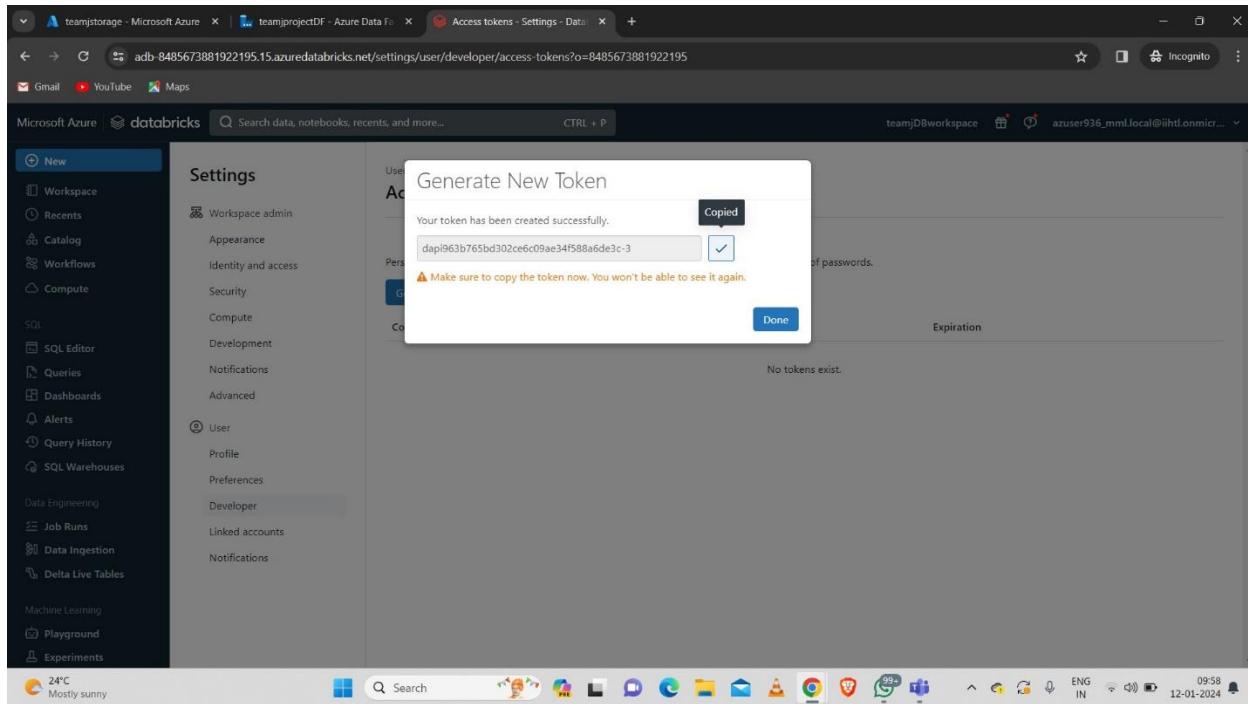
```
Cmd 1
1 dbutils.fs.mount(source = 'wasbs://teamjcontainer@teamjstorage.blob.core.windows.net',
2                   | mount_point = '/mnt/teamjstorage',
3                   | extra_configs = ('fs.azure.account.key.teamjstorage.blob.core.windows.net':'GkOxbgHTeHZHAgLXHlUZY3ruSEdP2J5EWiZ75gdSc
+HBG21ahBjVxHuvXOYex+AstCSR20Q=='))
```

```
Cmd 2
1 dbutils.fs.ls('/mnt/teamjstorage')
```

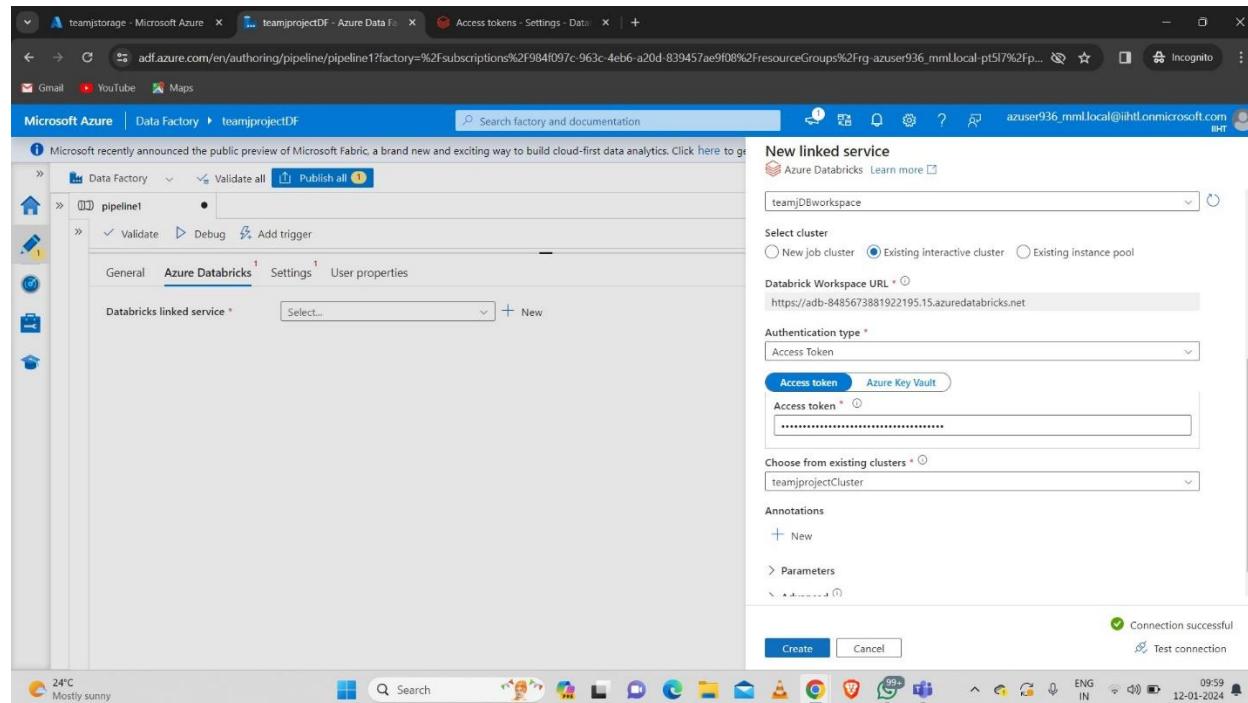
```
Cmd 3
1 df = spark.read.csv('/mnt/teamjstorage/project.csv',header=True)
2 df.show()
3 from pyspark.sql.functions import count,min,max
4 df.groupby('dept_id').agg(count('*').alias('count_of_emp'),\
5 | min('salary').alias('Min_salary'),\
6 | max('salary').alias('Max_salary')).show()
```

The right side features a "User Settings" panel with options like Admin Settings, Azure Portal, Manage Account, Privacy Policy, and Log out. The status bar at the bottom shows the URL <https://adb-8485673881922195.15.azuredatabricks.net/settings/user/profile?o=8485673881922195>, the date 12-01-2024, and the time 09:56.

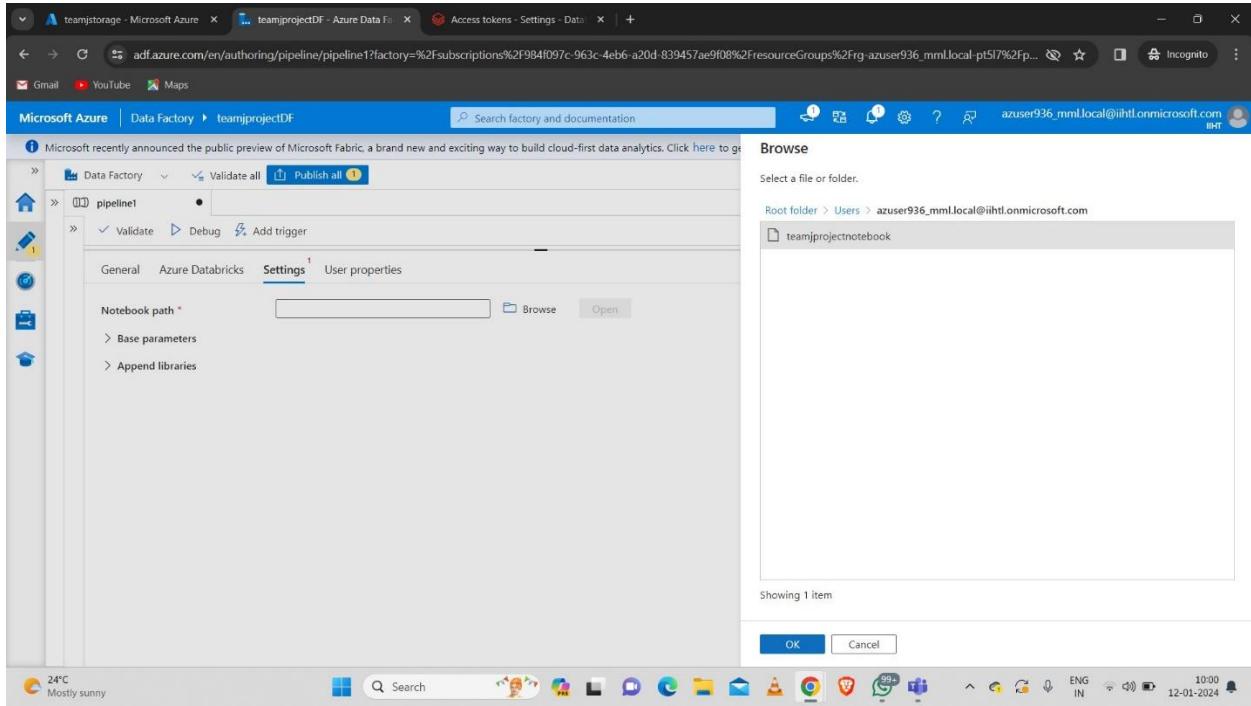
The screenshot shows the Azure Databricks Developer settings page. The left sidebar is identical to the first screenshot. The main area is titled "Developer" and "Manage your development settings". It includes sections for "Access tokens", "SQL query snippets", and "Editor settings". Under "Editor settings", there are "General" settings and two toggle switches: "Notebook Notifications" (On) and "Spark tips" (On). The status bar at the bottom shows the URL <https://adb-8485673881922195.15.azuredatabricks.net/settings/user/developer/access-tokens?o=8485673881922195>, the date 12-01-2024, and the time 09:57.



