ADF – Azure Data Factory

It's cloud-based ETL and data integration service provided by Microsoft azure.

What is Azure Data Factory?

- **Azure Data Factory is a cloud-based data integration service provided by the Microsoft azure.**
- **It helps us to manage, schedule and create data pipeline that move and transform data across multiple source and destination.**

Azure Data Factory is a cloud-based data integration service designed to orchestrate (arrange or order) and automate data workflows.

It is used to collect, transform, and deliver data, ensuring that insights are readily accessible for analytics and decision-making.

With its scalable and serverless architecture, ADF can handle workflows of any size—from simple data migrations to complex data transformation pipelines.

ADF stands out for its user-friendly, no-code interface, which allows both technical and non-technical users to build data pipeline easily

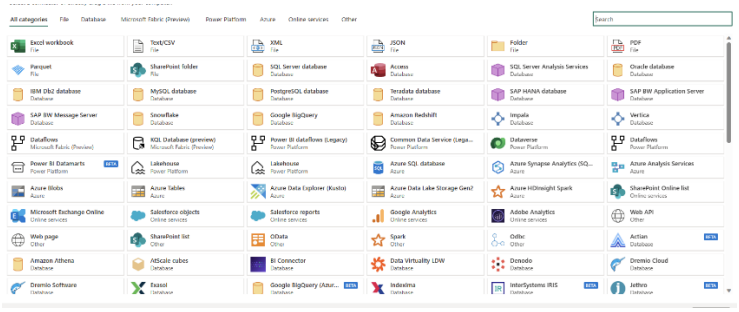
(Data integration is the process of combining data from different sources and providing a unified view of it.)

(Azure Data Factory is a cloud ETL and orchestration service that lets you create, schedule, and monitor data pipelines to move and transform data across on-prem and cloud systems without managing servers).

Features of ADF:

1. Data integration

Azure Data Factory supports integration with over 90 data sources, including cloud-based and on-premises systems. It includes support for SQL databases, NoSQL systems, REST APIs, and file-based data sources, allowing you to unify data workflows regardless of the source or format.

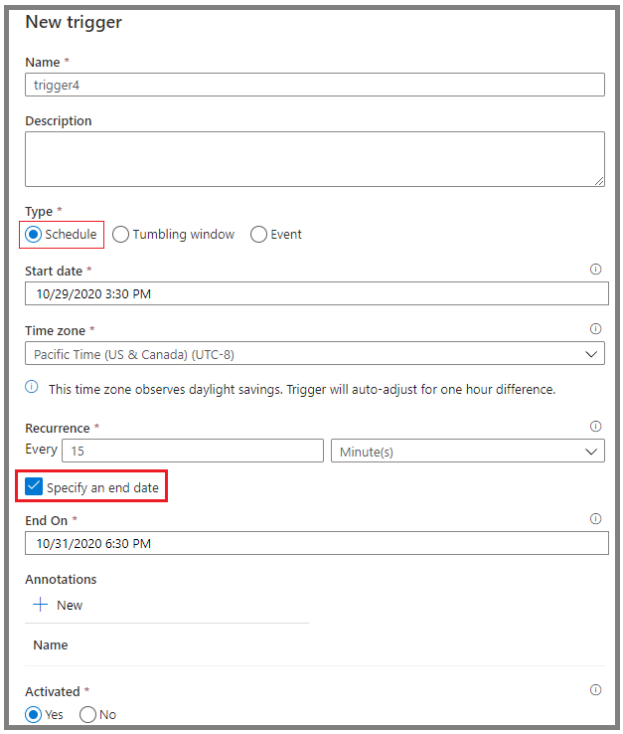


2. No-code pipeline authoring

ADF's drag-and-drop interface simplifies how users create data pipelines. With prebuilt templates, guided configuration wizards, and an intuitive visual editor, even users with no coding expertise can design comprehensive end-to-end workflows.

3. Scheduling

Azure Data Factory's scheduling tools offer workflow automation. Users can set up triggers based on specific conditions, such as a file's arrival in cloud storage or scheduled time intervals. These scheduling options eliminate the need for manual interventions and ensure workflows are executed consistently and reliably.



[Azure Data Factory: A Complete Guide for Beginners | DataCamp]

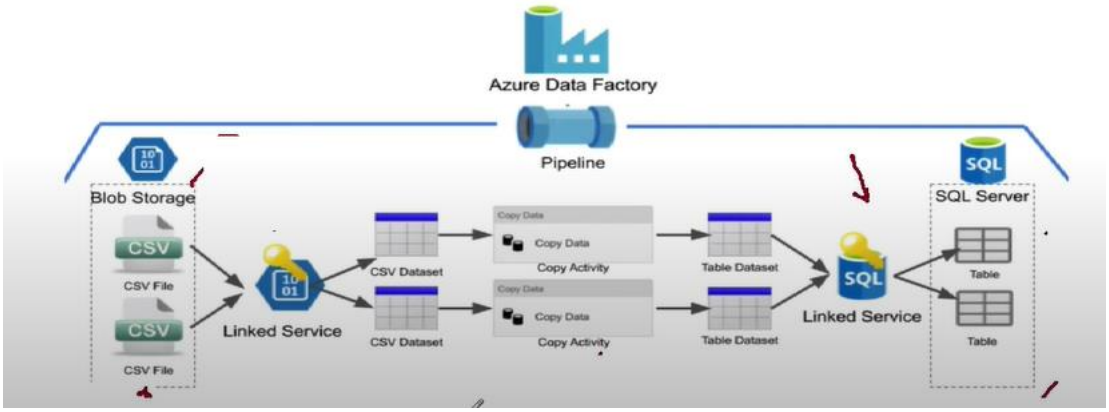
Features:

1. Data movement – copy data from on-premises system, SaaS apps, and cloud storages (e.g., SQL server, Oracle, Blob Storage, Data Lake).
2. Data transformation – supports:
   Build-in transformation with Mapping Data Flows (low-code / no-code)
   Running compute service like databricks, synapse or custom code.
3. Orchestration (order) & Monitoring – manage workflows, schedule pipelines, and monitor execution visually.
4. Integration runtime (IR) – provides the compute infrastructure for data movement and transformation.
5. Serverless – you don't need to manage servers; you pay only for the pipelines and runtime you use.

Common use cases:

- Migrating data from on-premises databases to the cloud.
- Building ETL/ELT pipelines for data warehousing.
- Moving data into azure data lake or synapse analytics for reporting.
- Automating data workflows for AI/ML model training.

- Real-time data ingestion with event triggers.



Functions in Azure Data Factory (ADF)

1. String Functions

Used to work with text.

Function	Description	Example
concat(a,b,...)	Joins multiple strings.	@concat('file_', pipeline().RunId, '.csv') → "file_1234.csv"
substring(string, start, length)	Extracts part of a string.	@substring('DataFactory', 0, 4) → "Data"
length(string)	Returns length of string.	@length('ADF') → 3
toLower(string)	Lowercase.	@toLower('ADF') → "adf"
toUpper(string)	Uppercase.	@toUpper('adf') → "ADF"
trim(string)	Removes spaces.	@trim(' hello ') → "hello"
replace(string, old, new)	Replaces text.	@replace('data_factory','_','') → "datafactory"
split(string, delimiter)	Splits into array.	@split('a,b,c',',') → ['a','b','c']
guid()	Generates unique ID.	@guid() → "7e8d...a2b"

2. Collection Functions (Arrays & Objects)

Work with arrays, objects, JSON.

Function	Description	Example
contains(collection, value)	Checks if value exists.	@contains(['a','b'],'b') → true

empty(collection)	Checks if empty.	@empty([]) → true
first(array)	First element.	@first(['x','y']) → "x"
last(array)	Last element.	@last(['x','y']) → "y"
intersection(arr1,arr2)	Common values.	@intersection([1,2],[2,3]) → [2]
union(arr1,arr2)	Combine unique values.	@union([1,2],[2,3]) → [1,2,3]
join(array, delimiter)	Join into string.	@join(['a','b'], '-') → "a-b"
length(array)	Array length.	@length([1,2,3]) → 3
addProperty(obj,key,value)	Adds property.	@addProperty(variables('obj'),'newKey','val')
removeProperty(obj,key)	Removes property.	@removeProperty(obj,'key')

3.Logical Functions

Used in conditions (If, Switch, Filter, etc.).

Function	Description	Example
and(a,b,...)	True if all true.	@and(equals(1,1),greater(2,1)) → true
or(a,b,...)	True if at least one true.	@or(equals(1,2),equals(2,2)) → true
not(expr)	Negates.	@not(equals(1,1)) → false
equals(a,b)	Checks equality.	@equals(5,5) → true
greater(a,b)	Greater than.	@greater(5,3) → true
less(a,b)	Less than.	@less(2,3) → true
if(cond, trueVal, falseVal)	Inline if.	@if(equals(2,2),'Yes','No') → "Yes"
coalesce(a,b,...)	First non-null value.	@coalesce(null,'ADF','DF') → "ADF"

4. Conversion Functions

Convert between data types.

Function	**Description**	**Example**
string(value)	Convert to string.	@string(123) → "123"
int(value)	Convert to integer.	@int('25') → 25
float(value)	Convert to float.	@float('12.34') → 12.34
bool(value)	Convert to boolean.	@bool('true') → true

json(string)	Convert to JSON.	@json('{"name":"abc"}') → object
base64(string)	Encode base64.	@base64('ADF') → "QURG"
base64ToString(string)	Decode base64.	@base64ToString('QURG') → "ADF"

5.Date & Time Functions

Handle datetime.

Function	**Description**	**Example**
utcNow()	**Current UTC datetime.**	**@utcNow() → "2025-09-02T06:30:00Z"**
addDays(datetime,n)	**Add days.**	**@addDays(utcNow(),5)**
addHours(datetime,n)	**Add hours.**	**@addHours(utcNow(),2)**
addMinutes(datetime,n)	**Add minutes.**	**@addMinutes(utcNow(),30)**
addMonths(datetime,n)	**Add months.**	**@addMonths('2025-01-01',1) → Feb 1**
addSeconds(datetime,n)	**Add seconds.**	**@addSeconds(utcNow(),10)**
subtractFromTime(datetime,n,unit)	**Subtract time.**	**@subtractFromTime(utcNow(),2,'Day')**
formatDateTime(datetime, format)	**Format datetime.**	**@formatDateTime(utcNow(),'yyyy-MM-dd')**
dayOfWeek(datetime)	**Returns day (0=Sunday).**	**@dayOfWeek('2025-09-02') → 2**
dayOfMonth(datetime)	**Day of month.**	**@dayOfMonth('2025-09-02') → 2**

6. Math Functions

Used for calculations.

Function	**Description**	**Example**
add(a,b)	Add.	@add(5,3) → 8
sub(a,b)	Subtract.	@sub(5,3) → 2
mul(a,b)	Multiply.	@mul(2,3) → 6
div(a,b)	Divide.	@div(6,3) → 2
mod(a,b)	Modulus.	@mod(10,3) → 1
min(a,b,...)	Minimum.	@min(3,5,1) → 1

max(a,b,...)	Maximum.	@max(3,5,1) → 5
rand(min,max)	Random number.	@rand(1,100) → e.g., 42
ceiling(value)	Round up.	@ceiling(2.3) → 3
floor(value)	Round down.	@floor(2.9) → 2
round(value,decimals)	Round with decimals.	@round(2.567,2) → 2.57

7. Pipeline Functions

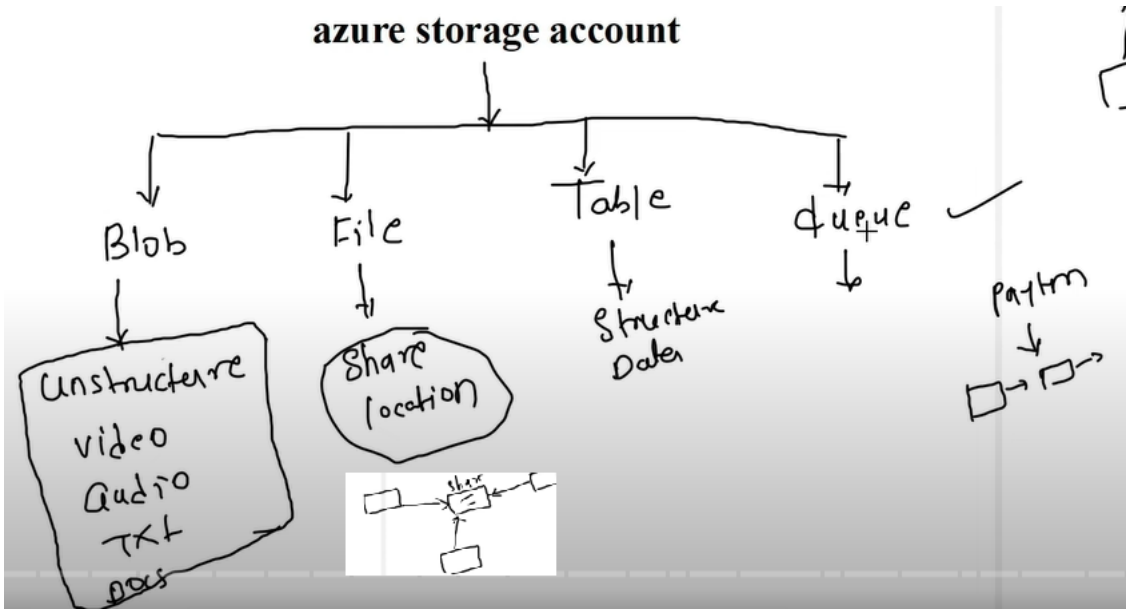
Access metadata inside pipelines.

Function	Description	Example
pipeline().DataFactory	Returns DF name.	"MyADF"
pipeline().Pipeline	Returns pipeline name.	"MyPipeline"
pipeline().RunId	Unique pipeline run ID.	"12345-abc"
pipeline().TriggerTime	Triggered datetime.	"2025-09-02T06:30:00Z"
activity('ActivityName'). output	Output of activity.	@activity('Lookup1').output.firstRow.id
trigger().outputs	Trigger values.	From event/schedule trigger
parameters('paramName')	Pipeline parameter.	@parameters('inputFile')
variables('varName')	Pipeline variable.	@variables('counter')

Activity dependencies

Condition	Meaning	Typical Use Case
On Success	Next activity runs only if the previous activity succeeds (status = Succeeded).	Run a data transformation only if the copy activity was successful.
On Failure	Next activity runs only if the previous activity fails (status = Failed).	Send an alert email or trigger a rollback when something fails.
On Skip	Next activity runs if the previous activity was skipped (not executed at all, e.g., due to an If Condition or dependency rule).	Log or handle skipped branches in complex pipelines.

Condition	Meaning	Typical Use Case
On Completion (the “another” one)	Next activity runs regardless of success, failure, or skip.	Always clean up resources, update a log table, or send a “job finished” notification.



ADLS stands for **Azure Data Lake Storage**.

- It’s a **cloud-based storage service** from Azure, specially designed for **big data analytics and processing**.
- Think of it like **Azure Blob Storage**, but **optimized for analytics, hierarchical structure, and security**.
- Is a scalable and secure data lake service designed to store and analyze large volumes of structured and unstructured data.

Types of ADLS

Earlier there were two generations, but now everything is under **ADLS Gen2** (built on top of Blob storage).

1. **ADLS Gen1** – older, separate service (now retired).
2. **ADLS Gen2** – combines the best of Blob storage + Data Lake features.

   ADLS Gen2 = Azure Blob Storage + ADLS Gen1

1. Hadoop compatible:
   ADLS Gen1 is based on Hadoop Distributed File System (HDFS),
   ADLS Gen2 is built on top of Azure Blob Storage.
2. Namespace and hierarchy:
   ADLS Gen1 has a flat namespace.

ADLS Gen2 has a hierarchical namespace. ADLS Gen2 uses directories to organize data, which provides a more efficient way to manage large amounts of data.

3. Security:
   ADLS Gen1 uses access control lists (ACLs) for authorization.
   ADLS Gen2 uses role-based access control (RBAC) and Azure Active Directory (AAD) authentication.
4. Data Lake Analytics:
   ADLS Gen1 integrates with Azure Data Lake Analytics, a cloud-based analytics service that can process big data using U-SQL, a language that combines SQL and C#.
   ADLS Gen2 also supports Data Lake Analytics, but it uses Apache Spark instead of U-SQL.

ADLS Gen 2 supports access methods such as REST APIs, .NET, Java, and Hadoop Distributed File System (HDFS).

ADF is like a pipeline tool in Azure that helps us **extract data from multiple sources, transform it, and load it into target systems like Data Lake or Synapse**.
It acts as an **orchestrator** – it doesn't always process the data itself, but it makes sure the right data moves and gets processed at the right time using services like Databricks, SQL, or Synapse."

One-liner:

- **ADF = Azure's ETL + Orchestration tool for building and managing data pipelines.**

Core components of Azure Data Factory

1. **Pipeline** – the workflow or flowchart or container that tells ADF how to move and transform data.
   In ADF, that whole workflow (Extract → Transform → Load → Report) = **Pipeline**.
2. **Activity** – the individual steps [tasks inside pipeline] (copy, transform, etc)
3. **Data sets** – represent the data structures (e.g., SQL table, CSV file, blob)
4. **Linked services** – connections to external resources (like storage accounts, APIs)
5. **Triggers** – define when the pipeline should run (schedule, event or manual).

Example Scenario

Let's say pipeline needs to:

1. Copy data from an **on-prem SQL database** to an **Azure Data Lake**.
2. Transform the data using a **Databricks notebook**.
3. Load the cleaned data into **Azure Synapse Analytics** for reporting.

In ADF, this would look like:

- **Pipeline** = *Daily Sales Data Pipeline*
- **Activities**:
  - Copy activity (SQL → Data Lake)
  - Notebook activity (Databricks transformation)
  - Copy activity (Data Lake → Synapse)
- **Trigger**: Schedule at 1 AM daily

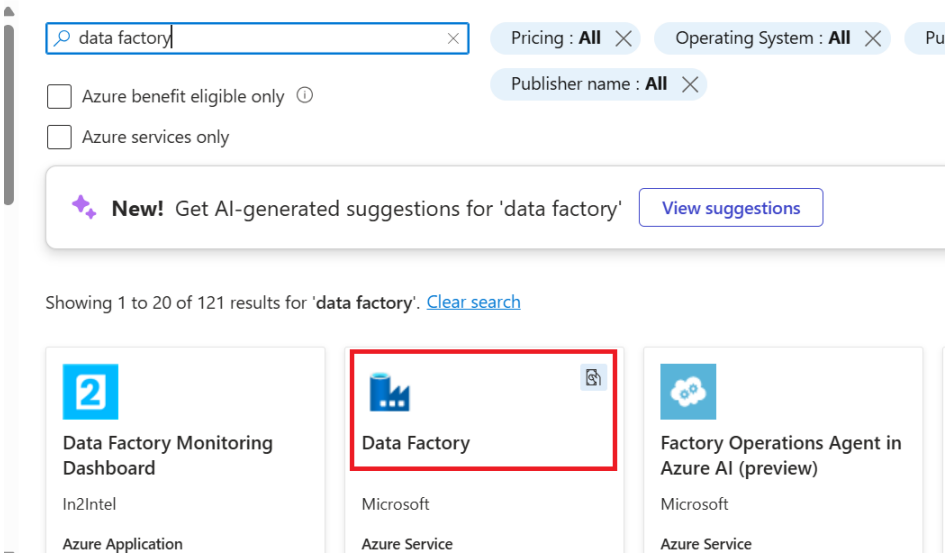
Pipeline

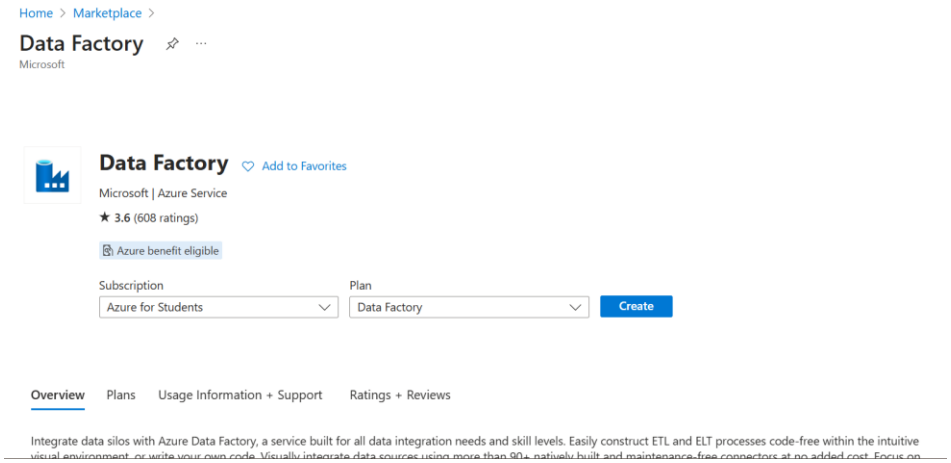
A data factory might have one or more pipelines. A pipeline is a logical grouping of activities that performs a unit of work. [we are only able to create 40 pipeline at max in single Data Factory]

Activity

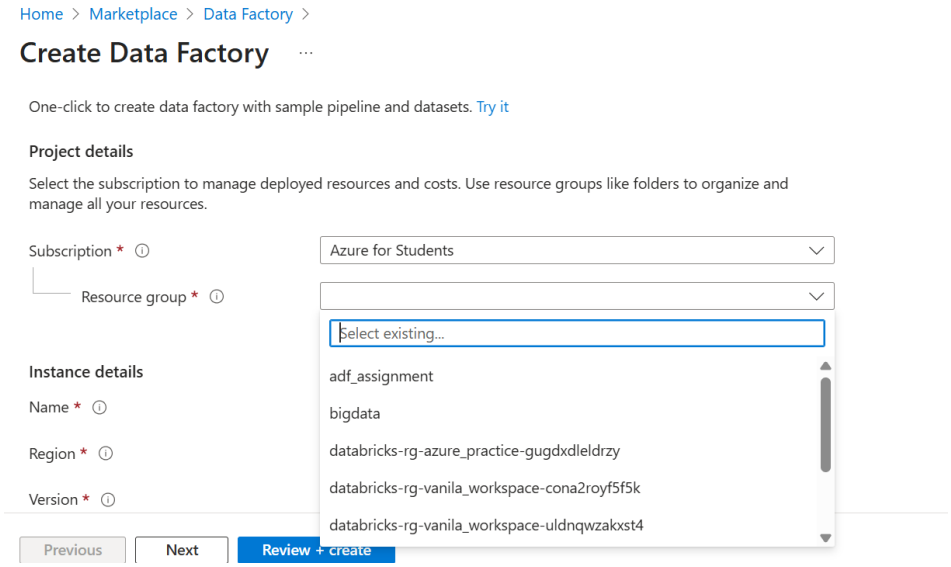
Activities represent a processing step in a pipeline. (copy activity to copy data store)

Search for “data factory”

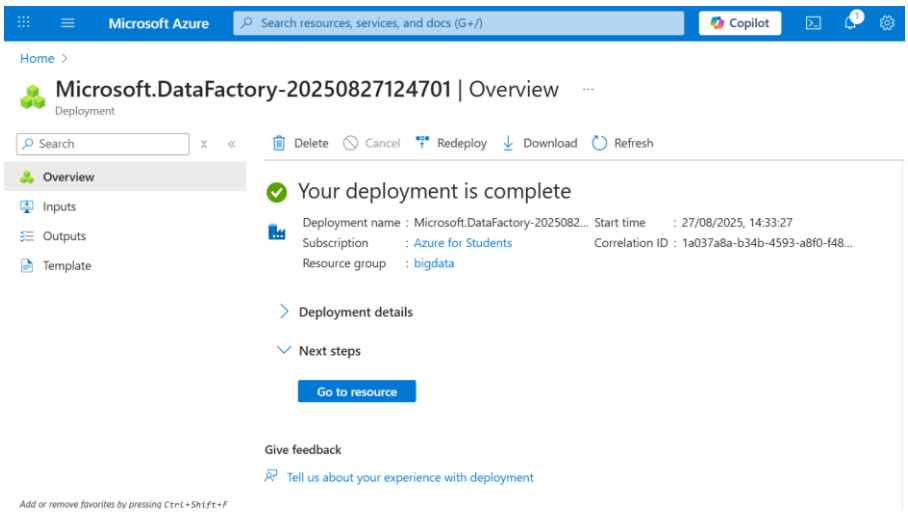


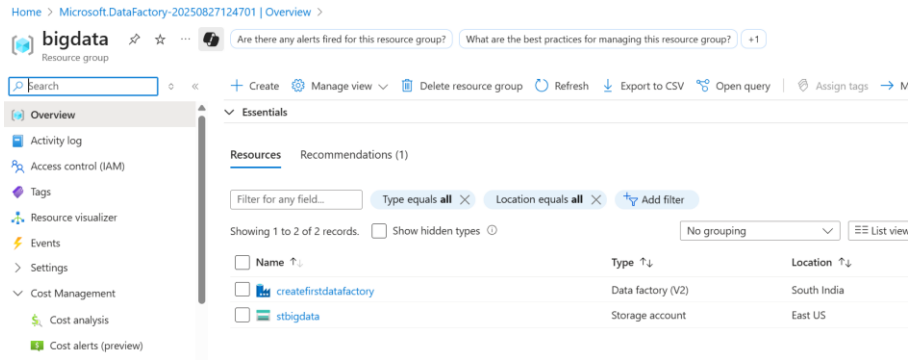


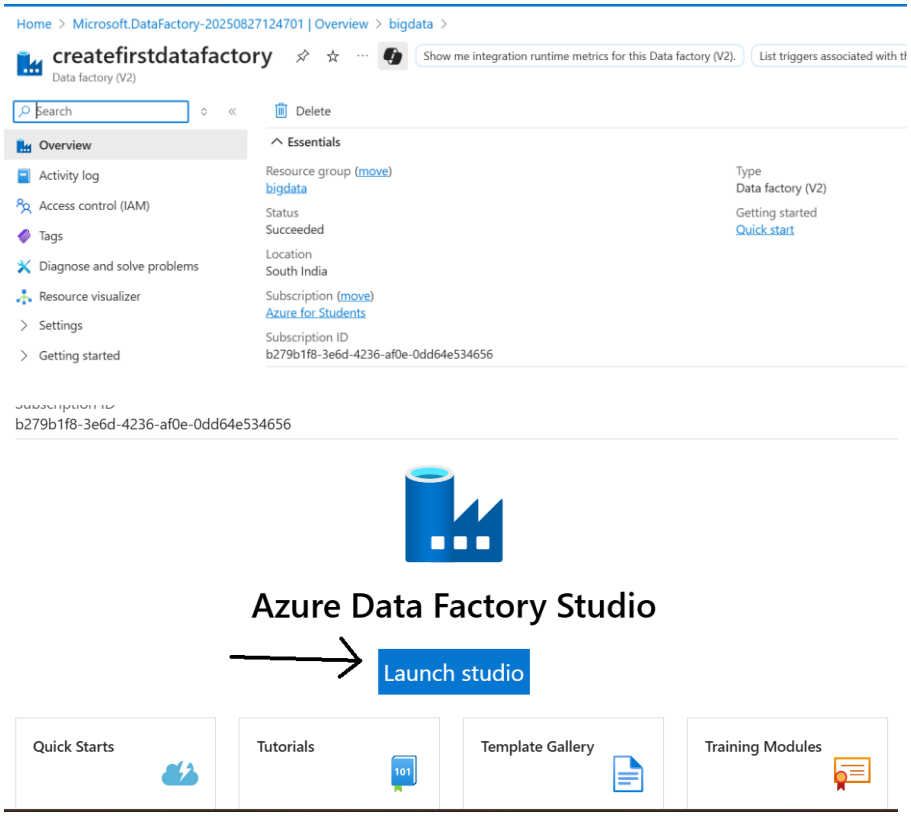
Select resource group



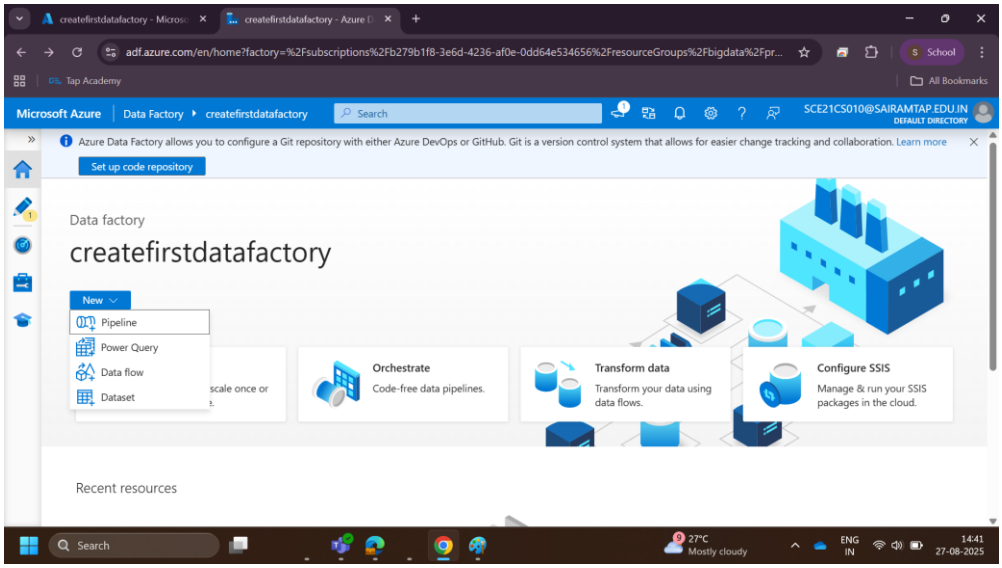
Data factory is created

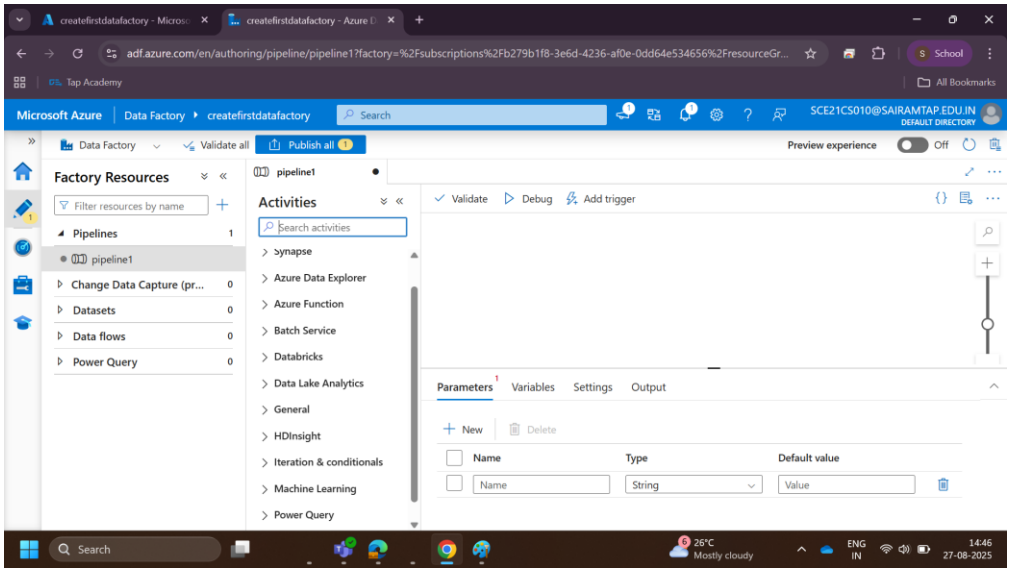



Home > Microsoft.DataFactory-20250827124701 | Overview >
bigdata
Resource group
Are there any alerts fired for this resource group?
What are the best practices for managing this resource group?
Search
Create
Manage view
Delete resource group
Refresh
Export to CSV
Open query
Overview
Activity log
Access control (IAM)
Tags
Resource visualizer
Events
Settings
Cost Management
Cost analysis
Cost alerts (preview)
Essentials
Resources
Recommendations (1)
Filter for any field...
Type equals all
Location equals all
Add filter
Showing 1 to 2 of 2 records.
Show hidden types
No grouping
Name
Type
Location
createfirstdatafactory
Data factory (V2)
South India
stbigdata
Storage account
East US

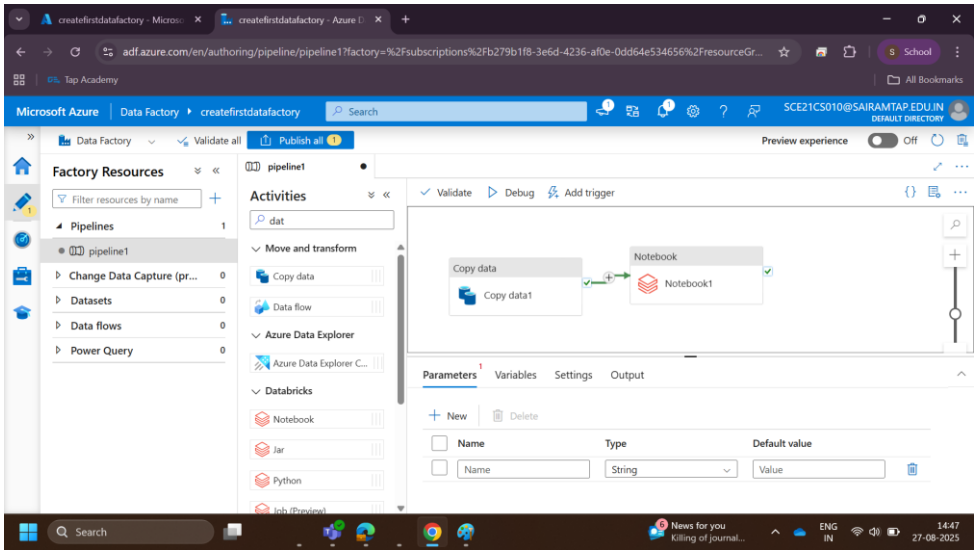

Home > Microsoft.DataFactory-20250827124701 | Overview > bigdata >
createfirstdatafactory
Data factory (V2)
Show me integration runtime metrics for this Data factory (V2).
List triggers associated with th
Search
Delete
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Resource visualizer
Settings
Getting started
Essentials
Resource group (move)
bigdata
Status
Succeeded
Location
South India
Subscription (move)
Azure for Students
Subscription ID
b279b1f8-3e6d-4236-af0e-0dd64e534656
Type
Data factory (V2)
Getting started
Quick start

b279b1f8-3e6d-4236-af0e-0dd64e534656
Azure Data Factory Studio
Launch studio
Quick Starts
Tutorials
Template Gallery
Training Modules


Microsoft Azure
Data Factory
createfirstdatafactory
Search
Azure Data Factory allows you to configure a Git repository with either Azure DevOps or GitHub. Git is a version control system that allows for easier change tracking and collaboration. Learn more
Set up code repository
Data factory
createfirstdatafactory
New
Pipeline
Power Query
Data flow
Dataset
Orchestrate
Code-free data pipelines.
Transform data
Transform your data using data flows.
Configure SSIS
Manage & run your SSIS packages in the cloud.
Recent resources



Drag and drop the activity



Linked services & data sets

Linked services are much like connection strings, which define the connection information that's needed for data factory to connect to external resources.

It defines how to connect to a data source**.**

Datasets represent data structures within the data stores, which simply point to or reference the data you want to use in your activities.

Think of it as a **pointer to actual data** inside the data source.
It represents the **structure & location** of the data you want to use.

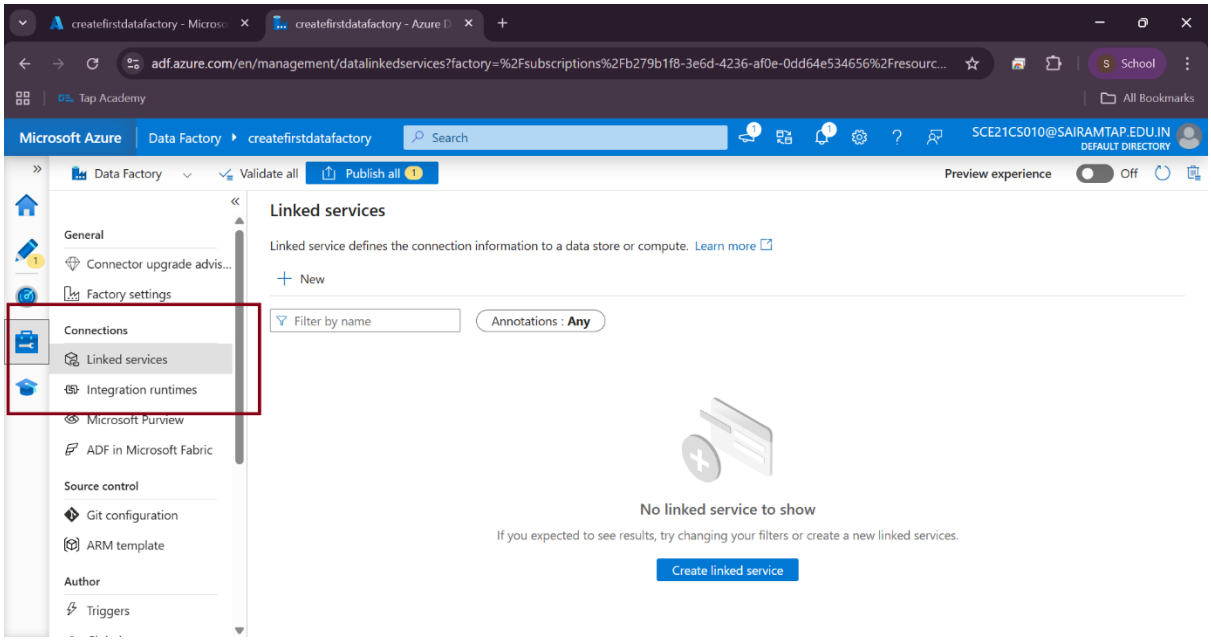
For example,

- **Linked Service** → Tells ADF how to connect.
- **Dataset** → Tells ADF what data inside that source to use.

Task: Copy customer data from SQL Database to Azure Blob.

1. **Linked Service (SQL)** → Connection details to SQL DB.
2. **Dataset (SQL)** → Points to *dbo.Customers* table.
3. **Linked Service (Blob)** → Connection details to Blob Storage.
4. **Dataset (Blob)** → Points to *customers.csv* in container *raw-data*.
5. **Copy Activity** → Moves data from Dataset (SQL) → Dataset (Blob).

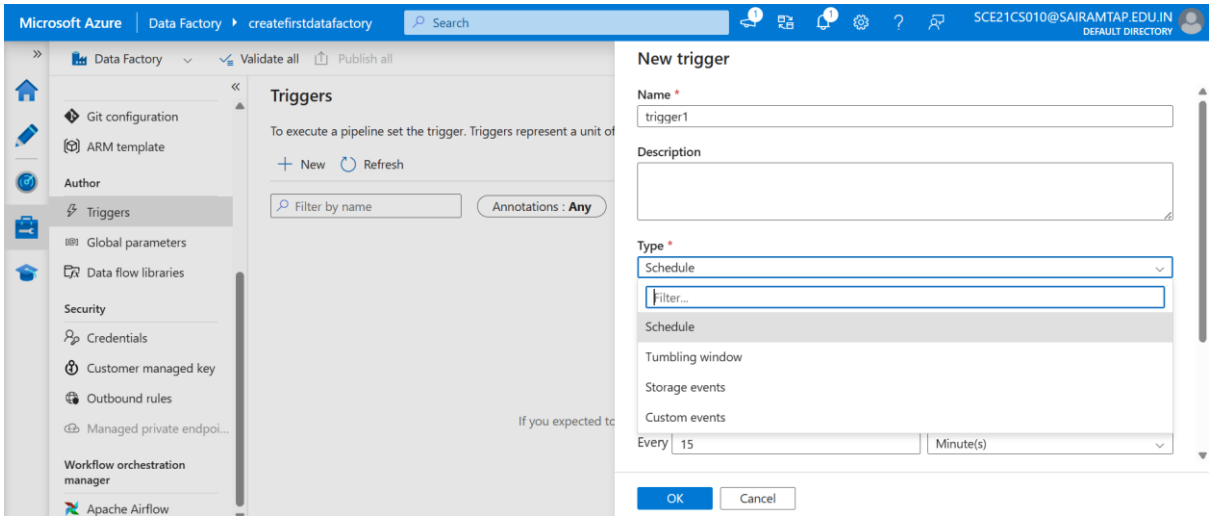
Click “New” to add the source data



Trigger

A trigger defines when and how a pipeline should run.
Pipelines don’t run automatically — you must either manually run them or use a trigger.

- Tiggers – you can execute your pipeline.
- It determines when a pipeline execution needs to be kicked off.
- Multiple triggers can kick off a single pipeline, or a single trigger can kick off multiple pipelines.
- Pipelines and triggers have a many-to-many relationship (except for the tumbling window trigger).



Types of triggers

- Schedule Trigger
- tumbling window Trigger
- Event Trigger or blob triggers
- custom events Trigger

Types of Triggers in ADF

1. **Schedule Trigger**

   - Runs a pipeline at **specific times** based on a schedule you define.
   - Example: Run every day at **9 AM**, or every **30 minutes**.
   - It is **time-based only**.
   - Doesn't care if previous runs are completed or not.
   - No dependency between runs (they are **independent**).
   - Best suited for:
     - Simple, fixed-time schedules.
     - Pipelines that do not depend on previous runs or data intervals.
   - Example: Run a data backup job at **midnight daily**.

2. **Tumbling Window Trigger**

   This trigger runs a pipeline on a periodic time interval from a specific start time.

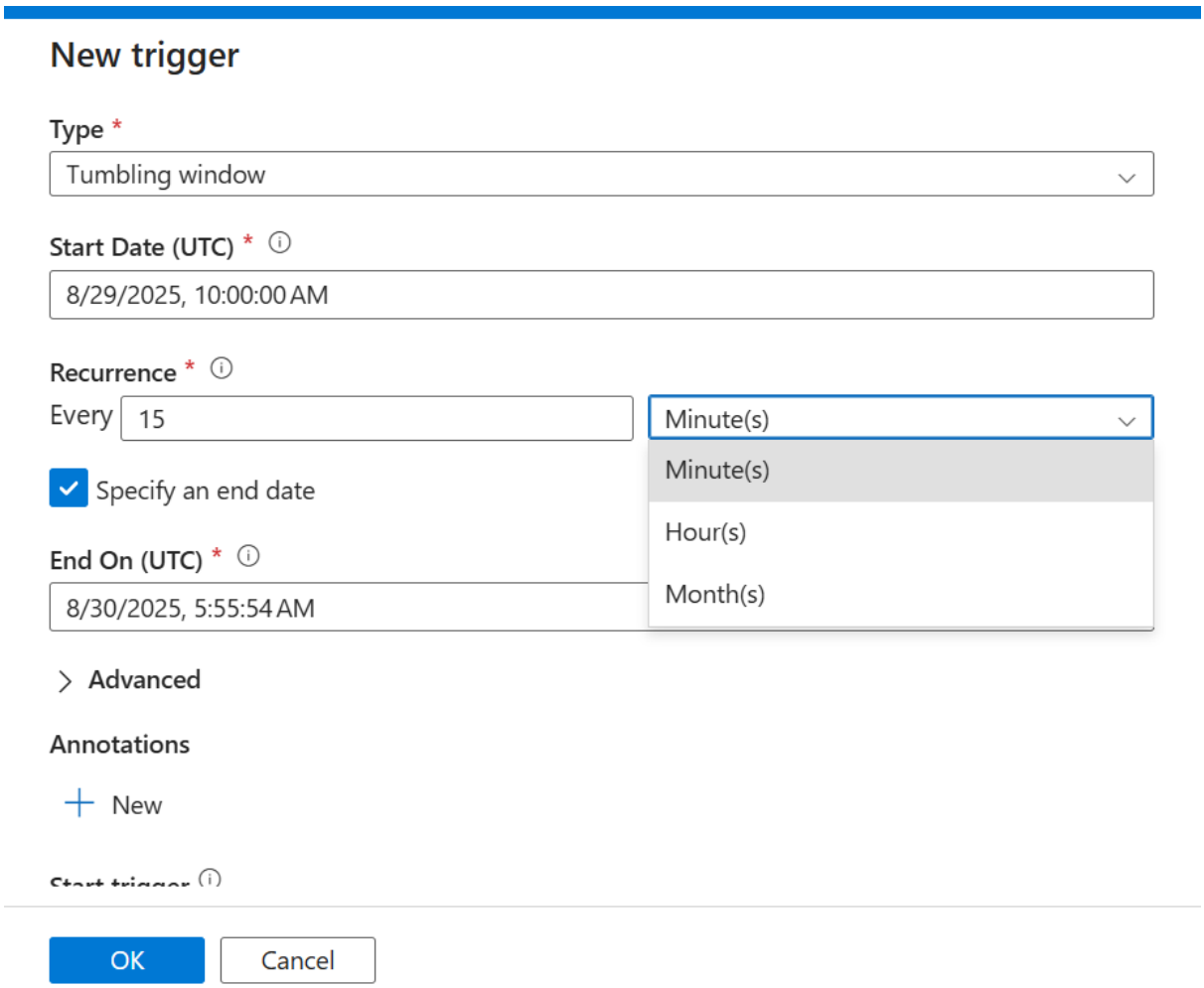
   - Runs a pipeline at a regular interval (window), but each run represents a specific, non-overlapping slice of time**.**
   - Example:

     Process data for each **hour**, where

     10:00–11:00 is one window,
     11:00–12:00 is the next window.