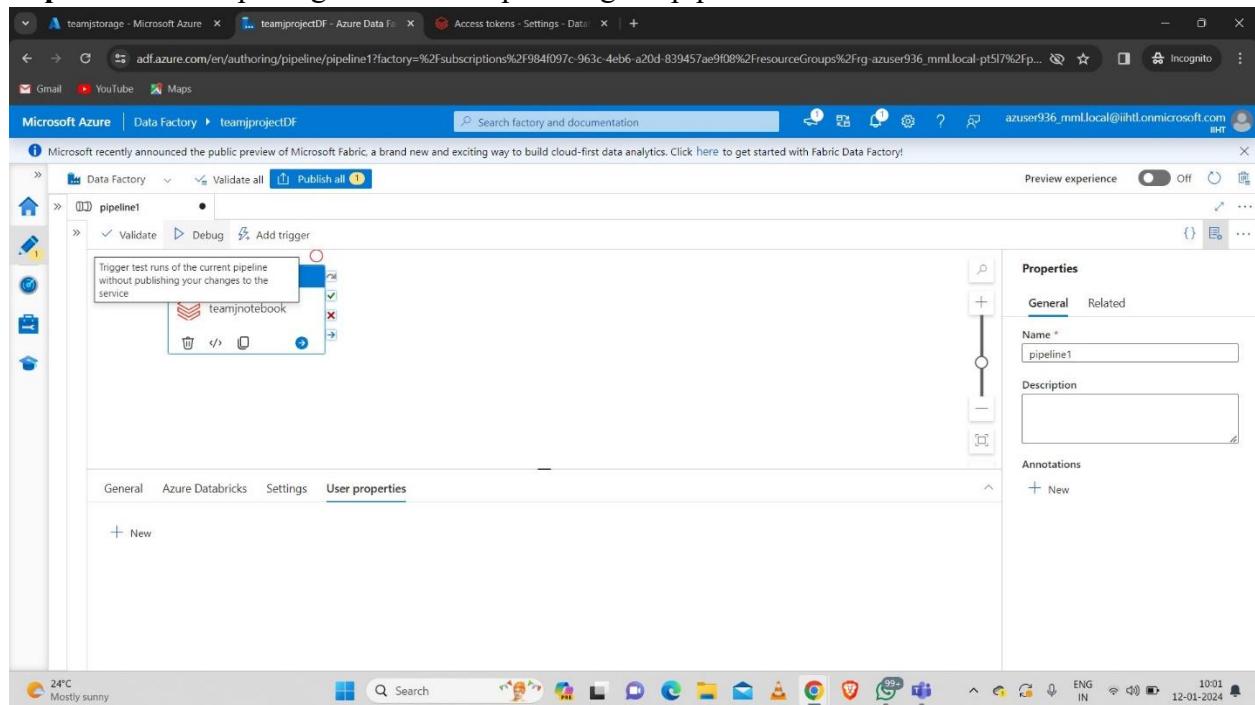
➤ Now create the databricks link service.



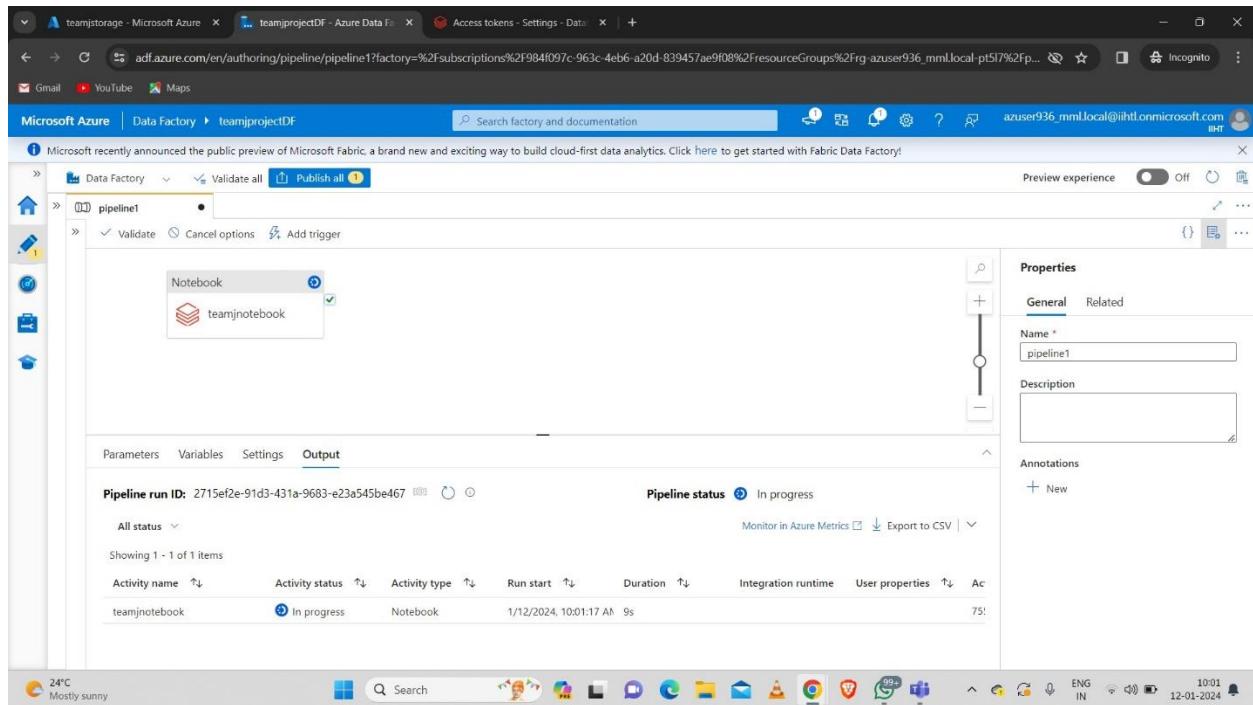
Step-3: Now give the path for the notebook by browsing it and giving the notebook path that we created as “teamjprojectnotebook”.



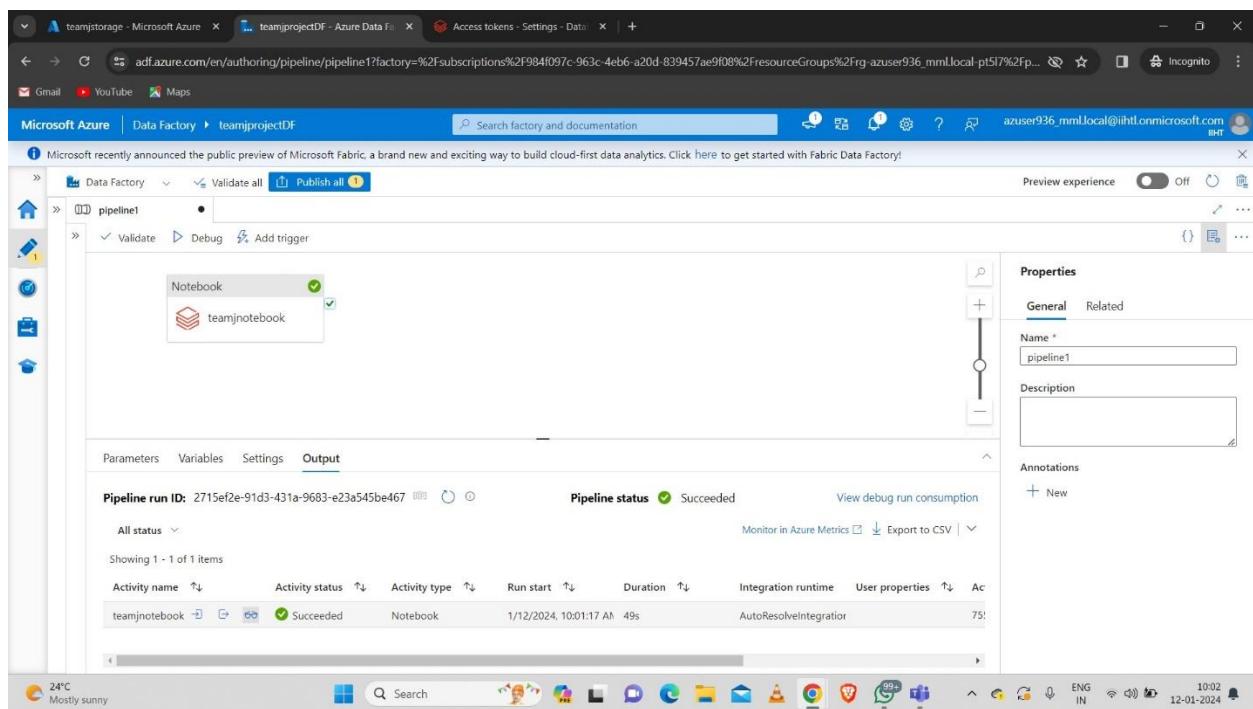
Step-4: After completing the above steps debug the pipeline.



- We can see the activity status of the pipeline as follows.



- We can see that the pipeline is debugged successfully.



6.9 Final Output

- Click on details at the output of the pipeline after debugging.
- Click on the url.

The screenshot shows the Microsoft Azure Data Factory interface. On the left, there's a sidebar with icons for Data Factory, Validate all, Publish all, and pipeline1. The main area displays a notebook named 'teamjnotebook'. A 'Details' modal is open, showing a duration of 00:00:46 and a 'Run page url' of <https://adb-8485673881922195.15.azuredatabricks.net/?o=8485673881922195#job/582366094629089/run/110582424004032>. The pipeline status is 'Succeeded'. The 'Output' tab is selected, showing a table with one item: 'teamjnotebook' (Status: Succeeded, Activity type: Notebook, Run start: 1/12/2024, Duration: 49s). The properties panel on the right shows the pipeline name as 'pipeline1'.

- After clicking on the above url we get the output of the notebook which we created as “teamjprojectnotebook”.

The screenshot shows the Azure Databricks workspace. The left sidebar includes 'Workflows', 'Compute', 'SQL', 'Job Runs', 'Data Ingestion', 'Delta Live Tables', 'Machine Learning', 'Playground', and 'Experiments'. The main area shows a workflow run titled 'ADF_teamjprojectDF_pipeline1_teamjnotebook_755feeb7-975e-4ff3-9db3-caeee9ccaa44 run'. The 'Output' section displays the results of a notebook run. The code shown is:

```
df = spark.read.csv('/mnt/teamjstorage/project.csv', header=True)
df.show()
from pyspark.sql.functions import count,min,max
df.groupBy('dept_id').agg(count('*').alias('count_of_emp'),\
                           min('salary').alias('Min_salary'),\
                           max('salary').alias('Max_salary')).show()
```

The output data is:

empid	name	sex	salary	dept_id
1	maheer	M	5000	IT
2	wafa	M	6000	IT
3	asi	F	2500	Payroll
4	sarfaraj	M	4000	HR
5	parijaat	F	2000	HR
6	Mahabooib	M	2000	Payroll
7	ayesha	F	3000	IT
8	ashish	M	50000	IT
9	awini	F	67346	HR
10	balaji	M	100000	support
11	harika	F	10000	sales

The screenshot shows the Azure Databricks interface with the 'Workflow' sidebar open. The main area displays the output of a PySpark DataFrame named 'df'. The data consists of 18 rows of employee information, including empid, name, sex, salary, and department. The output is presented as a table with columns: empid, name, sex, salary, and dept id.

empid	name	sex	salary	dept id
1	maheen	M	5000	IT
2	wafal	M	6000	IT
3	asil	F	2500	Payroll
4	sarfaraj	M	4000	HR
5	pyarjanan	F	2000	HR
6	Mahabool	M	2000	Payroll
7	ayeshal	F	3000	IT
8	ashishi	M	90000	IT
9	aswini	F	67346	HR
10	balaji	M	100000	support
11	harikali	F	10000	sales
12	yatin	M	50000	analyst
13	tarun	M	40000	business
14	sowjali	F	20000	Payroll
15	charan	M	70000	financial
16	kushith	M	30000	support
17	karishma	F	20000	sales
18	dileep	M	160000	developer

Command took 13.42 seconds

The screenshot shows the Azure Databricks interface with the 'Workflow' sidebar open. The main area displays the output of a PySpark DataFrame named 'df'. The data consists of 20 rows of aggregated employee information, including department ID, count of employees, minimum salary, and maximum salary. The output is presented as a table with columns: dept id, count_of_emp, Min_salary, and max_salary.

dept id	count_of_emp	Min_salary	max_salary
HR	6	11000	67346
IT	6	3000	50000
Payroll	5	2000	40000
analyst	4	100000	80500
business	2	40000	90000
data engineer	3	40000	70000
developer	2	160000	84678
financial	4	20000	89888
financial	1	30000	30000
sales	6	10000	76000
solution developer	2	50000	90000
support	2	100000	30000
system engineer	2	25000	50000
techiee	5	30000	80000

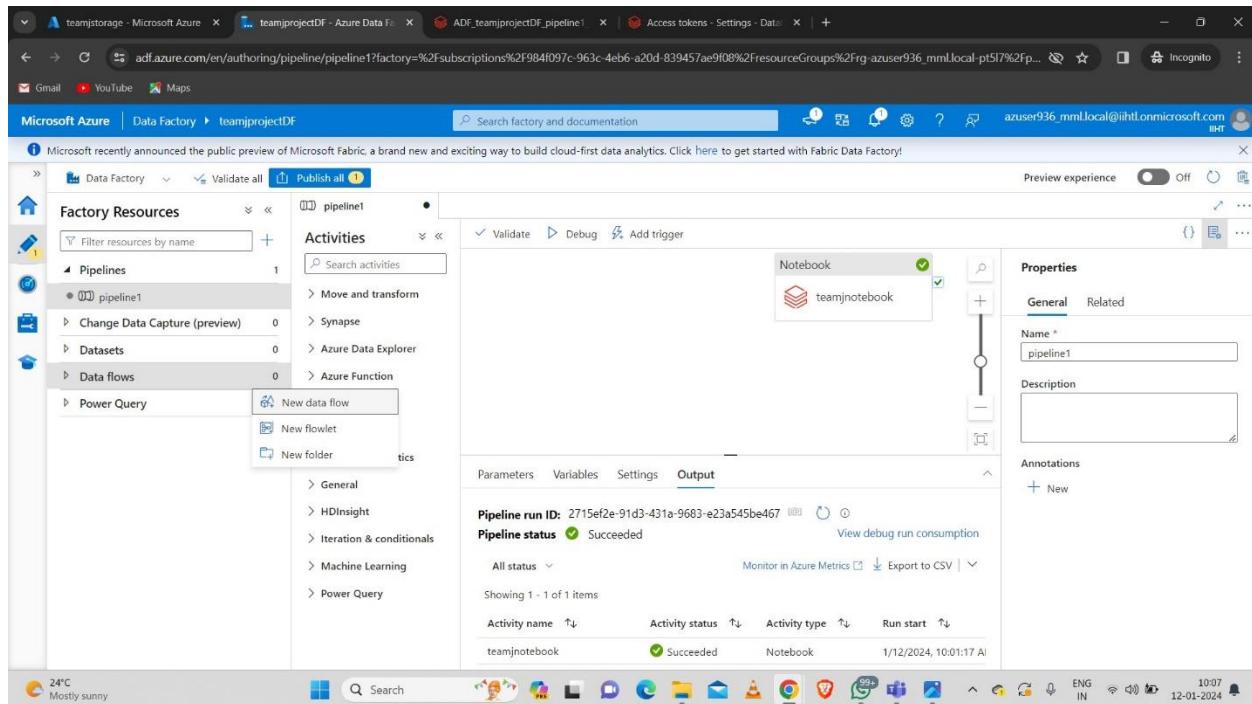
only showing top 20 rows

Command took 13.42 seconds

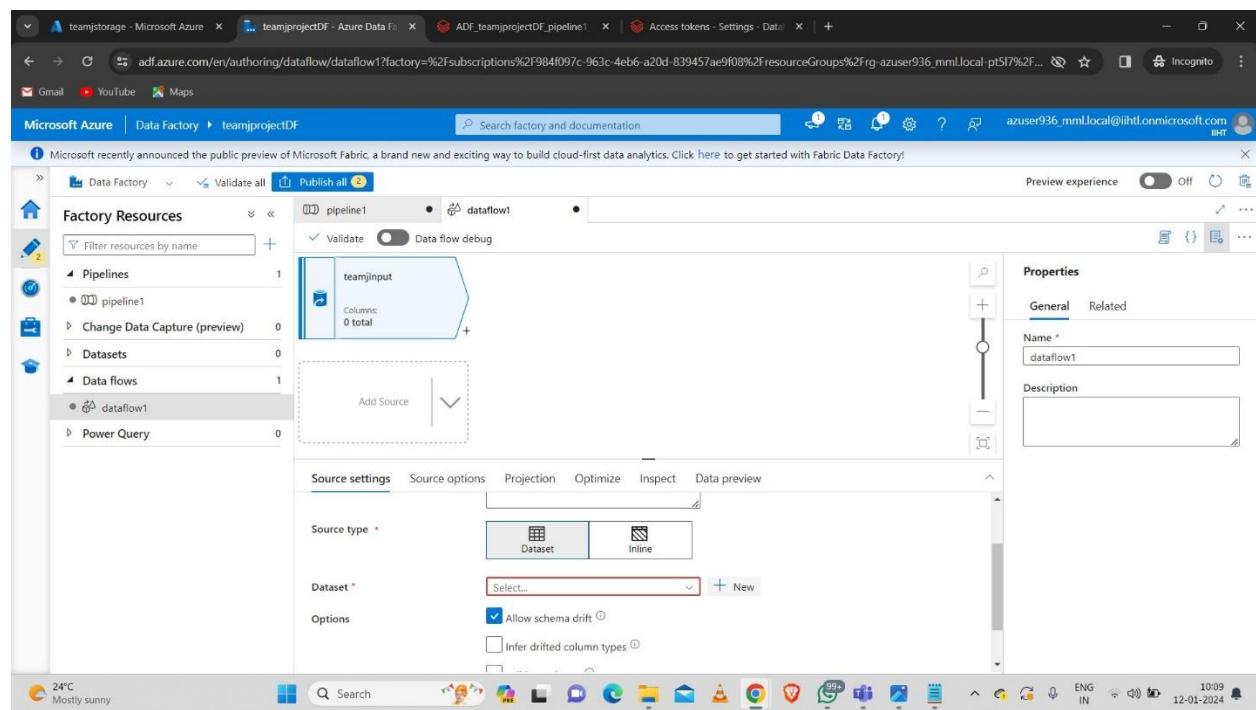
Hence we successfully implemented a daily data aggregation pipeline using Azure Data Factory to move raw data and Azure Databricks to aggregate and summarize the data based on daily intervals.

7.Copying the Aggregate data using Data Flow Activity

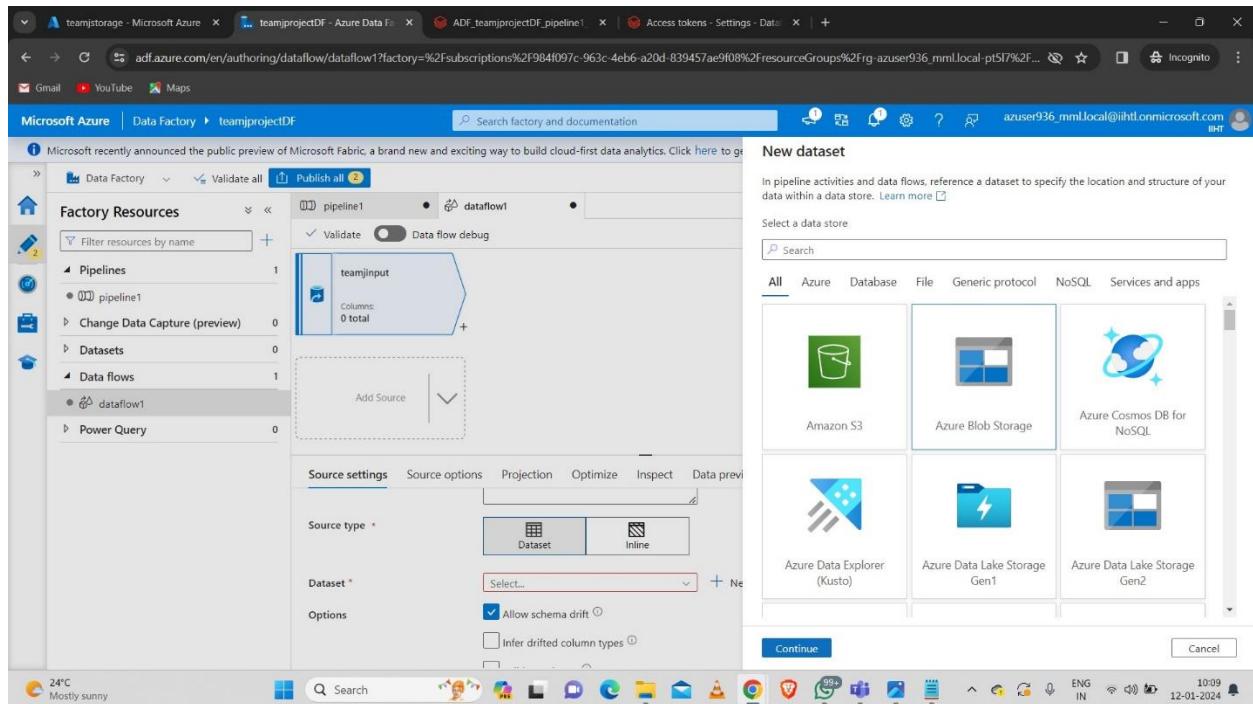
Step-1: Create a new data flow.