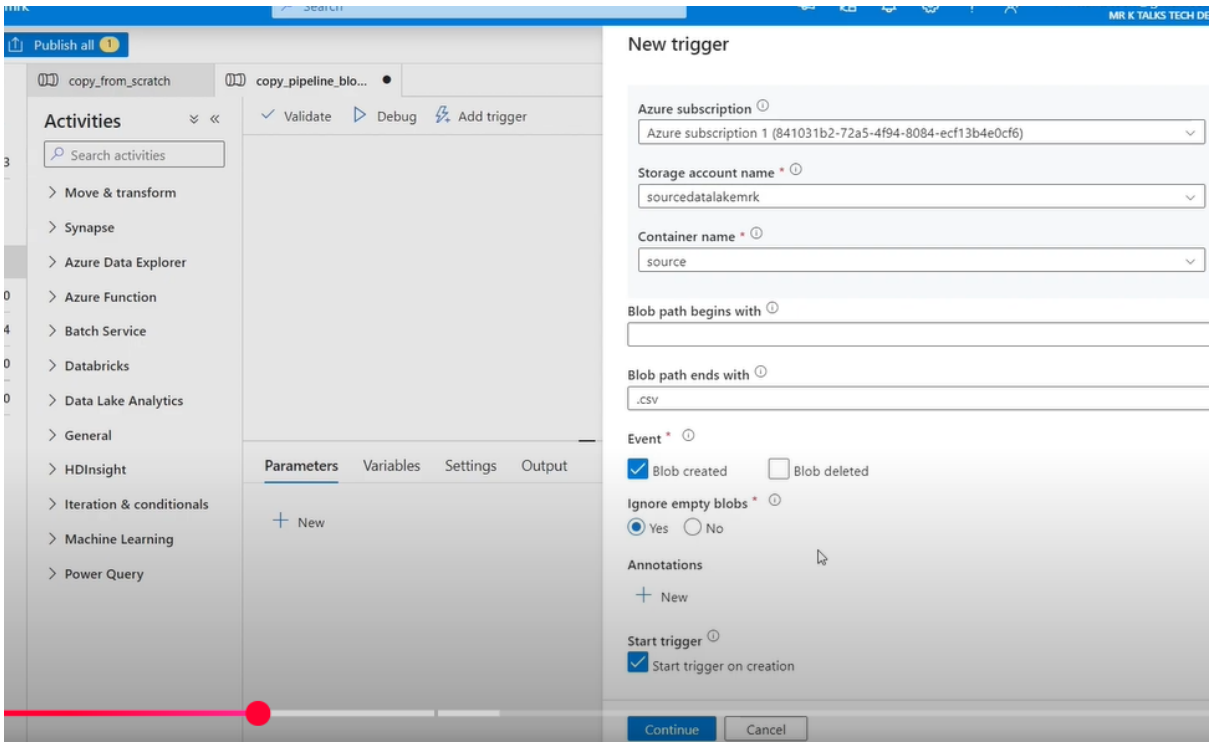
- Ensures exactly once execution per window (no overlaps).
- If a run for a window fails, the next window will wait until the failed one is re-run and succeeded.
- Supports dependency chaining (later runs can depend on the success of earlier windows).
- Useful for **incremental data processing**.
- You can **re-run** a specific failed window.
- Best suited for:
    - Time-partitioned data ingestion (hourly/daily batch data).
    - Scenarios where no data window should be skipped**.**
- Example: Process log files arriving **hourly**. If 2–3 PM failed, you can rerun just that window.

Feature	**Schedule Trigger**	**Tumbling Window Trigger**
Type	Calendar-based	Window-based (fixed intervals)
Dependency	No dependency between runs	Can depend on previous runs
Overlap	Runs can overlap	Non-overlapping windows
Use Case	Simple recurring jobs	Incremental, window-based processing
Retry/Re-run	Manual rerun of pipeline	Can rerun a specific failed window



3. Event-based Trigger

- Runs a pipeline **when an event occurs**, not on a schedule.
- Commonly used with **Azure Blob Storage** or **ADLS Gen2**.
- Typical events:
  - **Blob created** (new file uploaded)
  - **Blob deleted**
- Runs only when the event happens
- Example:
  - When a new file arrives in Azure Blob container, trigger pipeline to process it. (Run whenever a new orders.csv arrives in the raw-data container).
  - Ingest data automatically when a **new CSV file arrives** in raw/sales/ folder.
  - No need to schedule → the pipeline starts as soon as file lands.



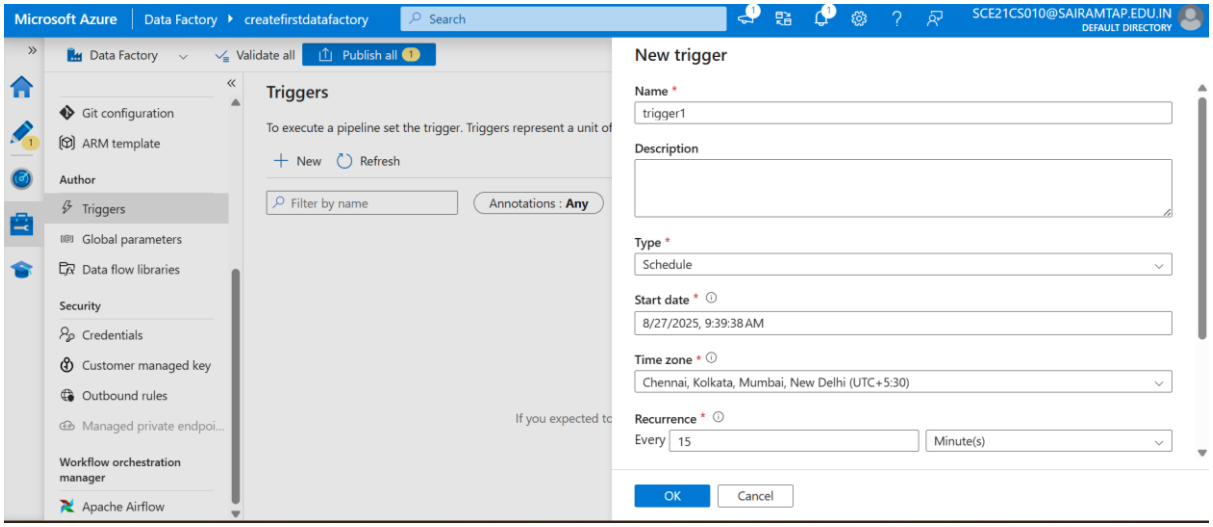
Event:

blob created – (if the file comes to from the folder, then triggered)

blob deleted – (if the file removed from the folder, then trigger it)

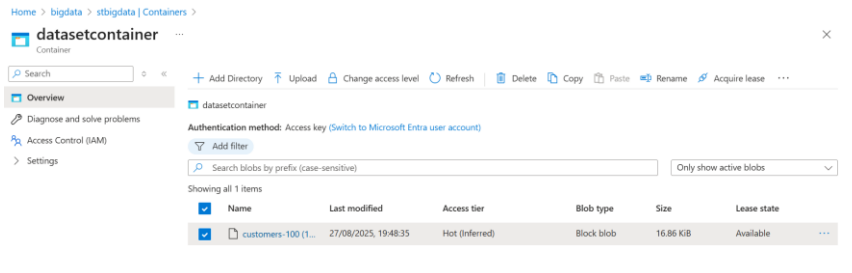
4. **Manual or custom Trigger**

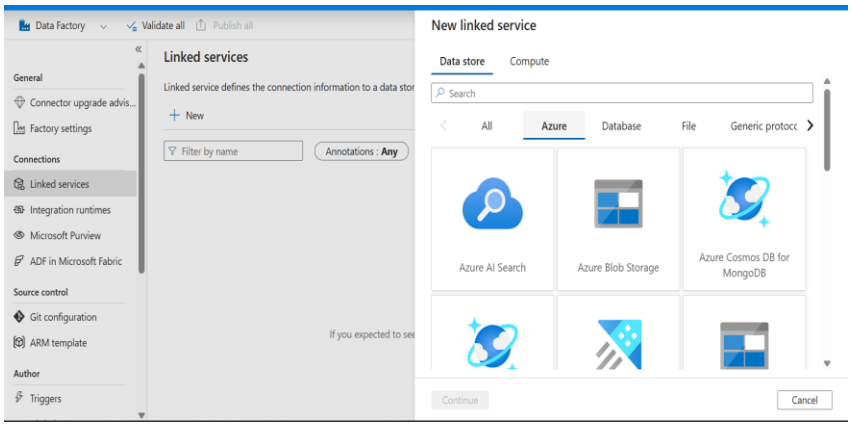
- **User-defined trigger** created using REST API, PowerShell, or SDK.
- Unlike Schedule, Tumbling, or Event triggers, you have **full control**.
- Useful when built-in triggers don’t fit your needs.
- You can invoke them programmatically with **custom logic** (from another system, app, or script).

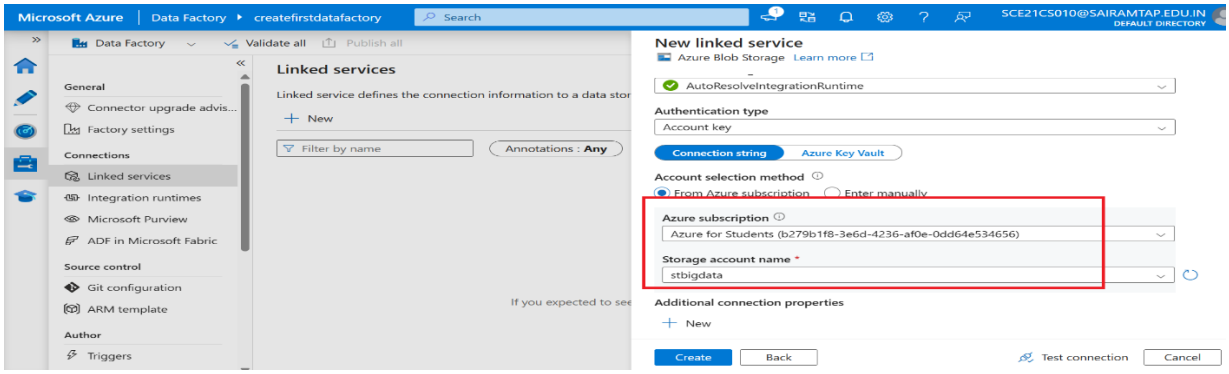


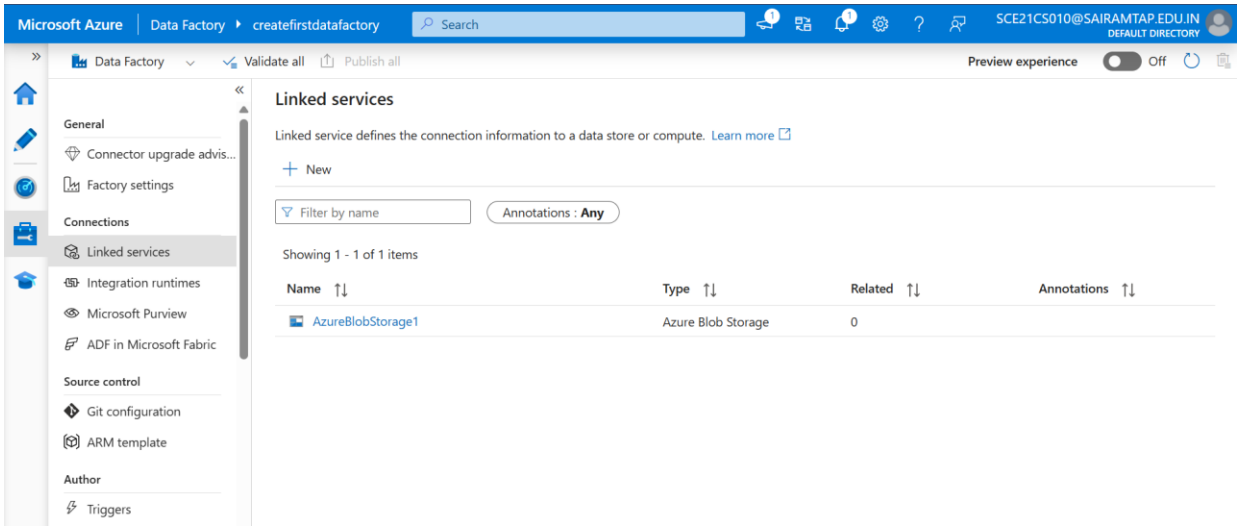
Practical demo of linked service

Connect the blob storage to ADF (use lined service) – customer.csv file

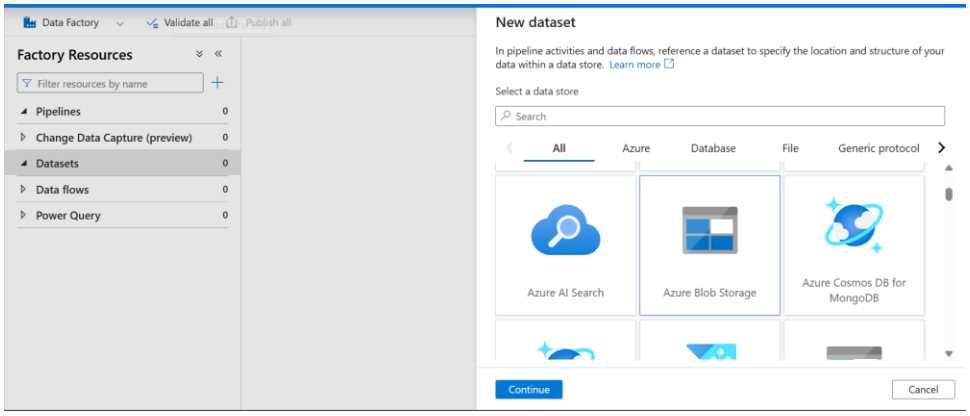




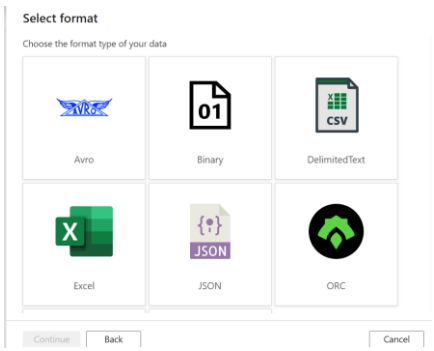


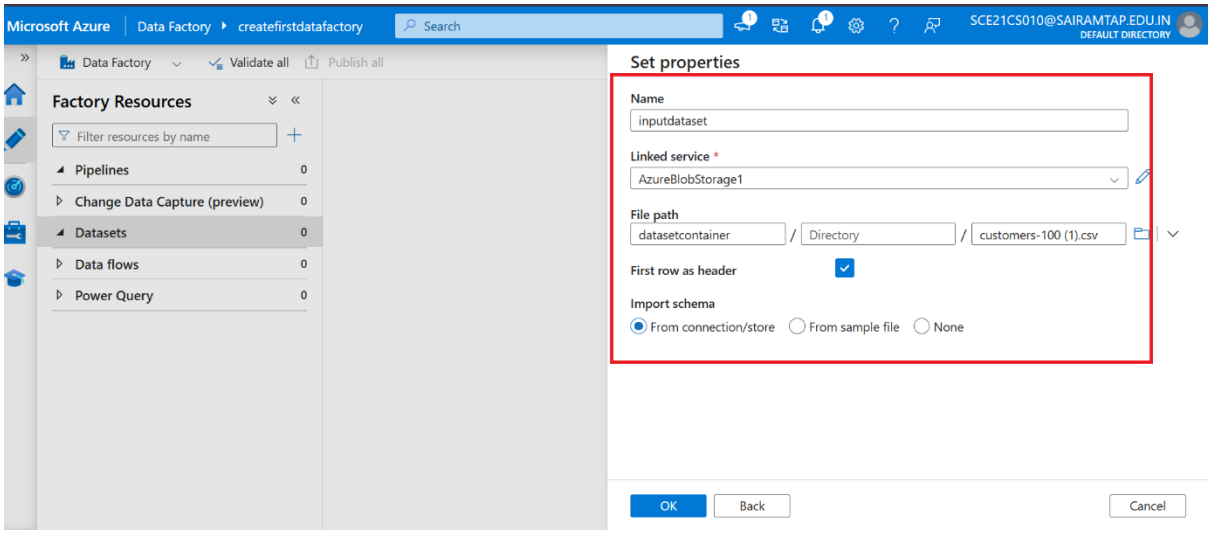


Create dataset

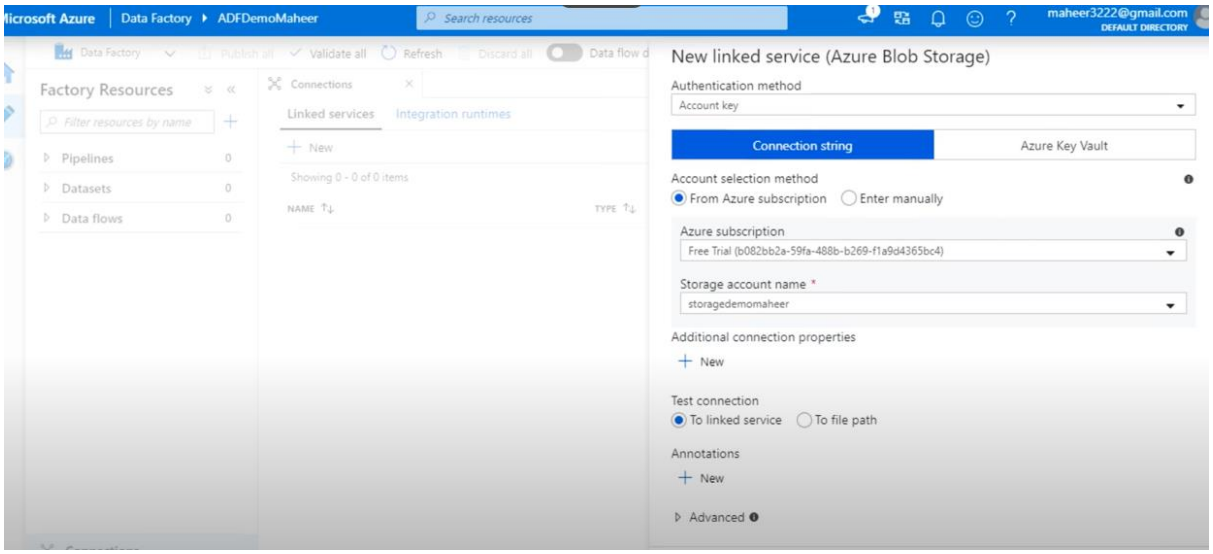


Select format

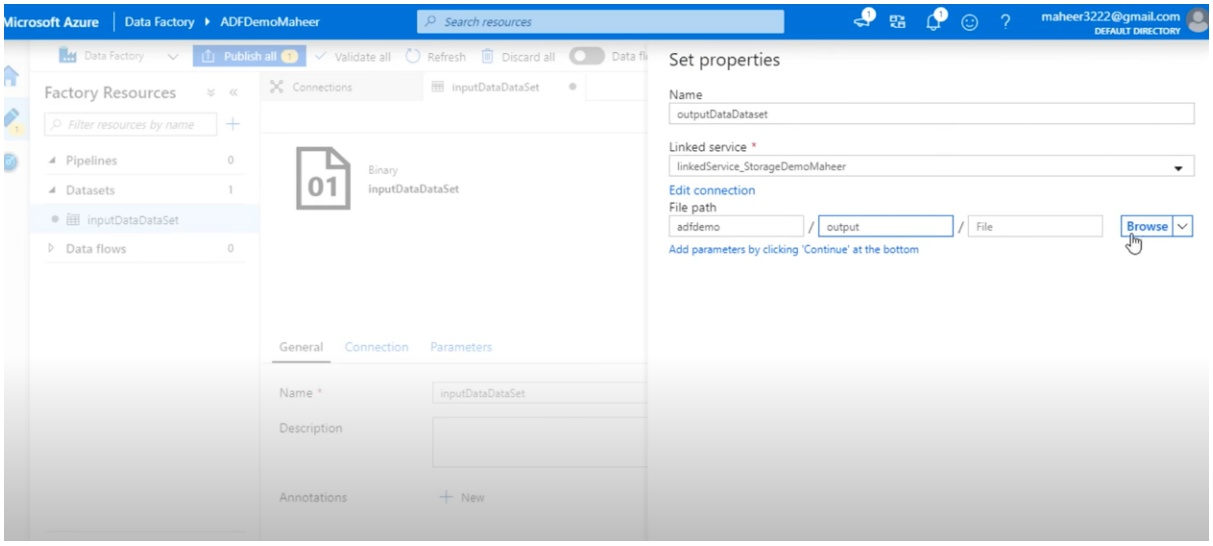


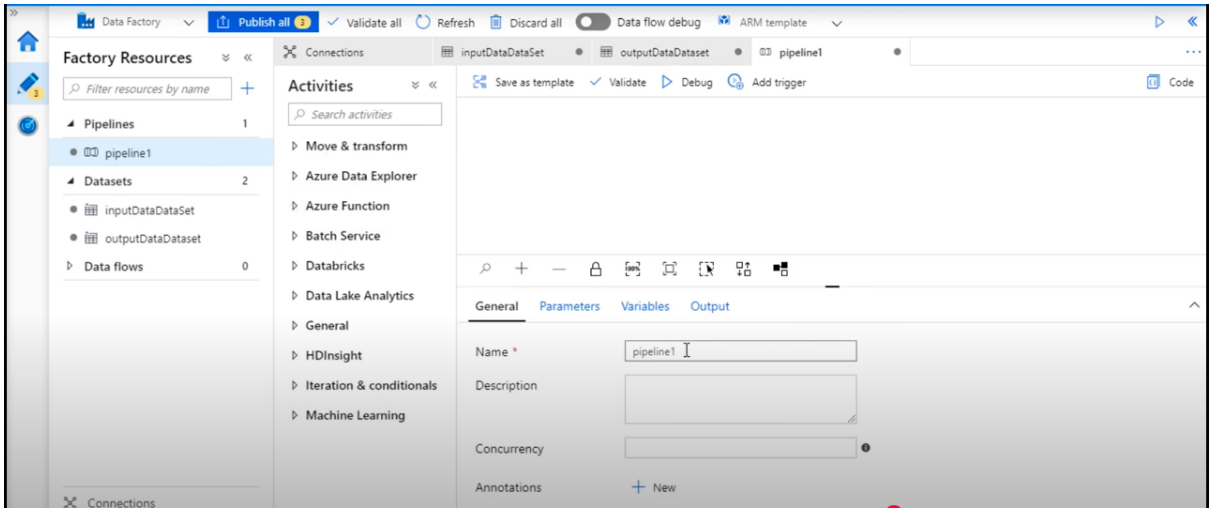


Same for input

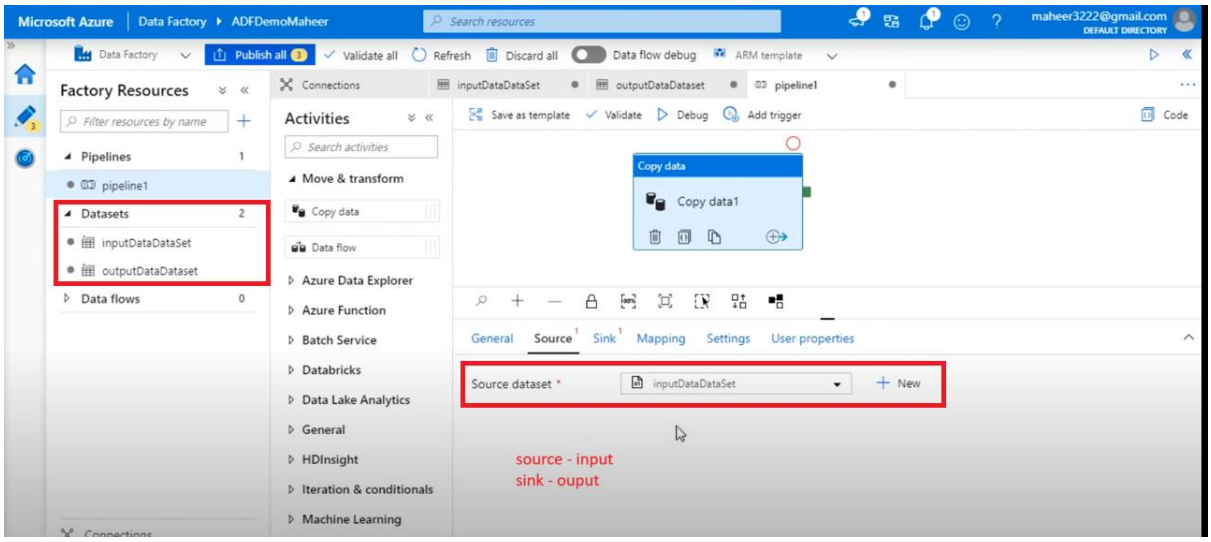


One for ouput





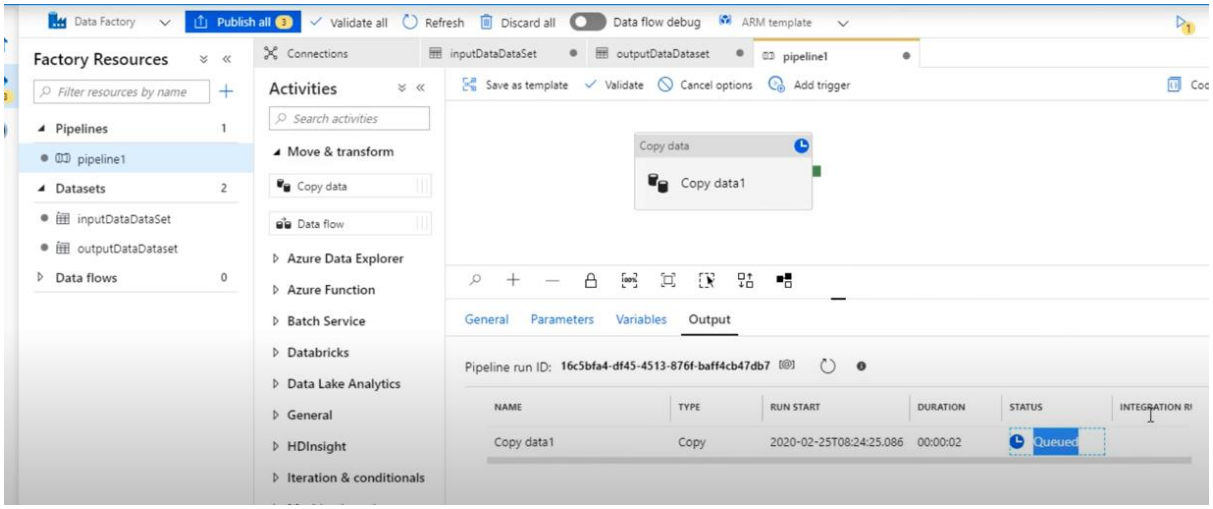
Copy activity – copy the data from the input folder to copy it output folder.

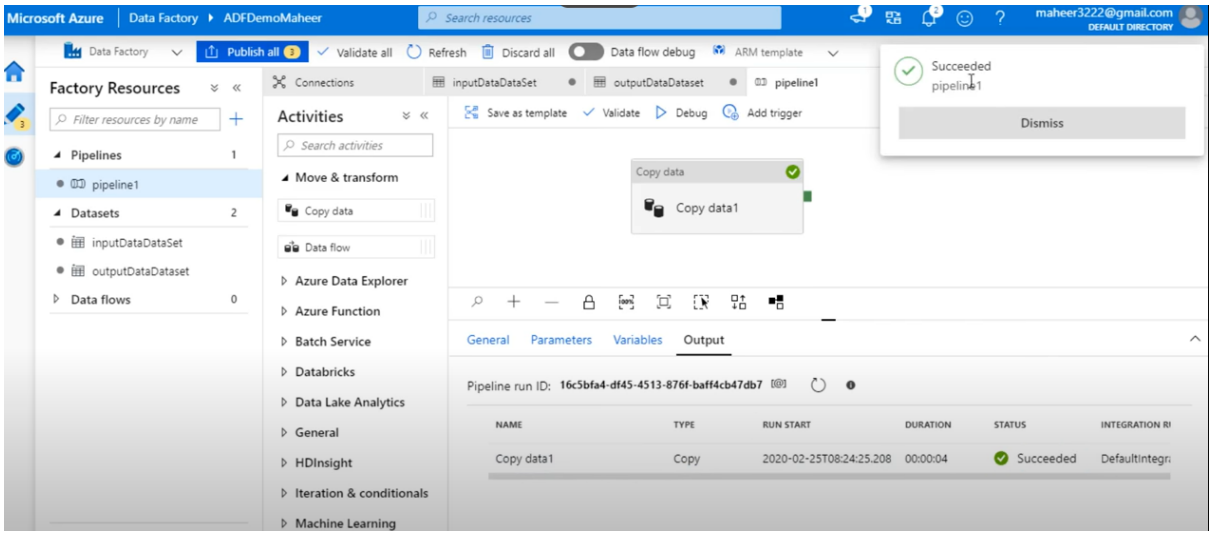


1st --- Validate

2nd --- Debug (whether data is copied or not)

It is ok then publish all





Activities

Types of Activities

- Data Movement Activities
- Data Transformation Activities
- Control flow activities

First, think about Azure Data Factory (ADF)

ADF is just an **orchestrator**.

- You design **pipelines** (like a recipe: "copy this data, transform that, load here").
- But ADF itself does **not** have the compute power to move or process the data.

So… something has to **do the actual work** → that's the **Integration Runtime (IR)**.

Integration Runtime (IR)

The Integration Runtime (IR) is the compute infrastructure used by Azure Data Factory.

Azure Synapse uses to **move data, run transformations, and connect to data sources**.

[Integration Runtime (IR) is the compute engine of ADF/Synapse that provides the processing power and connectivity needed to move and transform data].

Note: (Think of it like the **engine** that executes your pipeline activities. Without an IR, your pipeline has the steps defined, but nothing to **run them**.)

Azure Synapse pipelines to provide the following data integration capabilities across different network environments:

- **Data Flow**: Execute a data flow in a managed Azure compute environment.
- **Data movement**: Copy data across data stores in a public or private network (for both on-premises or virtual private networks). The service provides support for built-in connectors, format conversion, column mapping, and performant and scalable data transfer.
- **Activity dispatch**: Dispatch and monitor transformation activities running on various compute services such as Azure Databricks, Azure HDInsight, ML Studio (classic), Azure SQL Database, SQL Server, and more.
- **SSIS package execution**: Natively execute SQL Server Integration Services (SSIS) packages in a managed Azure compute environment.

Without IR, your pipeline is only instructions; nothing will actually run.

Environment

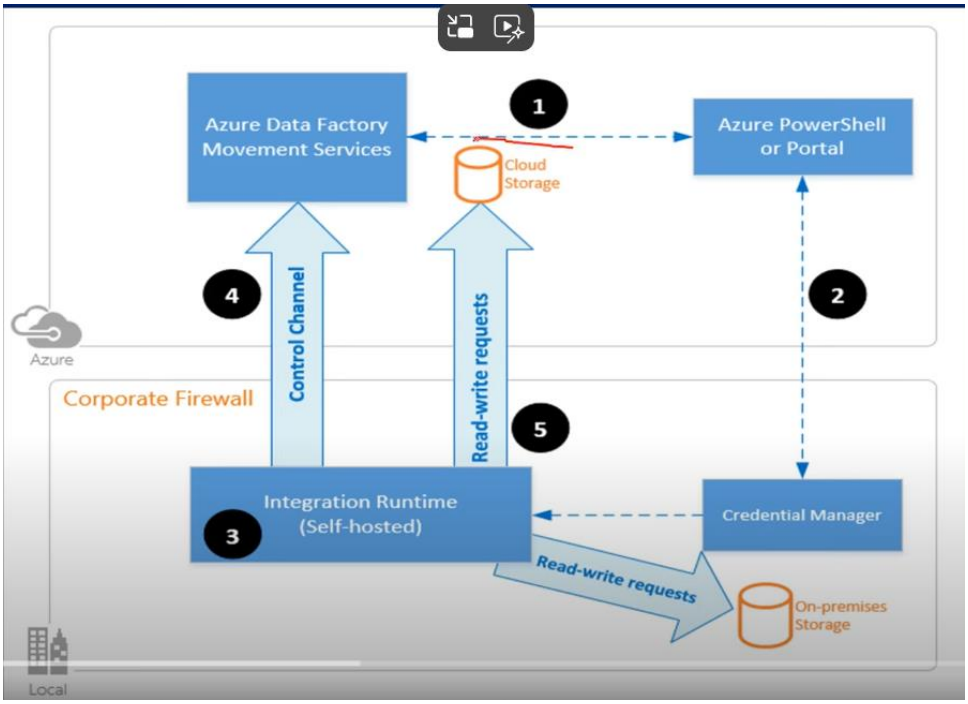
IR types:

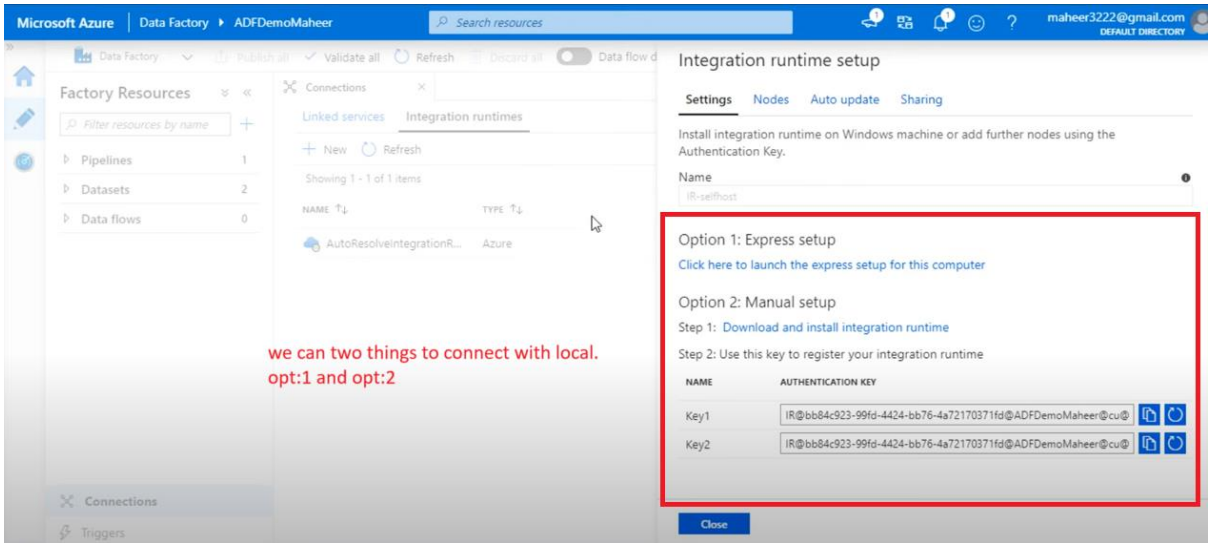
ADF offers 3 types of IR, and you should choose the types that best serves your data integration capabilities and network environment requirements.

- **Azure IR:** Handles cloud-based data integration tasks and is managed entirely by Azure. (default)

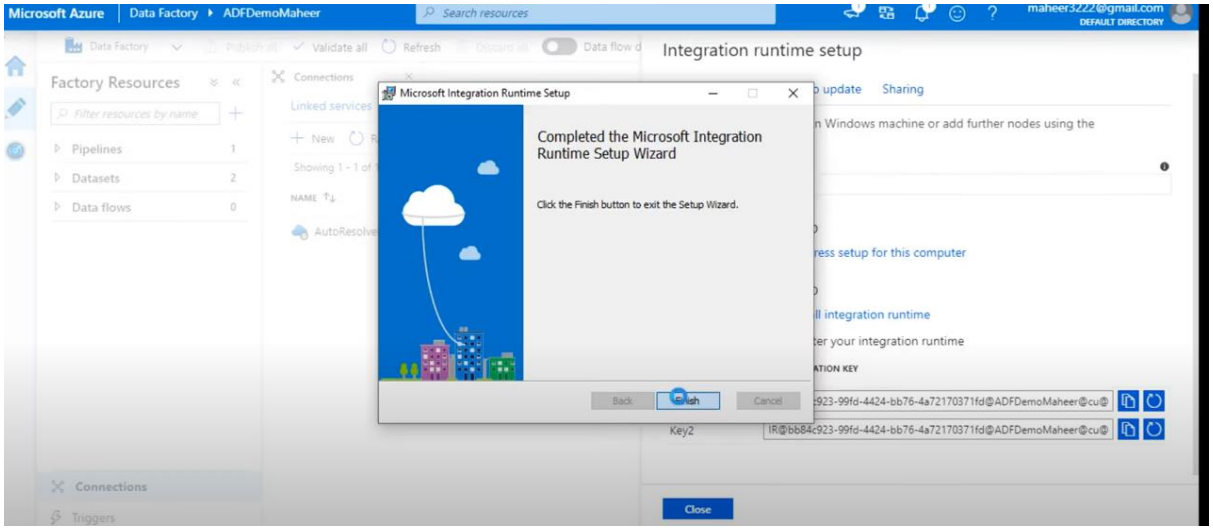
- [Running dataflow
- Running copy activities between cloud data stores.
- Running transform activities]
  - Best for **cloud-to-cloud data movement** and **transformations**.
  - Supports **mapping data flows** and copying data between Azure & other cloud sources.

- **Self-hosted IR:** Supports data movement between on-premises systems and the cloud, making it ideal for hybrid scenarios.
  - Installed on your **own machine (on-prem or in VM)**.
  - Acts like a **bridge** between ADF and your **private network**.
  - Required when you need to connect to:
    - On-premises databases (SQL Server, Oracle, MySQL, etc.).
    - Private network sources behind firewalls/VPN.
  - Can also be used for **cloud-to-cloud** copy if you want data to pass through your network.
  - Example: Copy data from On-Prem SQL Server → Azure Data Lake.

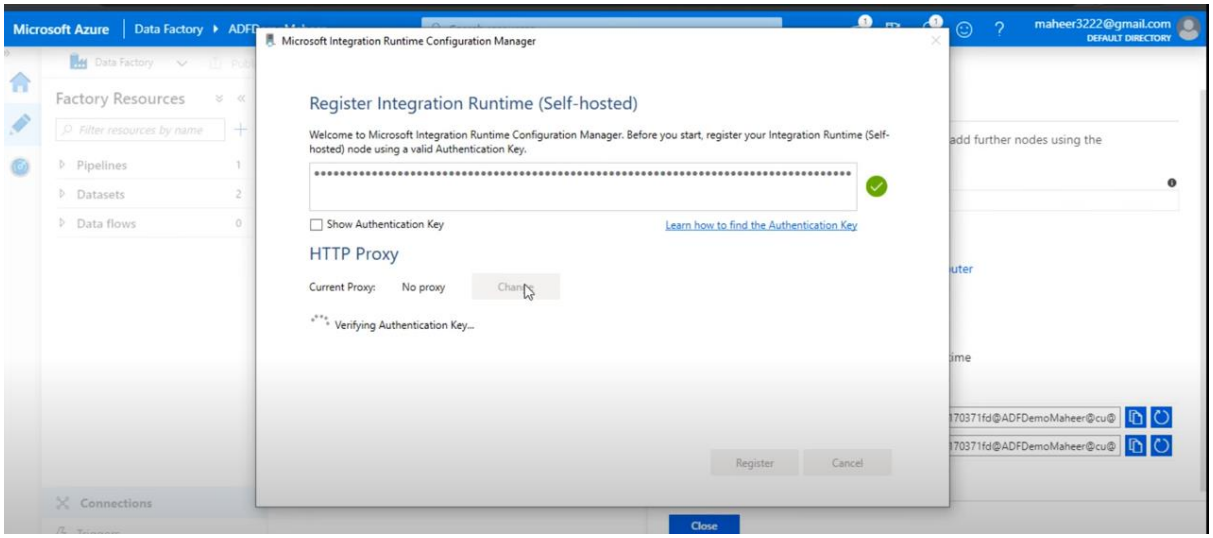


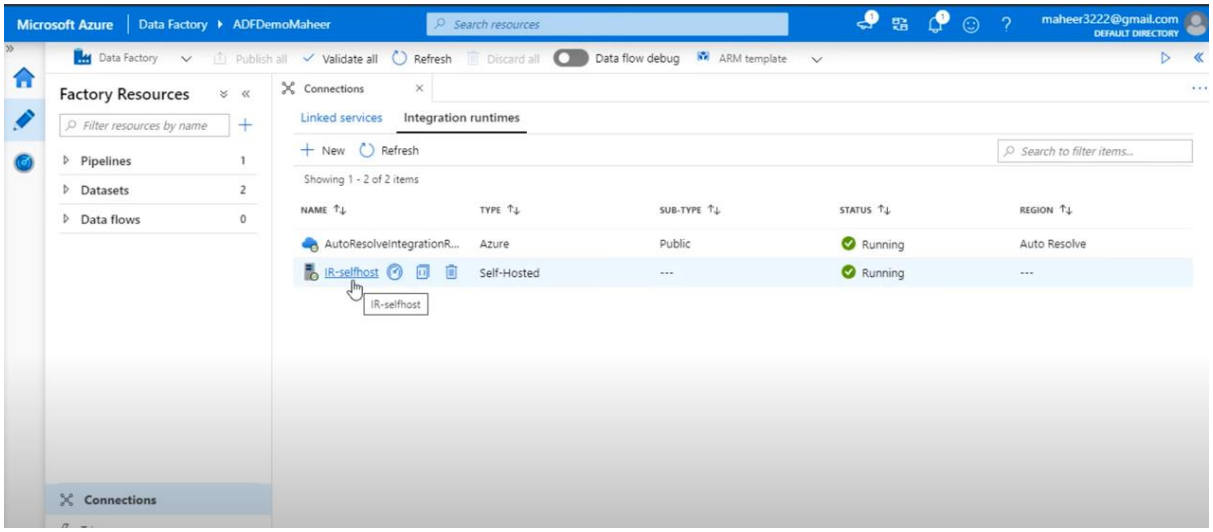


Need to install the **IR setup**.