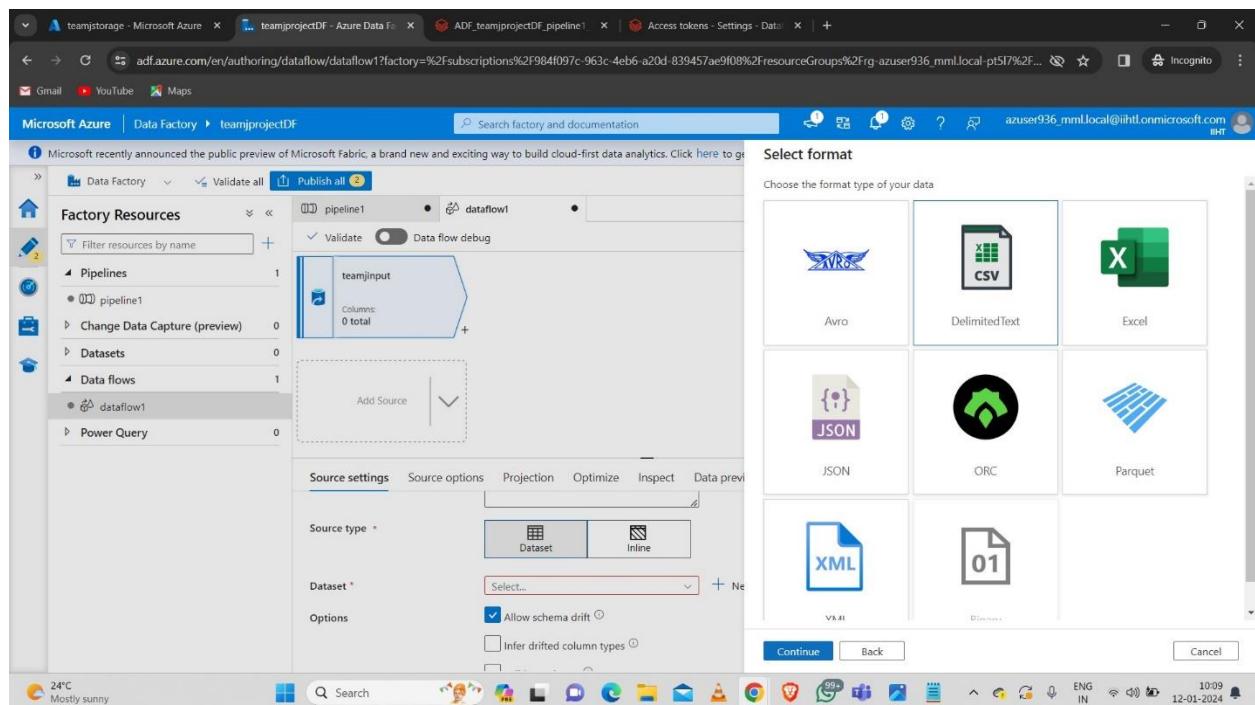
Step-2: Now inside the source settings link the new dataset.



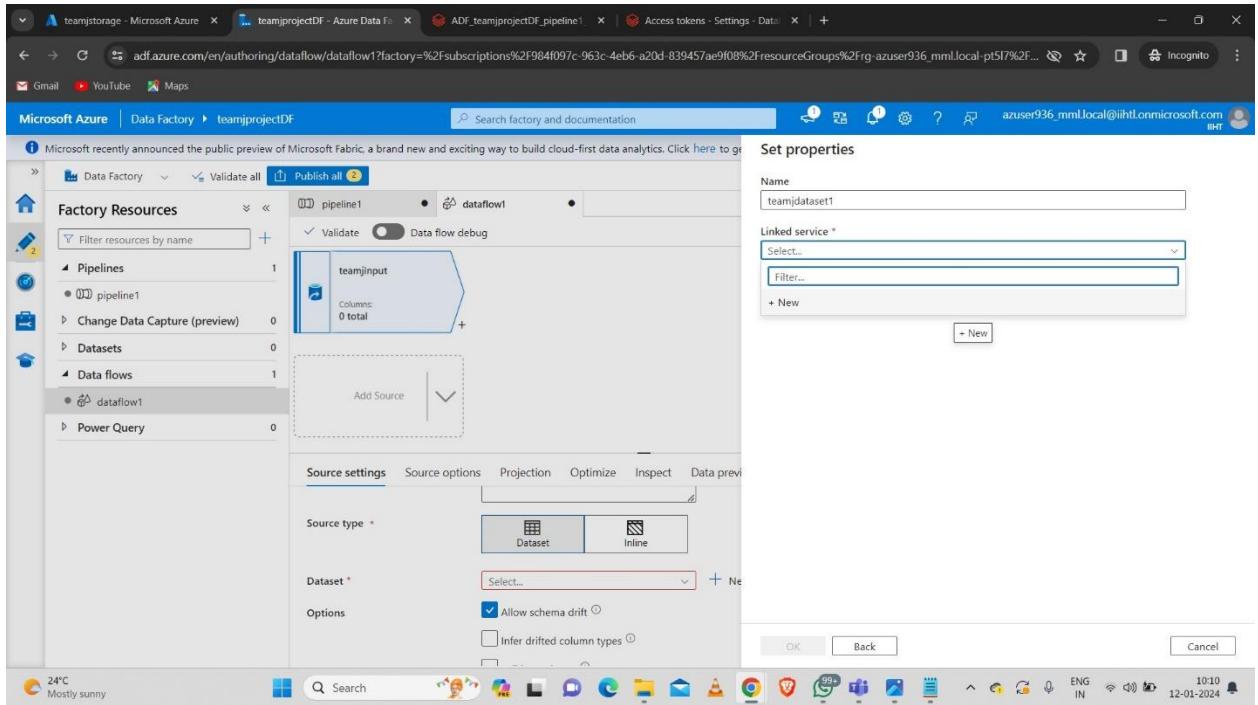
➤ Select the type of storage.



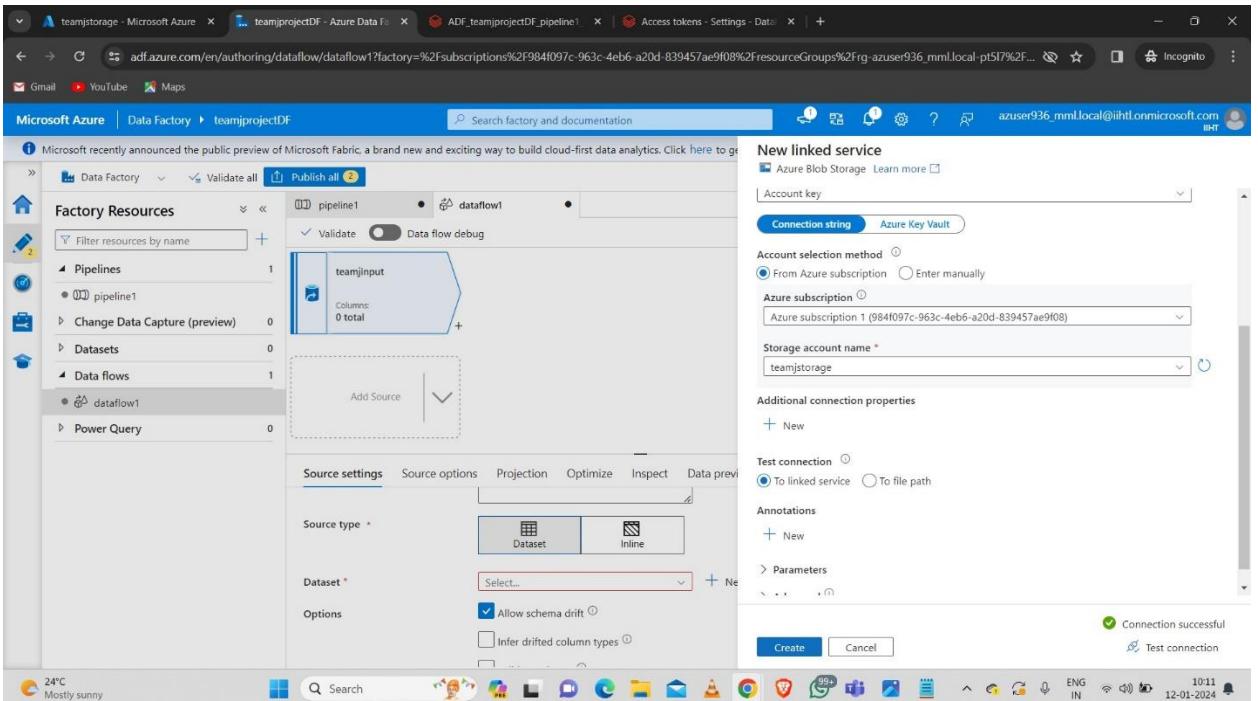
➤ Select the file type as CSV.



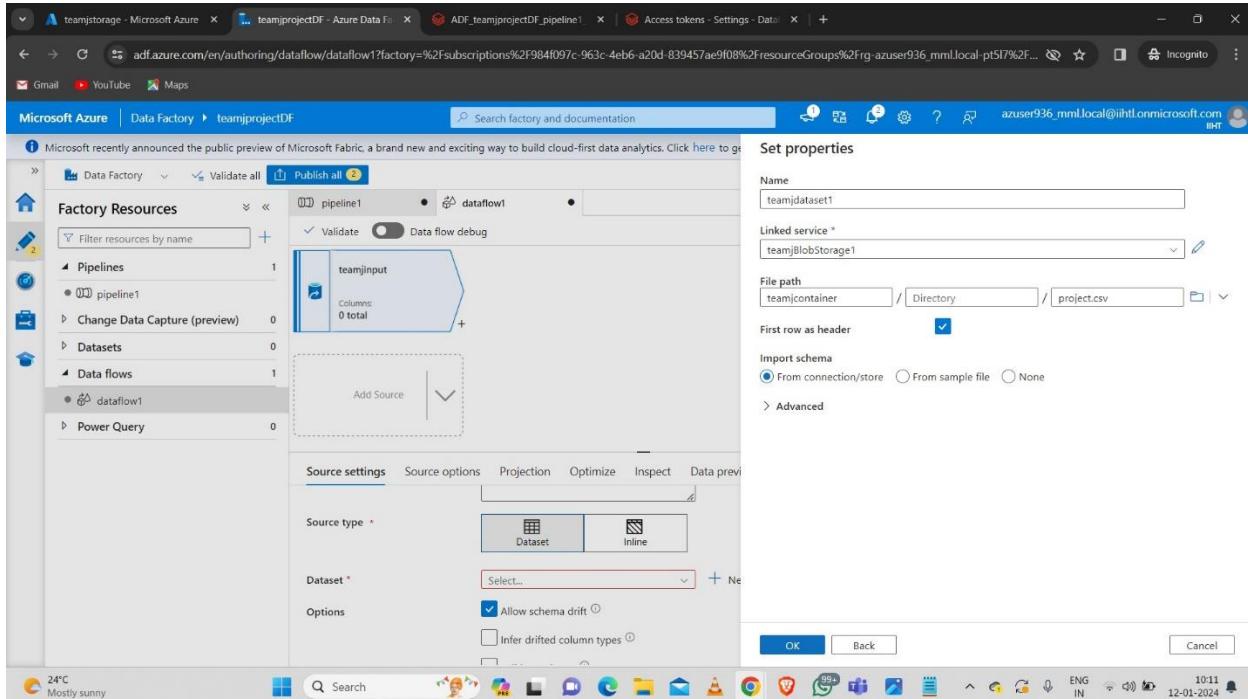
- Select the new linked service.



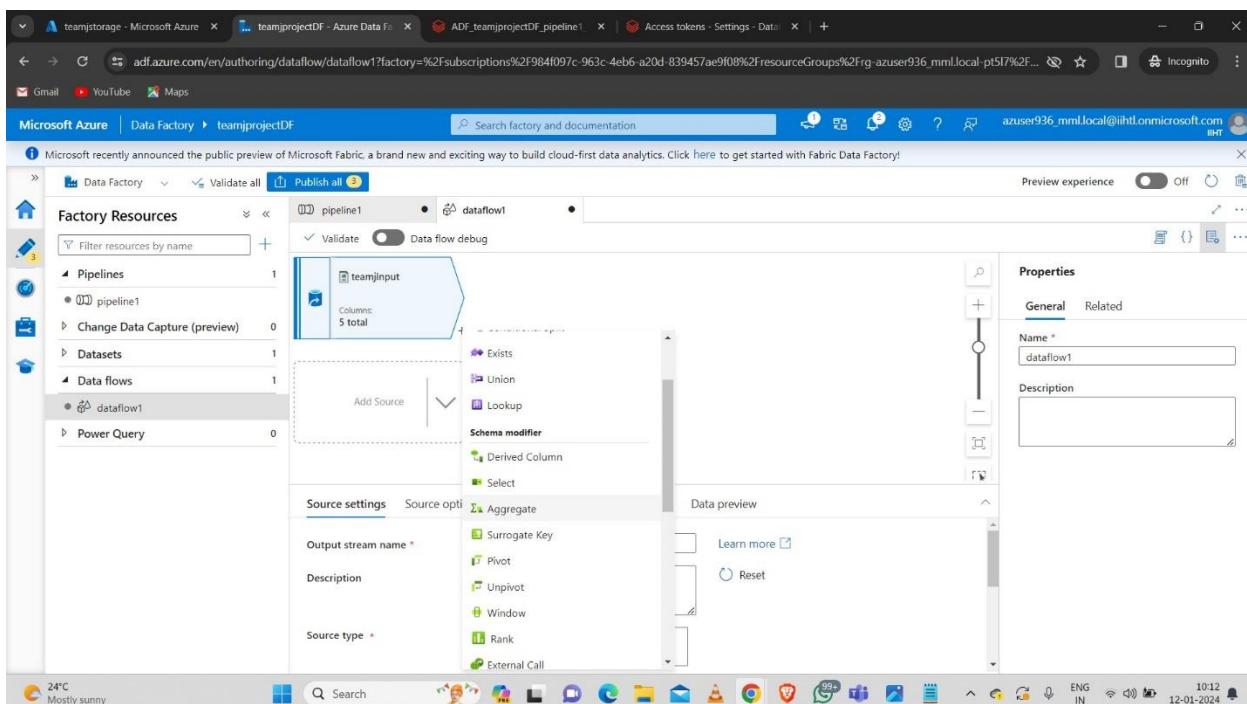
- Create the new linked service.



- Give the path of the source file that we uploaded in the storage account which is “project.csv”.



Step-3: Now add the aggregate schema modifier to the teamjinput.

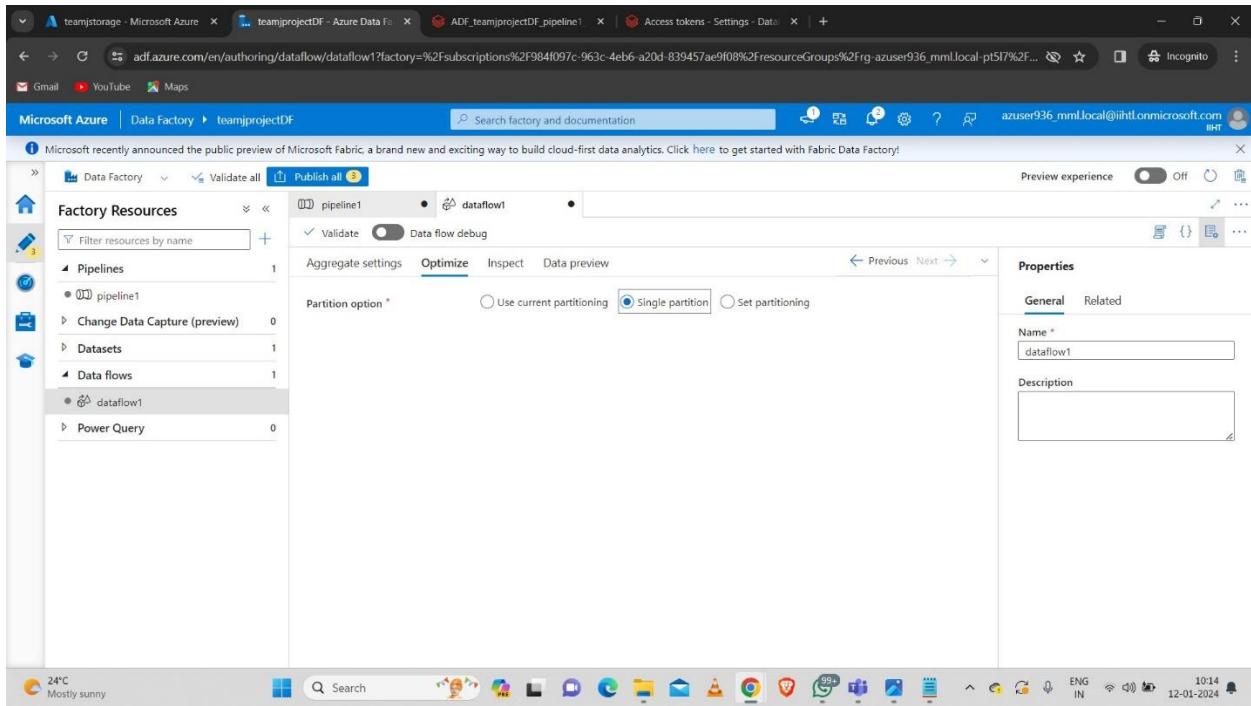


- Give the necessary data to perform the aggregation on the data.

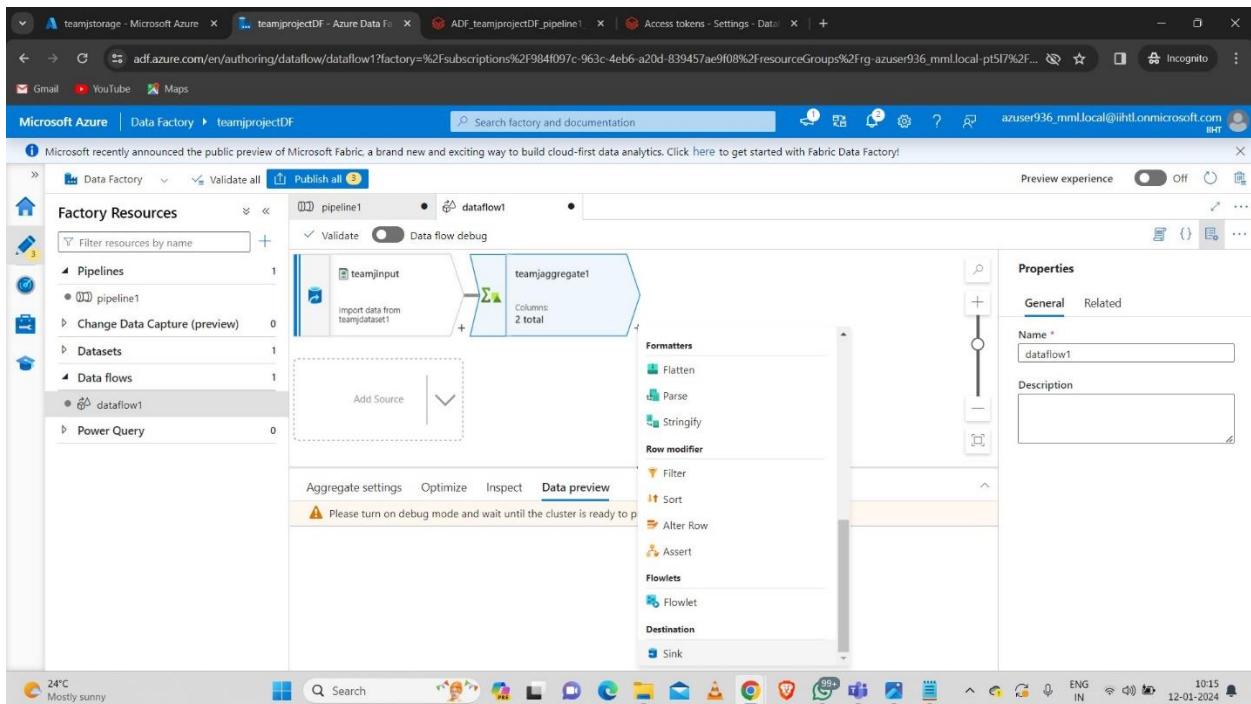
The screenshot shows the Microsoft Azure Data Factory Data Flow blade. On the left, the 'Factory Resources' sidebar lists 'Pipelines' (pipeline1), 'Datasets' (1), 'Data flows' (1, dataflow1), and 'Power Query' (0). The main area displays the 'dataflow1' configuration under the 'Aggregate settings' tab. The 'Incoming stream' is set to 'teamjinput'. In the 'Columns' section, 'dept id' is listed with 'Name as' 'dept id'. The 'Properties' panel on the right shows the 'Name' as 'dataflow1'.