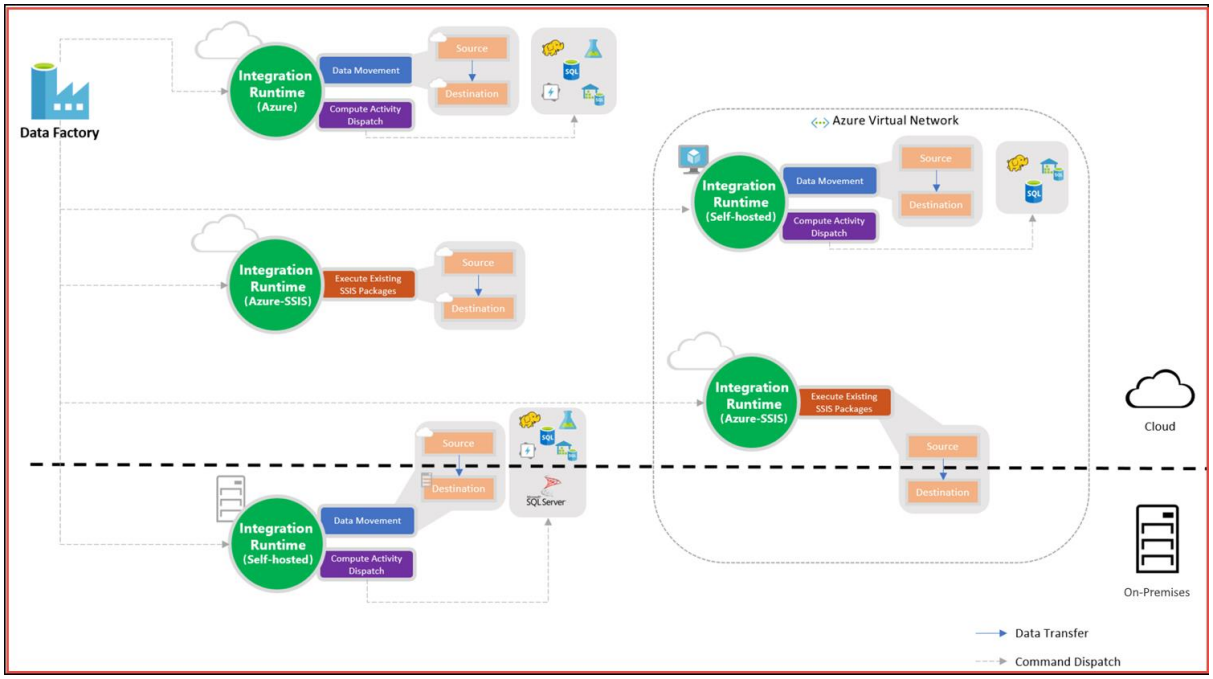


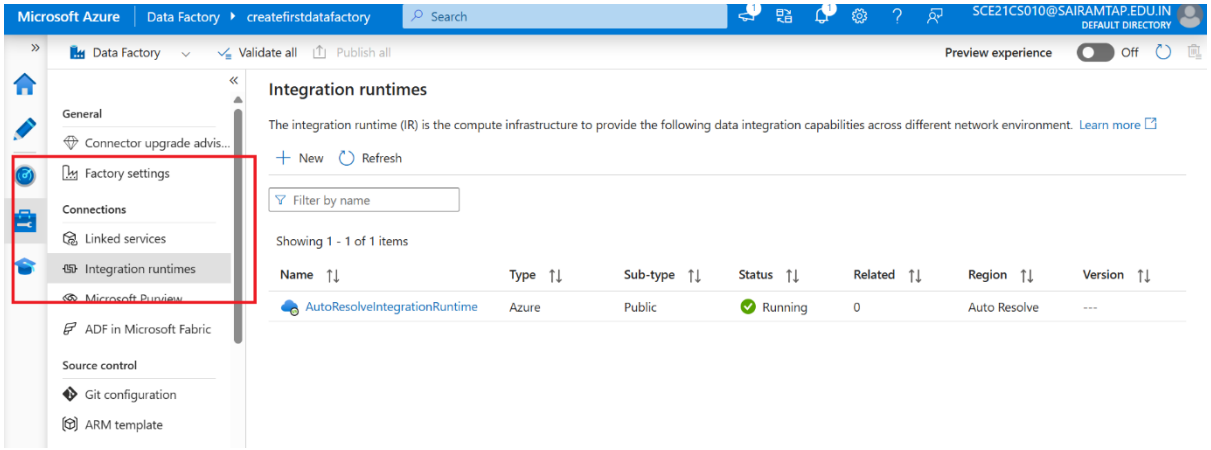
Copy the key and paste it.

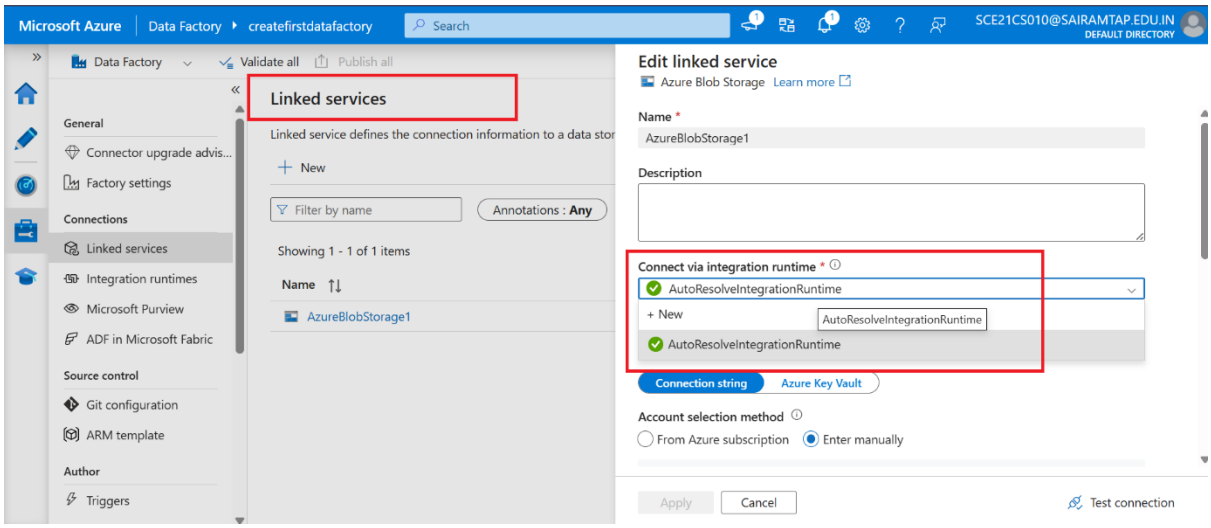




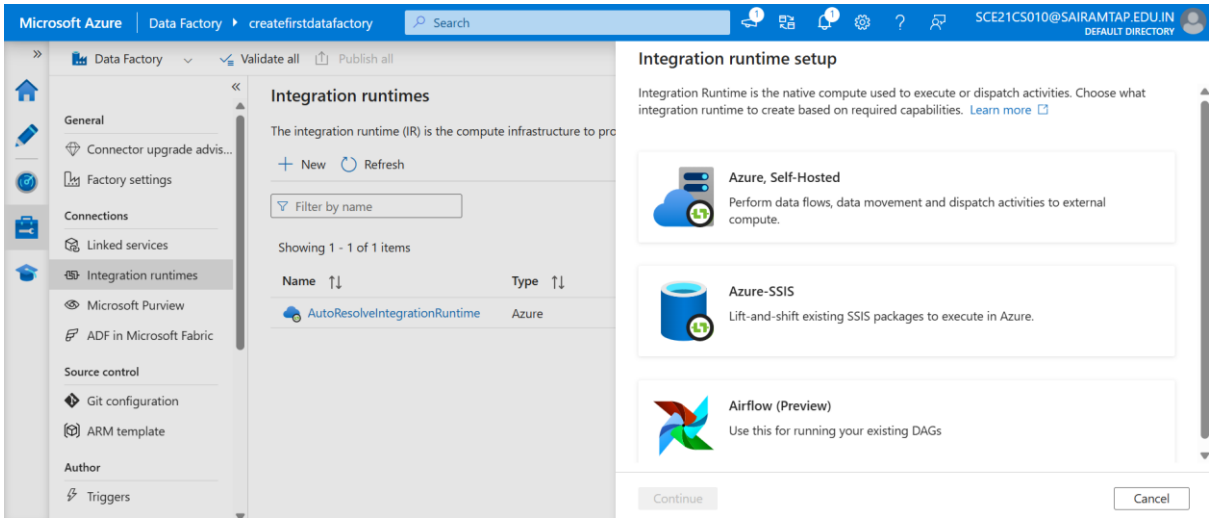
- **SSIS IR:** Enables the execution of SQL Server Integration Service (SSIS) packages within Azure, allowing you to reuse existing SSIS workflows in the cloud.







Create new IR



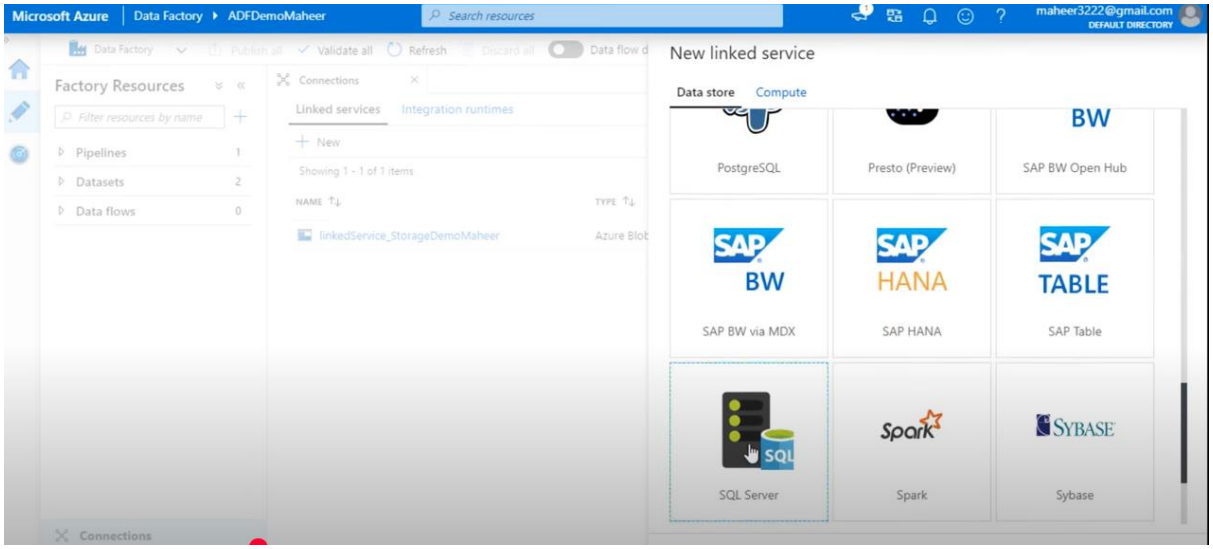
Parameterize Linked Services

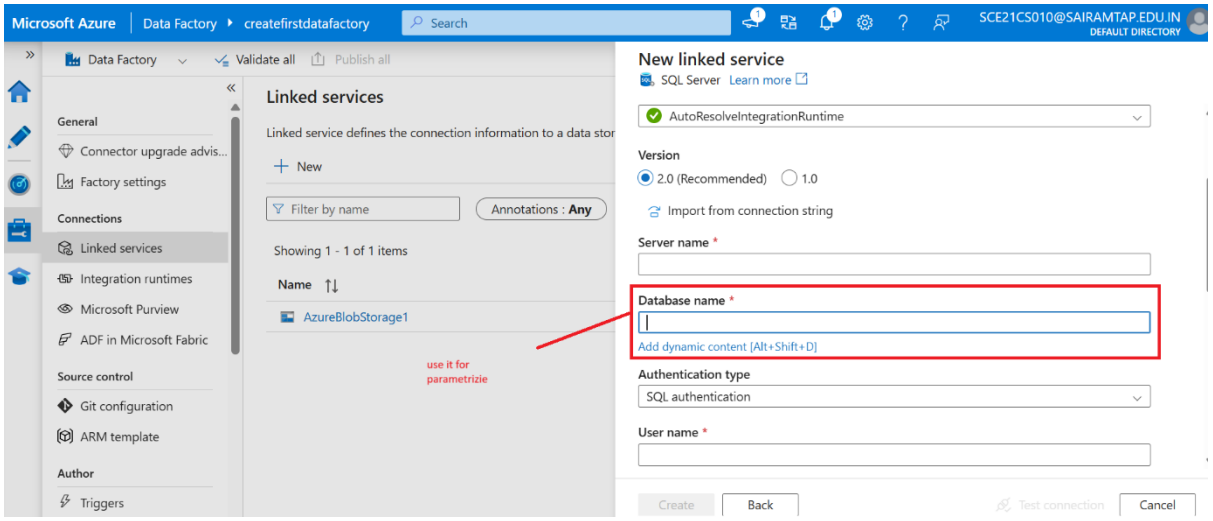
Consider SQL server:

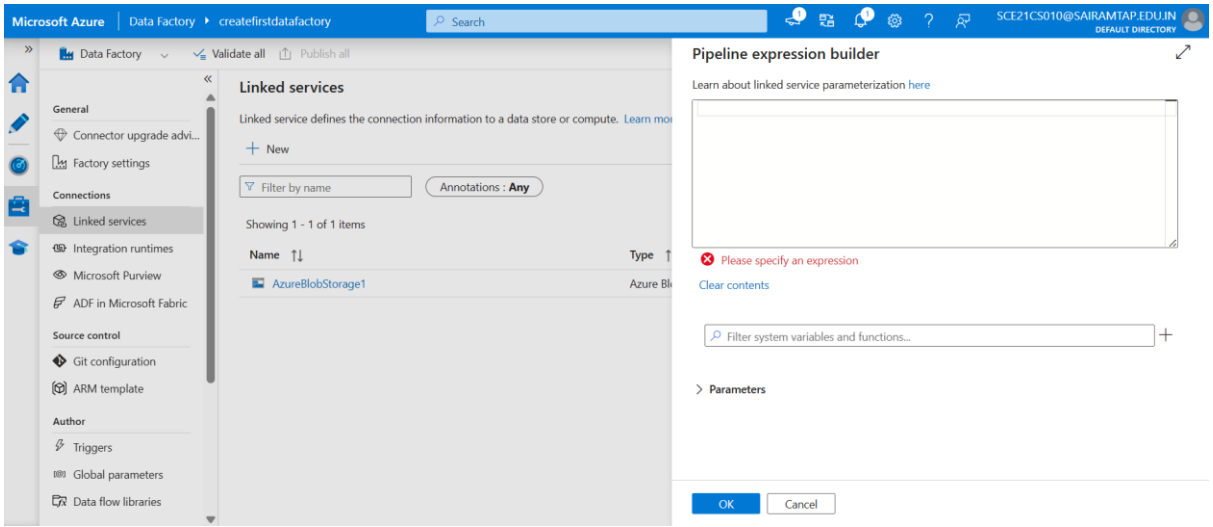
contains 10 databases, for that creating 10 linked service for all 10 DBs.

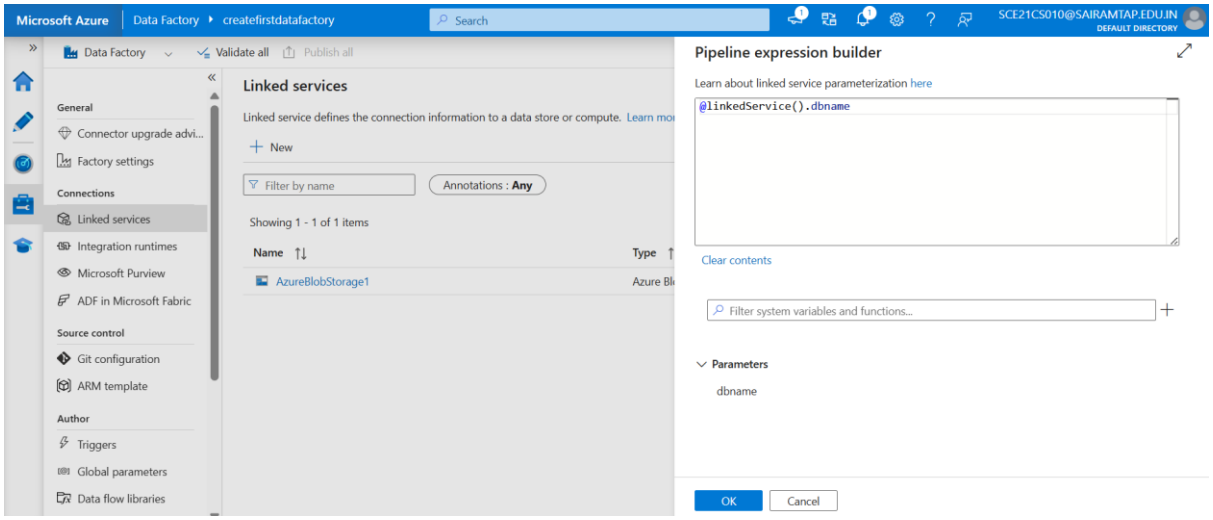
To avoid to this we use, **parameterization** (by this create only one linked service and use it for 10 dbs).

Create storage (sql server):

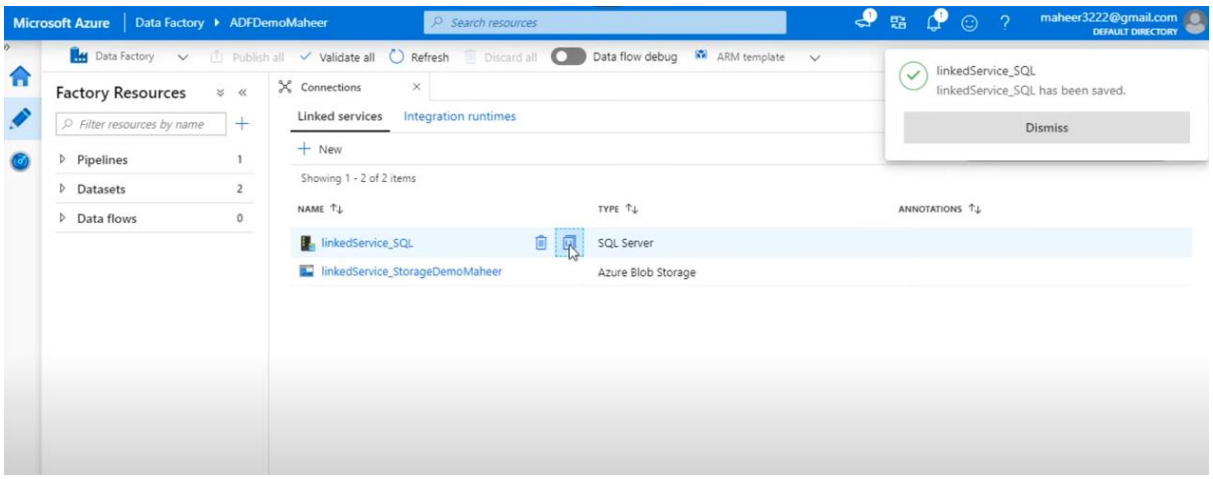


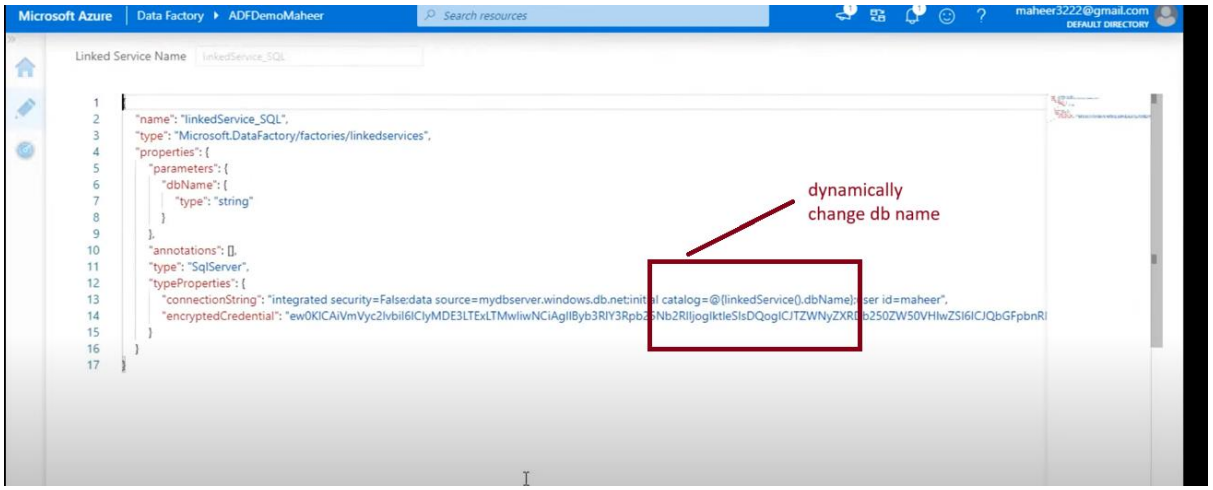




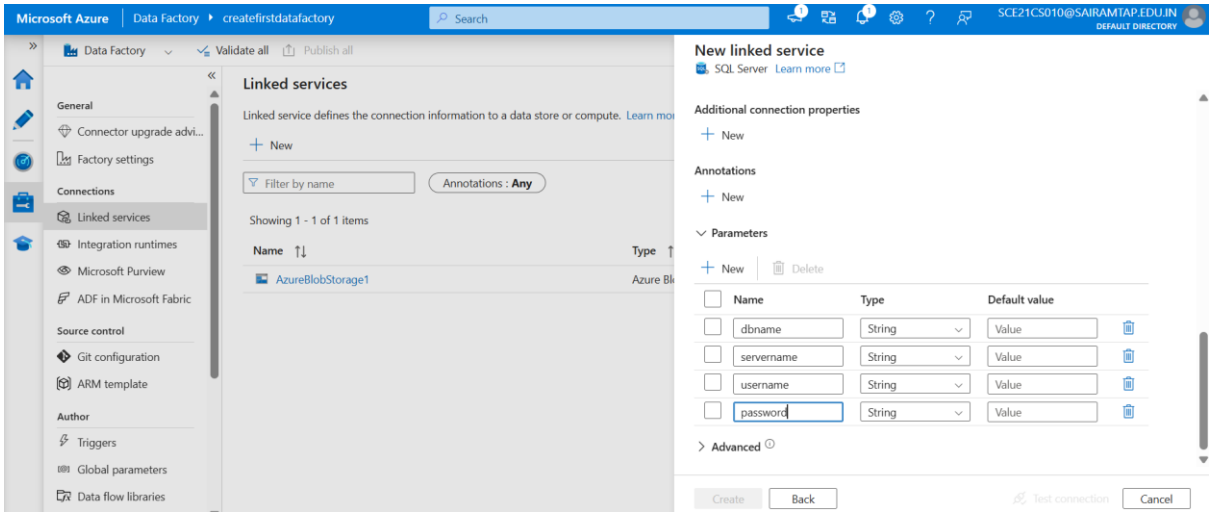


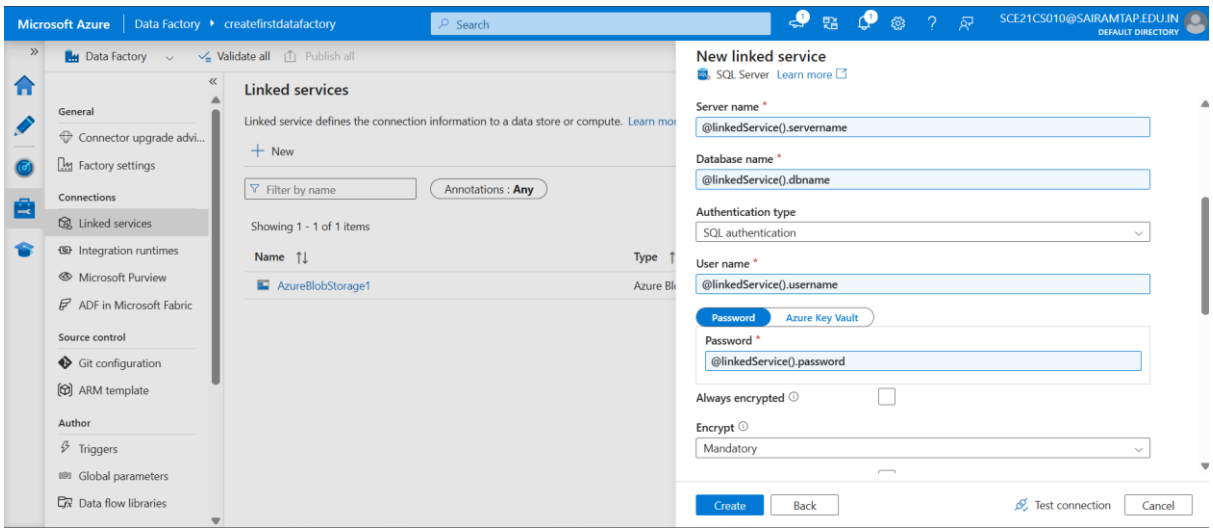
Once created





We can create the parameters for all the things like username, password, server name and database name.





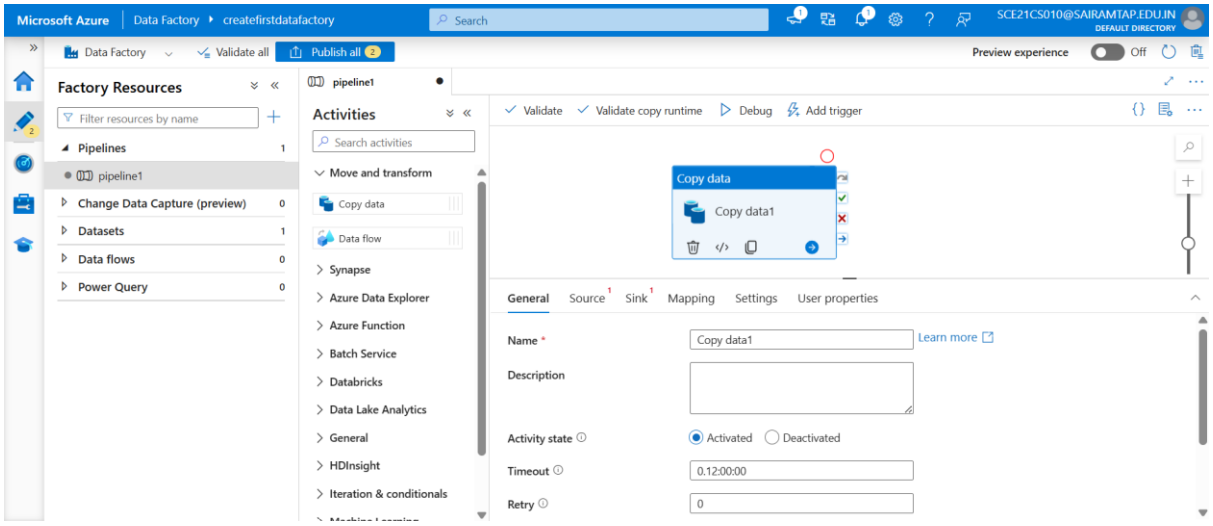
Parameterize Dataset and pipeline in ADF

You can parameterize a dataset and pass dynamic values at run time.

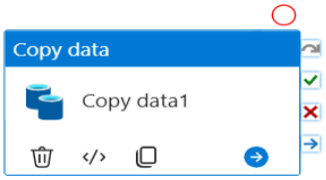
Supported file format in ADF

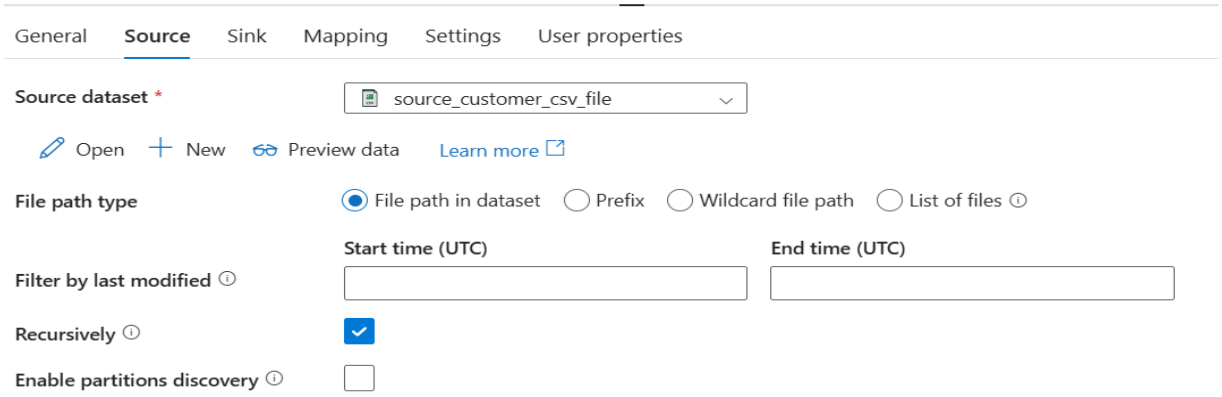
- **Avro format**
- **Binary format –** text files
- **Delimited text format –** csv file
- **JSON format**
- **ORC format**
- **Parquet format**

Copy Data Activity

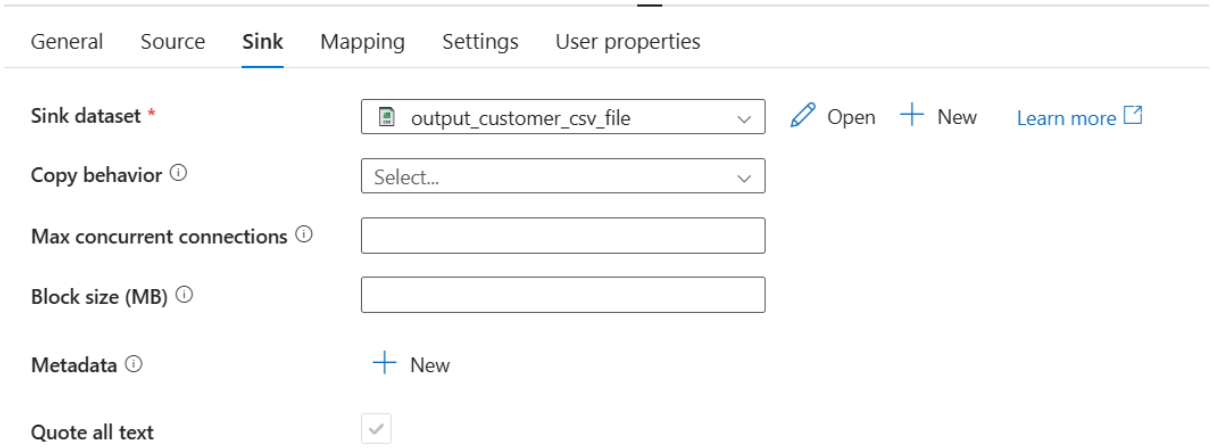


Source

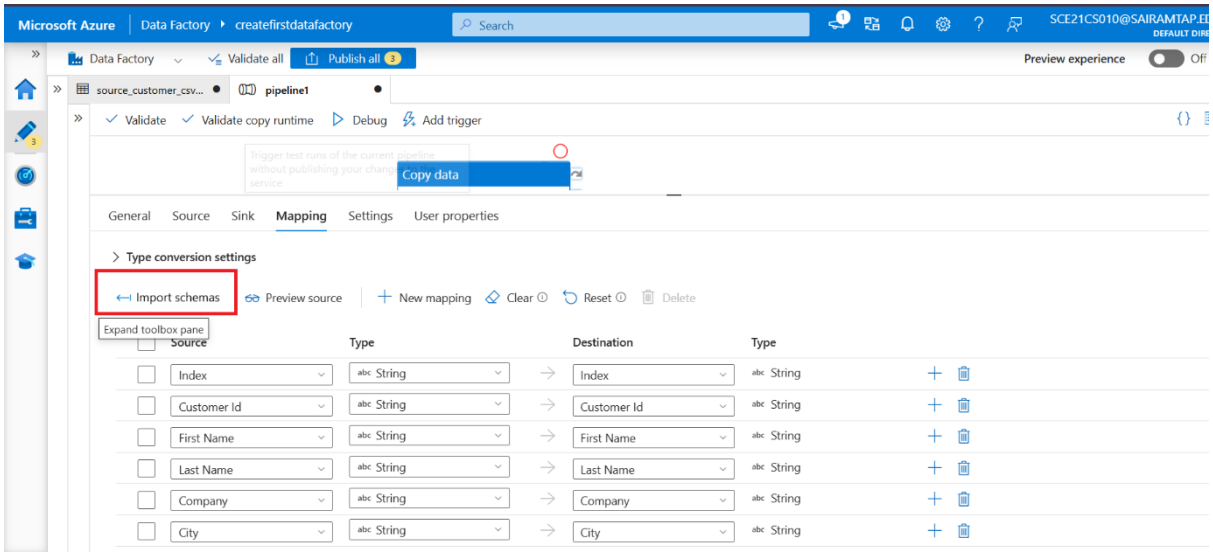




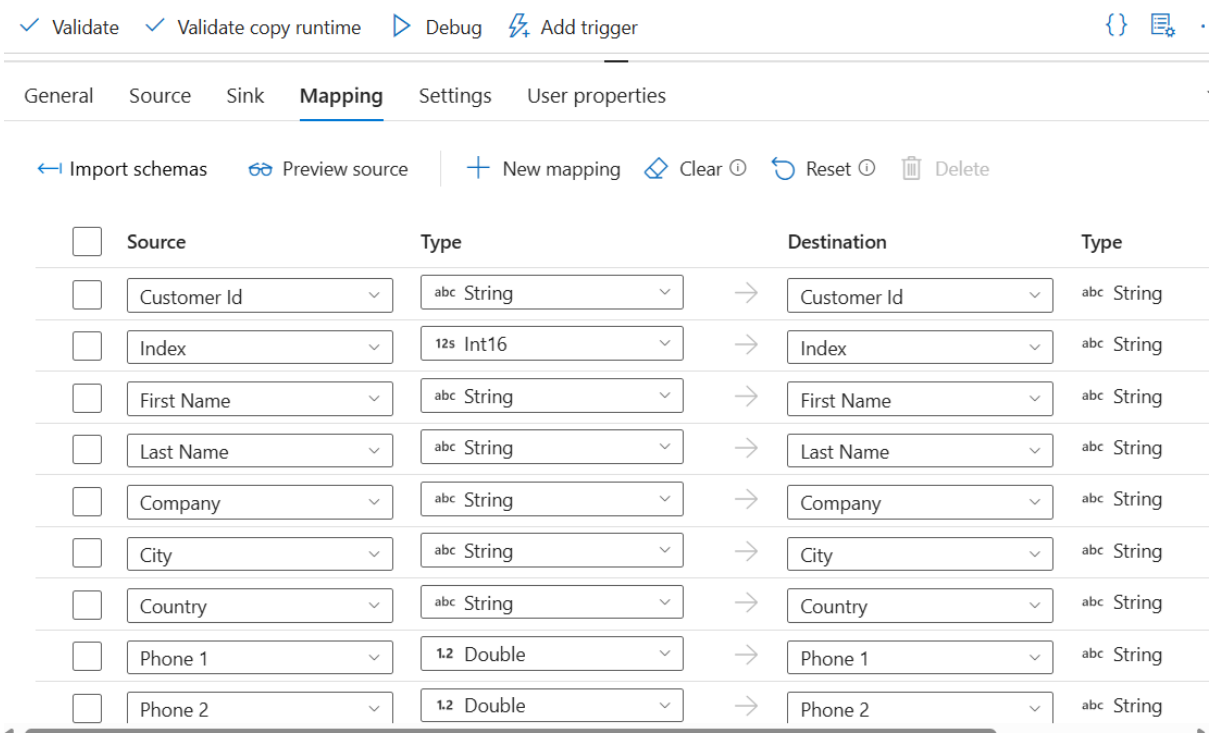
Destination - sink

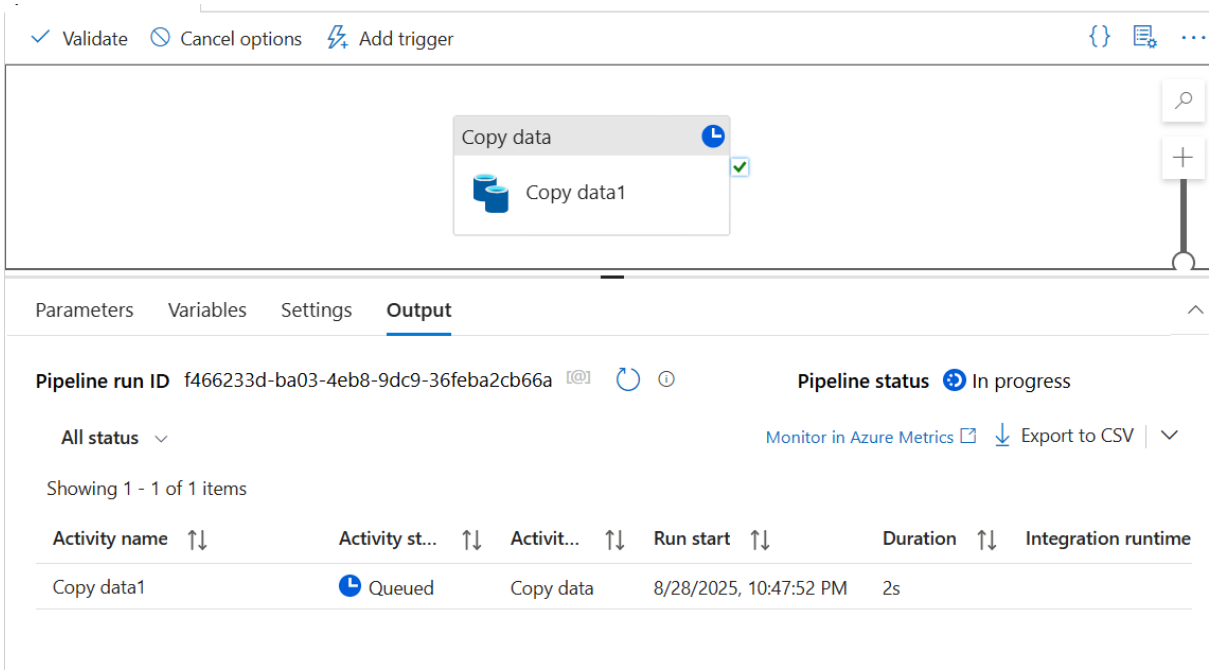


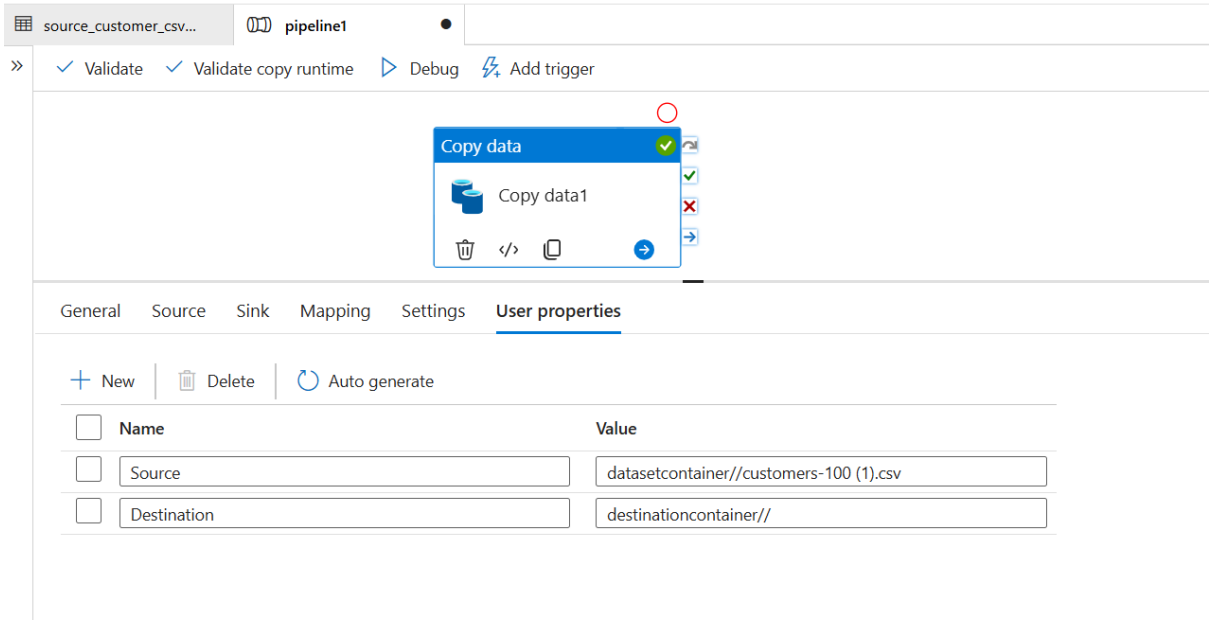
click on import schema

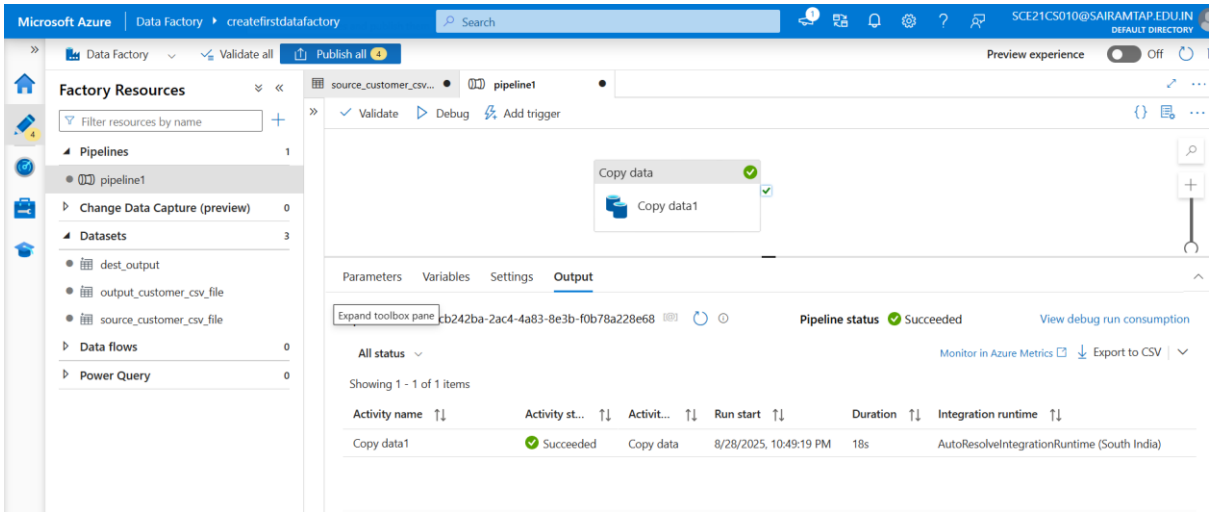


We change the column order and data type. Also delete the not required column in destination.