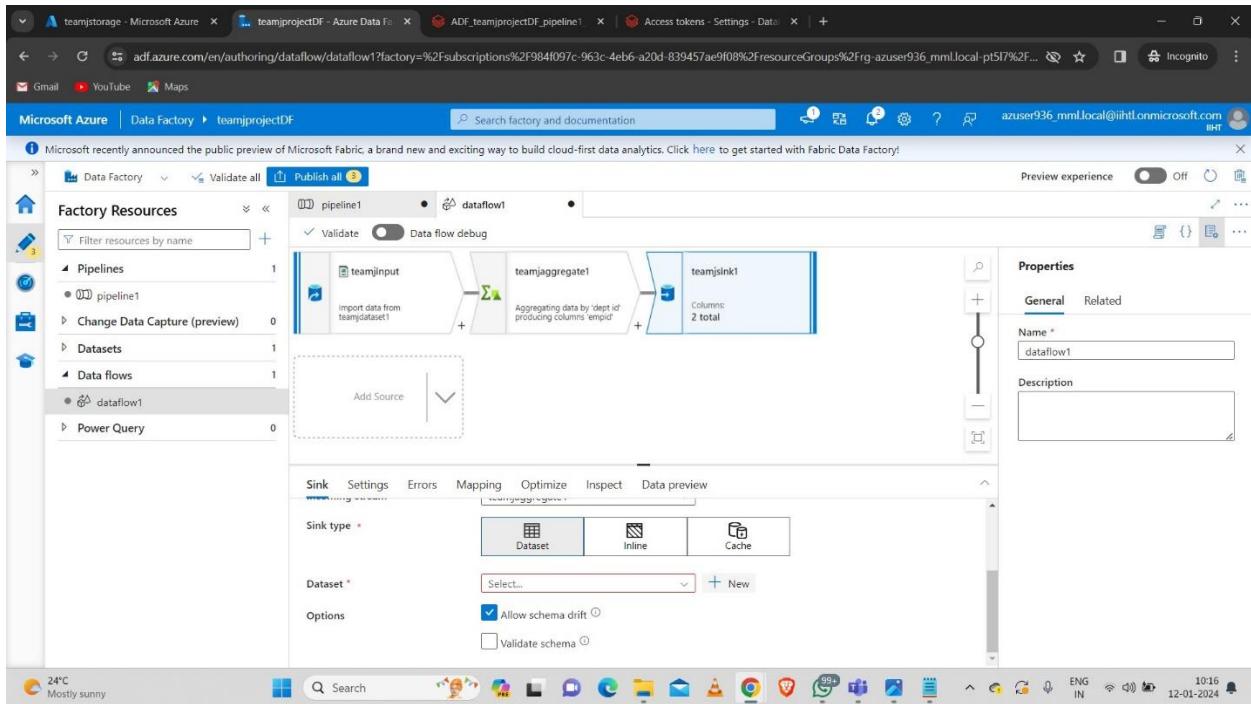
- Here we are performing the “GroupBy” operation as groupby department by aggregating as per count(emplid).

The screenshot shows the Microsoft Azure Data Factory Data Flow blade. The configuration is identical to the previous one, but now the 'Grouped by' section shows 'dept id'. In the 'Column' section, 'emplid' is selected with the expression 'count(emplid)'.

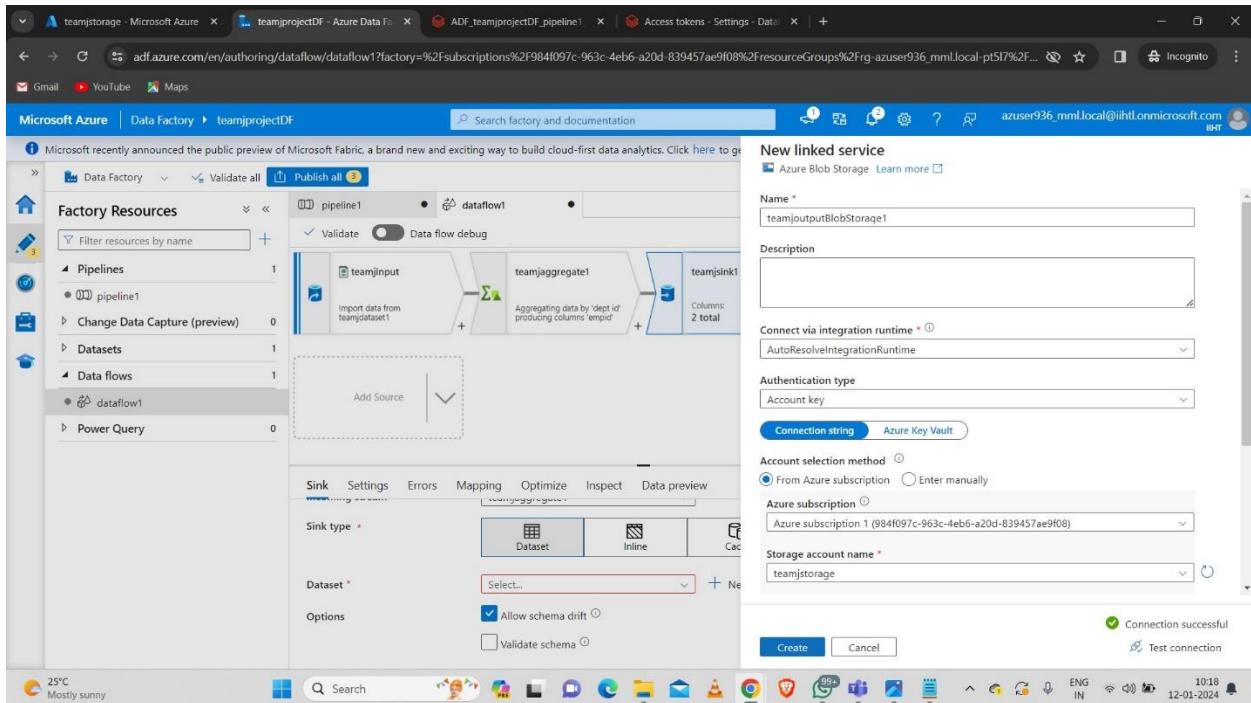


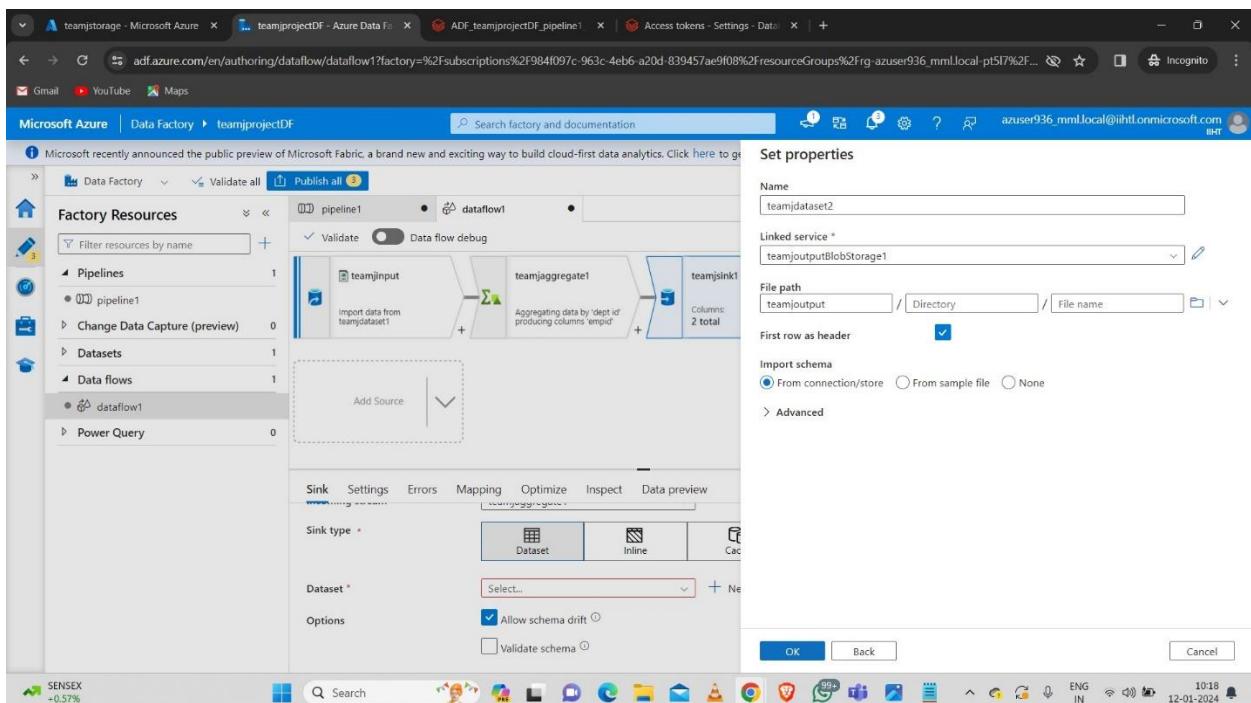
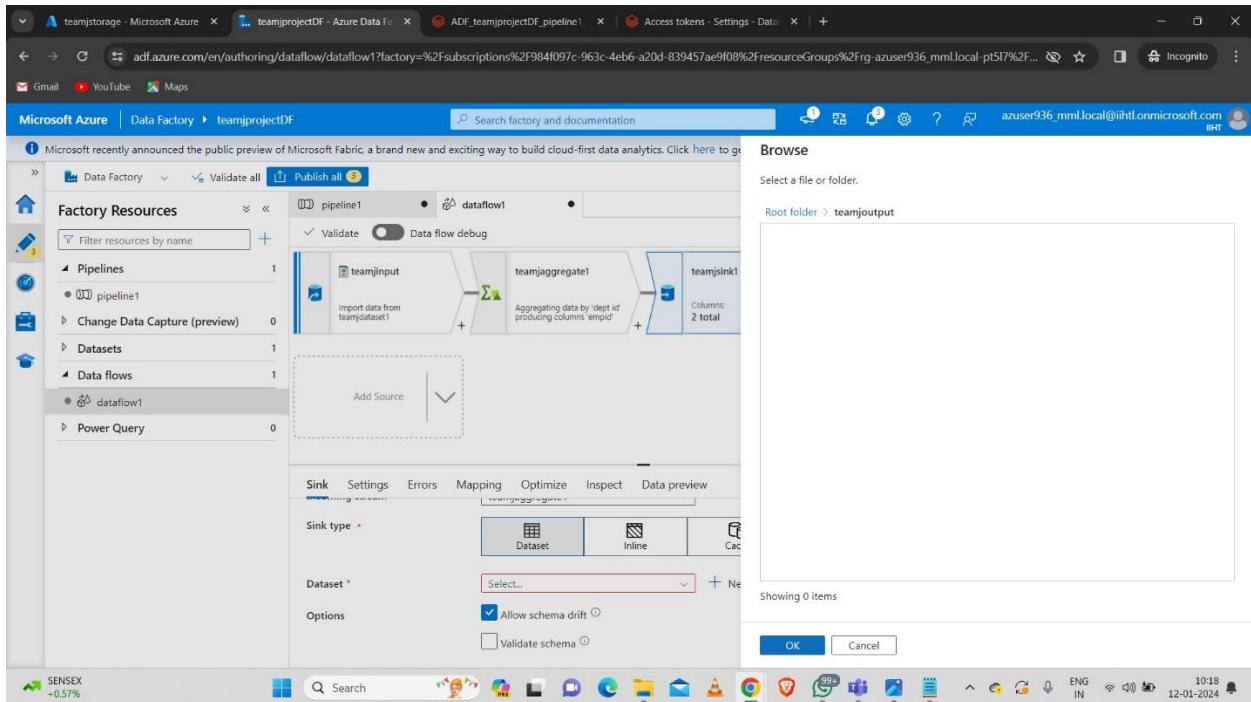
Step-4: Now we sink the aggregated data to another empty container.



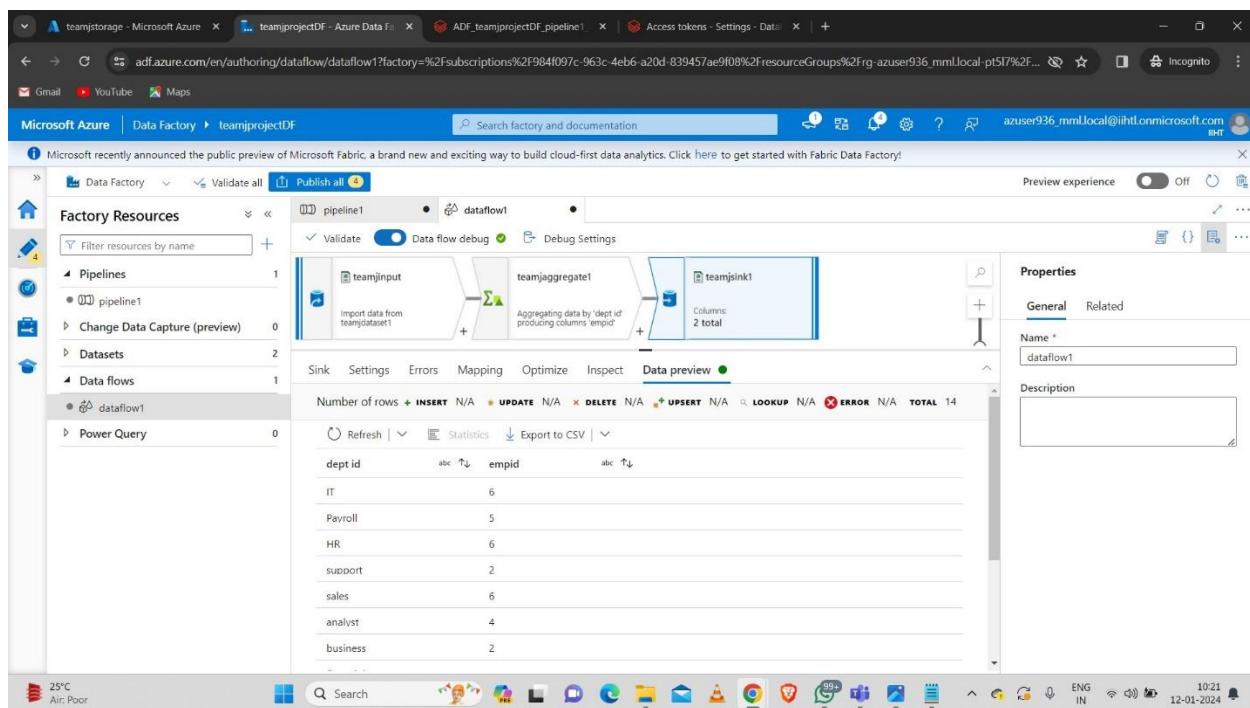
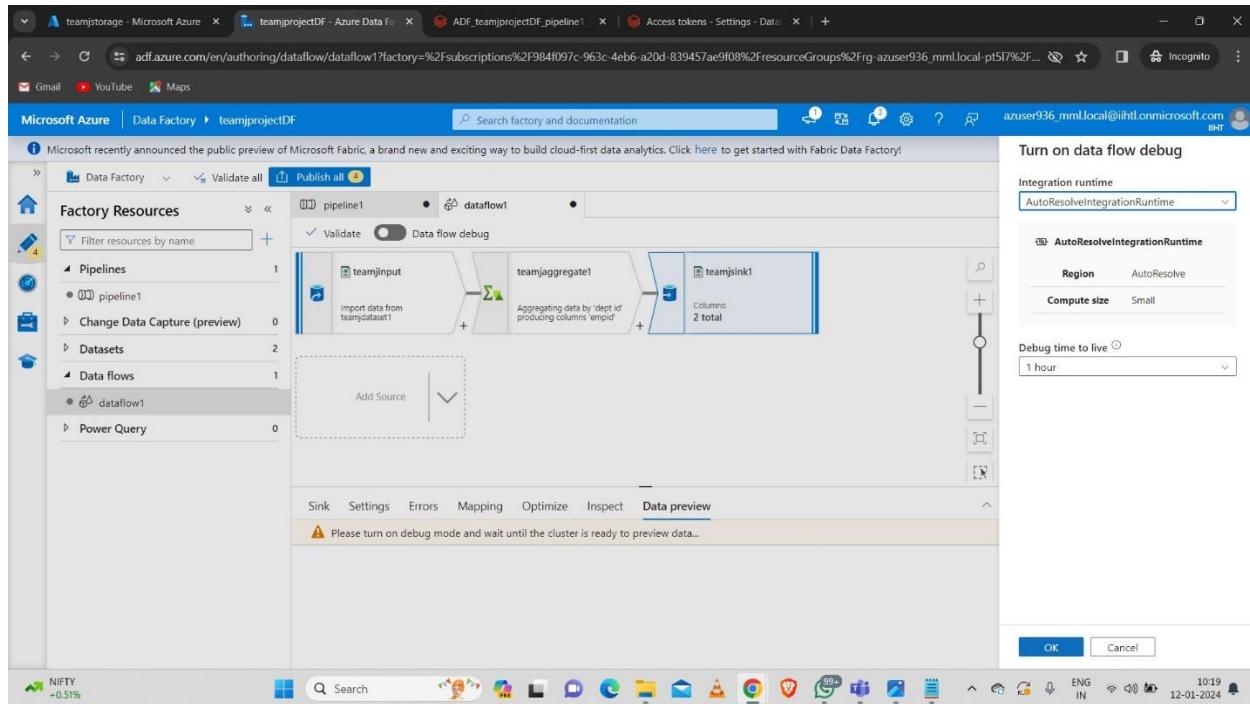


- Now create a new Dataset and link the dataset with output empty container.

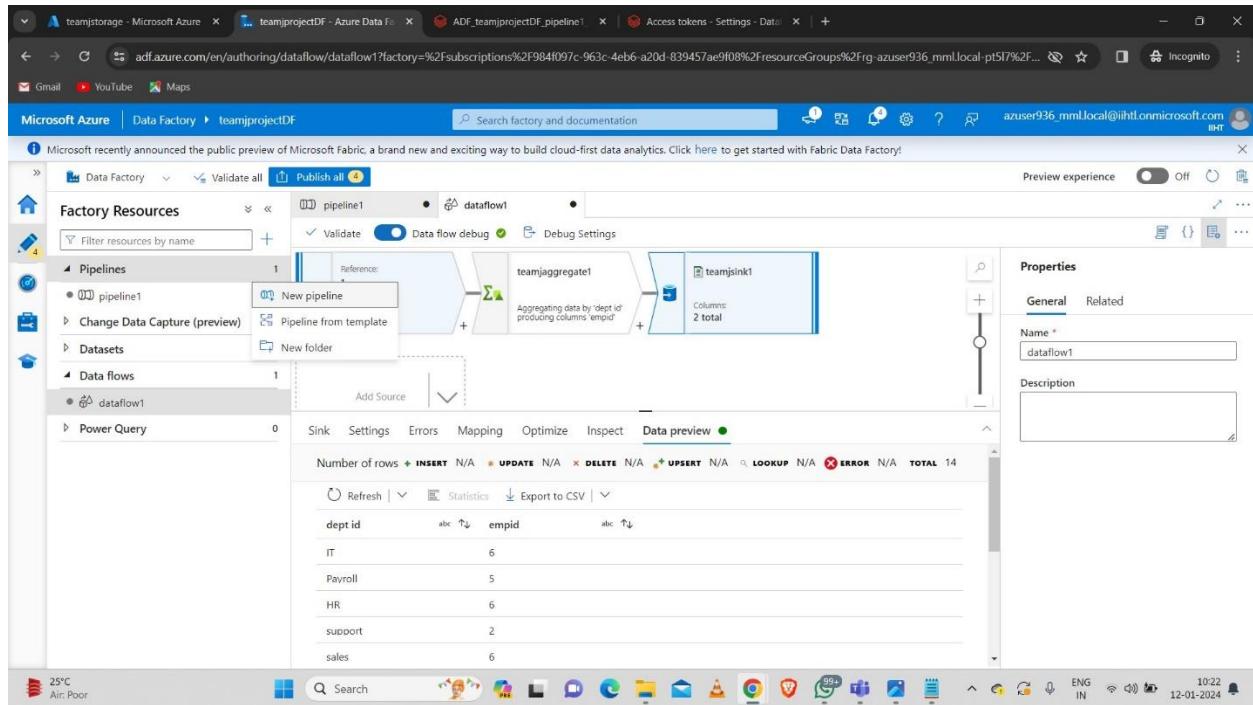




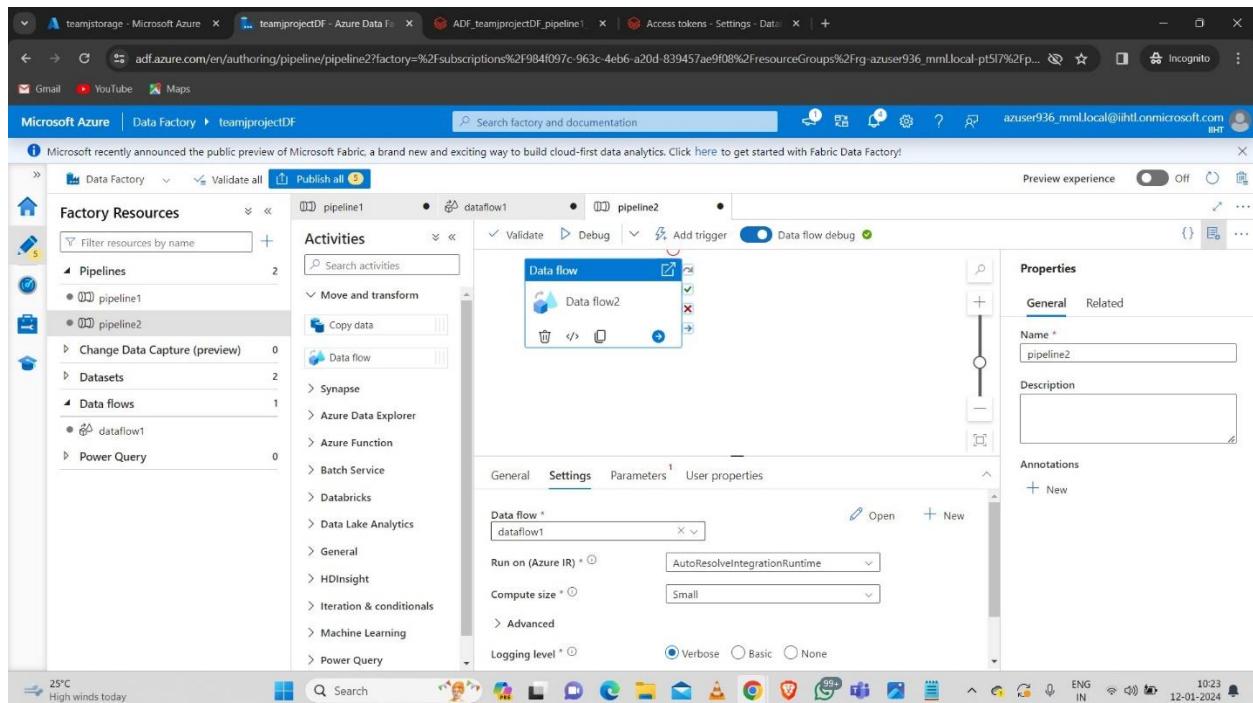
- Enable the Dataflow debug to preview the data as follows.