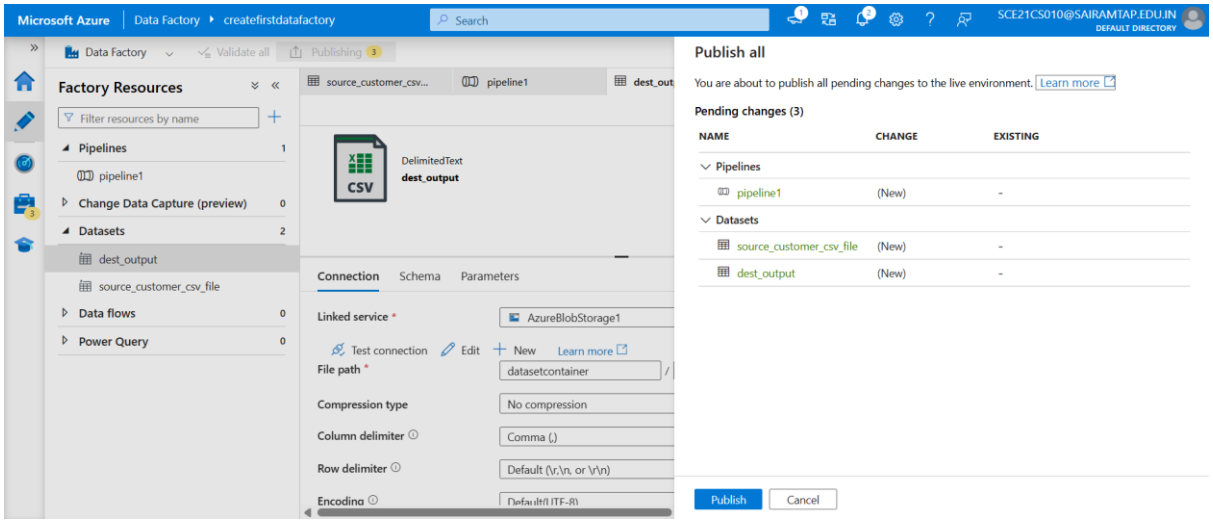



Validate
Cancel options
Add trigger
Copy data
Copy data1
Parameters
Variables
Settings
Output
Pipeline run ID f466233d-ba03-4eb8-9dc9-36feba2cb66a
Pipeline status In progress
All status
Monitor in Azure Metrics
Export to CSV
Showing 1 - 1 of 1 items
Activity name
Activity st...
Activit...
Run start
Duration
Integration runtime
Copy data1
Queued
Copy data
8/28/2025, 10:47:52 PM
2s

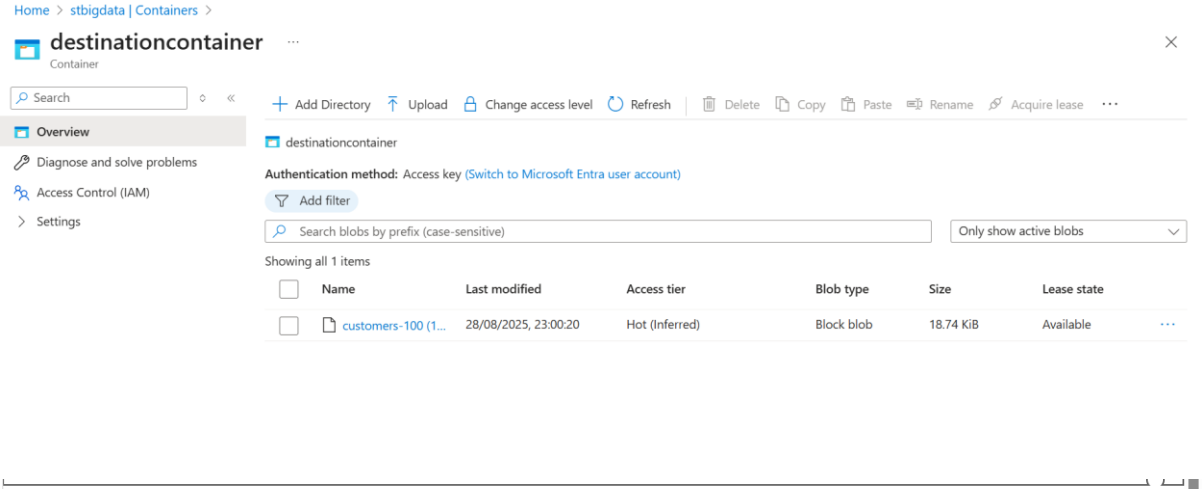

source_customer_csv...
pipeline1
Validate
Validate copy runtime
Debug
Add trigger
Copy data
Copy data1
General
Source
Sink
Mapping
Settings
User properties
New
Delete
Auto generate
Name
Value
Source
datasetcontainer//customers-100 (1).csv
Destination
destinationcontainer//

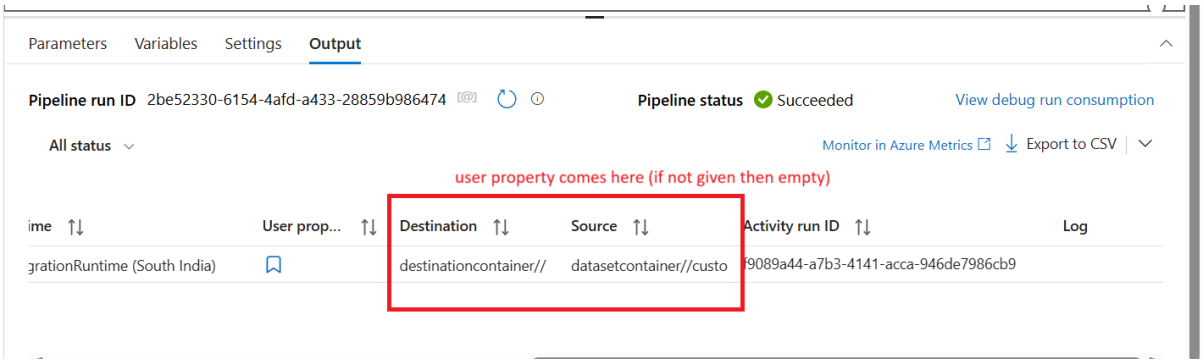

Microsoft Azure
Data Factory
createfirstdatafactory
Search
SCE21CS010@SAIRAMTAP.EDU.IN
DEFAULT DIRECTORY
Data Factory
Validate all
Publish all 4
Preview experience
Off
Factory Resources
Filter resources by name
Pipelines 1
pipeline1
Change Data Capture (preview) 0
Datasets 3
dest_output
output_customer_csv_file
source_customer_csv_file
Data flows 0
Power Query 0
Validate
Debug
Add trigger
Copy data
Copy data1
Parameters
Variables
Settings
Output
Expand toolbox pane
Pipeline status Succeeded
View debug run consumption
All status
Monitor in Azure Metrics
Export to CSV
Showing 1 - 1 of 1 items
Activity name
Activity st...
Activit...
Run start
Duration
Integration runtime
Copy data1
Succeeded
Copy data
8/28/2025, 10:49:19 PM
18s
AutoResolveIntegrationRuntime (South India)

Publish

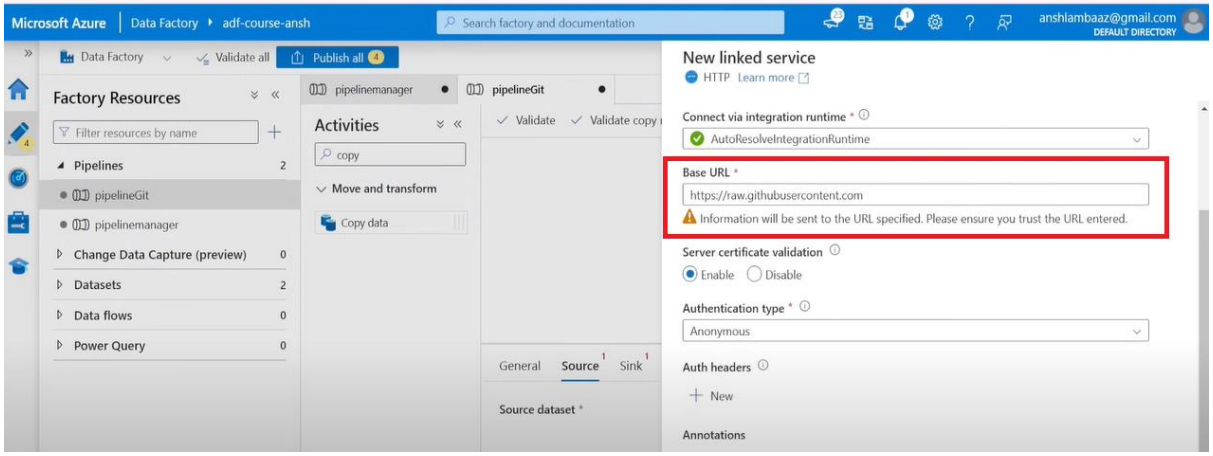


Copied from src to destination

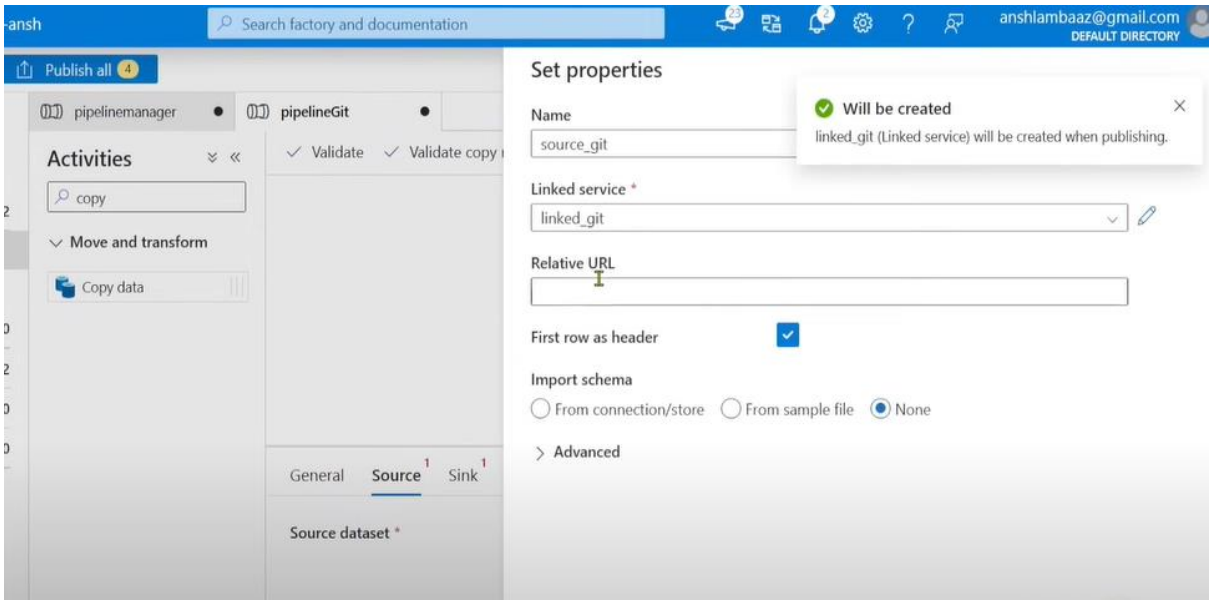




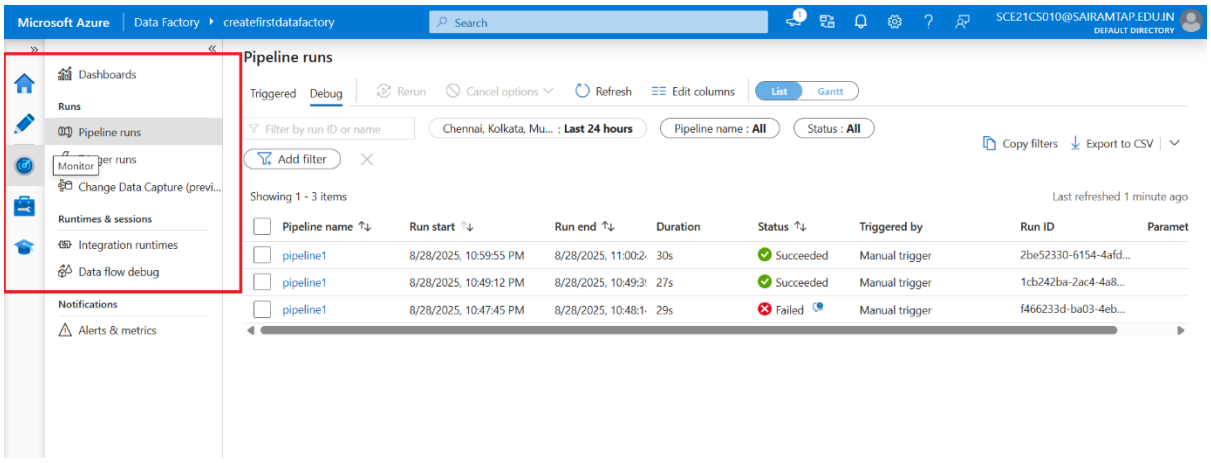
For copy the data from the API (use the base url)



Relative url



Monitor copy activity



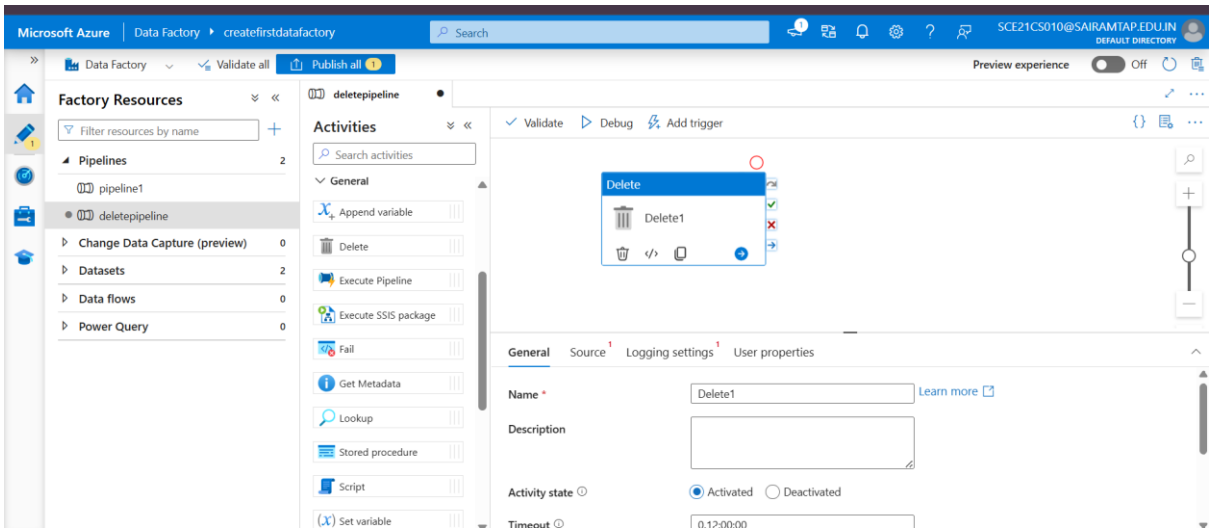
Check the details of the pipeline

Once you've created and published a pipeline in ADF, you can associate it with a trigger or manually kick off an hoc run. You can monitor all of your pipeline runs natively in the ADF user experience.

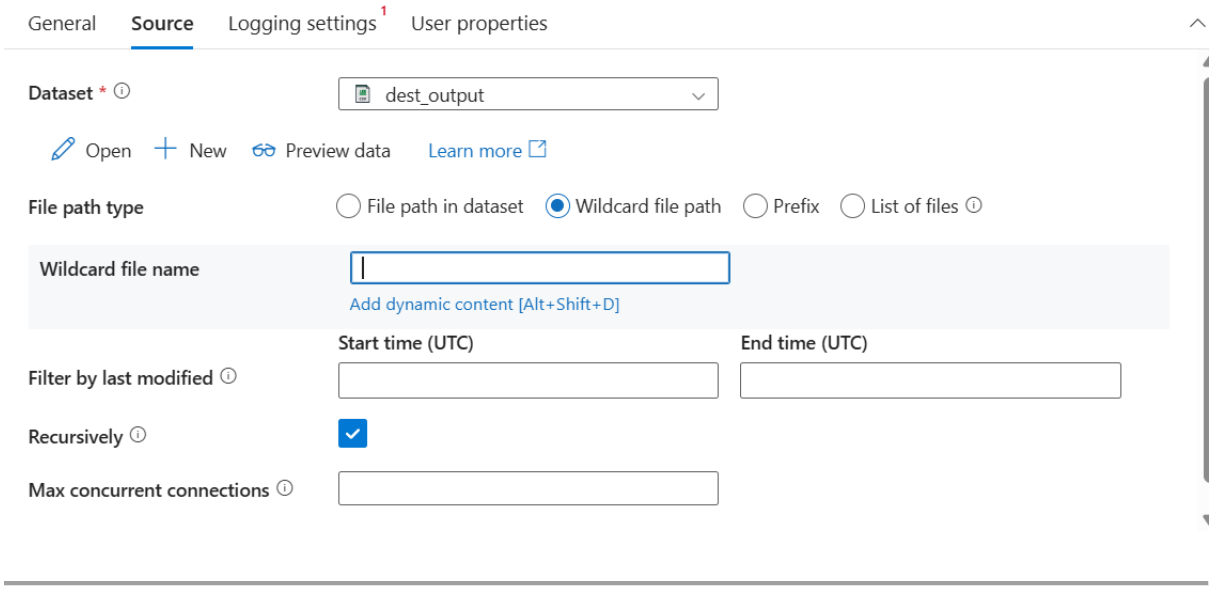
Delete Activity

You can use the delete activity in ADF to delete files or folders from on-premises storage or cloud storage.

Deleted files or folders cannot be restored.



We can customize the file (*.txt – means delete only txt)



Lookup Activity

The Lookup activity in Azure Data Factory is a powerful tool used to retrieve data from an external source, such as a database or file system, and pass it to subsequent activities in a pipeline.

Lookup activity reads and returns the content of a configuration file or table. It also returns the result of executing a query or stored procedure.

The output from lookup activity can be used in a subsequent activity.

What Lookup Activity Does:

- It runs a **query** or **reads a file** and returns the result.
- The result can be:
  - **Single row** → returns key-value pairs.
  - **Multiple rows** → returns an array of JSON objects.

Limitation or drawback of lookup activity

- **Row Limit**: It can fetch a maximum of **5,000 rows**.
- **Data Size Limit**: The total size of the fetched data cannot exceed **4 MB**.

Example Scenario

- **Use Case**: Retrieve a file name from a database table to dynamically copy the file from Azure Blob Storage.

Scenario

We had to build a pipeline that **processes multiple files dynamically**.

- File details (name, path, active flag) were stored in a **control/config table** in Azure SQL DB.

- Each file had to be copied from **source Blob storage** to **curated ADLS zone** only if it was **active**.

Step 1 – Lookup Activity

- First, we used a **Lookup activity** to fetch all the file details from the control table.
- Lookup returned JSON like:

```
{
  "value": [
    { "FileName": "sales.csv", "Path": "/input/sales/", "IsActive": 1 },
    { "FileName": "customer.csv", "Path": "/input/customers/", "IsActive": 0 }
  ]
}
```

- This allowed us to **parameterize the pipeline** and make it **dynamic**.
- Without Lookup, we would have to hardcode the file names.

Step 2 – ForEach Activity

- We passed the Lookup output array into a **ForEach activity**.
- ForEach iterated over each file record.

Step 3 – Copy Activity (inside ForEach)

- Inside ForEach, we placed a **Copy activity**.
- Source = Blob file path (parameterized from Lookup result).
- Sink = ADLS curated folder.
- Copy moved the actual file contents (GBs of data) from source → destination.

Get Metadata

The Get Metadata activity in Azure Data Factory (ADF) is a control flow activity that retrieves metadata from a specified data source.

This metadata can include details like file names, sizes, structure, or even the existence of a file or folder.

It is particularly useful for **dynamic pipelines** where decisions depend on the properties of the data.

1. **Supported Data Sources**: Works with various sources like Azure Blob Storage, Azure Data Lake, SQL databases, and more.

2. **Metadata Properties**: **(Field list)** You can retrieve properties such as:

- File name
- File size
- Last modified date
- Child items (for folders) – items which are available with this folder
- Column structure (for tables)
- Add dynamic

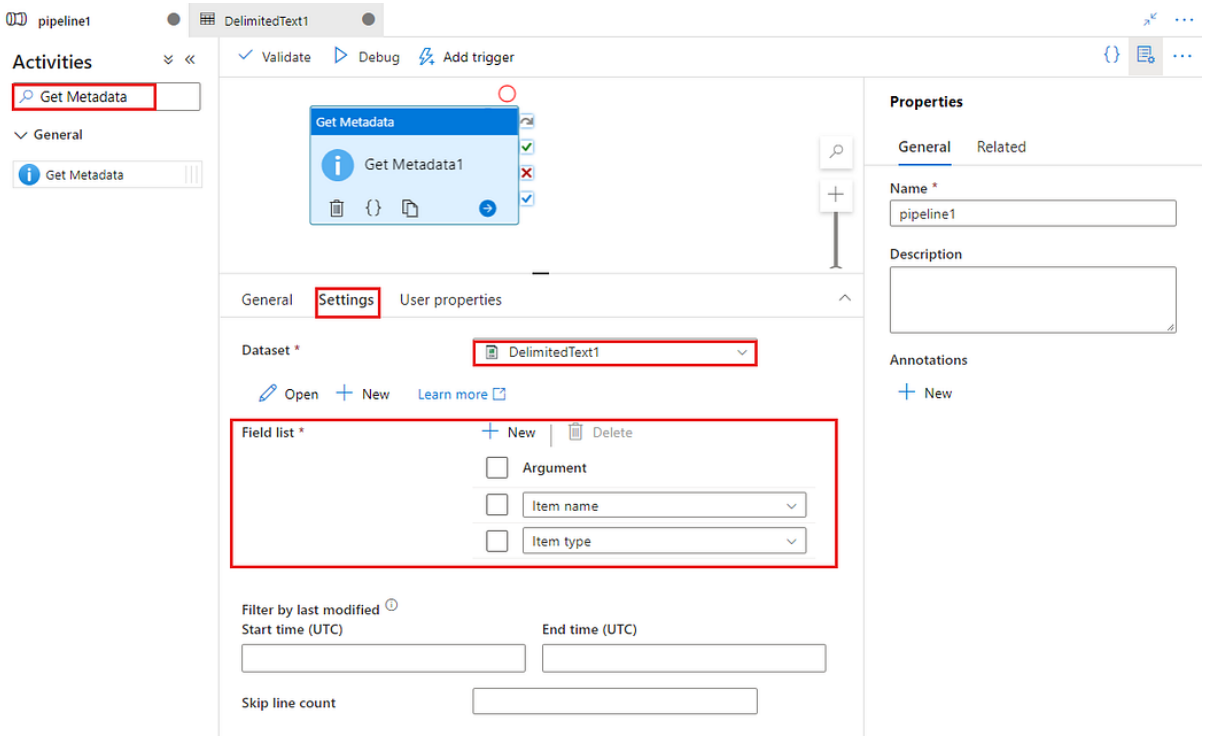
Metadata type	Description
itemName	Name of the file or folder.
itemType	Type of the file or folder. Returned value is `File` or `Folder`.
size	Size of the file, in bytes. Applicable only to files.
created	Created datetime of the file or folder.
lastModified	Last modified datetime of the file or folder.
childItems	List of subfolders and files in the given folder. Applicable only to folders. Returned value is a list of the name and type of each child item.
contentMD5	MD5 of the file. Applicable only to files.
structure	Data structure of the file or relational database table. Returned value is a list of column names and column types.
columnCount	Number of columns in the file or relational table.
exists	Whether a file, folder, or table exists. Note that if `exists` is specified in the Get Metadata field list, the activity won't fail even if the file, folder, or table doesn't exist. Instead, `exists: false` is returned in the output.

3. **Dynamic Pipelines**: Enables conditional logic based on metadata, such as checking if a file exists before processing.

Example Use Cases

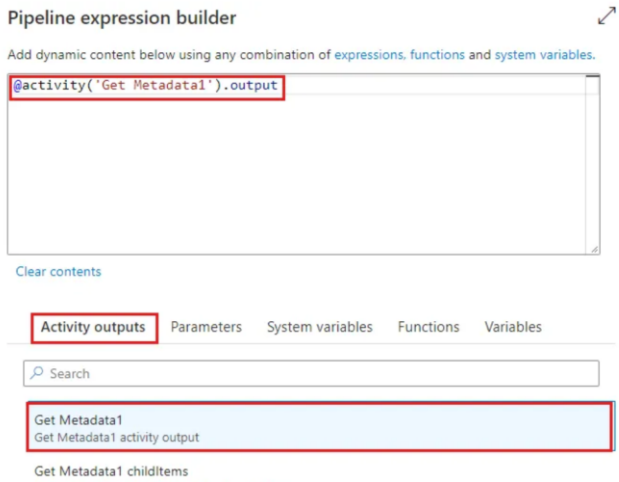
- **File Validation**: Check if a file exists before copying or transforming it.
- **Dynamic Processing**: Retrieve column names or structure to dynamically configure downstream activities.
- **Folder Iteration: List all files in a folder for batch processing.**

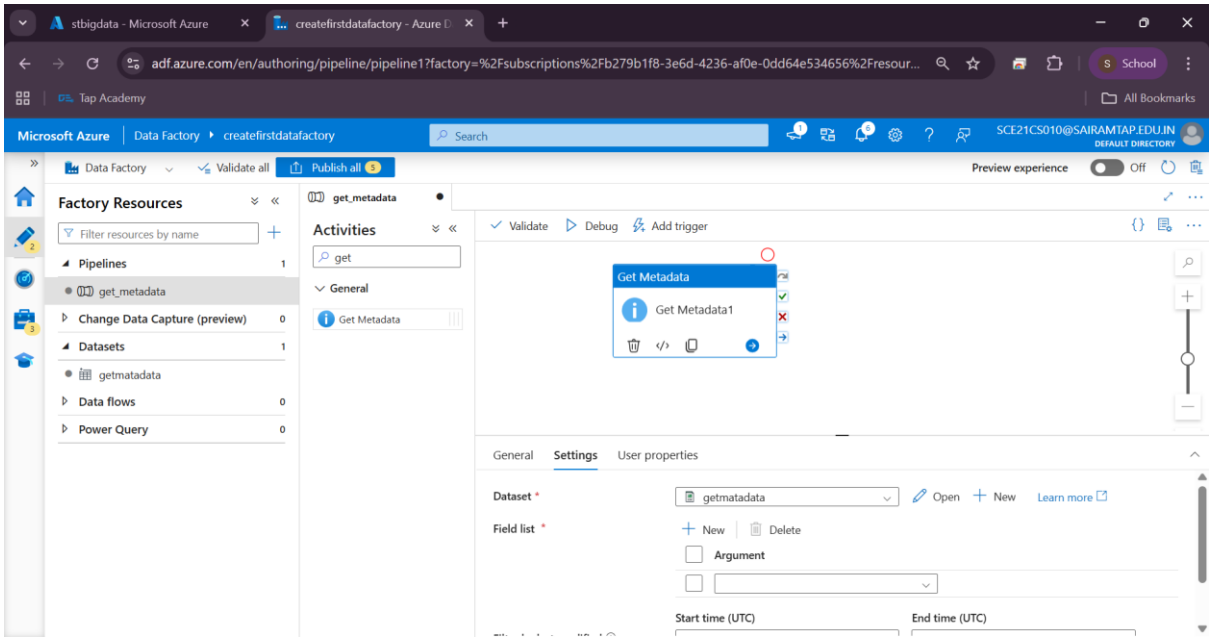
- **Usage:** Useful when you need to dynamically get information about files or datasets, such as file names, size, last modified timestamps, or even schema information.

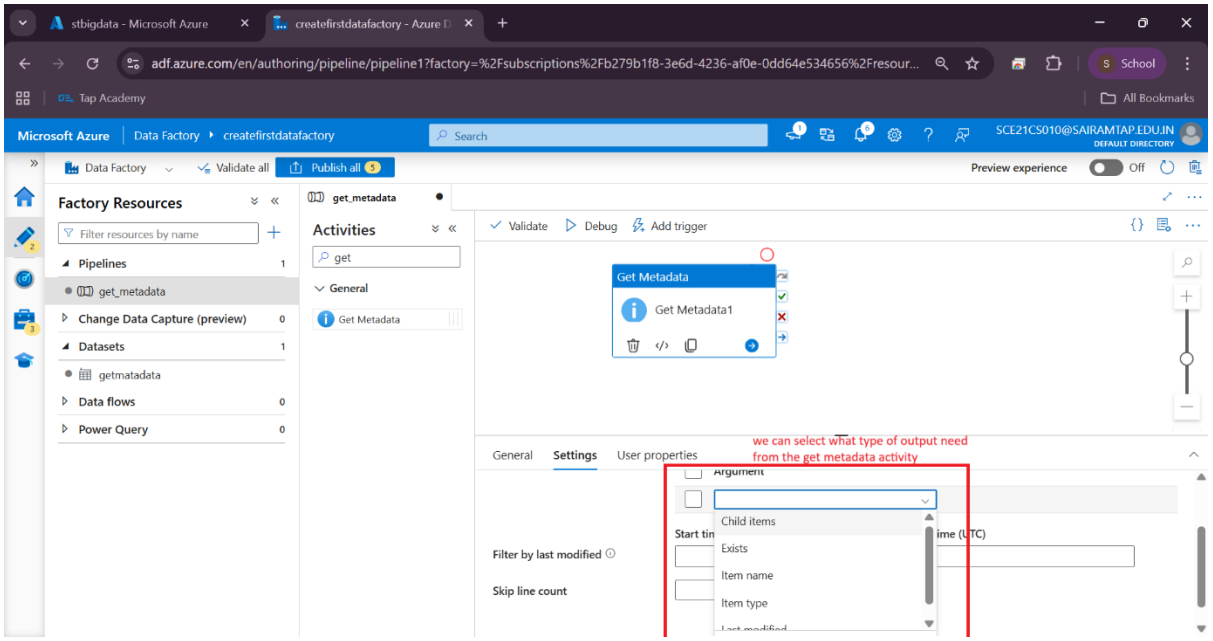


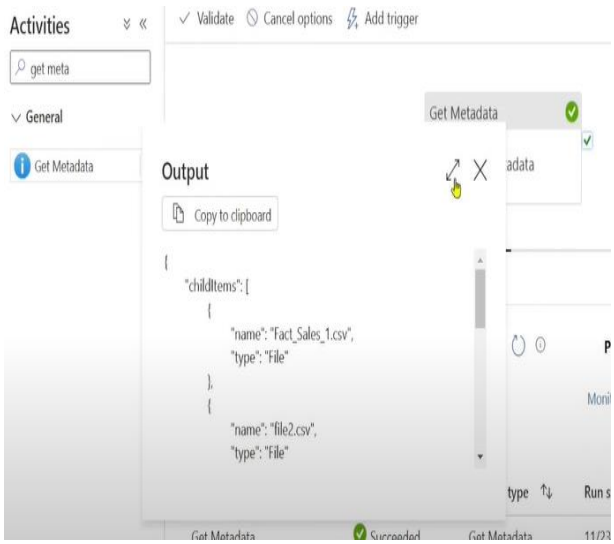
This activity is a powerful tool for building dynamic and intelligent data pipelines in ADF.

Use the Output: The output of this activity will be an array of file names. You can use this output to dynamically create pipelines or copy activities for each file.




stbigdata - Microsoft Azure
createfirstdatafactory - Azure D
adf.azure.com/en/authoring/pipeline/pipeline1?factory=%2Fsubscriptions%2Fb279b1f8-3e6d-4236-af0e-0dd64e534656%2Fresour...
Microsoft Azure
Data Factory
createfirstdatafactory
Search
SCE21CS010@SAIRAMTAP.EDU.IN
DEFAULT DIRECTORY
Validate all
Publish all 5
Preview experience
Off
Factory Resources
Filter resources by name
Pipelines 1
get_metadata
Change Data Capture (preview) 0
Datasets 1
getmatadata
Data flows 0
Power Query 0
Activities
get
General
Get Metadata
Validate
Debug
Add trigger
Get Metadata
Get Metadata1
General
Settings
User properties
Dataset *
getmatadata
Open
New
Learn more
Field list *
New
Delete
Argument
Start time (UTC)
End time (UTC)


stbigdata - Microsoft Azure
createfirstdatafactory - Azure D
adf.azure.com/en/authoring/pipeline/pipeline1?factory=%2Fsubscriptions%2Fb279b1f8-3e6d-4236-af0e-0dd64e534656%2Fresour...
Microsoft Azure
Data Factory
createfirstdatafactory
Search
SCE21CS010@SAIRAMTAP.EDU.IN
DEFAULT DIRECTORY
Validate all
Publish all 5
Preview experience
Off
Factory Resources
Filter resources by name
Pipelines 1
get_metadata
Change Data Capture (preview) 0
Datasets 1
getmatadata
Data flows 0
Power Query 0
Activities
get
General
Get Metadata
Validate
Debug
Add trigger
Get Metadata
Get Metadata1
General
Settings
User properties
we can select what type of output need from the get metadata activity
Argument
Child items
Exists
Item name
Item type
Filter by last modified
Skip line count

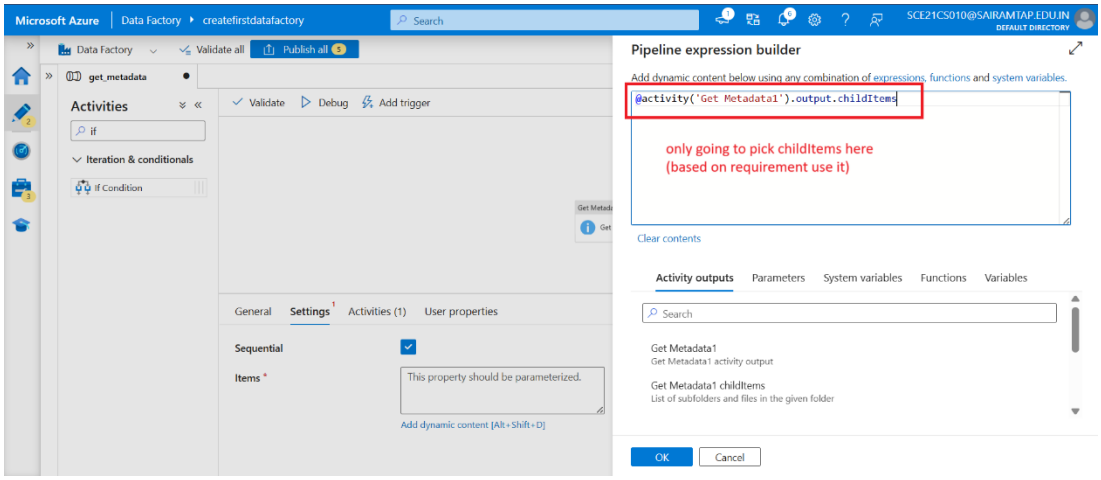

Activities
get meta
General
Get Metadata
Validate
Cancel options
Add trigger
Get Metadata
Output
Copy to clipboard
{
"childItems": [
{
"name": "Fact_Sales_1.csv",
"type": "File"
},
{
"name": "file2.csv",
"type": "File"

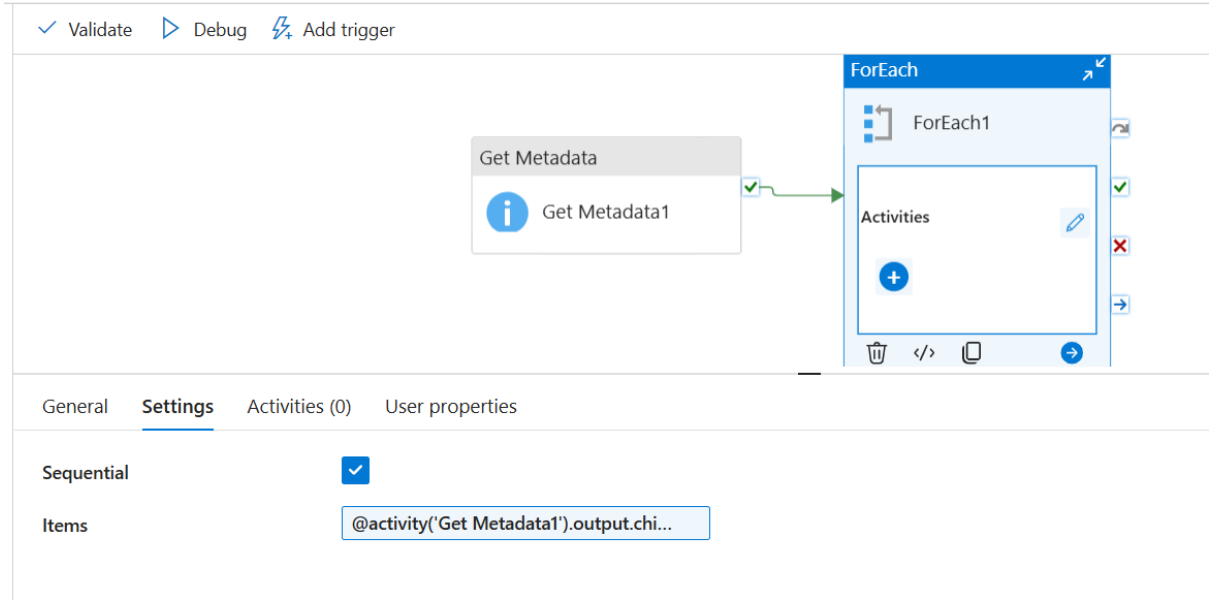
Output

Copy to clipboard

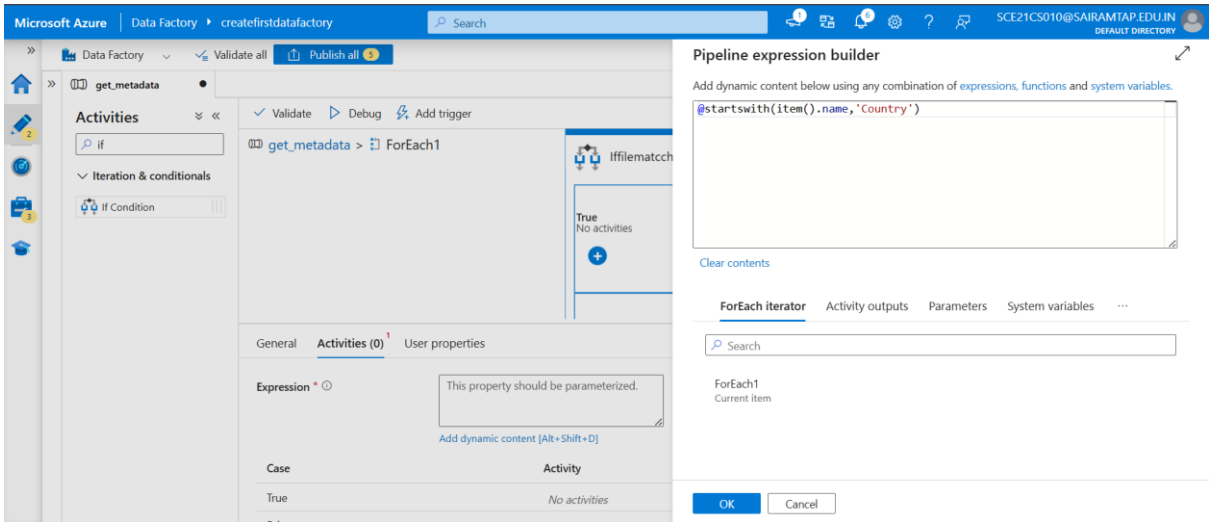

```
{
    "childItems": [
        {
            "name": "Fact_Sales_1.csv",
            "type": "File"
        },
        {
            "name": "file2.csv",
            "type": "File"
        }
    ],
    "effectiveIntegrationRuntime": "AutoResolveIntegrationRuntime (East US)",
    "executionDuration": 1,
    "durationInQueue": {
        "integrationRuntimeQueue": 0
    },
    "billingReference": {
        "activityType": "PipelineActivity",
        "billableDuration": [
            {
                "meterType": "AzureIR",
                "duration": 0.016666666666666666,
                "unit": "Hours"
            }
        ]
    }
}
```

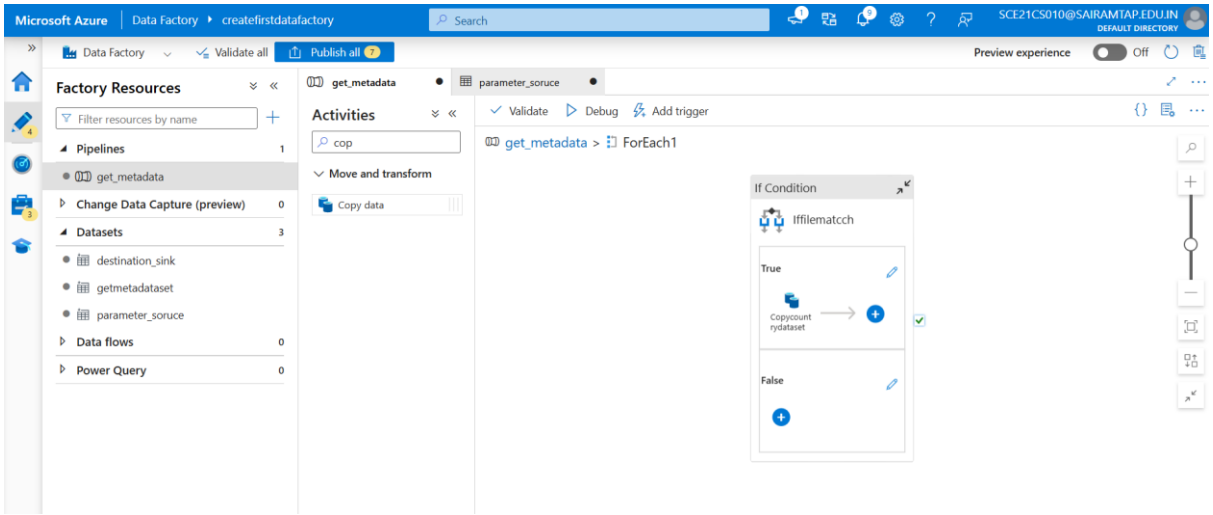

Now to connect the output of the get metadata to the for each activity (for taking the file name)





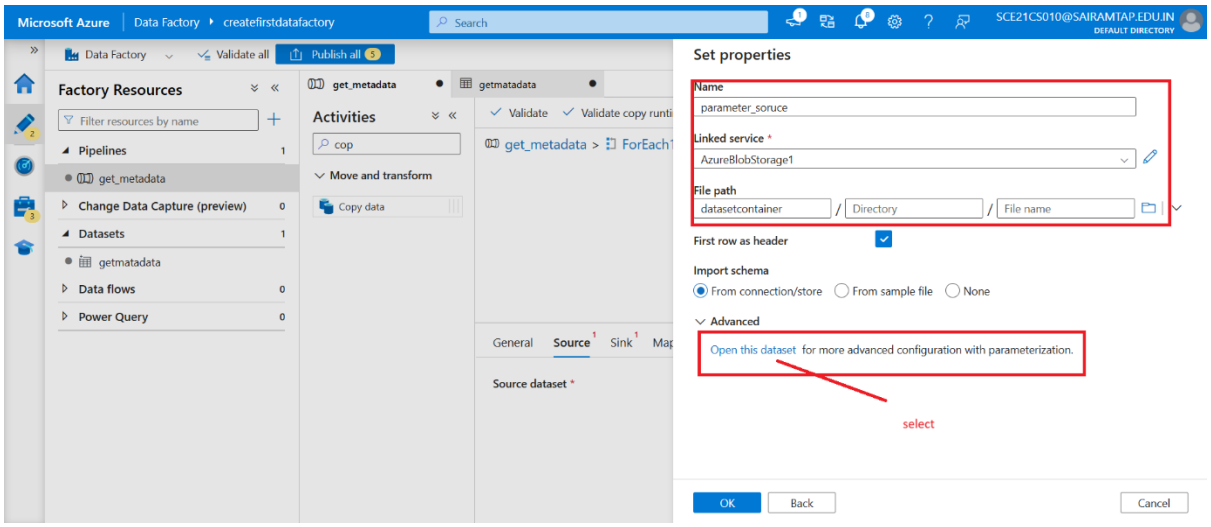
Set the condition inside the “if activity”

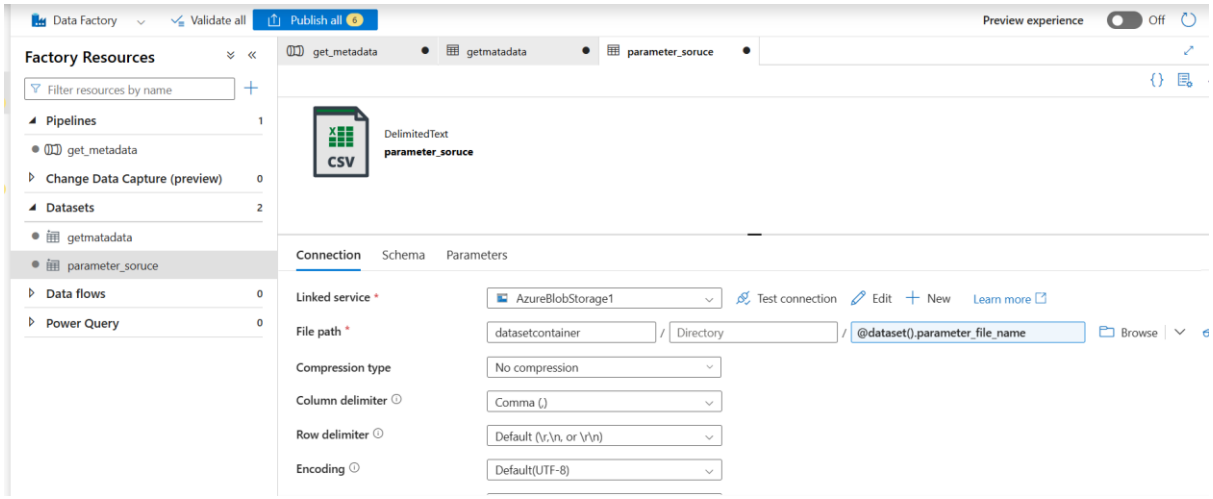


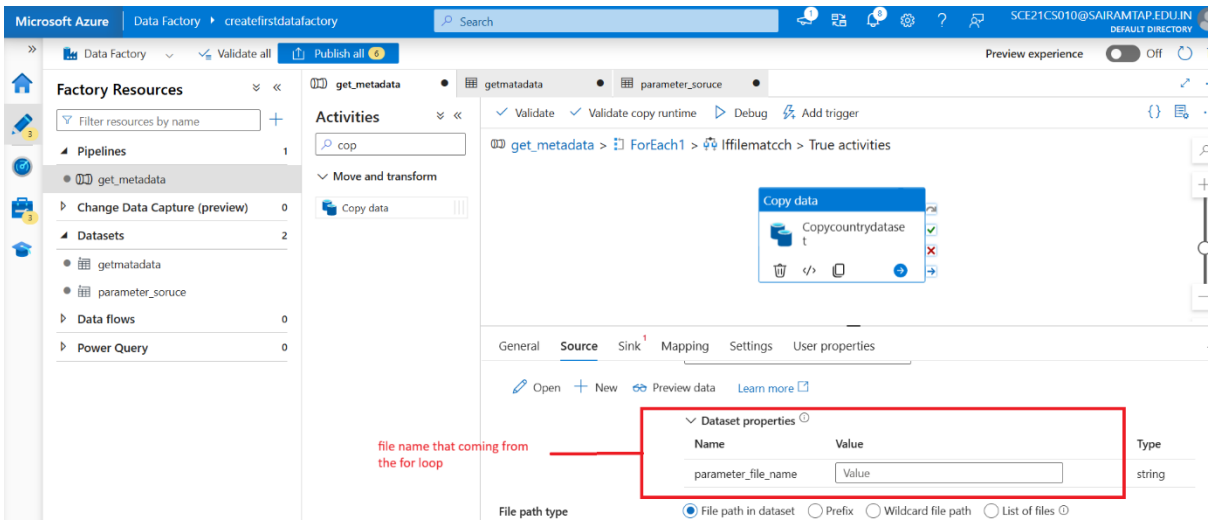


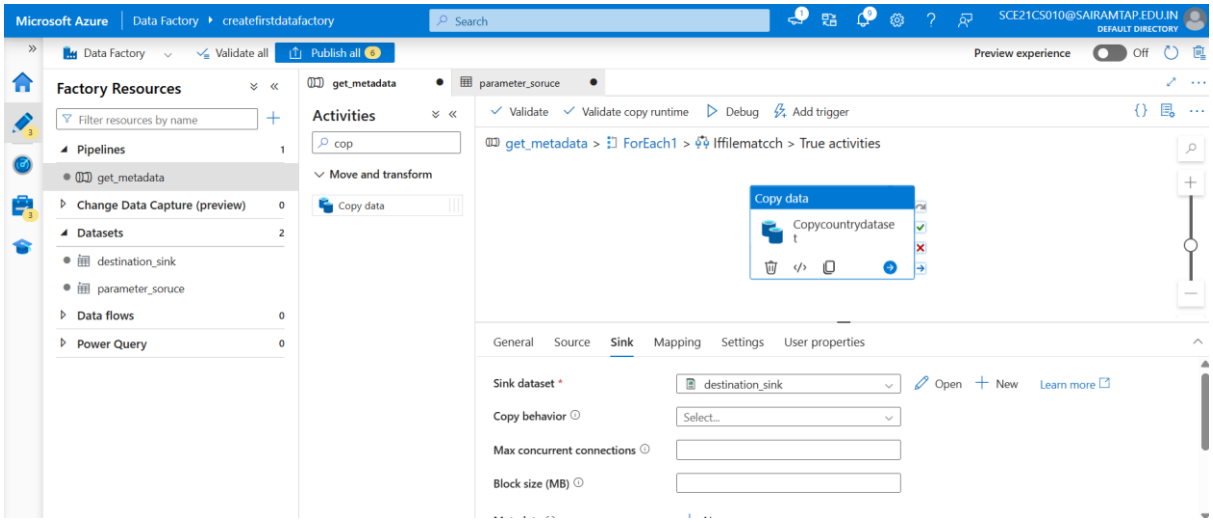
Step to Add the file name Dynamically

Create the dataset, while create it select the folder, directory, leave file name empty



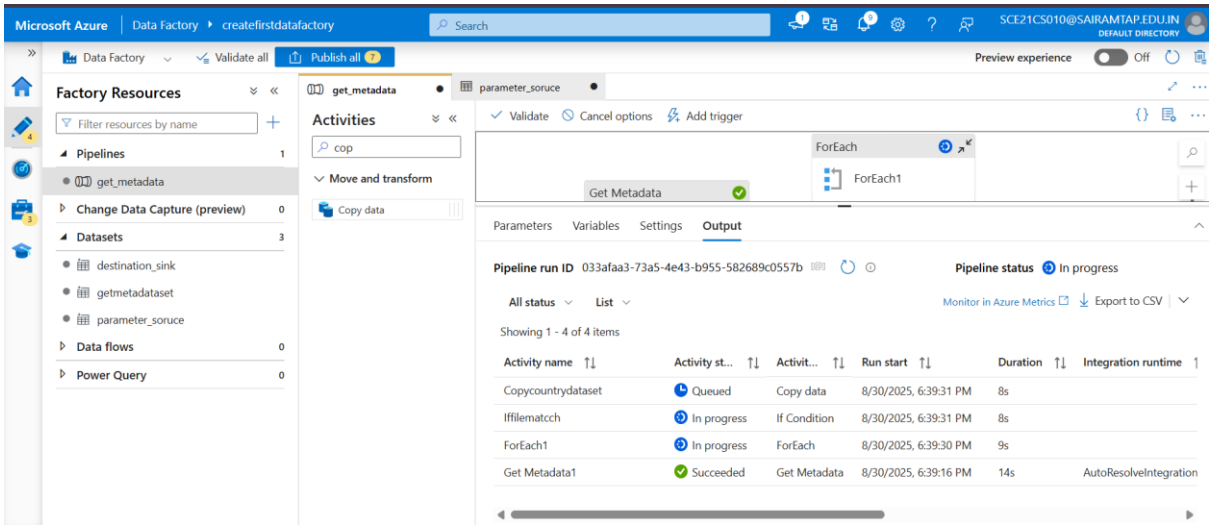


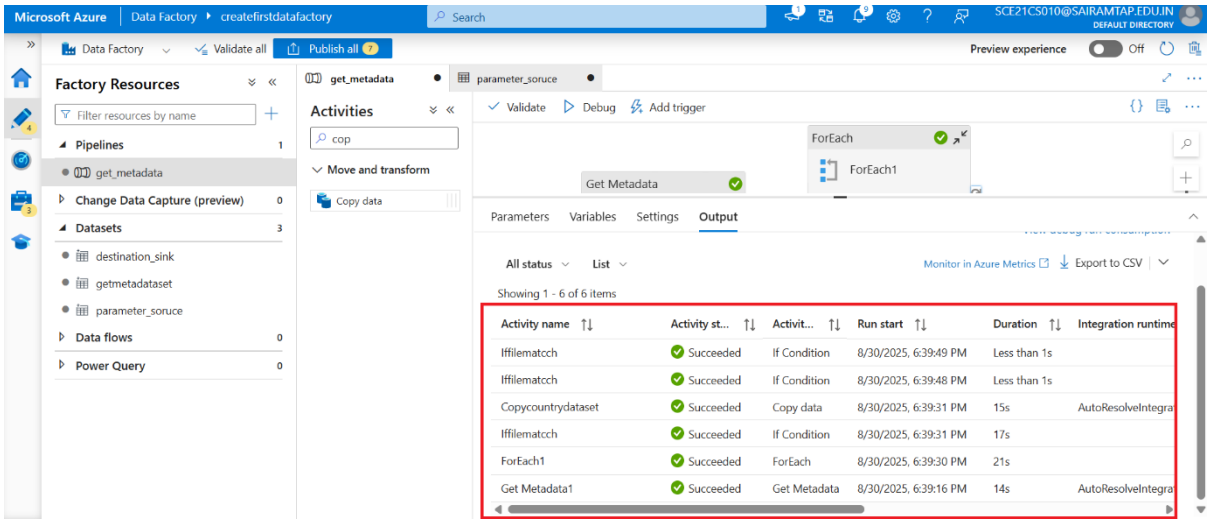




Get metadata → for each → if (True) → copy activity

output


Microsoft Azure
Data Factory
createfirstdatafactory
Search
SCE21CS010@SAIRAMTAP.EDU.IN
DEFAULT DIRECTORY
Validate all
Publish all 7
Preview experience
Off
Factory Resources
Filter resources by name
Pipelines 1
get_metadata
Change Data Capture (preview) 0
Datasets 3
destination_sink
getmetadataset
parameter_soruce
Data flows 0
Power Query 0
get_metadata
parameter_soruce
Activities
cop
Move and transform
Copy data
Validate
Cancel options
Add trigger
ForEach
ForEach1
Get Metadata
Parameters
Variables
Settings
Output
Pipeline run ID 033afaa3-73a5-4e43-b955-582689c0557b
Pipeline status In progress
All status
List
Monitor in Azure Metrics
Export to CSV
Showing 1 - 4 of 4 items
Activity name
Activity st...
Activit...
Run start
Duration
Integration runtime
Copycountrydataset Queued Copy data 8/30/2025, 6:39:31 PM 8s
Iffilematcch In progress If Condition 8/30/2025, 6:39:31 PM 8s
ForEach1 In progress ForEach 8/30/2025, 6:39:30 PM 9s
Get Metadata1 Succeeded Get Metadata 8/30/2025, 6:39:16 PM 14s AutoResolveIntegration

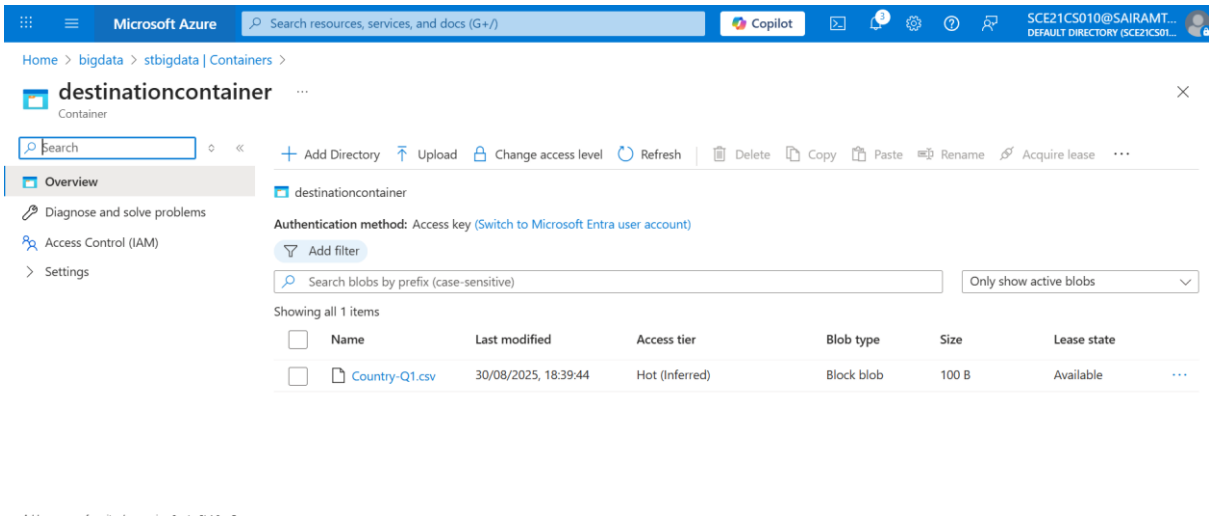

Microsoft Azure
Data Factory
createfirstdatafactory
Search
SCE21CS010@SAIRAMTAP.EDU.IN
DEFAULT DIRECTORY
Validate all
Publish all 7
Preview experience
Off
Factory Resources
Filter resources by name
Pipelines 1
get_metadata
Change Data Capture (preview) 0
Datasets 3
destination_sink
getmetadataset
parameter_soruce
Data flows 0
Power Query 0
get_metadata
parameter_soruce
Activities
cop
Move and transform
Copy data
Validate
Debug
Add trigger
ForEach
ForEach1
Get Metadata
Parameters
Variables
Settings
Output
All status
List
Monitor in Azure Metrics
Export to CSV
Showing 1 - 6 of 6 items
Activity name
Activity st...
Activit...
Run start
Duration
Integration runtime
Iffilematcch Succeeded If Condition 8/30/2025, 6:39:49 PM Less than 1s
Iffilematcch Succeeded If Condition 8/30/2025, 6:39:48 PM Less than 1s
Copycountrydataset Succeeded Copy data 8/30/2025, 6:39:31 PM 15s
Iffilematcch Succeeded If Condition 8/30/2025, 6:39:31 PM 17s
ForEach1 Succeeded ForEach 8/30/2025, 6:39:30 PM 21s
Get Metadata1 Succeeded Get Metadata 8/30/2025, 6:39:16 PM 14s

Output

Copy to clipboard

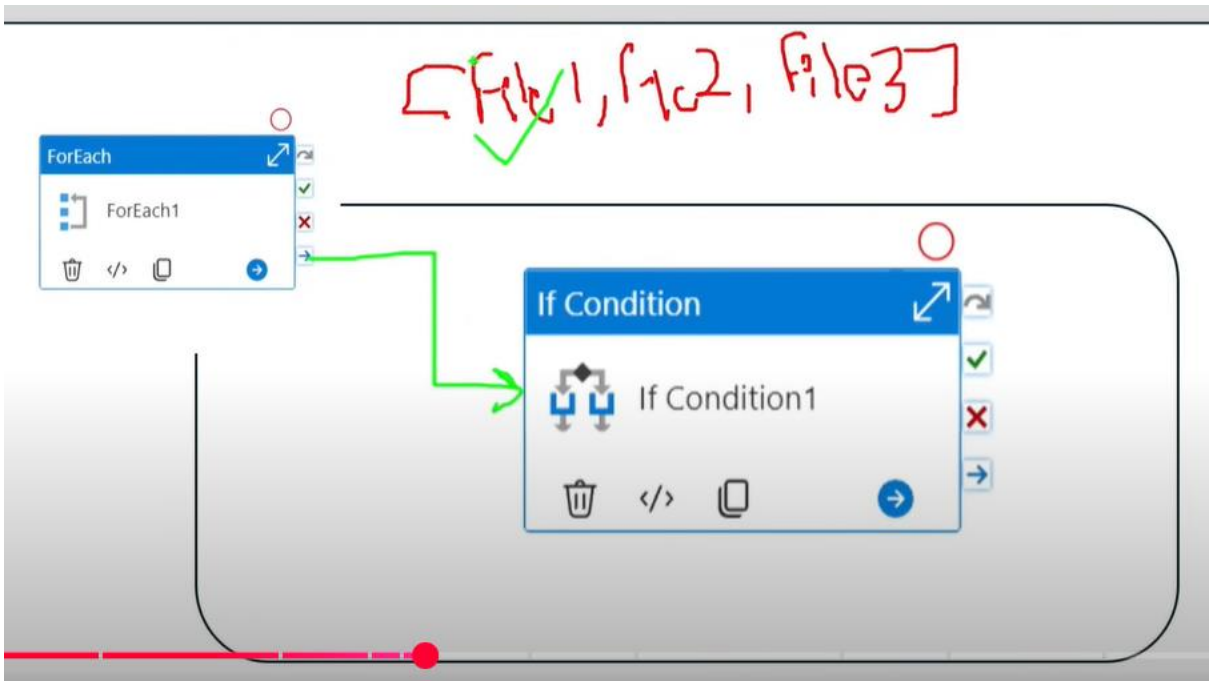

```
{
    "childItems": [
        {
            "name": "Country-Q1.csv",
            "type": "File"
        },
        {
            "name": "Department-Q1.csv",
            "type": "File"
        },
        {
            "name": "Employee-Q1.csv",
            "type": "File"
        }
    ],
    "effectiveIntegrationRuntime": "AutoResolveIntegrationRuntime (South India)",
    "executionDuration": 1,
    "durationInQueue": {
        "integrationRuntimeQueue": 10
    },
    "billingReference": {
        "activityType": "PipelineActivity",
        "billableDuration": [
            {
                "meterType": "AzureIR",
                "duration": 0.016666666666666666,
```


Only country dataset is copied



Feature	Lookup Activity	Get Metadata Activity

Returns	**Data values** (rows from source) (5000rows or 4 MB)	**Properties/metadata** of dataset
Scope	Works on **structured data sources** (SQL, files, API)	Works on **files, folders, tables**
Typical Use	Drive pipeline logic based on data	Drive pipeline logic based on file/dataset properties
Example	Get list of customer IDs from SQL	Get list of files in blob storage. (If you want to **get all CSV file names in a folder)**



Key Differences:

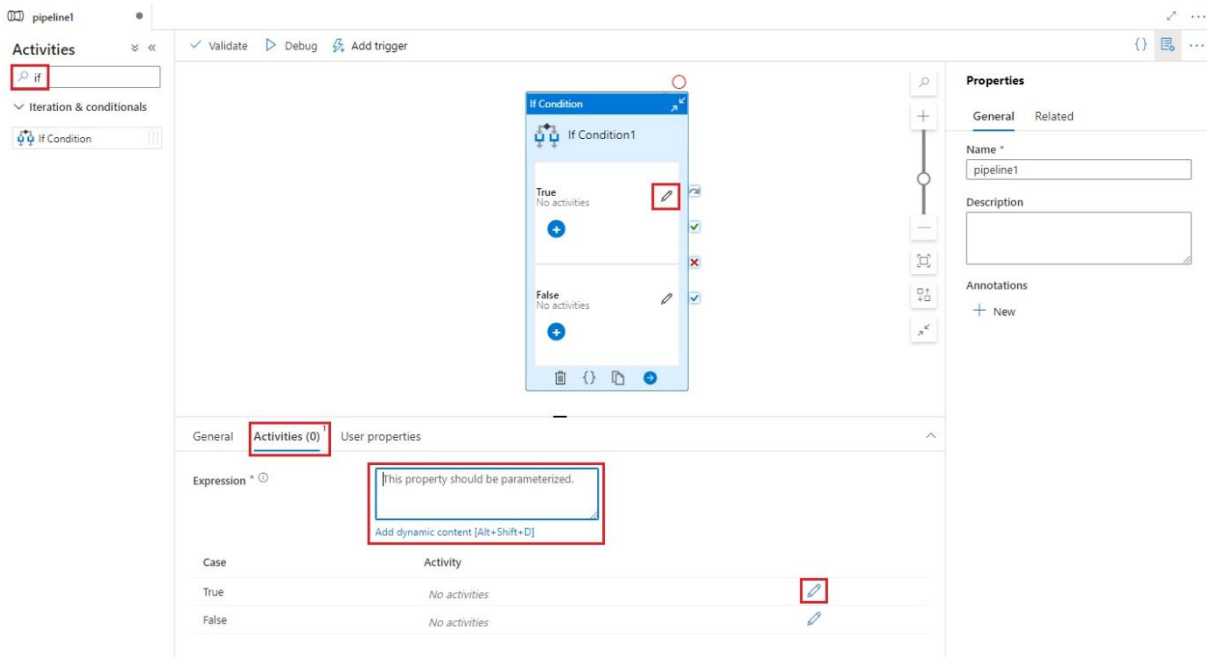
- **Functionality:** GetMetadata retrieves metadata about files or datasets, whereas Lookup retrieves data from a data source based on a query.
- **Use Case:** Use GetMetadata when you need information about files or datasets before processing. Use Lookup when you need to fetch specific data values or small datasets to use within your pipeline.
- **Output:** GetMetadata outputs metadata properties of files or datasets (like file names or schema details). Lookup outputs data rows from a query result.

If condition activity

The **If Condition activity** in Azure Data Factory (ADF) is a control flow activity that allows you to execute different sets of activities based on a logical condition. It evaluates an expression and executes one of two activity groups: one for when the condition evaluates to **true** and another for when it evaluates to **false**.

Key Features of If Condition Activity

- **Logical Evaluation**: You define an expression that evaluates to either true or false.
- **True and False Paths**:
  - If the condition is **true**, the activities in the **True** branch are executed.
  - If the condition is **false**, the activities in the **False** branch are executed.
- **Dynamic Content**: You can use dynamic expressions to evaluate variables, parameters, or outputs from other activities.

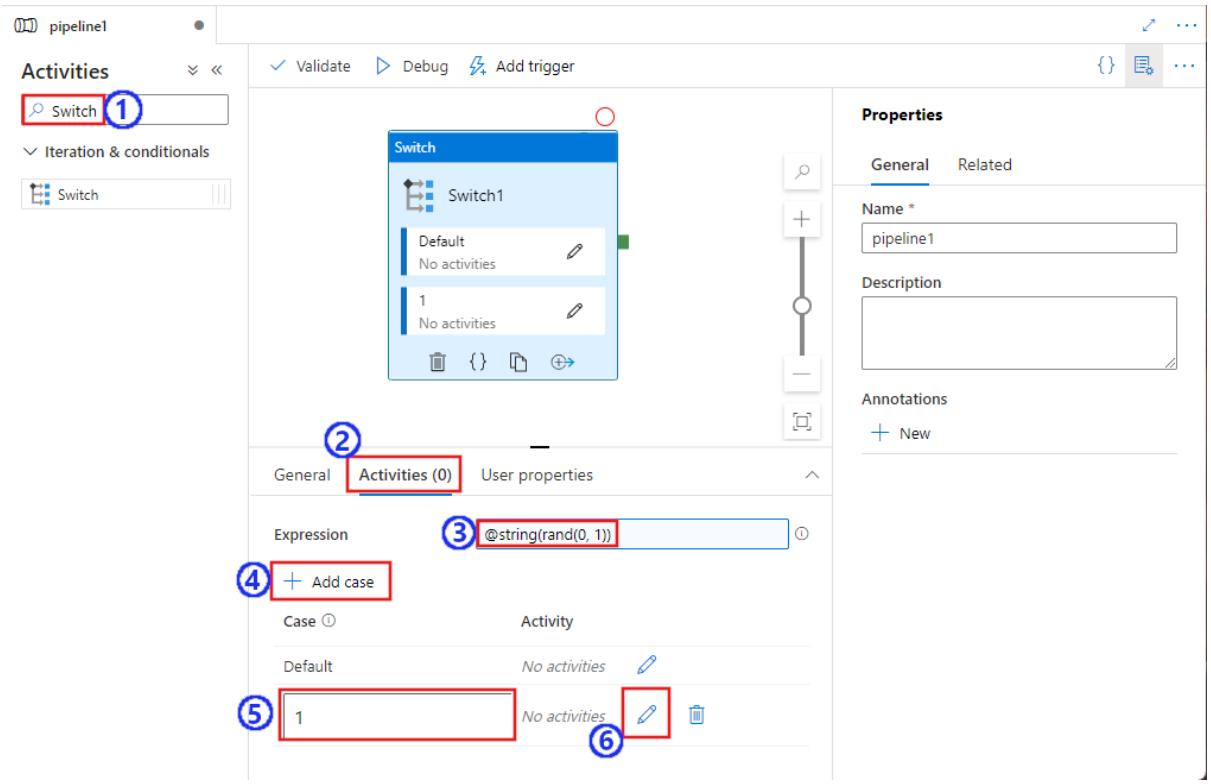


Key Points:

- **Expression-Based:** It evaluates an expression (typically using Azure Data Factory expressions) and executes activities based on the result (true or false).
- **Single Condition:** Supports only one condition evaluation per activity.
- **Control Flow:** Directs the workflow based on the evaluated condition.

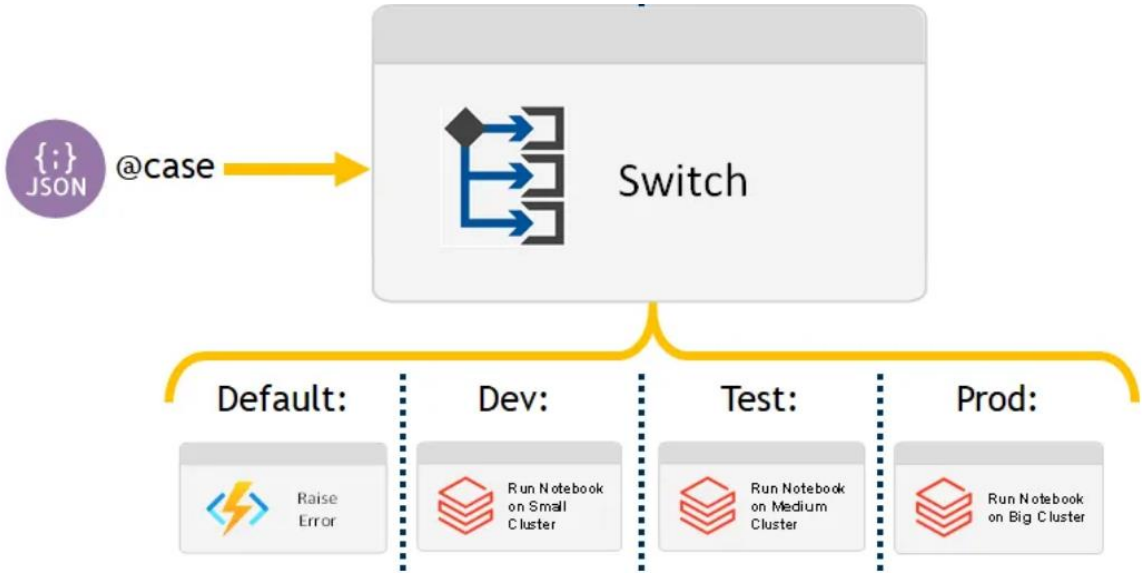
Switch Activity

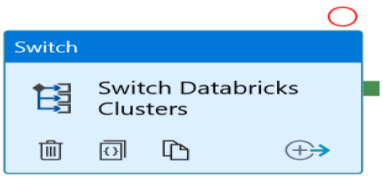
When dealing with bigger sets of circumstances, this method is frequently thought to be more effective than a block of If, Else If, Else If, Else type logic tests.

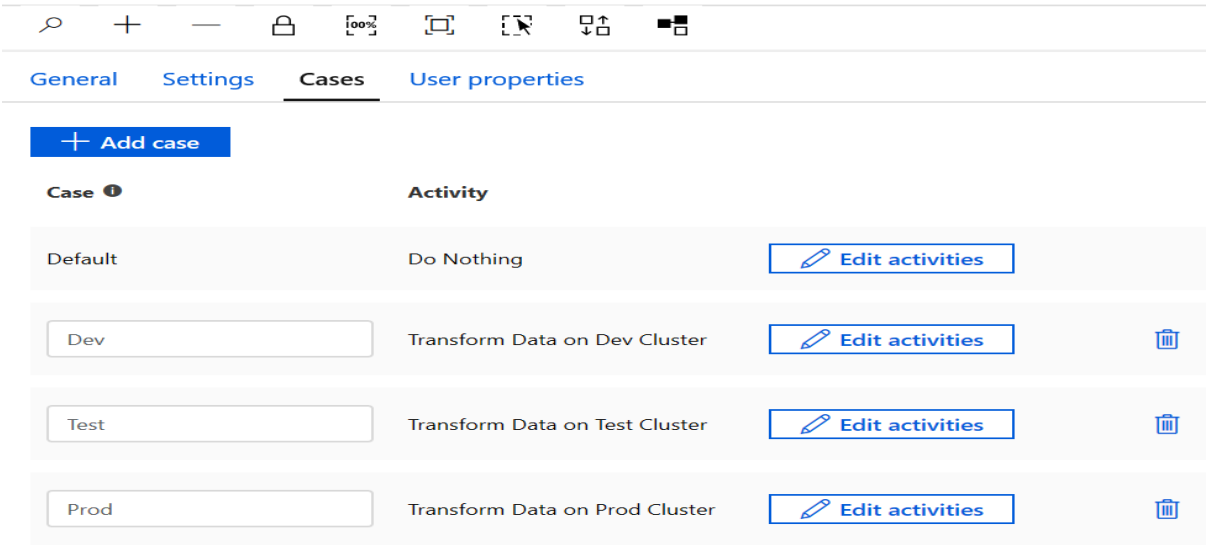


- **Use Cases:**

1. **Multiple Conditions:** Use Switch when you have multiple conditions to evaluate and based on each condition, you want to execute different sets of activities. For instance, processing files of different types or performing different transformations based on data characteristics.
2. **Dynamic Branching:** It provides a structured way to handle complex branching logic where you need to evaluate multiple expressions and route based on different outcomes.



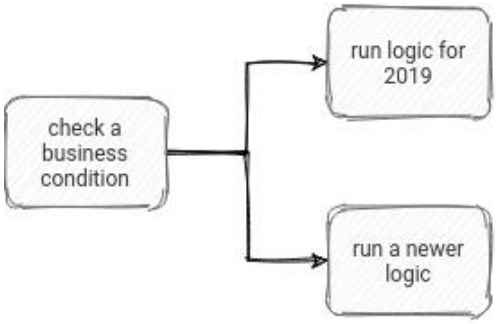




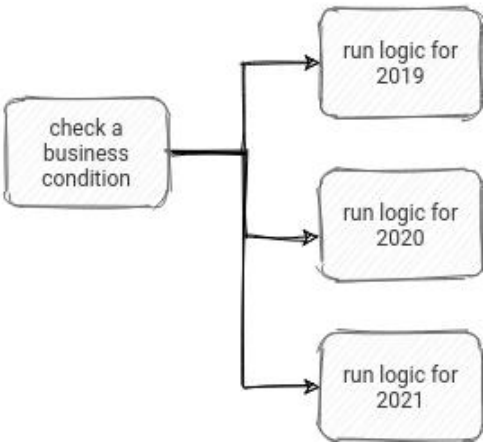
Key Points:

- **Multi-Condition:** Allows defining multiple conditions (cases) and corresponding activities or control flows for each case.
- **Structured Flow:** Offers a more structured approach to handle multiple scenarios compared to nested If Condition activities.
- **Enhanced Readability:** Improves pipeline readability when dealing with several branching scenarios by organizing them into distinct cases.

If-else conditional

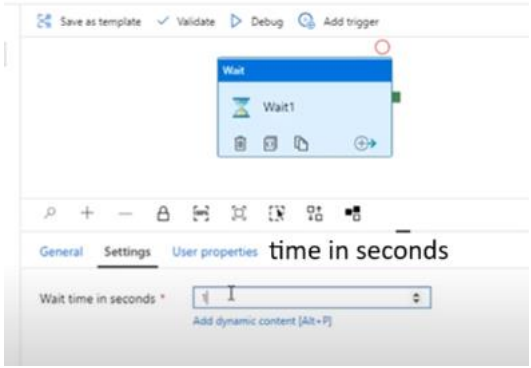


switch conditional



Wait Activity in ADF

When you use a wait activity in a pipeline, the pipeline waits for the specified period of time before continuing with execution of subsequent activities.

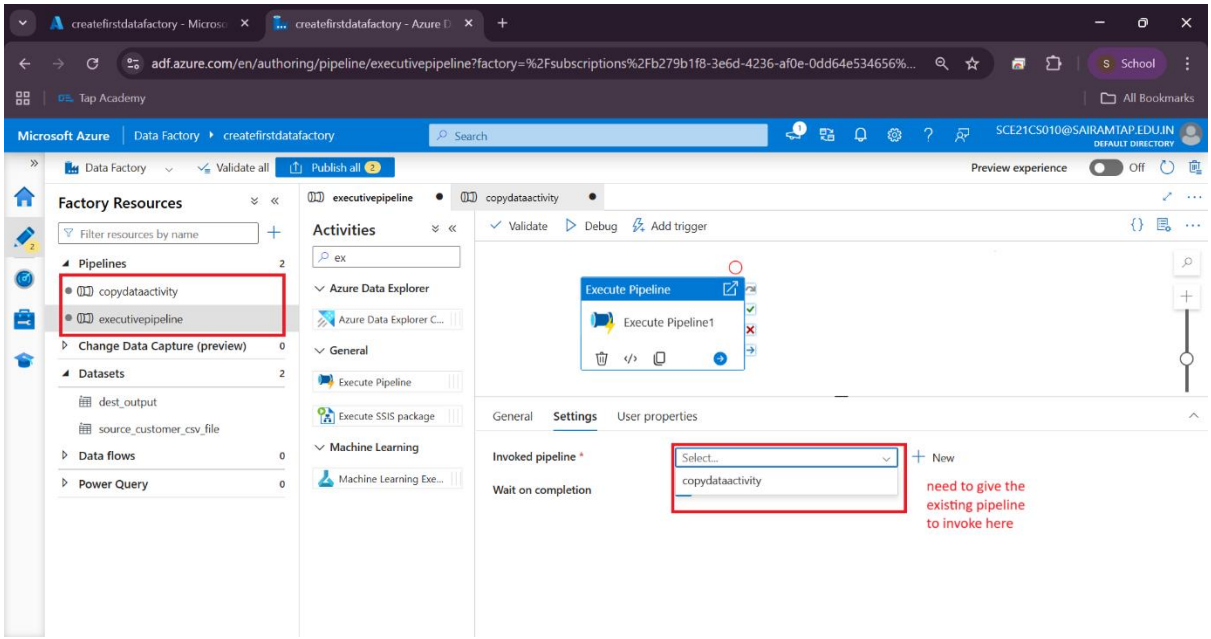


Execute Pipeline Activity:

This activity is essential when you need to chain pipelines together and execute them in a sequence, ensuring a smooth workflow from one process to another.

Execute Pipeline Activity in Azure Data Factory allows one pipeline to invoke and execute another pipeline, enabling modular, reusable, and sequential workflows.

This activity allows you to call and run another pipeline within a pipeline. It helps you break down complex workflows into smaller, modular components, which can then be executed in a sequence or based on certain conditions.



Wait on completion: wait for the selected pipeline(child pipeline) to complete (if checked)

There is difference between parameter and variable:

For Parameter value is passed in the Triggers

Parameter are external values passed into pipelines, datasets or linked services. The value cannot be changed inside a pipeline.

Variable → set values for these variables inside pipelines

variable are internal values set inside a pipeline. The value can be changed inside the pipeline using set variable or append variable activity.

Variable in ADF:

Variables are like internal to pipeline and they can be changed inside your pipeline.

Variables support 4 data type: **string, bool, array, integer**

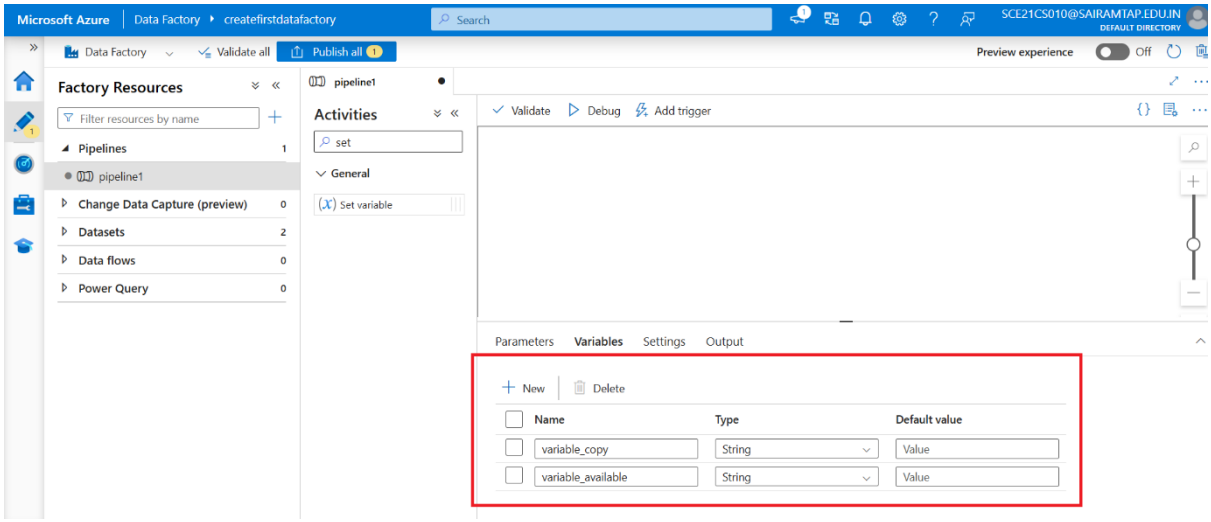
@variables('variableName')

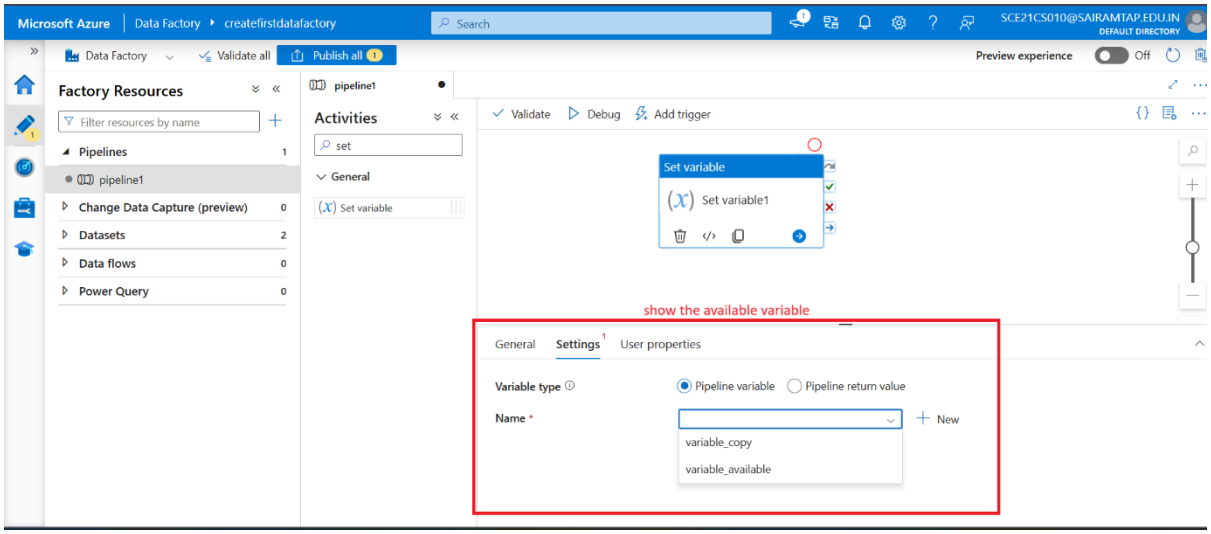
Set variable activity

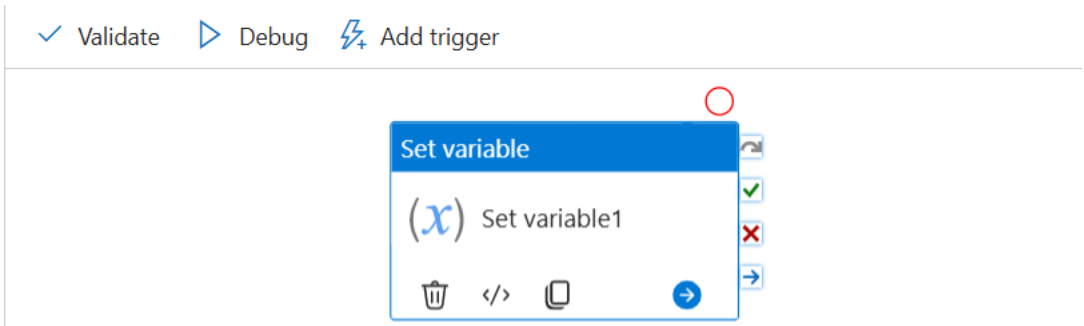
If we want to set the value for the variable during the pipeline execution, we can use this.

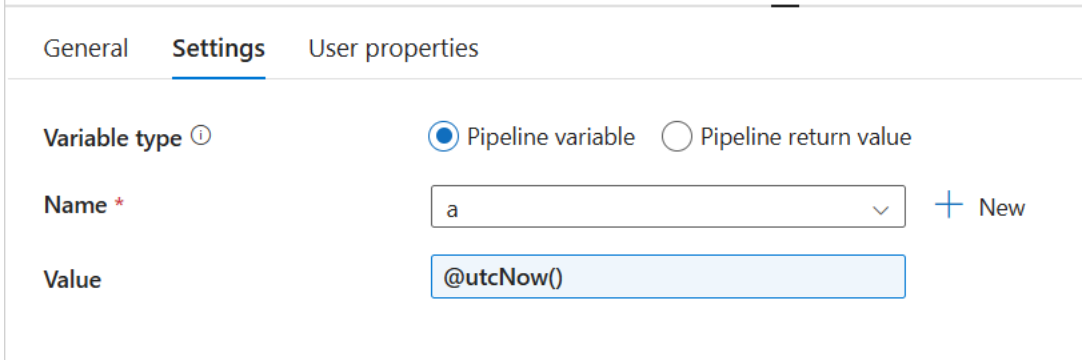
Use the set variable activity to set the value of an existing variable of

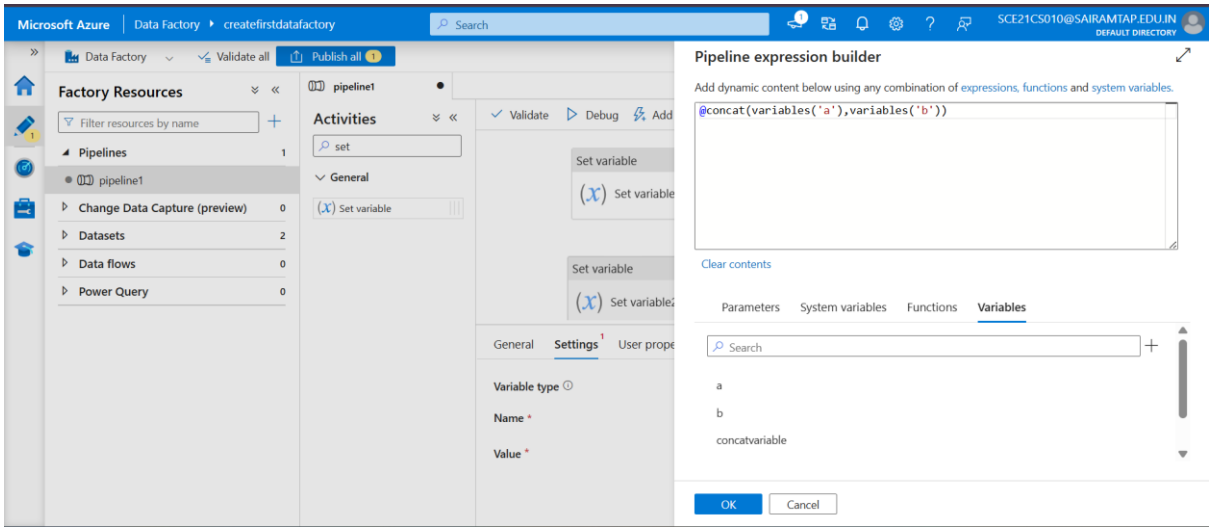
type **string, bool or array** defined in a Data factory pipeline.