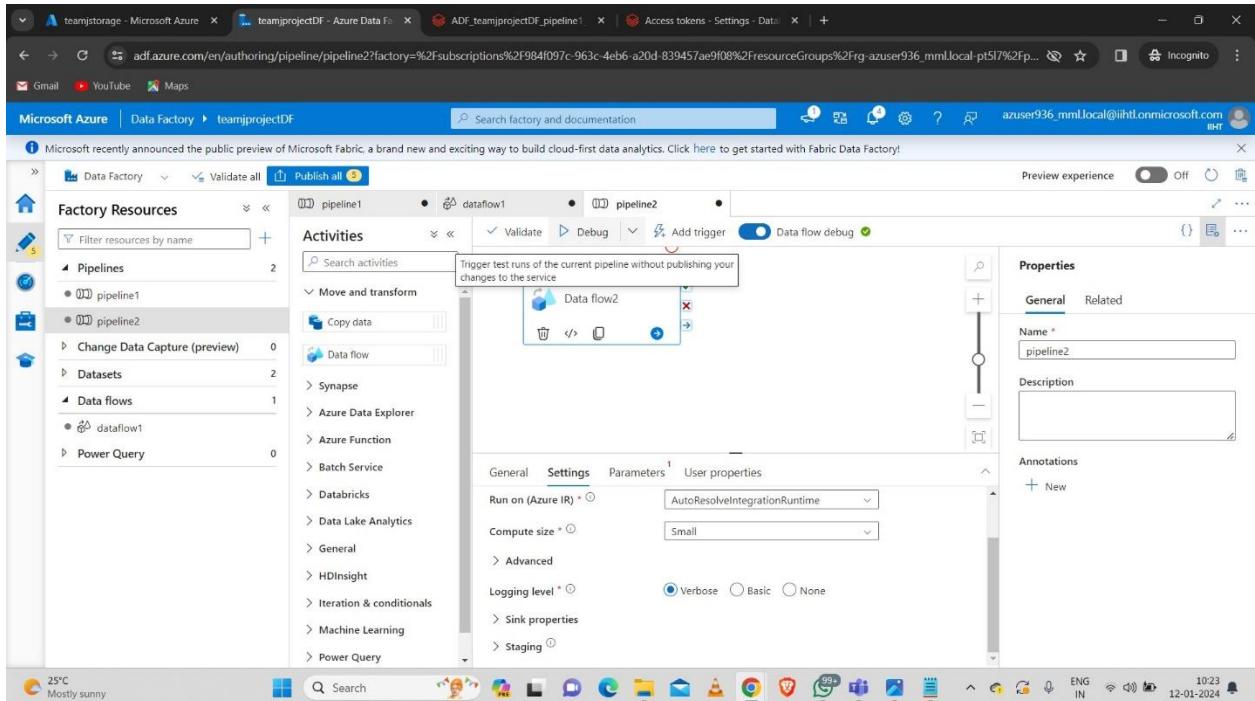
Step-5: Now create a new pipeline.



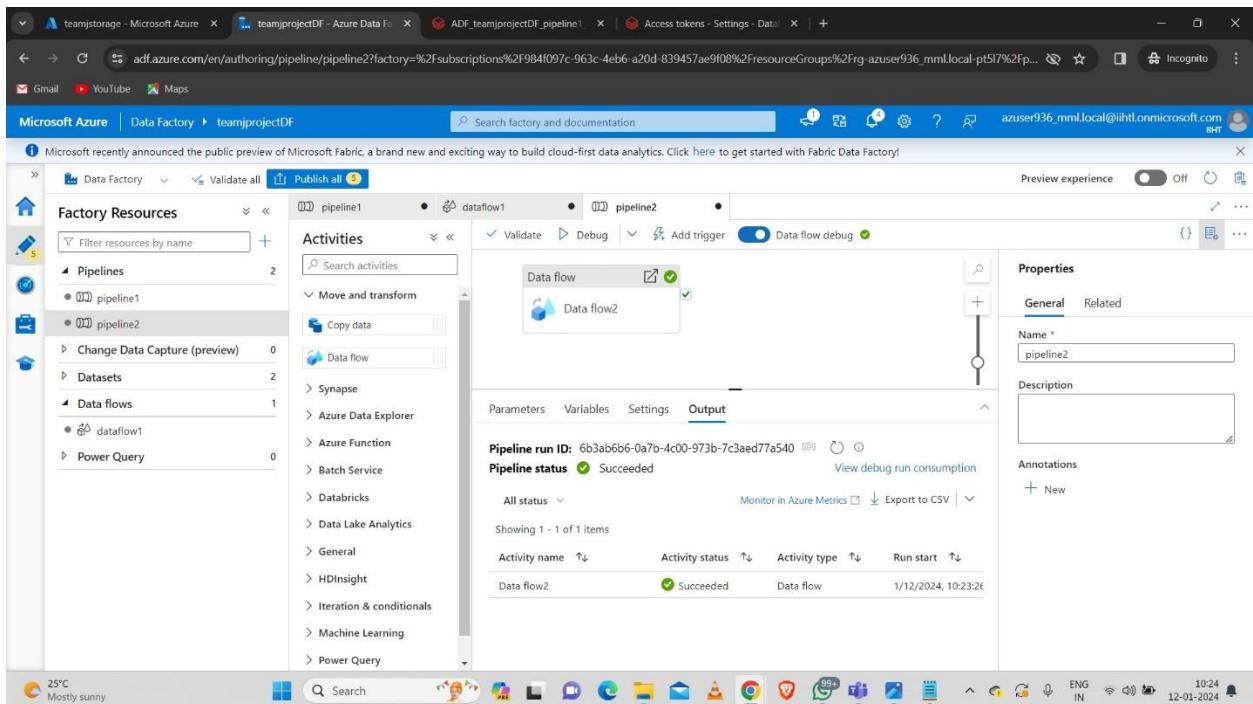
- Drag the data flow activity and give the path of data flow as “Dataflow-1” which we created before.



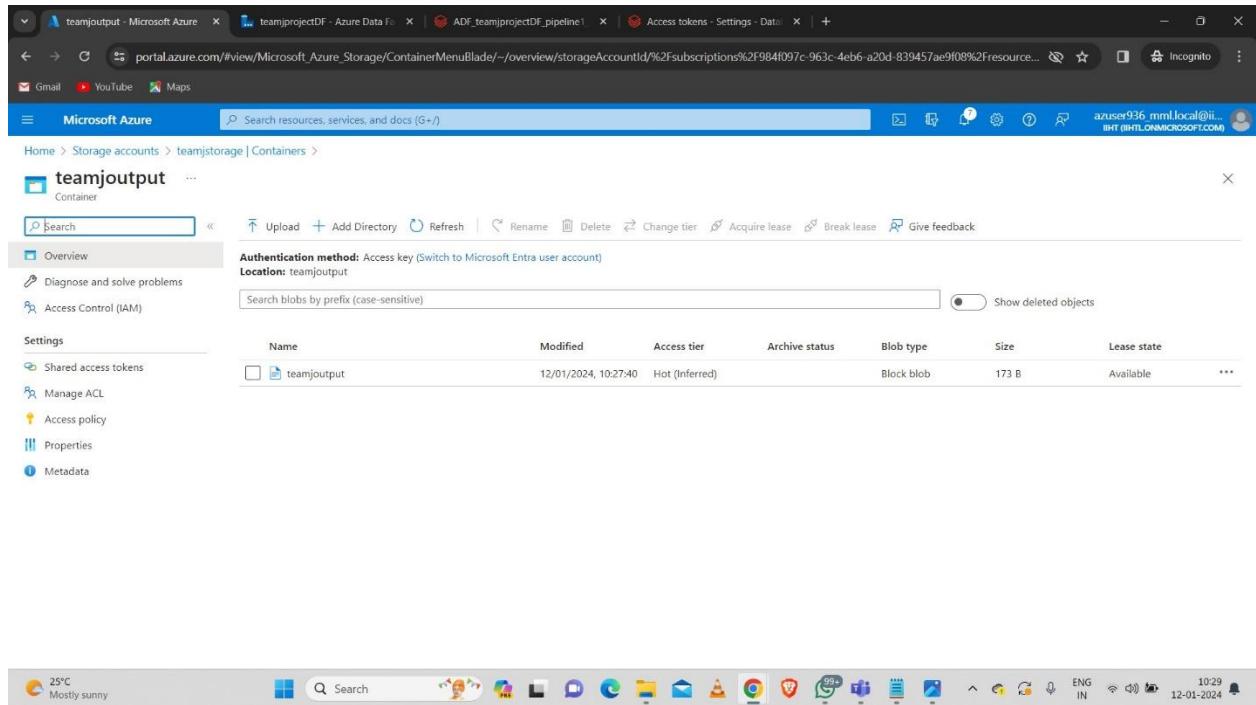
➤ Debug the pipeline.



➤ Now we can see that the debug is succeeded.



Hence the aggregated file is successfully copied using Data flow activity.



The screenshot shows the Microsoft Azure Storage Explorer interface. The left sidebar lists 'Overview', 'Diagnose and solve problems', 'Access Control (IAM)', and 'Settings' (Shared access tokens, Manage ACL, Access policy, Properties, Metadata). The main area displays the 'teamjoutput' container details. It shows the authentication method as 'Access key' and the location as 'teamjoutput'. A search bar at the top right allows searching by blob prefix. Below is a table with one row:

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
teamjoutput	12/01/2024, 10:27:40	Hot (Inferred)		Block blob	173 B	Available

The taskbar at the bottom shows various pinned icons and system status indicators.

Conclusion: In conclusion, implementing a daily data aggregation pipeline using Azure Data Factory and Azure Databricks offers a robust and scalable solution for handling raw data and deriving meaningful insights on a daily basis. This integration provides a seamless workflow that combines the data movement capabilities of Azure Data Factory with the powerful data processing and analytics features of Azure Databricks.