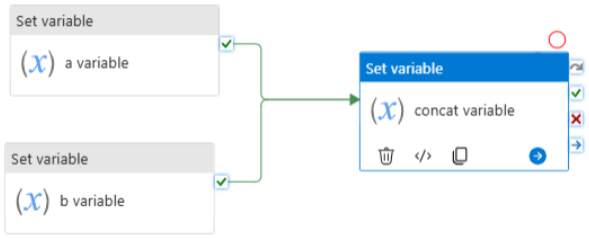

Microsoft Azure
Data Factory
createfirstdatafactory
Search
Validate all
Publish all
Preview experience
Off
Factory Resources
Filter resources by name
Pipelines
pipeline1
Change Data Capture (preview)
Datasets
Data flows
Power Query
Activities
set
General
Set variable
Validate
Debug
Add trigger
Set variable1
show the available variable
General
Settings
User properties
Variable type
Pipeline variable
Pipeline return value
Name *
New
variable_copy
variable_available

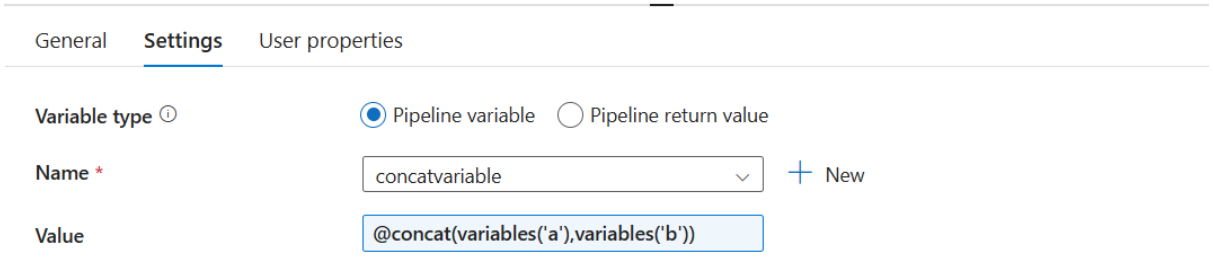

Validate
Debug
Add trigger
Set variable
Set variable1

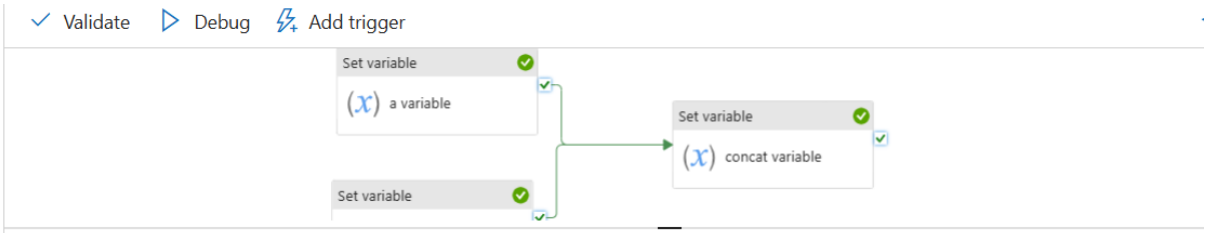

General
Settings
User properties
Variable type
Pipeline variable
Pipeline return value
Name *
a
New
Value
@utcNow()


Microsoft Azure
Data Factory
createfirstdatafactory
Search
SCE21CS010@SAIRAMTAP.EDU.IN
DEFAULT DIRECTORY
Data Factory
Validate all
Publish all
Factory Resources
Filter resources by name
Pipelines
pipeline1
Change Data Capture (preview)
Datasets
Data flows
Power Query
Activities
set
General
Set variable
Validate
Debug
Add
Set variable
Set variable
General
Settings
User prope
Variable type
Name *
Value *
Pipeline expression builder
Add dynamic content below using any combination of expressions, functions and system variables.
@concat(variables('a'),variables('b'))
Clear contents
Parameters
System variables
Functions
Variables
Search
a
b
concatvariable
OK
Cancel


Validate
Debug
Add trigger


Set variable
a variable
Set variable
b variable
Set variable
concat variable


General
Settings
User properties
Variable type
Pipeline variable
Pipeline return value
Name *
concatvariable
New
Value
@concat(variables('a'),variables('b'))



Parameters Variables Settings **Output**

Pipeline run ID aac1da36-5e66-4edc-9adf-7360930348fd **Pipeline status** Succeeded View debug run co

All status Monitor in Azure Metrics Export to

Showing 1 - 3 of 3 items

Activity name	Activity st...	Activit...	Run start	Duration	Integration runtim
concat variable	Succeeded	Set variable	9/2/2025, 12:43:04 PM	Less than 1s	
b variable	Succeeded	Set variable	9/2/2025, 12:43:04 PM	Less than 1s	
a variable	Succeeded	Set variable	9/2/2025, 12:43:04 PM	Less than 1s	

Output

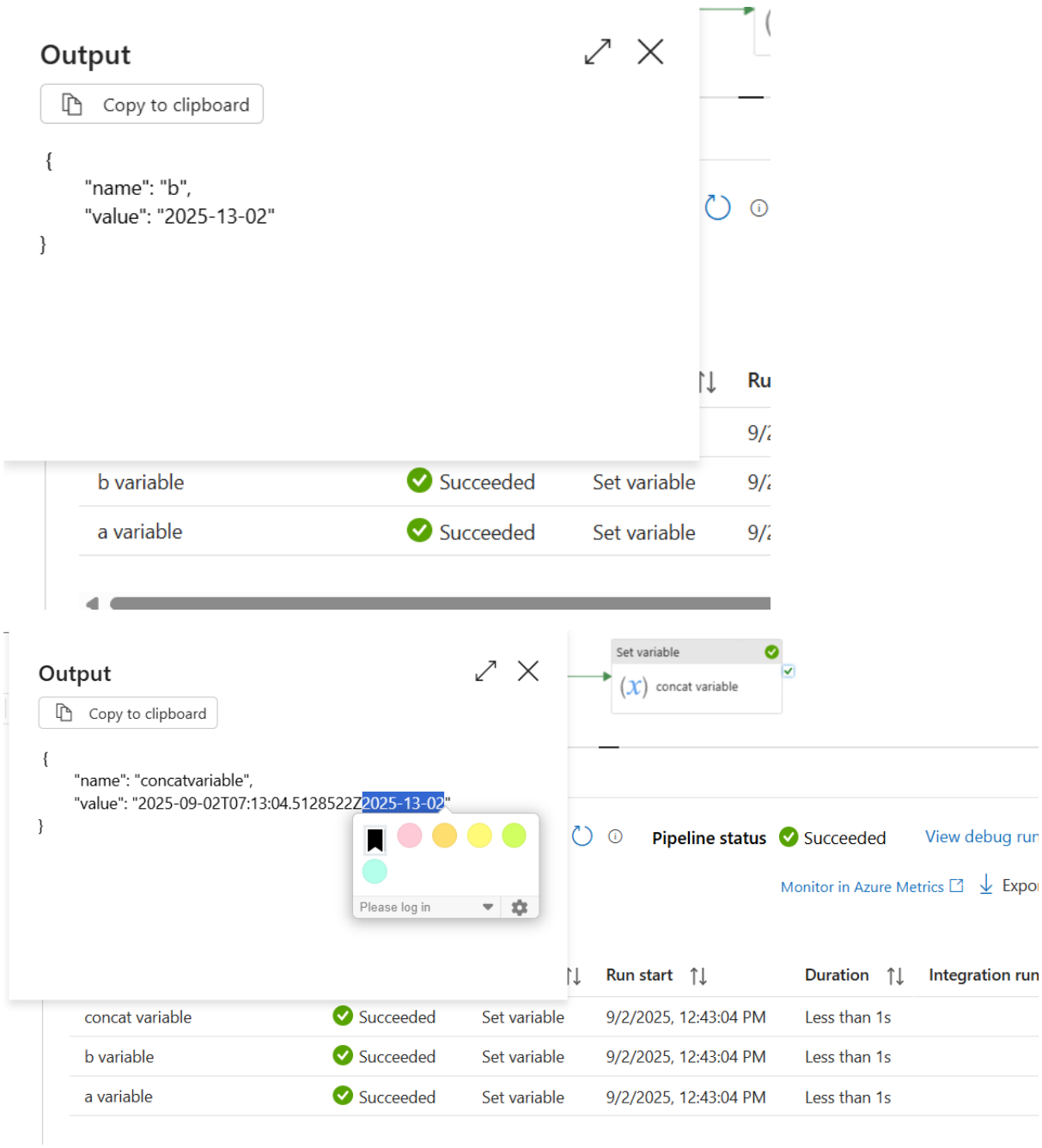
Copy to clipboard


```
{
    "name": "a",
    "value": "2025-09-02T07:13:04.5128522Z"
}
```


Pipeline status Succeeded

Monitor in Azure Met

		Run start	Duration
		9/2/2025, 12:43:04 PM	Less than 1s
		9/2/2025, 12:43:04 PM	Less than 1s
a variable	Succeeded	9/2/2025, 12:43:04 PM	Less than 1s



Append Variable

The **Append Variable** activity is used to add (append) values to an **array variable** during pipeline execution.

- You **cannot use it with string or int variables**, only with **Array type variables**.
- Each execution appends a new element to the existing array.

Steps to Use Append Variable

1. **Create a Variable** in your pipeline (type = Array).

"variables": {

```
  "FileList": {
    "type": "Array",
    "defaultValue": []
  }
}
```

2. **Add Append Variable Activity**
   - Select the variable (FileList).
   - Provide the value to append (can be dynamic expression).

@concat('file_', pipeline().RunId, '.csv')

Example

Let's say you loop through a list of file names and want to store them:

- Initial variable (FileList): []
- First Append Variable: append "customer.csv" → ["customer.csv"]
- Second Append Variable: append "sales.csv" → ["customer.csv", "sales.csv"]
- Third Append Variable: append "product.csv" → ["customer.csv", "sales.csv", "product.csv"]

Real-time scenarios

- Collect file names processed in a loop.
- Store failed records file names for logging.
- Build a dynamic list of values to use later in the pipeline (for email notifications, logging tables, etc.).

Store Procedure

A stored procedure is a prepared SQL code that you can save, so the code can be reused again.

[What is stored procedure (sql)? A **stored procedure** is a **pre-compiled set of SQL statements** (and optional control logic) that you save inside a database and execute by name whenever you need it].

Stored Procedure activity

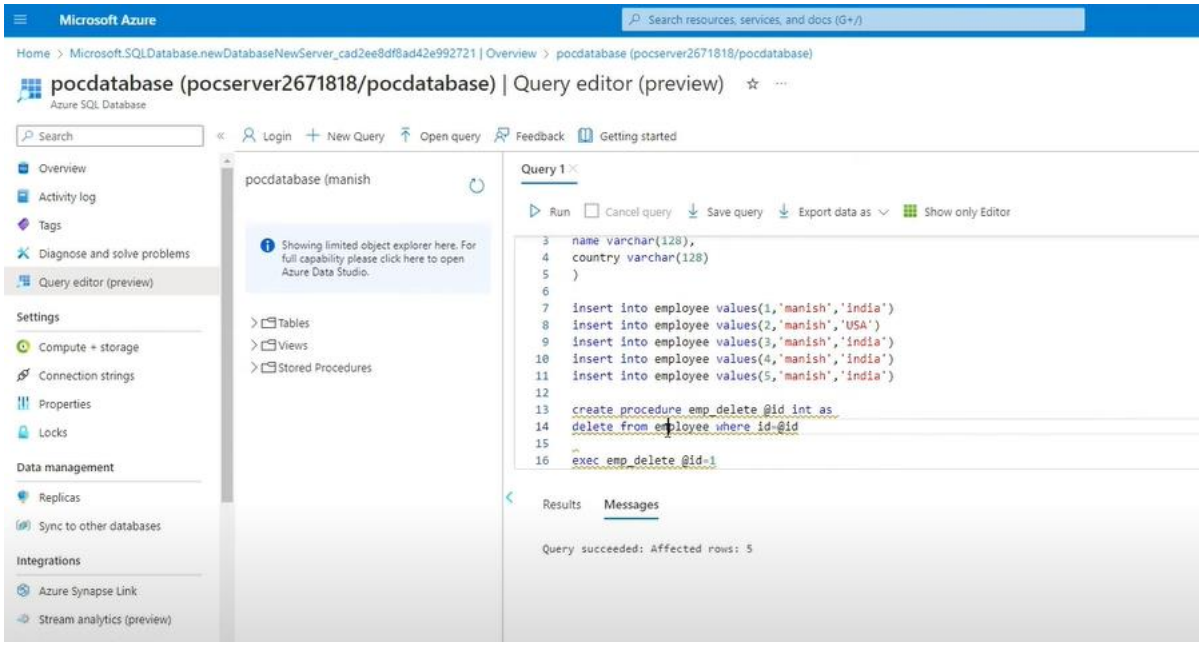
The **Stored Procedure activity** lets you call a stored procedure that already exists inside a database (for example Azure SQL Database, Azure Synapse SQL Pool, or SQL Server).

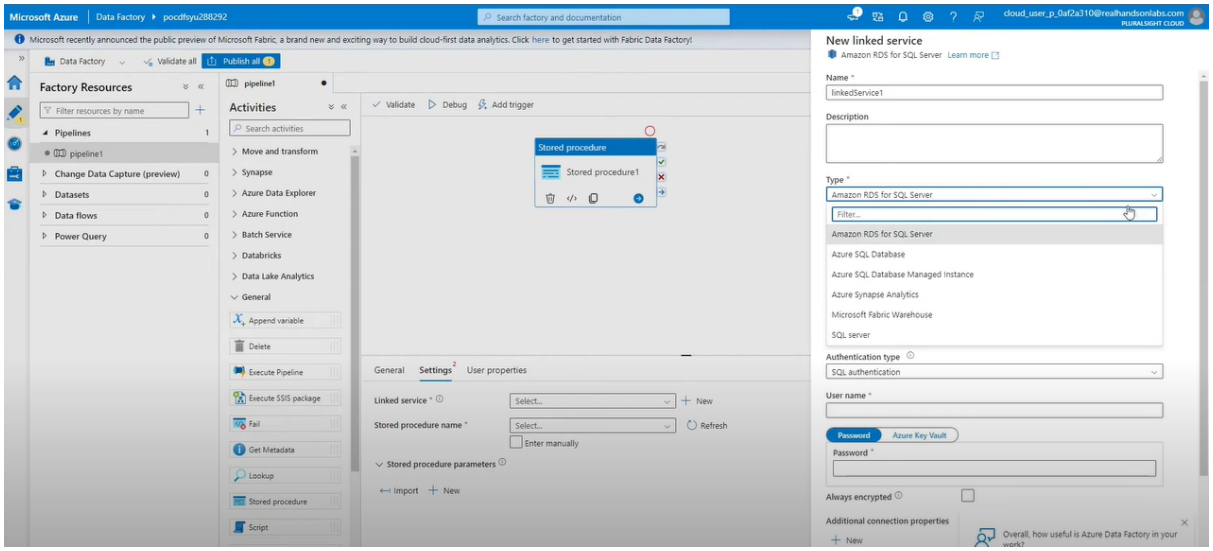
So, if you have an SQL query that you write again, save it as a stored procedure, and then just call it to execute it.

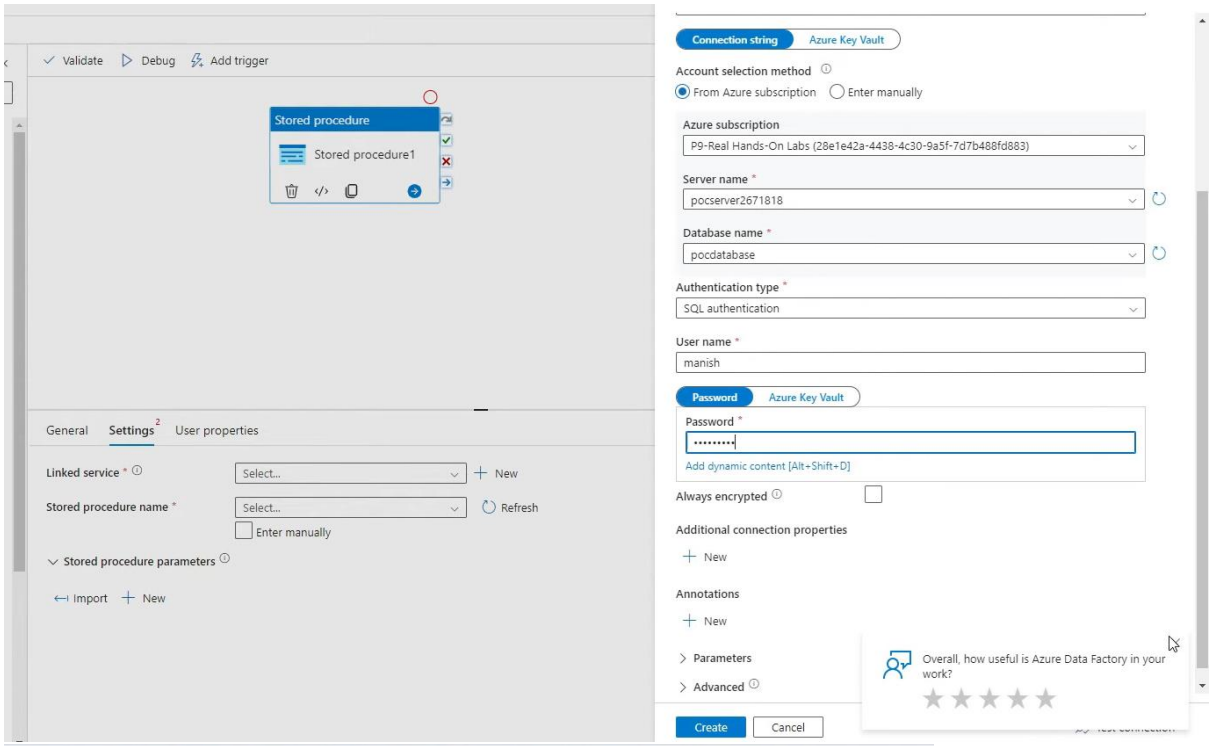
You can also pass parameters to a stored procedure, so that the stored procedure can act based on the parameters value that is passed.

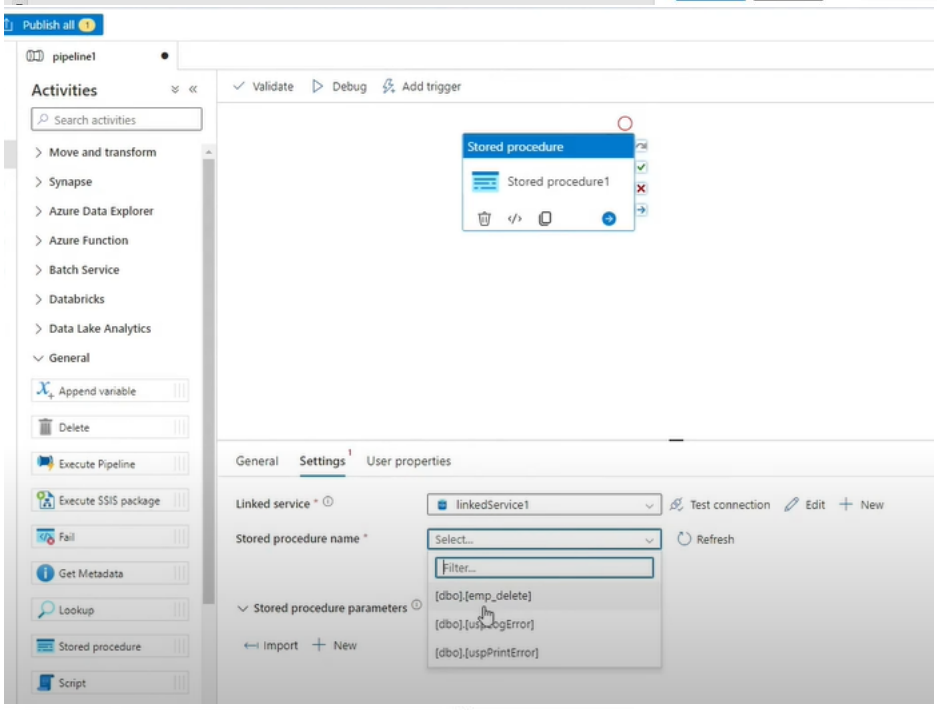
You can use the stored procedure in

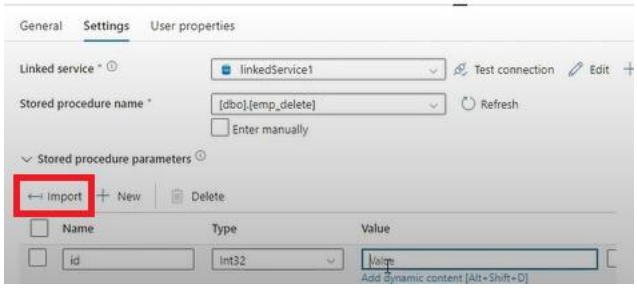
- Azure SQL Database
- Azure synapse analytics
- SQL server database


Microsoft Azure
Search resources, services, and docs (G+/)
Home > Microsoft.SQLDatabase.newDatabaseNewServer_cad2ee8df8ad42e992721 | Overview > pocdatabase (pocserver2671818/pocdatabase)
pocdatabase (pocserver2671818/pocdatabase) | Query editor (preview)
Azure SQL Database
Search
Overview
Activity log
Tags
Diagnose and solve problems
Query editor (preview)
Settings
Compute + storage
Connection strings
Properties
Locks
Data management
Replicas
Sync to other databases
Integrations
Azure Synapse Link
Stream analytics (preview)
Login
New Query
Open query
Feedback
Getting started
pocdatabase (manish
Showing limited object explorer here. For full capability please click here to open Azure Data Studio.
Tables
Views
Stored Procedures
Query 1
Run
Cancel query
Save query
Export data as
Show only Editor
name varchar(128),
country varchar(128)
)
insert into employee values(1,'manish','india')
insert into employee values(2,'manish','USA')
insert into employee values(3,'manish','india')
insert into employee values(4,'manish','india')
insert into employee values(5,'manish','india')
create procedure emp_delete @id int as
delete from employee where id=@id
exec emp_delete @id=1
Results
Messages
Query succeeded: Affected rows: 5


Microsoft Azure
Data Factory
Factory Resources
Pipelines
pipeline1
Change Data Capture (preview)
Datasets
Data flows
Power Query
Activities
Move and transform
Synapse
Azure Data Explorer
Azure Function
Batch Service
Databricks
Data Lake Analytics
General
Append variable
Delete
Execute Pipeline
Execute SSIS package
Fail
Get Metadata
Lookup
Stored procedure
Script
Validate
Debug
Add trigger
Stored procedure
Stored procedure1
General
Settings
User properties
Linked service *
Stored procedure name *
Enter manually
Stored procedure parameters
Import
New
Refresh
New linked service
Amazon RDS for SQL Server
Learn more
Name *
linkedService1
Description
Type *
Amazon RDS for SQL Server
Filter...
Amazon RDS for SQL Server
Azure SQL Database
Azure SQL Database Managed Instance
Azure Synapse Analytics
Microsoft Fabric Warehouse
SQL server
Authentication type
SQL authentication
User name *
Password
Azure Key Vault
Password *
Always encrypted
Additional connection properties
New







Until activity

The until activity is a looping activity in ADF pipeline

It repeatedly executes one or more activity until a condition is met (like a WHILE LOOP)

(Until it work the given condition becomes TRUE)

Expression (Condition):

- Defines when the loop should **stop**.
- Written using ADF expression functions (e.g., @equals(variables('status'),'Success')).

Activities inside Until:

- You can put multiple activities inside (Copy, Lookup, Stored Proc, Web activity, etc.).
- They will repeat until the condition is true.

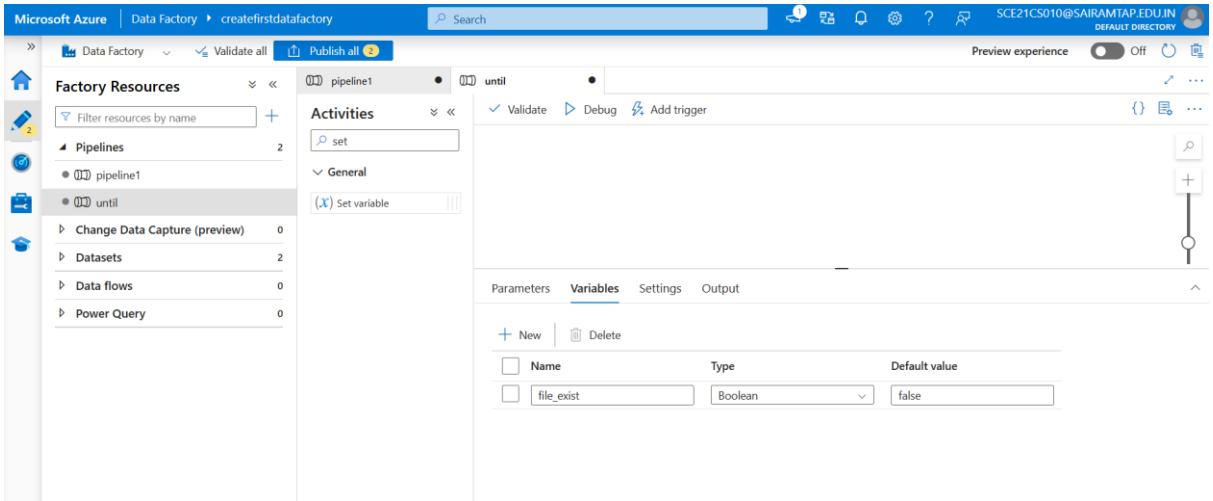
Timeout:

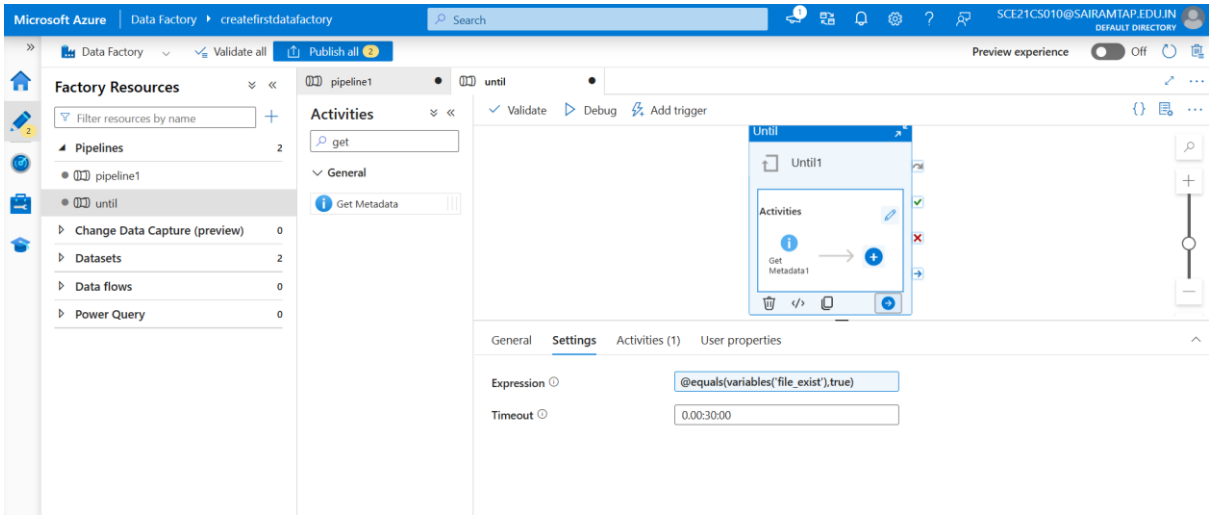
- Prevents infinite looping.
- Example: 7.00:00:00 → loop stops after 7 days even if condition is not met.

Real-Time Uses

- Waiting for files to land in storage.
  (Wait Until File Arrives
  Suppose ADF must wait until a file is available in Azure Blob before proceeding)
- Polling an external API until processing completes.
- Checking for flag/status in a control table.
- Retrying logic until resource is ready.

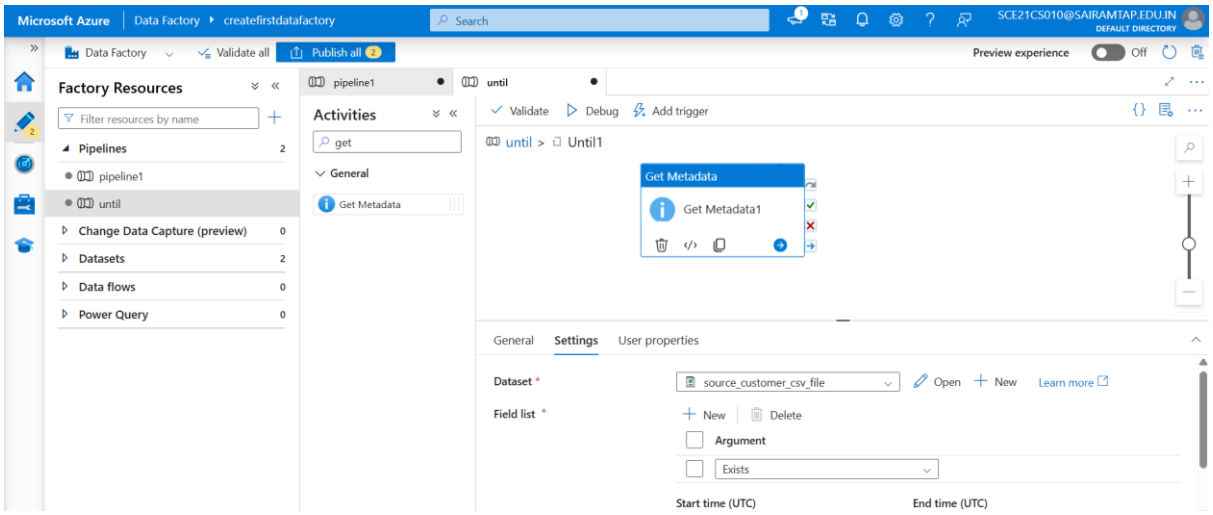
Set the variable as **false (until it becomes TRUE this will until activity will work)**



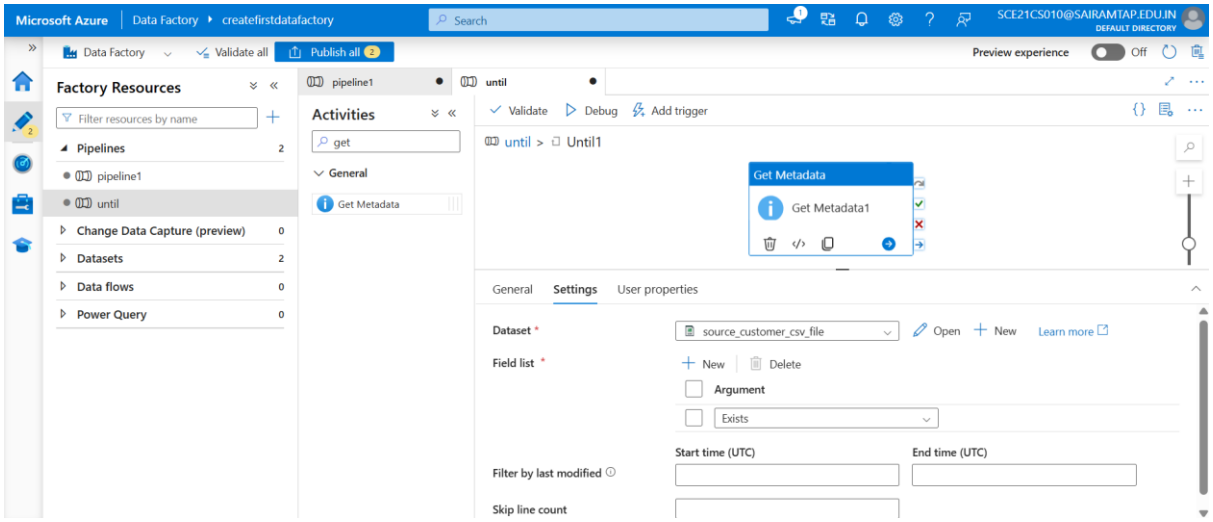


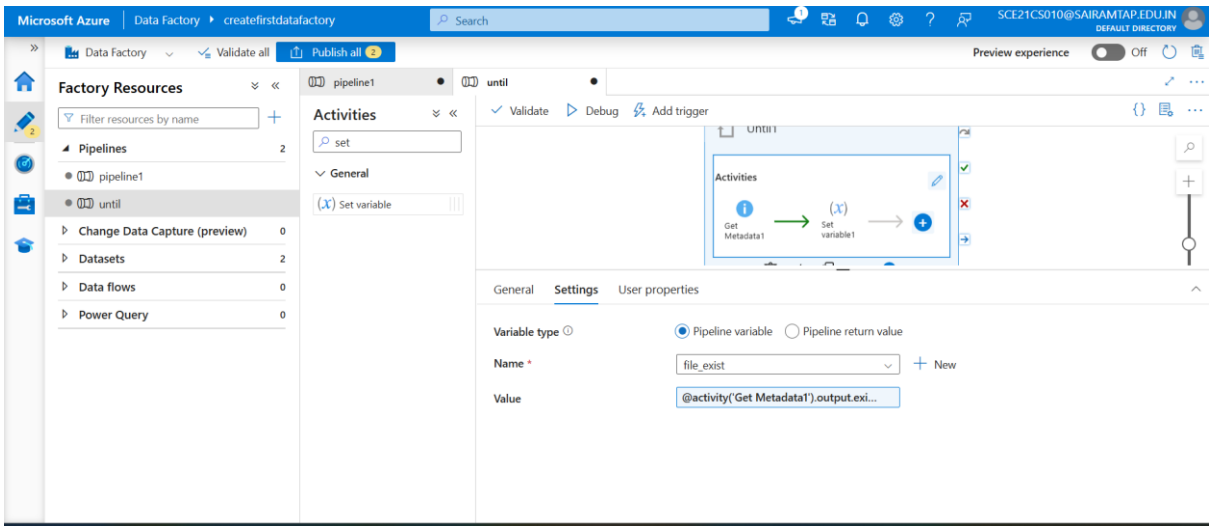
Get Metadata Activity

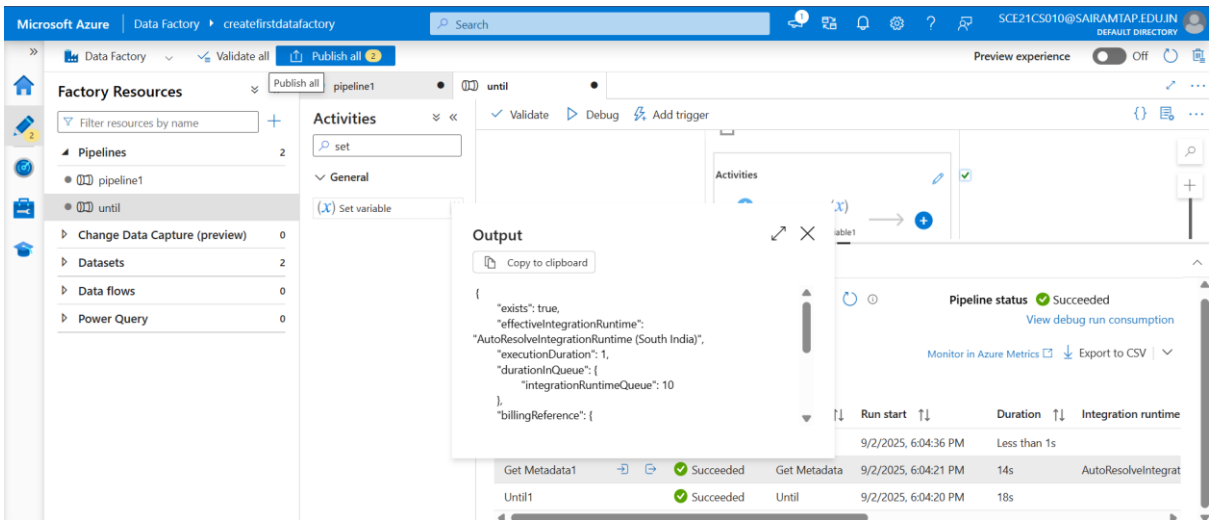
- Point it to the **file path** in your Blob dataset.
- Under *Field list*, check **Exists**.
- This will return "exists": true/false.

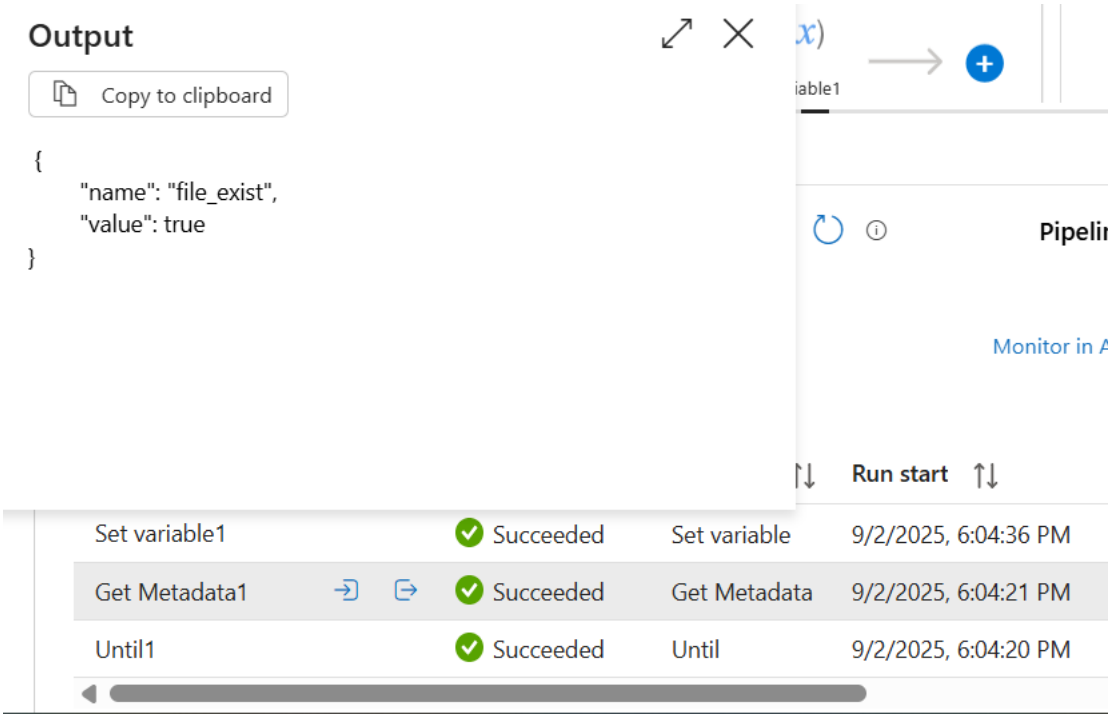


Output of this get metadata activity given to the set variable to update it dynamically.


Microsoft Azure
Data Factory
createfirstdatafactory
Search
Validate all
Publish all
Preview experience
Off
Factory Resources
Filter resources by name
Pipelines
pipeline1
until
Change Data Capture (preview)
Datasets
Data flows
Power Query
Activities
get
General
Get Metadata
Validate
Debug
Add trigger
until > Until1
Get Metadata1
General
Settings
User properties
Dataset *
source_customer_csv_file
Open
New
Learn more
Field list *
Delete
Argument
Exists
Start time (UTC)
End time (UTC)
Filter by last modified
Skip line count


Microsoft Azure
Data Factory
createfirstdatafactory
Search
Validate all
Publish all
Preview experience
Off
Factory Resources
Filter resources by name
Pipelines
pipeline1
until
Change Data Capture (preview)
Datasets
Data flows
Power Query
Activities
set
General
Set variable
Validate
Debug
Add trigger
Get Metadata1
Set variable1
General
Settings
User properties
Variable type
Pipeline variable
Pipeline return value
Name *
file_exist
New
Value
@activity('Get Metadata1').output.exi...


Microsoft Azure
Data Factory
createfirstdatafactory
Search
Validate all
Publish all
Preview experience
Off
Factory Resources
Filter resources by name
Pipelines
pipeline1
until
Change Data Capture (preview)
Datasets
Data flows
Power Query
Activities
set
General
Set variable
Validate
Debug
Add trigger
Output
Copy to clipboard
{
"exists": true,
"effectiveIntegrationRuntime":
"AutoResolveIntegrationRuntime (South India)",
"executionDuration": 1,
"durationInQueue": {
"integrationRuntimeQueue": 10
},
"billingReference": {
Pipeline status
Succeeded
View debug run consumption
Monitor in Azure Metrics
Export to CSV
Run start
Duration
Integration runtime
9/2/2025, 6:04:36 PM
Less than 1s
Get Metadata1
Succeeded
Get Metadata
9/2/2025, 6:04:21 PM
14s
AutoResolveIntegrat
Until1
Succeeded
Until
9/2/2025, 6:04:20 PM
18s



Web Activity in ADF

The **Web Activity** lets us to **call REST API** (HTTP APIs) from inside your pipeline. It's useful when you need to **interact with external services or trigger jobs** from ADF.

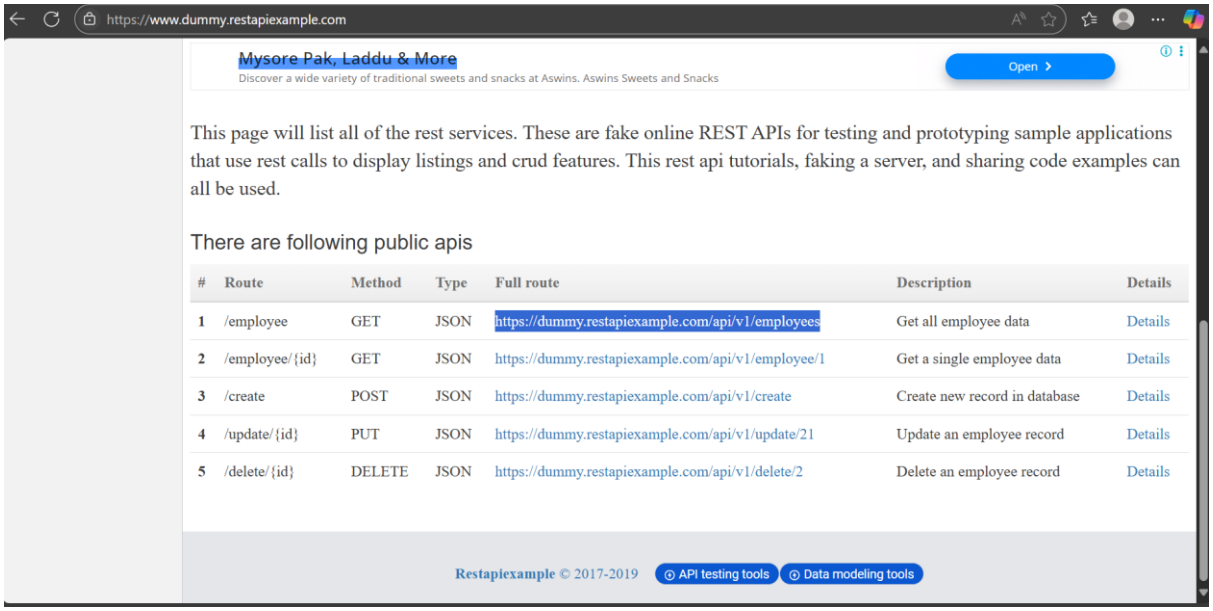
Used to make a **simple HTTP request** and optionally capture the response for later activities.

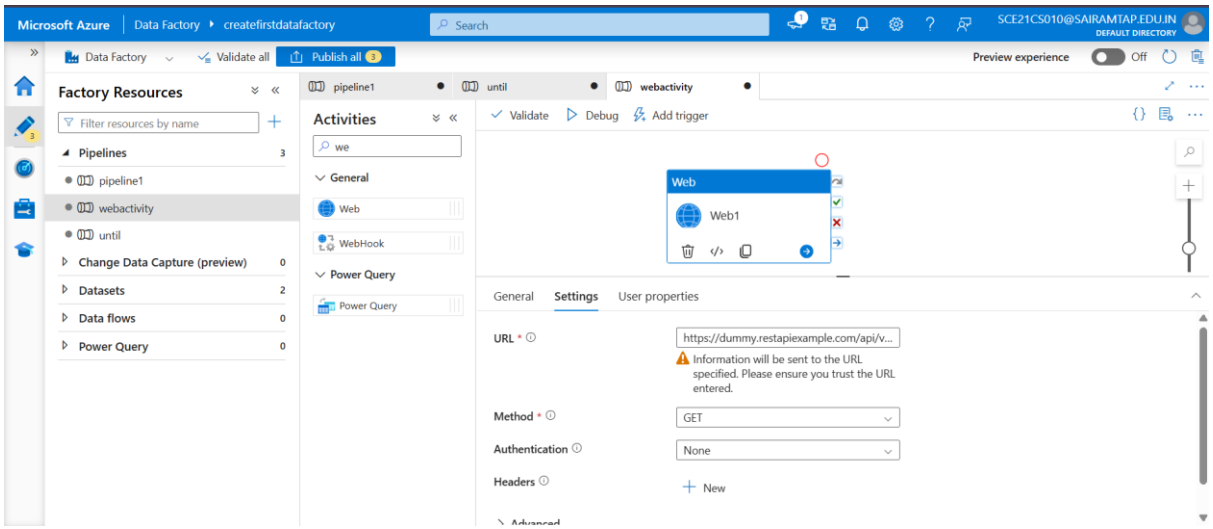
You can pass datasets and linked services to be consumed and accessed by the activity.

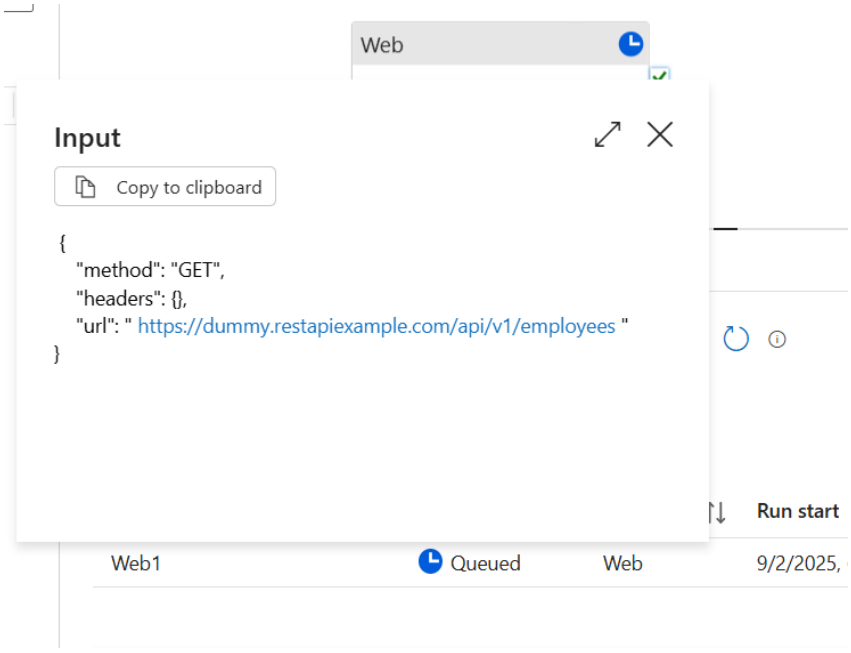
Key Properties

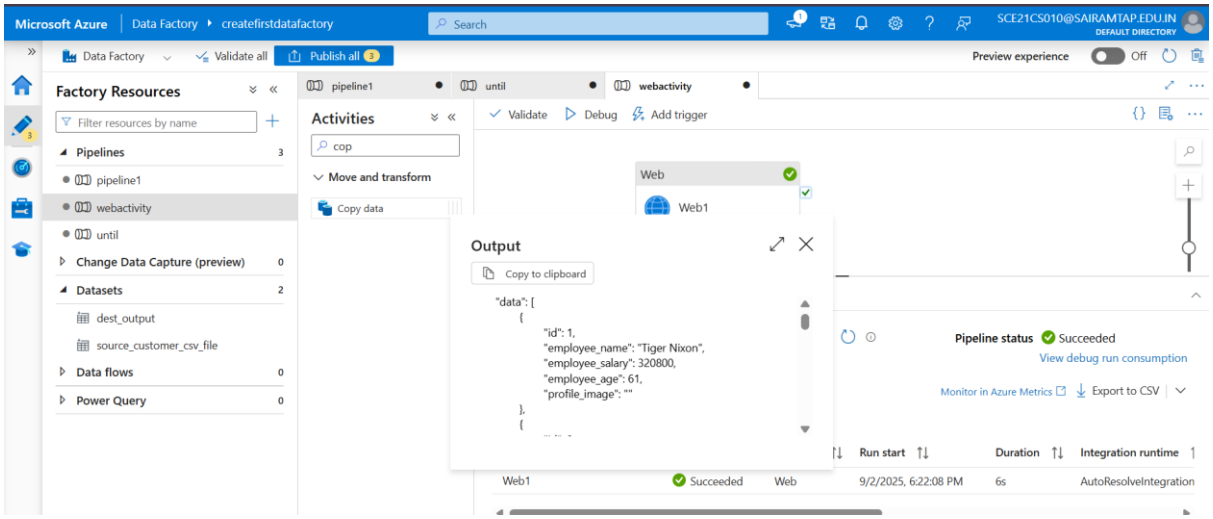
1. **URL** → The endpoint you're calling (e.g., https://www.dummy.restapiexample.com/).
2. **Method** → GET, POST, PUT, DELETE, Patch(partially update a resource.)
3. **Headers** → You can pass things like Content-Type: application/json,

   Authorization: Bearer <token>.
4. **Body** → Used in POST or PUT calls to send data (must be valid JSON).
5. **Authentication** (supports multiple types):
   - None
   - Basic (username/password)
   - MSI (Managed Identity)
   - Service Principal

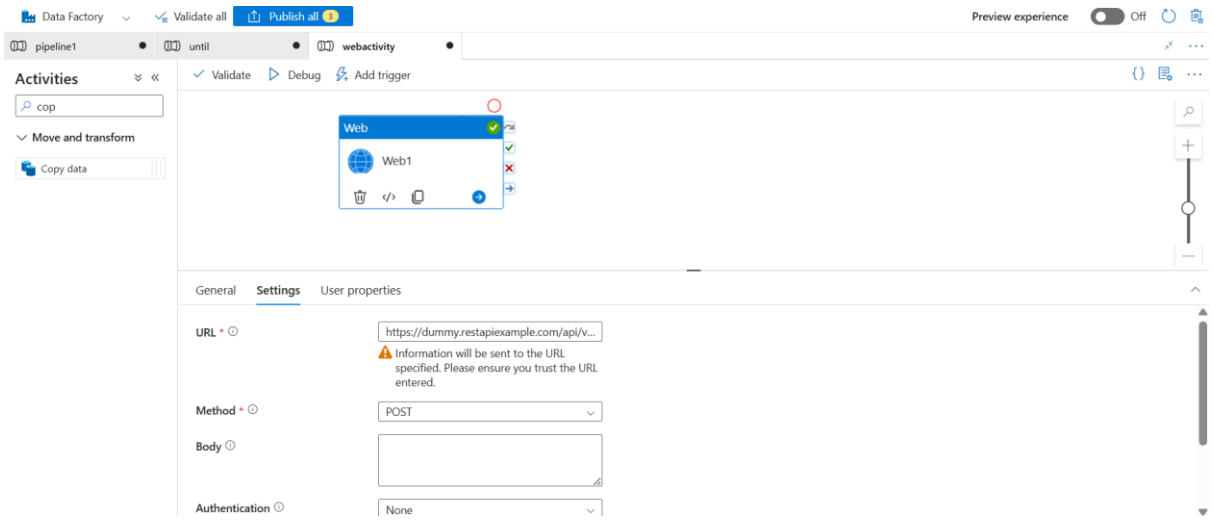

https://www.dummy.restapiexample.com
Mysore Pak, Laddu & More
Discover a wide variety of traditional sweets and snacks at Aswins. Aswins Sweets and Snacks
Open
This page will list all of the rest services. These are fake online REST APIs for testing and prototyping sample applications that use rest calls to display listings and crud features. This rest api tutorials, faking a server, and sharing code examples can all be used.
There are following public apis
Route Method Type Full route Description Details
1 /employee GET JSON https://dummy.restapiexample.com/api/v1/employees Get all employee data Details
2 /employee/{id} GET JSON https://dummy.restapiexample.com/api/v1/employee/1 Get a single employee data Details
3 /create POST JSON https://dummy.restapiexample.com/api/v1/create Create new record in database Details
4 /update/{id} PUT JSON https://dummy.restapiexample.com/api/v1/update/21 Update an employee record Details
5 /delete/{id} DELETE JSON https://dummy.restapiexample.com/api/v1/delete/2 Delete an employee record Details
Restapiexample © 2017-2019
API testing tools
Data modeling tools


Microsoft Azure
Data Factory
createfirstdatafactory
Search
SCE21CS010@SAIRAMTAP.EDU.IN
DEFAULT DIRECTORY
Validate all
Publish all
Preview experience
Off
Factory Resources
Filter resources by name
Pipelines
pipeline1
webactivity
until
Change Data Capture (preview)
Datasets
Data flows
Power Query
Activities
General
Web
WebHook
Power Query
Validate
Debug
Add trigger
Web1
General
Settings
User properties
URL
https://dummy.restapiexample.com/api/v...
Information will be sent to the URL specified. Please ensure you trust the URL entered.
Method
GET
Authentication
None
Headers
New


Web
Input
Copy to clipboard
{
"method": "GET",
"headers": {},
"url": " https://dummy.restapiexample.com/api/v1/employees "
}
Run start
Web1
Queued
Web
9/2/2025,



For post



Web Hook activity

The **Webhook activity** in Azure Data Factory is a powerful feature that allows you to interact with external systems or services by **calling an endpoint and waiting for a callback response.**

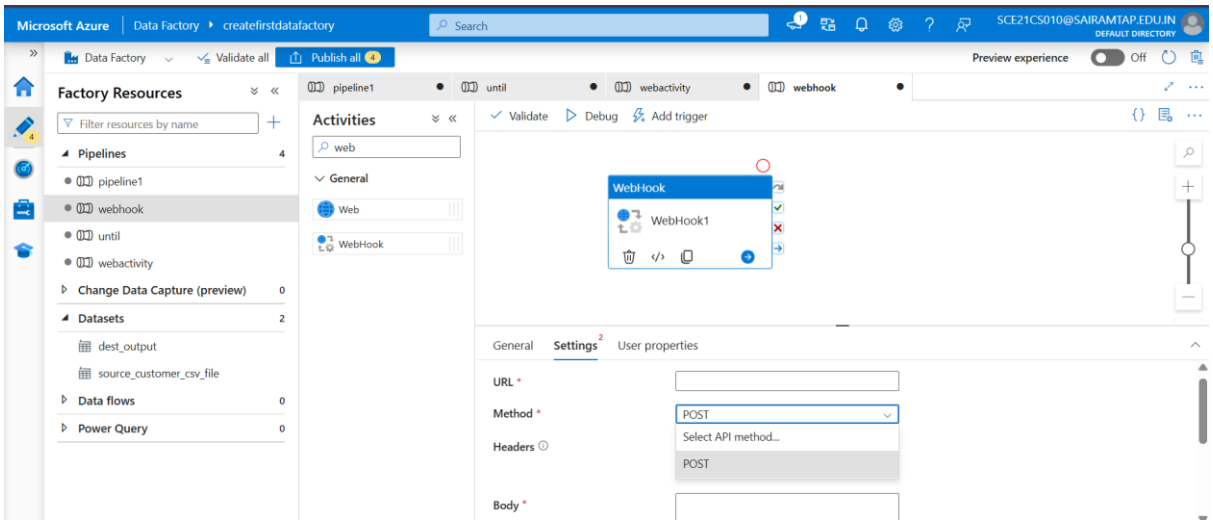
Used when you need to wait for an external callback (for example, to wait for a long-running external process to signal completion).

Web Activity → just calls an API and moves on (doesn't wait). **Webhook Activity** → calls an API (usually long-running jobs) and **waits for a callback response** before pipeline continues.

How It Works

1. **Trigger External Service**: The Webhook activity sends a request to an external endpoint (e.g., REST API) (post request).

2. **Callback URL**: ADF provides a callback URL in the request payload. The external service must invoke this URL once its processing is complete.
3. **Wait for Completion**: The pipeline execution pauses until the callback URL is invoked or a timeout occurs.



- **Scenario**: Trigger a Logic App to process data and wait for its completion before moving to the next pipeline step.

Supported HTTP Methods:

- POST
- PUT

Important notes:

- The external service **must call back** to the callbackUrl provided by ADF, sending a status payload (usually HTTP 200) to mark the activity as complete.
- If the service never calls back within the timeout, the activity fails.

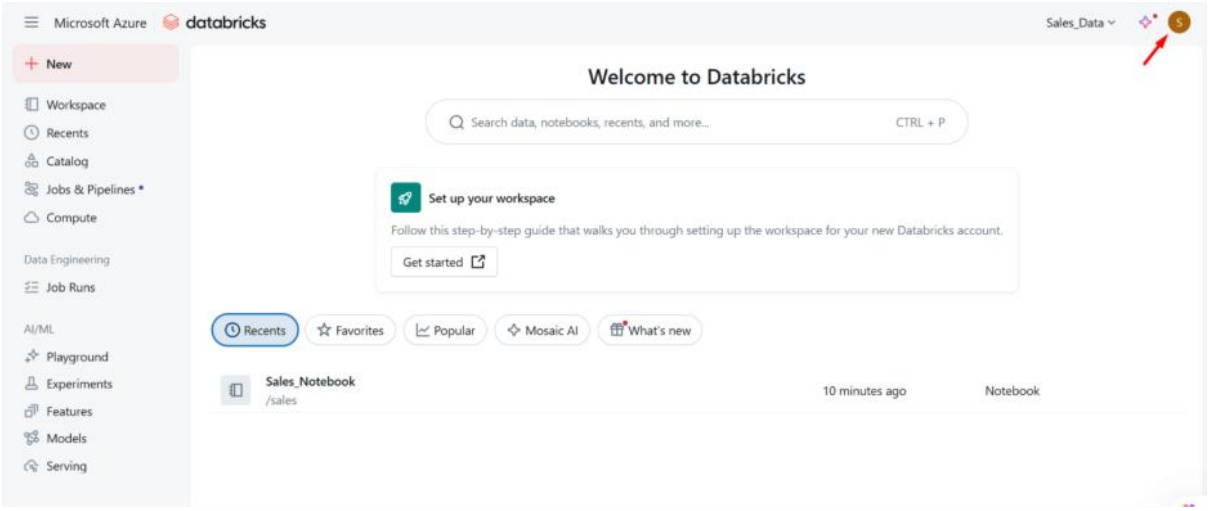
Aspect	Web Activity	Webhook Activity
Purpose	Make a simple HTTP call and immediately get a response.	Start an external process and wait for that process to call back when it's done.
Execution flow	Pipeline continues as soon as the HTTP response is returned.	Pipeline stays in *Running* state until an external system sends a callback to the provided URL or the timeout expires.
Supported HTTP methods	GET, POST, PUT, DELETE, PATCH	POST, PUT

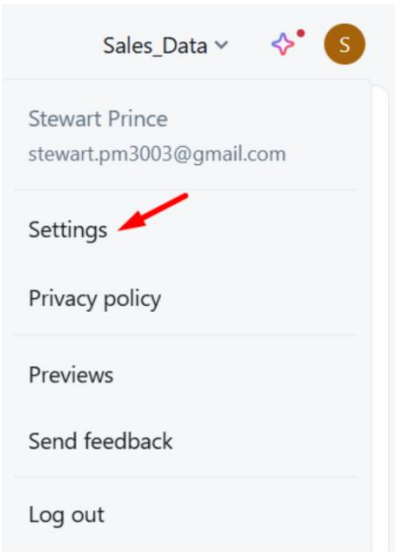
Aspect	Web Activity	Webhook Activity
Callback required	No	Yes—external service must POST to the callbackUrl included in the request body/headers.
Typical use cases	• Call a REST API to fetch or update data.• Trigger a microservice and move on.	• Trigger a long-running job (e.g., build, ML training) and wait for a “job complete” notification.• Ensure downstream steps don’t start until an external workflow signals success/failure.
Timeout behavior	Fails only if the initial HTTP call fails or times out.	Fails if the external service doesn’t call back within the configured timeout.
Return value	Immediate response body is available to later activities.	Final status is determined by the callback payload.

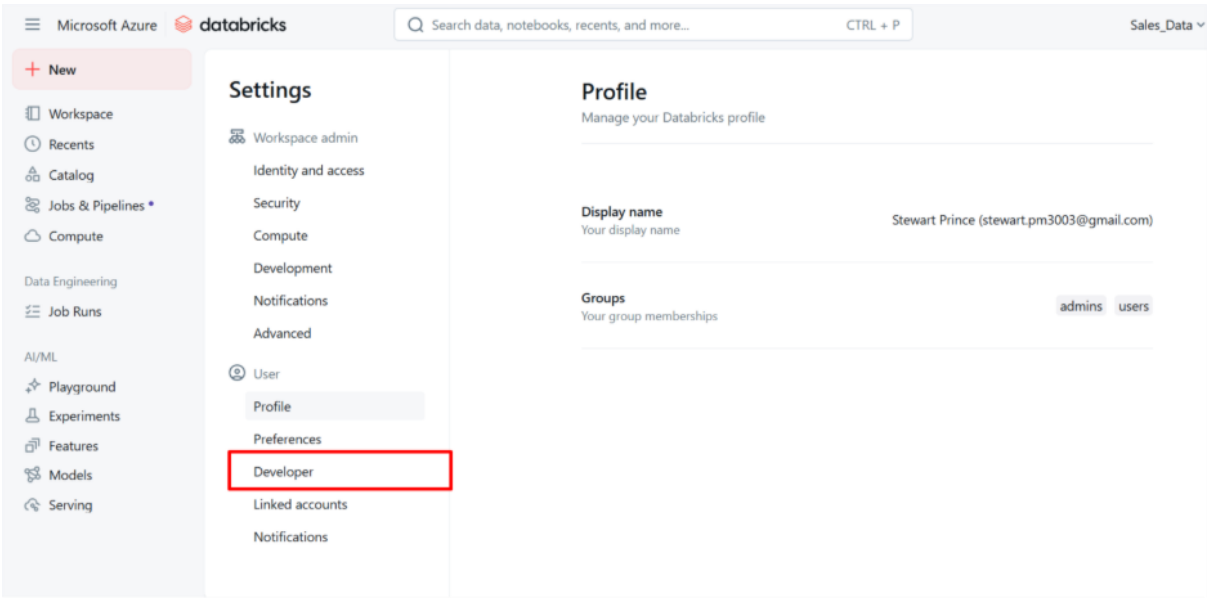
Use Web Activity when you just need to send/receive a single HTTP request/response.

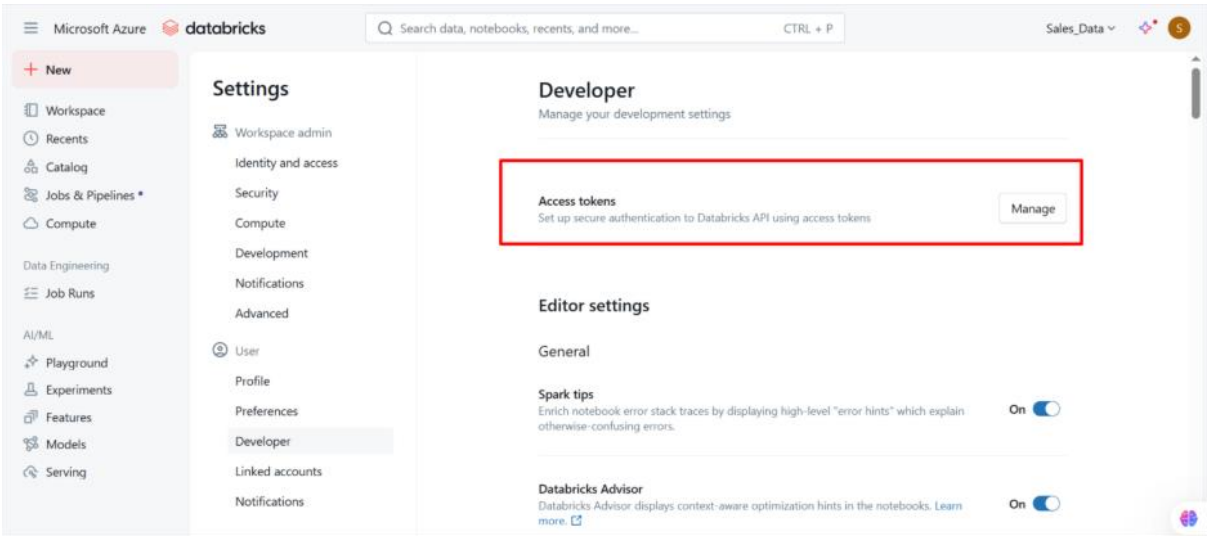
Use Webhook Activity when you must wait for an external system to confirm completion before the pipeline proceeds.

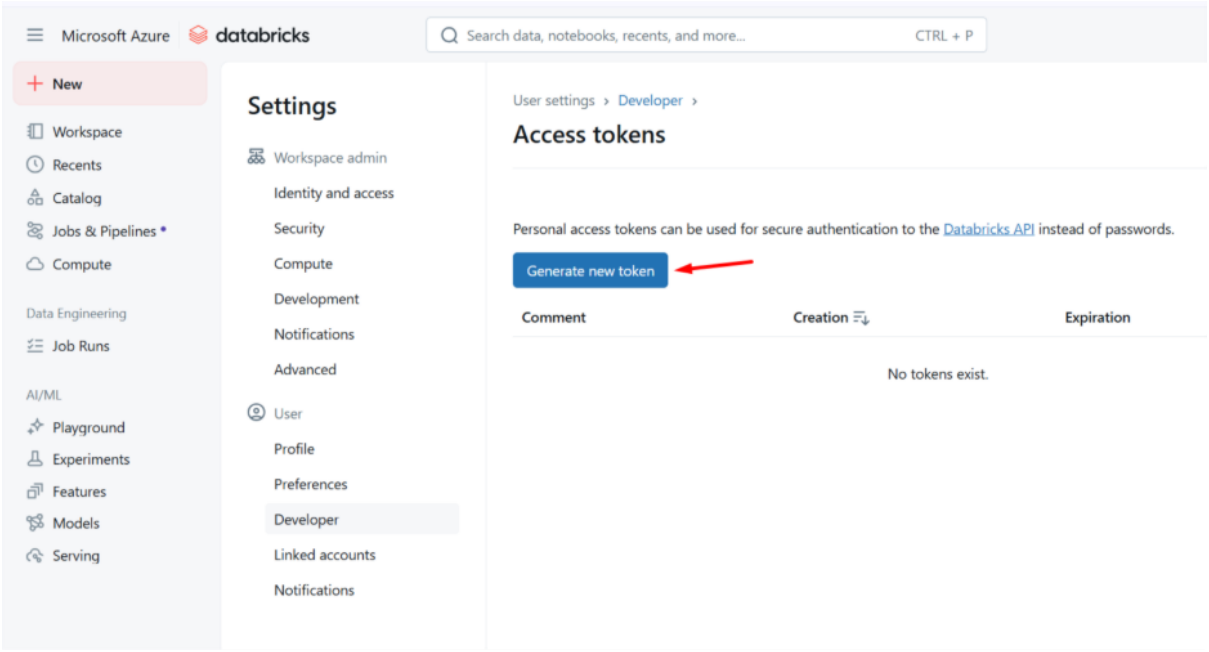
Databricks activity

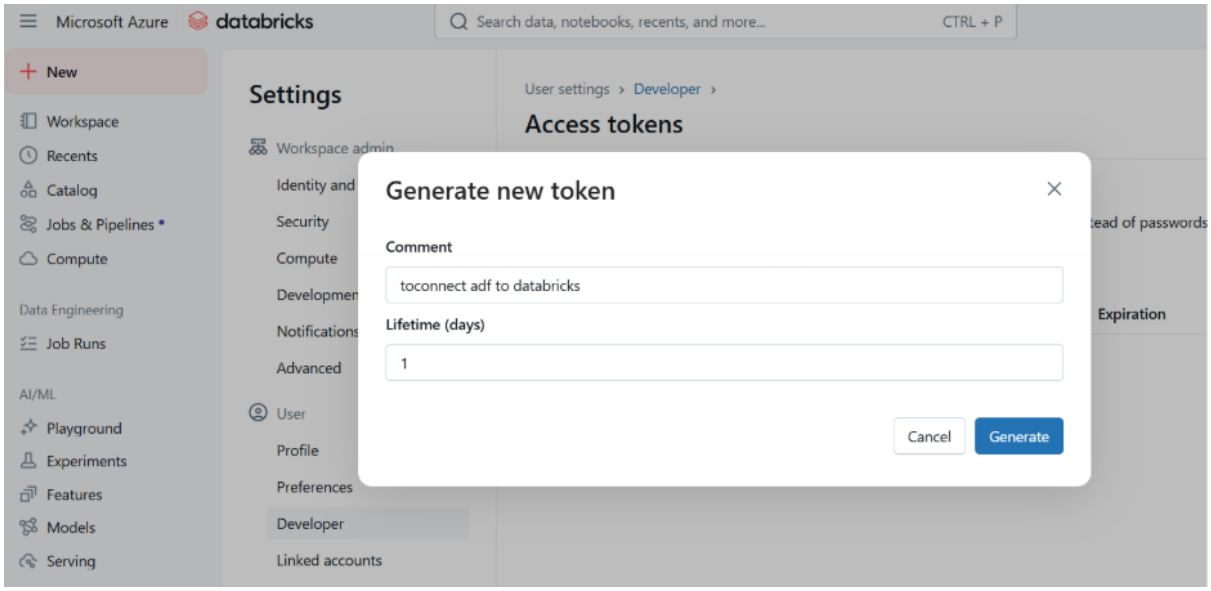

Microsoft Azure
databricks
Sales_Data
New
Workspace
Recents
Catalog
Jobs & Pipelines
Compute
Data Engineering
Job Runs
AI/ML
Playground
Experiments
Features
Models
Serving
Welcome to Databricks
Search data, notebooks, recents, and more...
CTRL + P
Set up your workspace
Follow this step-by-step guide that walks you through setting up the workspace for your new Databricks account.
Get started
Recents
Favorites
Popular
Mosaic AI
What's new
Sales_Notebook
/sales
10 minutes ago
Notebook

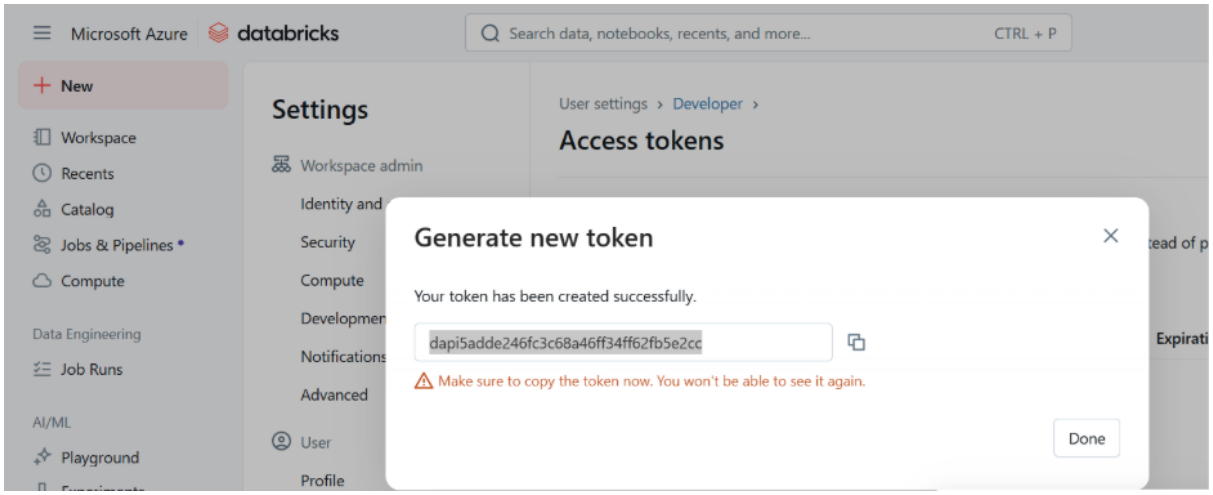

Sales_Data
S
Stewart Prince
stewart.pm3003@gmail.com
Settings
Privacy policy
Previews
Send feedback
Log out

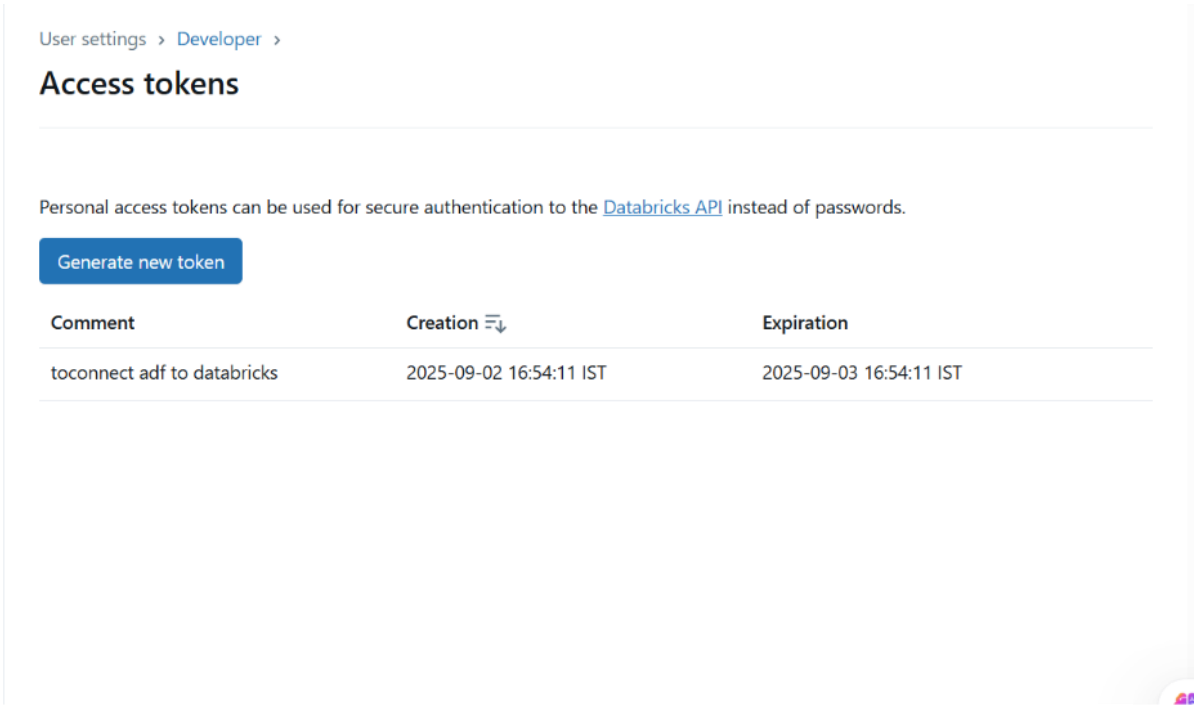

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
Sales_Data
New
Workspace
Recents
Catalog
Jobs & Pipelines
Compute
Data Engineering
Job Runs
AI/ML
Playground
Experiments
Features
Models
Serving
Settings
Workspace admin
Identity and access
Security
Compute
Development
Notifications
Advanced
User
Profile
Preferences
Developer
Linked accounts
Notifications
Profile
Manage your Databricks profile
Display name
Your display name
Stewart Prince (stewart.pm3003@gmail.com)
Groups
Your group memberships
admins
users


Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
Sales_Data
New
Workspace
Recents
Catalog
Jobs & Pipelines
Compute
Data Engineering
Job Runs
AI/ML
Playground
Experiments
Features
Models
Serving
Settings
Workspace admin
Identity and access
Security
Compute
Development
Notifications
Advanced
User
Profile
Preferences
Developer
Linked accounts
Notifications
Developer
Manage your development settings
Access tokens
Set up secure authentication to Databricks API using access tokens
Manage
Editor settings
General
Spark tips
Enrich notebook error stack traces by displaying high-level "error hints" which explain otherwise-confusing errors.
On
Databricks Advisor
Databricks Advisor displays context-aware optimization hints in the notebooks. Learn more.
On

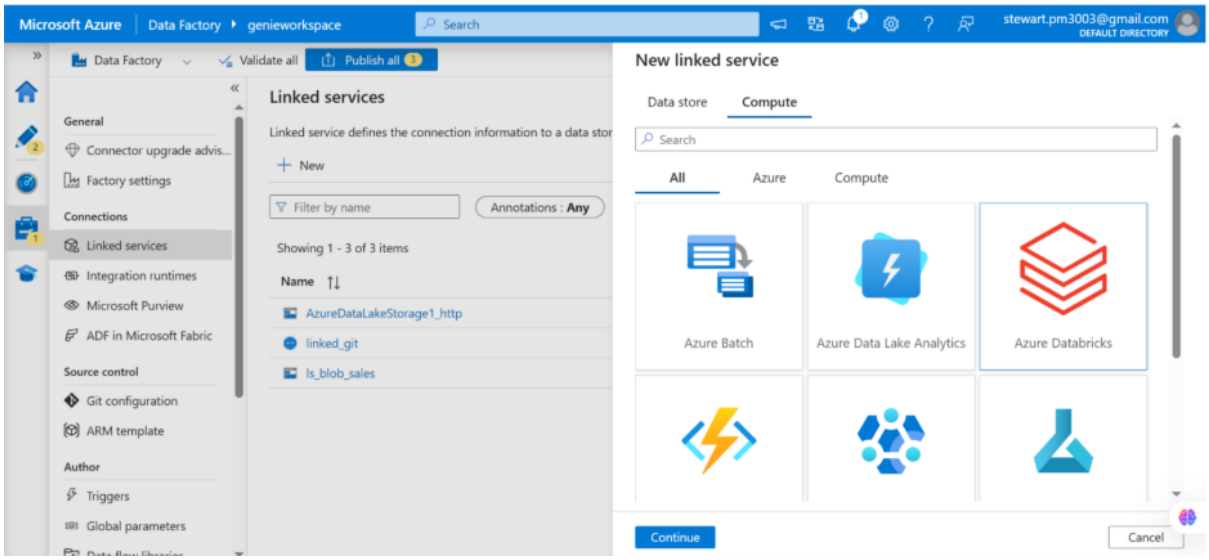

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
New
Workspace
Recents
Catalog
Jobs & Pipelines
Compute
Data Engineering
Job Runs
AI/ML
Playground
Experiments
Features
Models
Serving
Settings
Workspace admin
Identity and access
Security
Compute
Development
Notifications
Advanced
User
Profile
Preferences
Developer
Linked accounts
Notifications
User settings › Developer ›
Access tokens
Personal access tokens can be used for secure authentication to the Databricks API instead of passwords.
Generate new token
Comment
Creation
Expiration
No tokens exist.

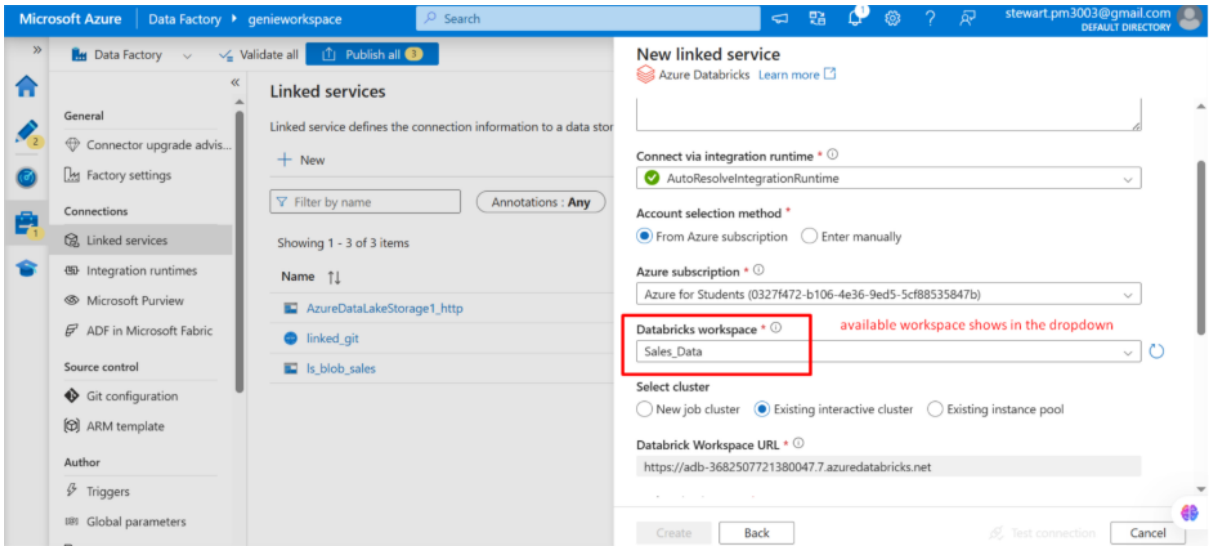

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
New
Workspace
Recents
Catalog
Jobs & Pipelines
Compute
Data Engineering
Job Runs
AI/ML
Playground
Experiments
Features
Models
Serving
Settings
Workspace admin
Identity and
Security
Compute
Developmen
Notifications
Advanced
User
Profile
Preferences
Developer
Linked accounts
User settings › Developer ›
Access tokens
Generate new token
Comment
toconnect adf to databricks
Lifetime (days)
1
Cancel
Generate
Expiration

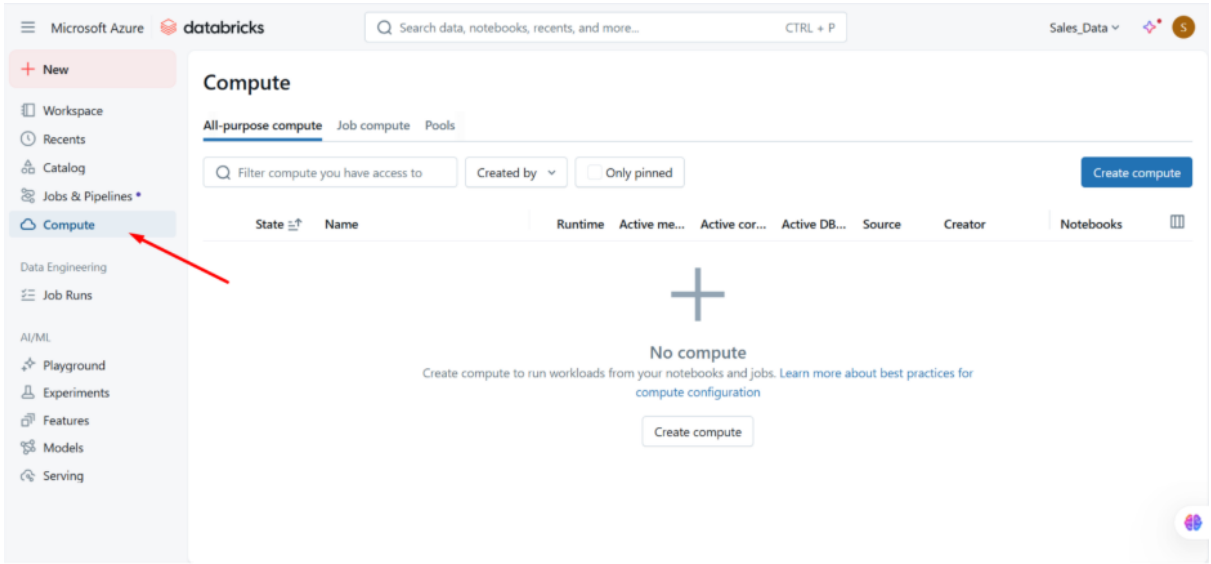

Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
New
Workspace
Recents
Catalog
Jobs & Pipelines
Compute
Data Engineering
Job Runs
AI/ML
Playground
Settings
Workspace admin
Identity and
Security
Compute
Developmen
Notifications
Advanced
User
Profile
User settings › Developer ›
Access tokens
Generate new token
Your token has been created successfully.
dapi5adde246fc3c68a46ff34ff62fb5e2cc
Make sure to copy the token now. You won't be able to see it again.
Done

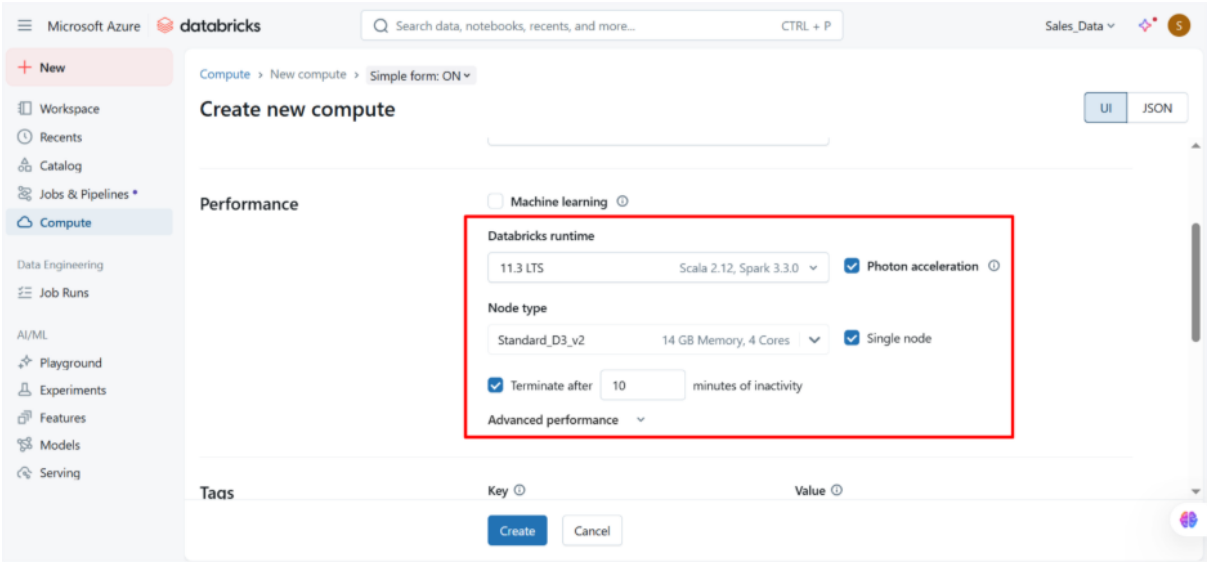

User settings › Developer ›
Access tokens
Personal access tokens can be used for secure authentication to the Databricks API instead of passwords.
Generate new token
Comment
Creation
Expiration
toconnect adf to databricks
2025-09-02 16:54:11 IST
2025-09-03 16:54:11 IST

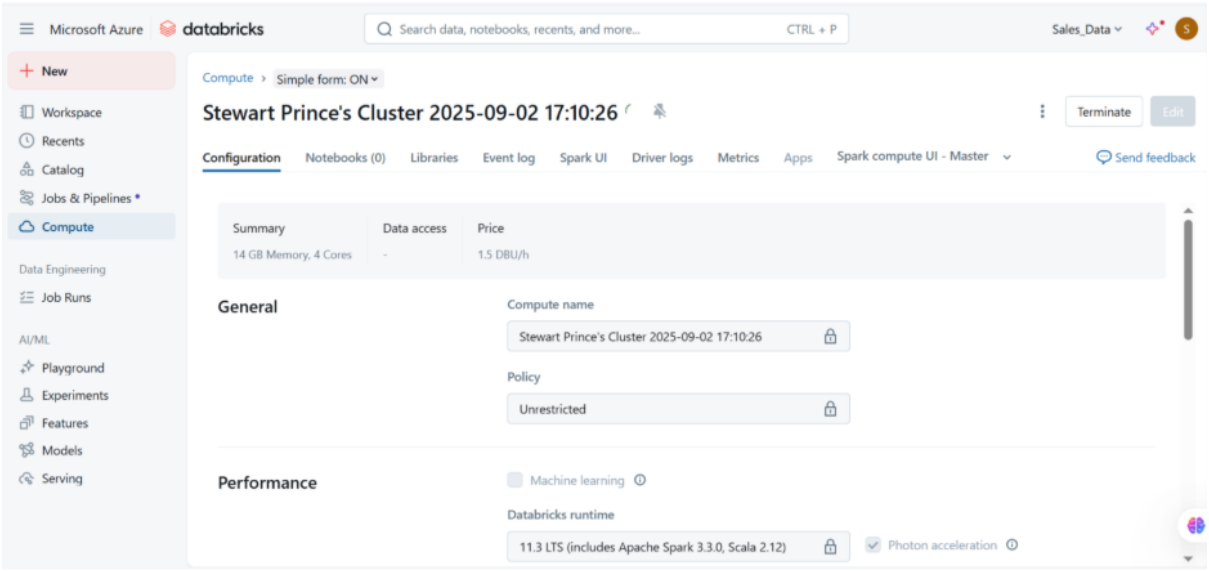
Create linked service for databricks (go to compute)





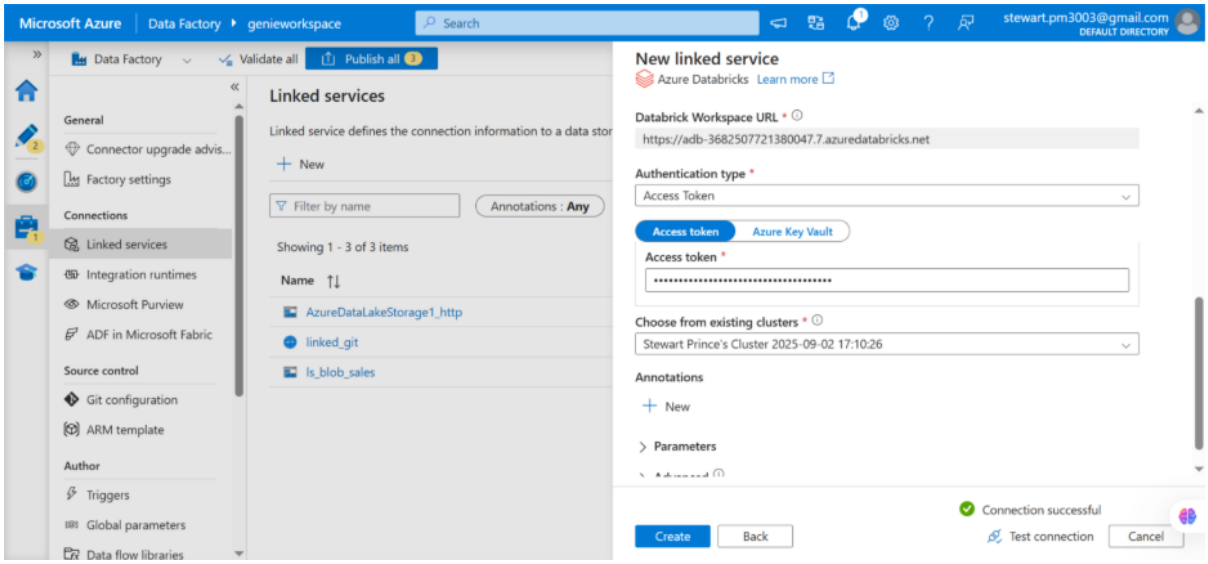






Cluster created

Connect linked service



Linked services

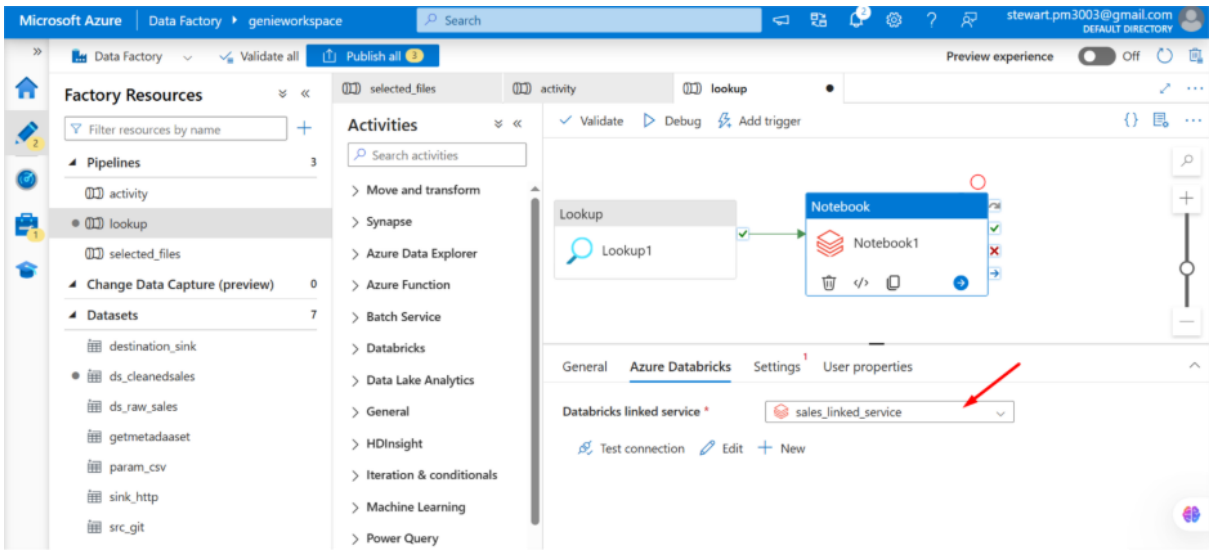
Linked service defines the connection information to a data store or compute. Learn more

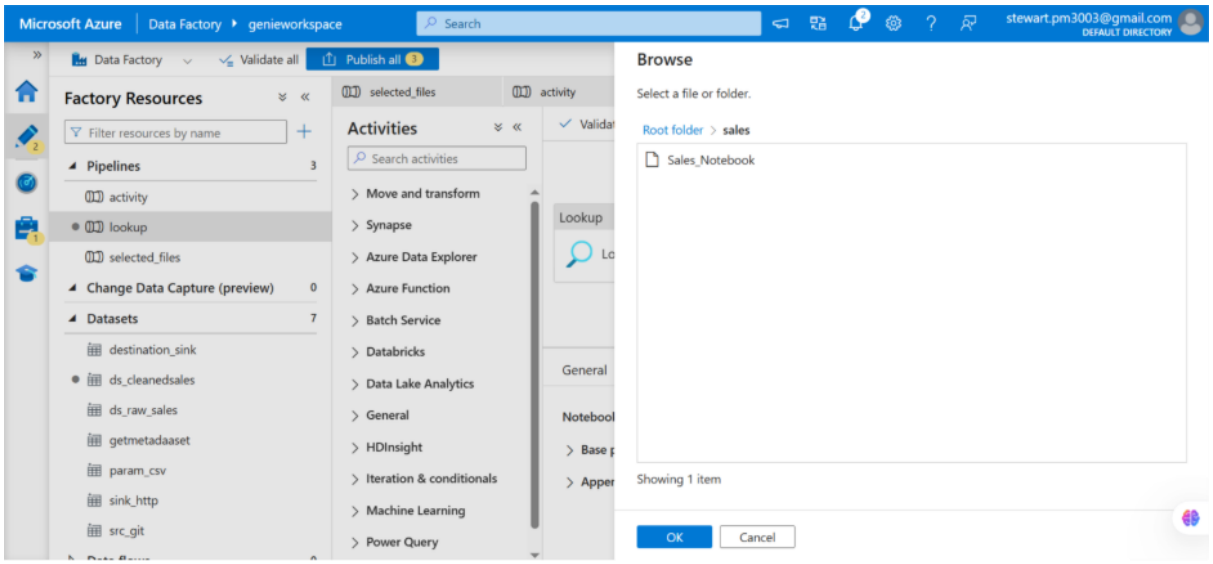
+ New

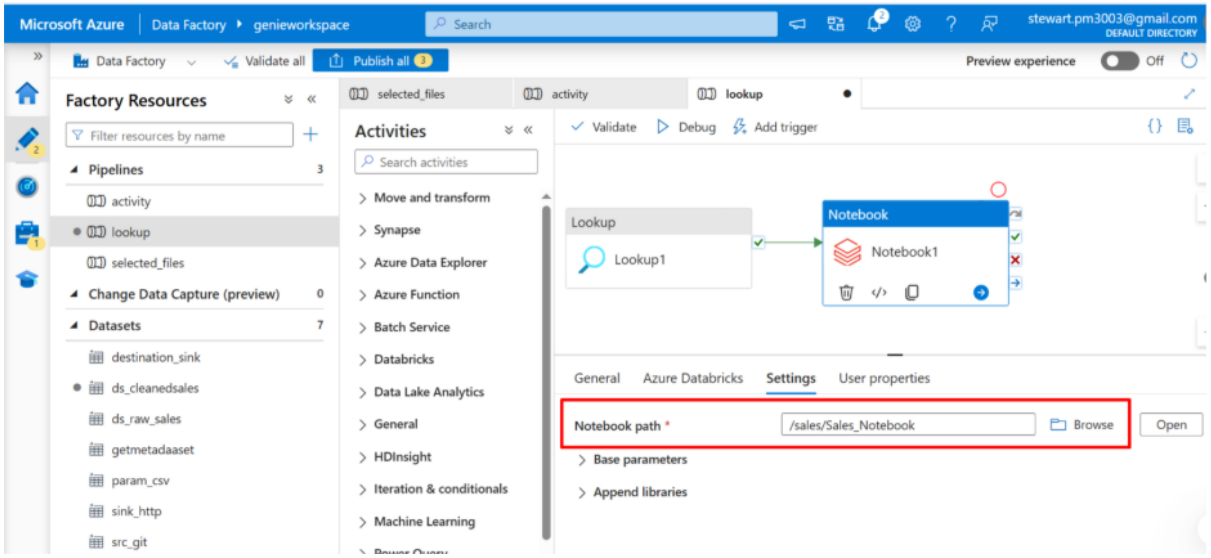
Filter by name | Annotations : **Any**

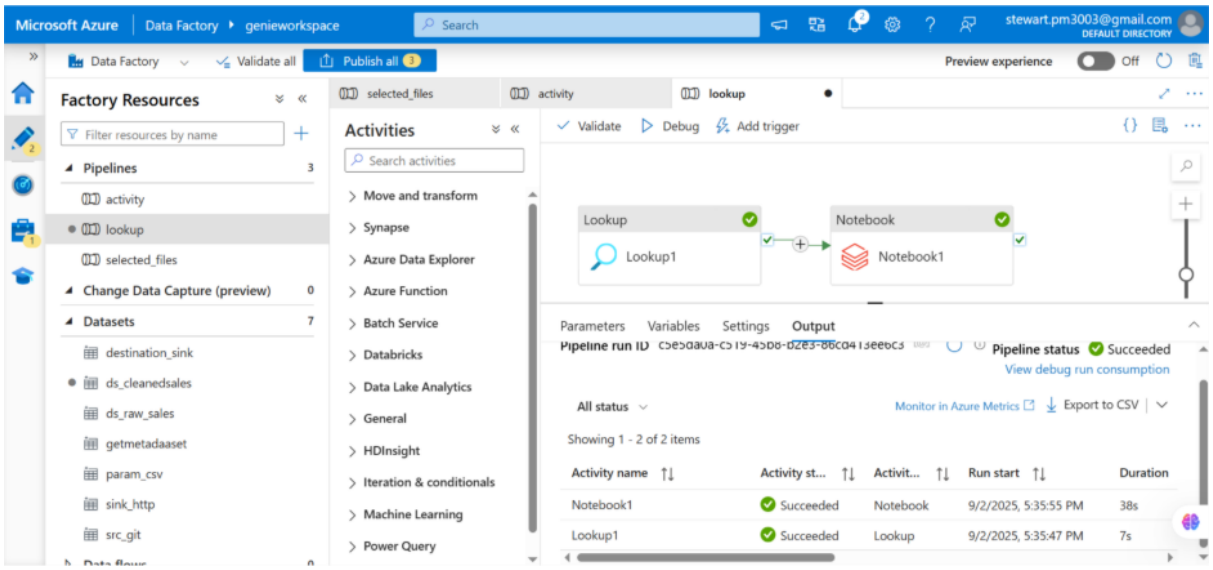
Showing 1 - 4 of 4 items

Name	Type	Related	Annotations
AzureDataLakeStorage1_http	Azure Data Lake Storage Gen2	4	
linked_git	HTTP	1	
ls_blob_sales	Azure Blob Storage	2	
sales_linked_service	Azure Databricks	0	

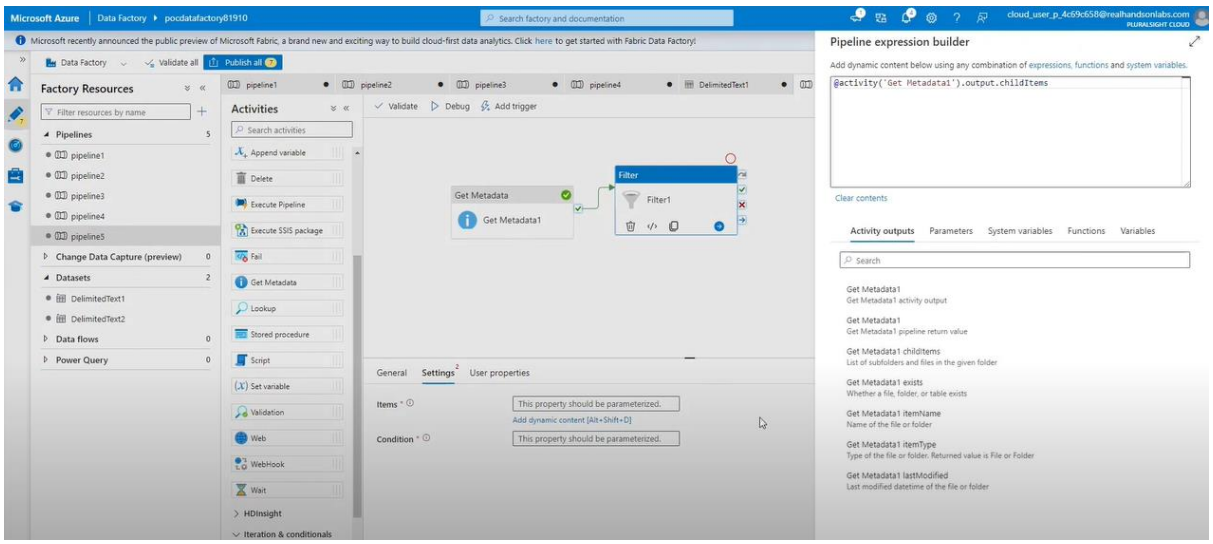


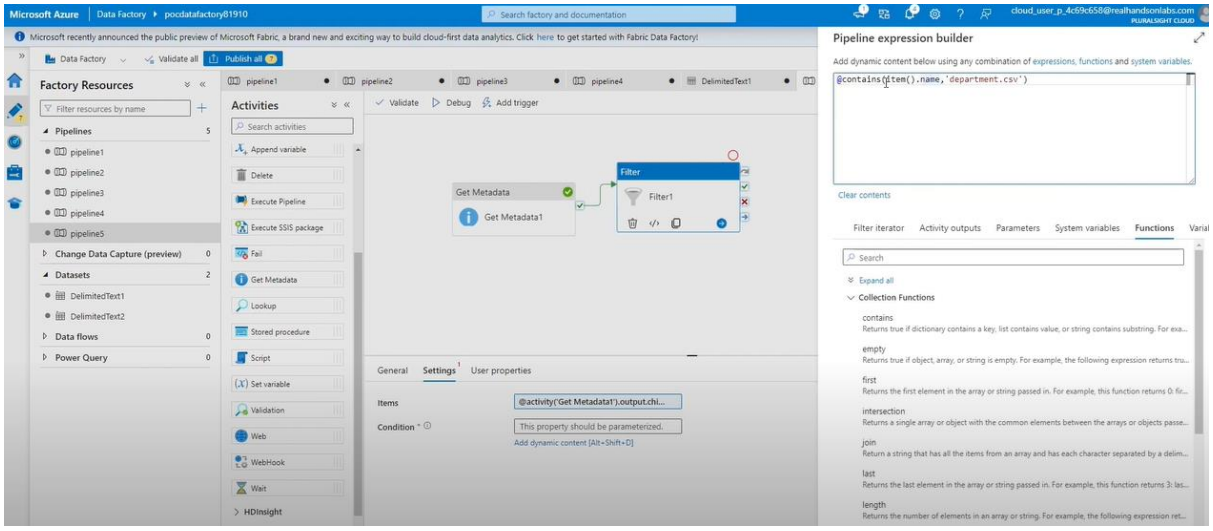


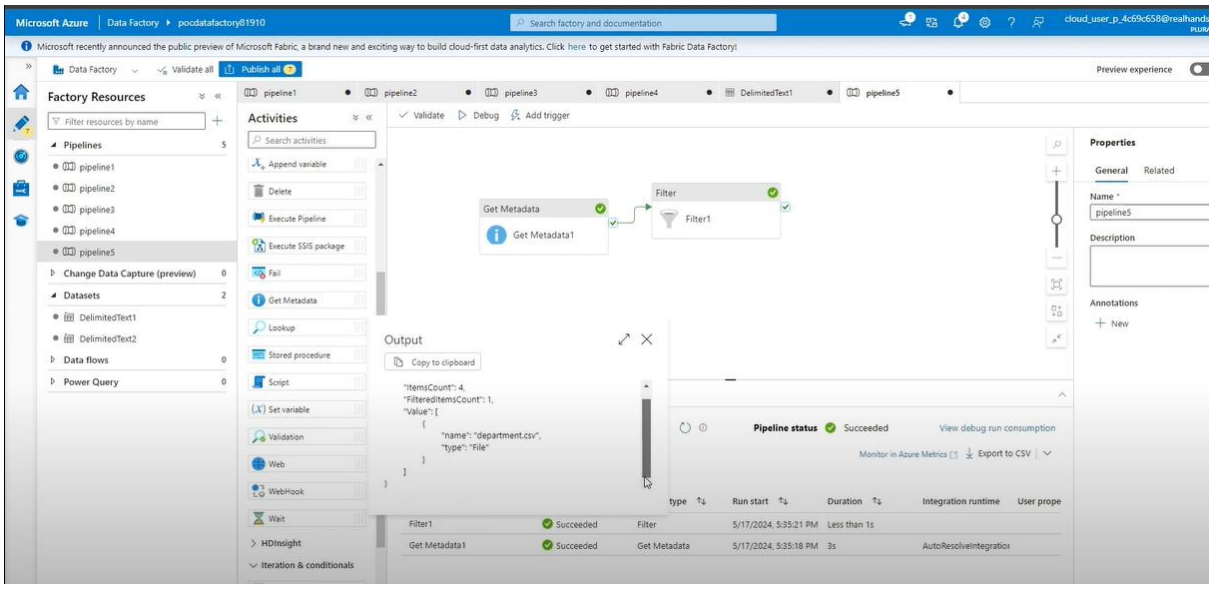




Filter activity







Azure Data Factory Pipeline Activities

- Move and transform

Activity	What it does	Real-time scenario
Copy Data	Moves data from a source to a sink (with optional schema mapping).	Copy CSV from Blob → Azure SQL Table.
Data Flow	Runs Mapping Data Flows (row-level transformations like join, aggregate, conditional split).	Cleanse customer data, handle SCD Type 2, load fact tables.

2. **Azure Data Lake / Databricks**

Activity	What it does	Scenario
Databricks Notebook / Jar / Python	Runs Databricks jobs.	Run PySpark to flatten JSON or apply SCD logic.
HDInsight Hive / Pig / MapReduce / Spark	Run big data jobs on HDInsight cluster.	Legacy Hadoop workloads.

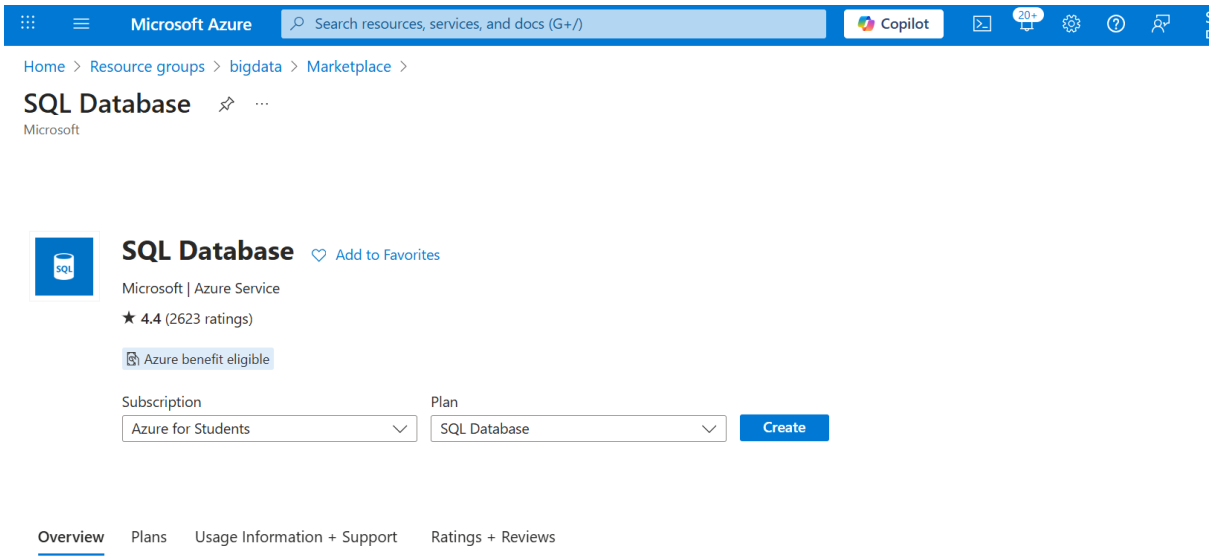
3. **General Activities**

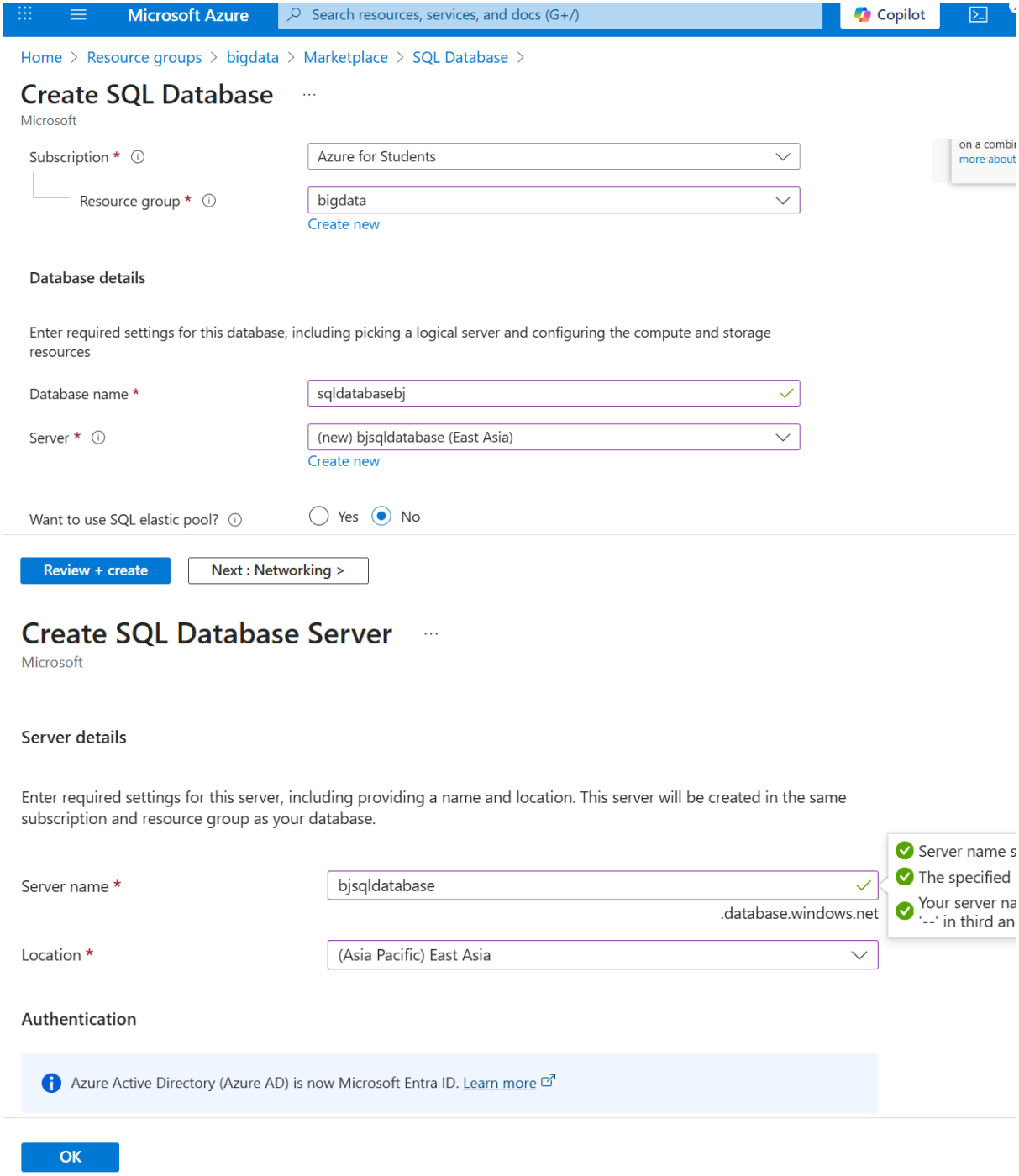
Activity	What it does	Scenario
Lookup	Reads a single row or dataset from a source for use in pipeline.	Get latest watermark value from SQL table.
Get Metadata	Fetches file/table properties (size, modified date, schema).	Check if file exists before processing.
Delete	Deletes files/folders from storage.	Clean up old files after load.
Web	Calls REST APIs (fire and forget).	Trigger an API that refreshes Power BI dataset.
Webhook	Calls API and waits for callback.	Trigger ML job and wait until training finishes.
Stored Procedure	Calls SQL stored procedure.	Run merge procedure after staging load.
Execute Pipeline	Runs another pipeline.	Modularize pipelines (ingest pipeline → transform pipeline).
Append Variable	Adds a value to an array variable.	Collect list of file names during iteration.
Set Variable	Assigns value to a variable.	Store dynamic file path for use later.
Validation	Checks existence/availability of data before continuing.	Ensure file is present before Copy Activity.
Wait	Pauses pipeline for set time.	Add 10 min delay before retrying API.

4. **Iteration & Conditionals**

Activity	What it does	Scenario
If Condition	Branch logic (if-else).	If file size > 1GB, process in Databricks, else Copy directly.
Switch	Case-based branching.	Route data load to bronze, silver, or gold container.
ForEach	Loops over a collection of items.	Process each file in a folder one by one.
Until	Loops until condition = true.	Keep checking for a file until it arrives.

Create sql database




Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
Home > Resource groups > bigdata > Marketplace > SQL Database >
Create SQL Database
Microsoft
Subscription *
Azure for Students
Resource group *
bigdata
Create new
Database details
Enter required settings for this database, including picking a logical server and configuring the compute and storage resources
Database name *
sqldatabasebj
Server *
(new) bjsqldatabase (East Asia)
Create new
Want to use SQL elastic pool?
Yes
No
Review + create
Next : Networking >
Create SQL Database Server
Microsoft
Server details
Enter required settings for this server, including providing a name and location. This server will be created in the same subscription and resource group as your database.
Server name *
bjsqldatabase
.database.windows.net
Location *
(Asia Pacific) East Asia
Authentication
Azure Active Directory (Azure AD) is now Microsoft Entra ID. Learn more
OK

bjSQL@5@7

Create SQL Database Server

Microsoft

Select your preferred authentication methods for accessing this server. Create a server admin login and password to access your server with SQL authentication, select only Microsoft Entra authentication Learn more using an existing Microsoft Entra user, group, or application as Microsoft Entra admin Learn more, or select both SQL and Microsoft Entra authentication.

Authentication method

- ○ Use Microsoft Entra-only authentication
- ○ Use both SQL and Microsoft Entra authentication
- ◉ Use SQL authentication

Server admin login * bjsqlserver

Password * ••••••••

Confirm password * ••••••••

OK

Microsoft Azure | Search resources, services, and docs (G+/)

Home > Resource groups > bigdata > Marketplace > SQL Database >

Create SQL Database

Microsoft

Workload environment

- ◉ Development
- ○ Production

Default settings provided for Development workloads. Configurations can be modified as needed.

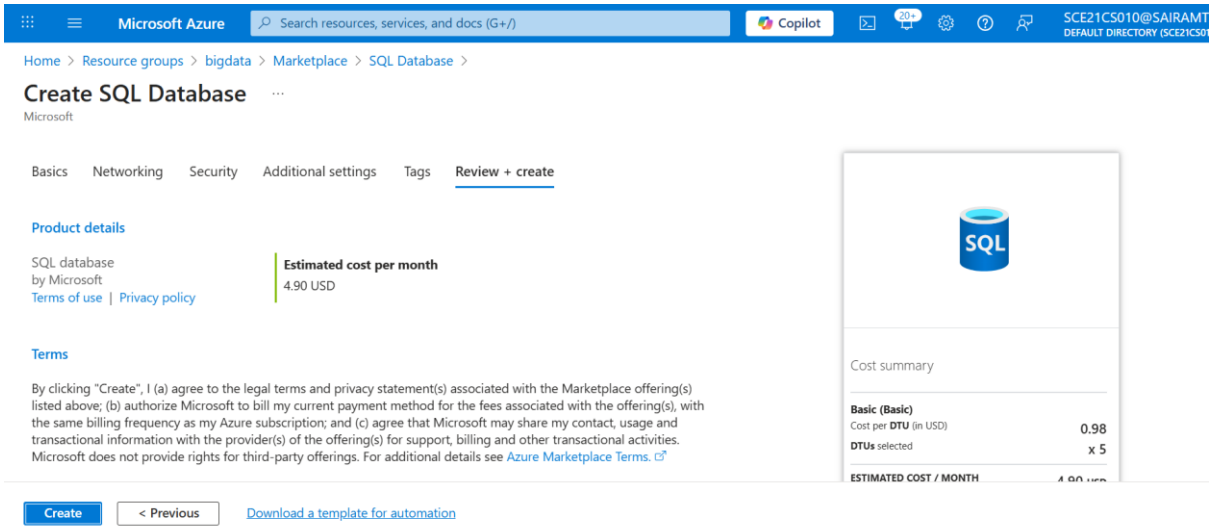
Compute + storage *

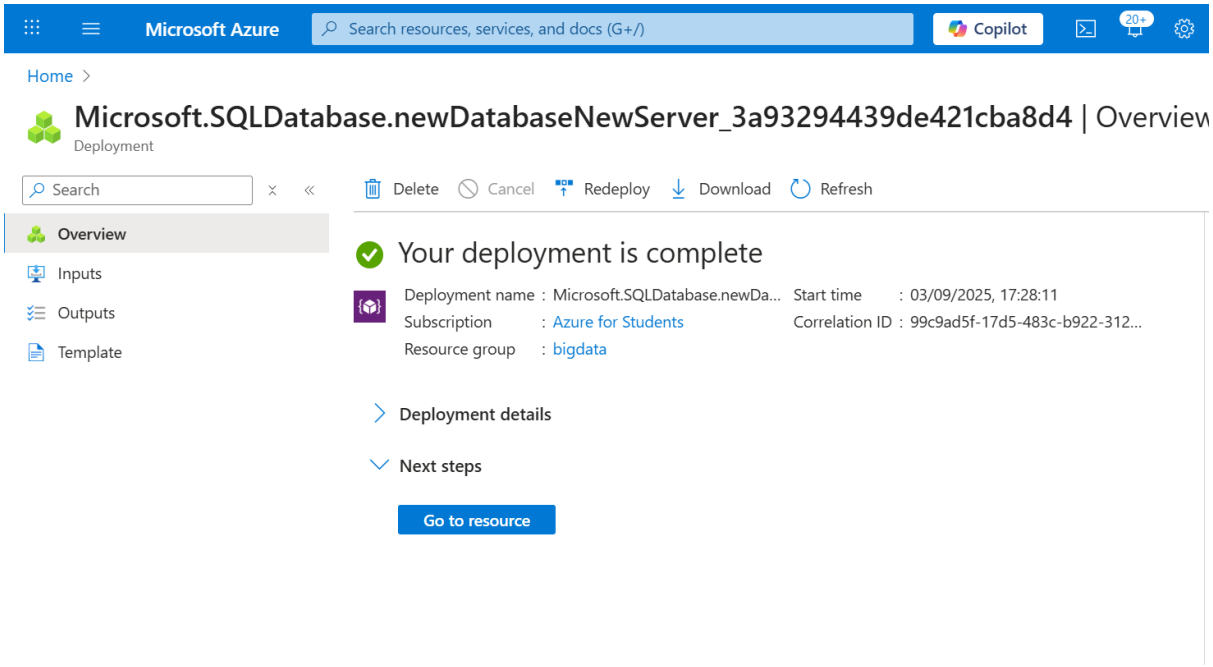
Basic
2 GB storage
Configure database

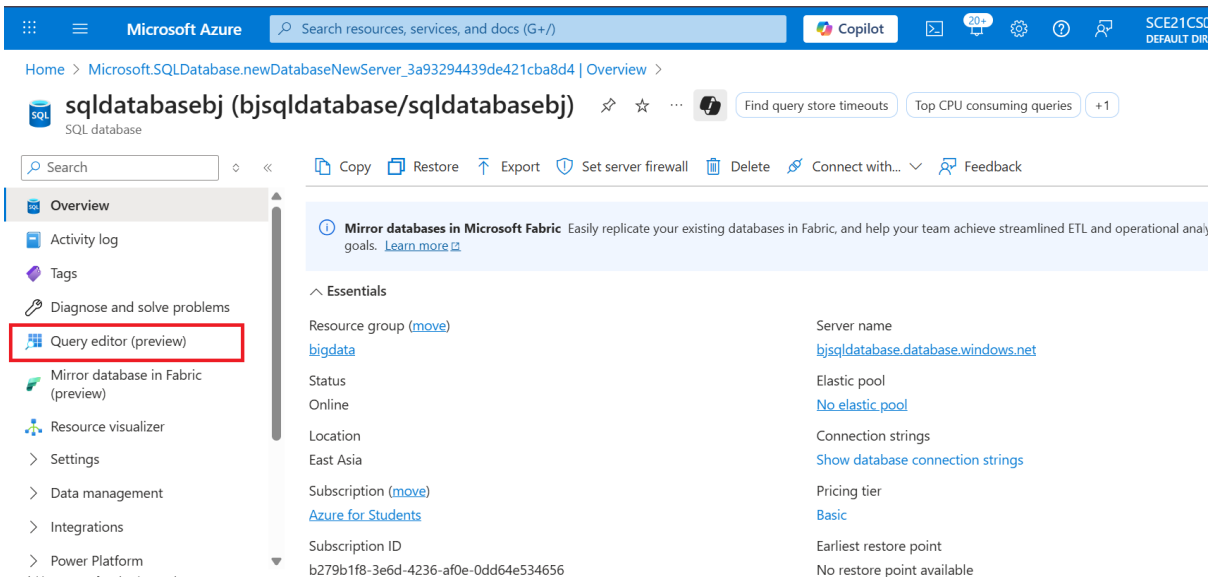
Backup storage redundancy

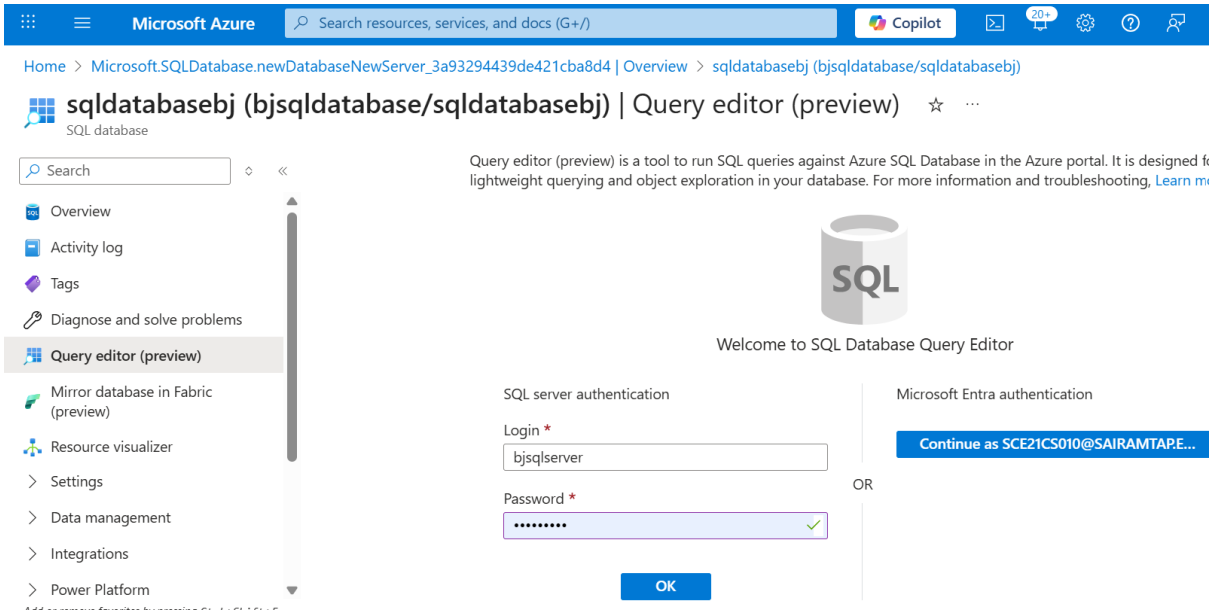
Choose how your PITR and LTR backups are replicated. Geo restore or ability to recover from regional outage is only available when geo-redundant storage is selected.

Review + create | Next : Networking >


Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
Home > Resource groups > bigdata > Marketplace > SQL Database >
Create SQL Database
Microsoft
Basics
Networking
Security
Additional settings
Tags
Review + create
Product details
SQL database
by Microsoft
Terms of use | Privacy policy
Estimated cost per month
4.90 USD
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. For additional details see Azure Marketplace Terms.
Cost summary
Basic (Basic)
Cost per DTU (in USD)
0.98
DTUs selected
x 5
Create
< Previous
Download a template for automation


Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
Home >
Microsoft.SQLDatabase.newDatabaseNewServer_3a93294439de421cba8d4 | Overview
Deployment
Search
Overview
Inputs
Outputs
Template
Delete
Cancel
Redeploy
Download
Refresh
Your deployment is complete
Deployment name : Microsoft.SQLDatabase.newDa...
Subscription : Azure for Students
Resource group : bigdata
Start time : 03/09/2025, 17:28:11
Correlation ID : 99c9ad5f-17d5-483c-b922-312...
Deployment details
Next steps
Go to resource


Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
Home > Microsoft.SQLDatabase.newDatabaseNewServer_3a93294439de421cba8d4 | Overview >
sqldatabasebj (bjsqldatabase/sqldatabasebj)
SQL database
Find query store timeouts
Top CPU consuming queries
+1
Search
Overview
Activity log
Tags
Diagnose and solve problems
Query editor (preview)
Mirror database in Fabric (preview)
Resource visualizer
Settings
Data management
Integrations
Power Platform
Copy
Restore
Export
Set server firewall
Delete
Connect with...
Feedback
Mirror databases in Microsoft Fabric Easily replicate your existing databases in Fabric, and help your team achieve streamlined ETL and operational
goals. Learn more
Essentials
Resource group (move)
bigdata
Status
Online
Location
East Asia
Subscription (move)
Azure for Students
Subscription ID
b279b1f8-3e6d-4236-af0e-0dd64e534656
Server name
bjsqldatabase.database.windows.net
Elastic pool
No elastic pool
Connection strings
Show database connection strings
Pricing tier
Basic
Earliest restore point
No restore point available



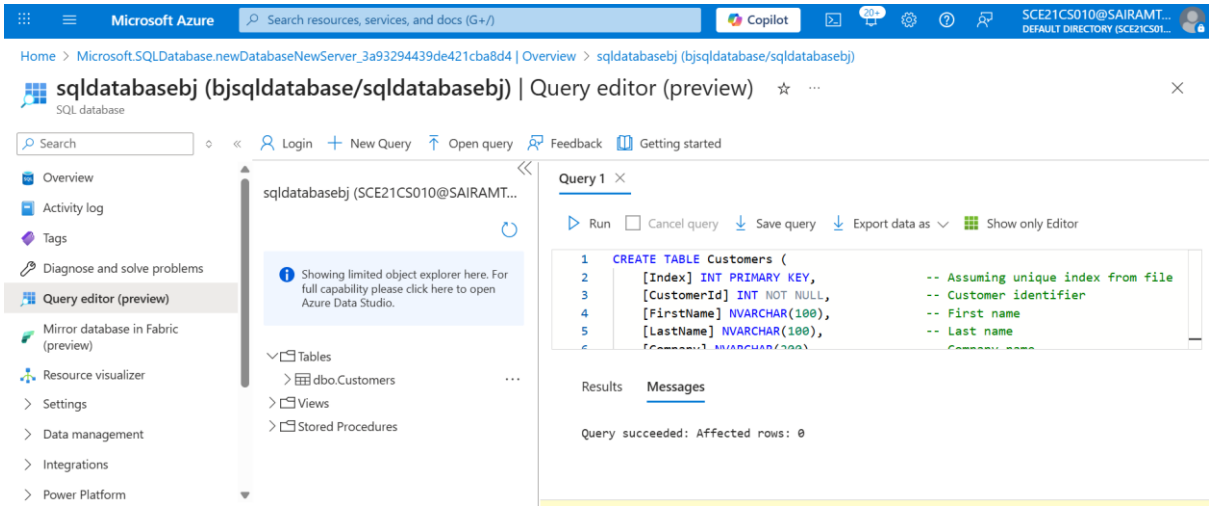
Scenario

1. We have data available in blob or ADLS as csv format
2. We need to copy data from storage to sql database (table format)

Source is created

Sink

(table created)



New linked service

Azure SQL Database Learn more

Version

2.0 (Recommended) 1.0

Import from connection string

Account selection method

From Azure subscription Enter manually

Azure subscription

Azure for Students (b279b1f8-3e6d-4236-af0e-0dd64e534656)

Server name *

bjsqldatabase

Database name *

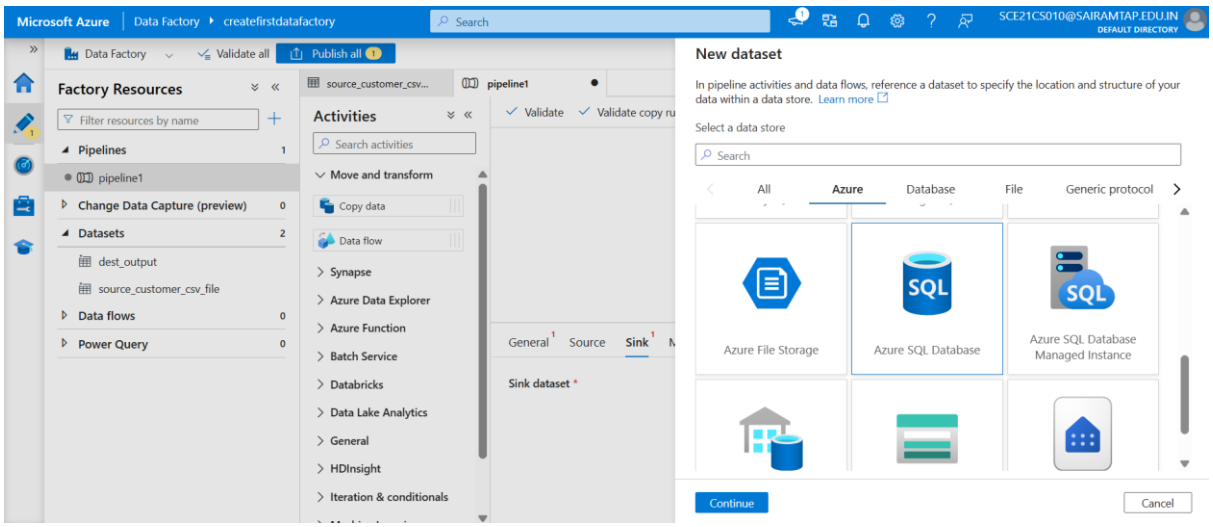
sqldatabasebj

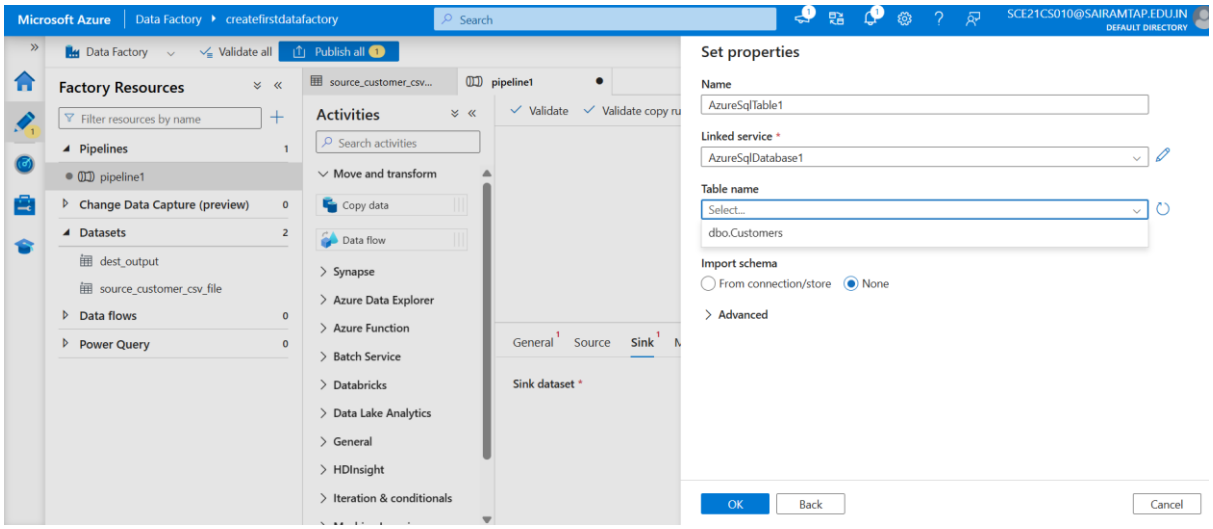
Authentication type *

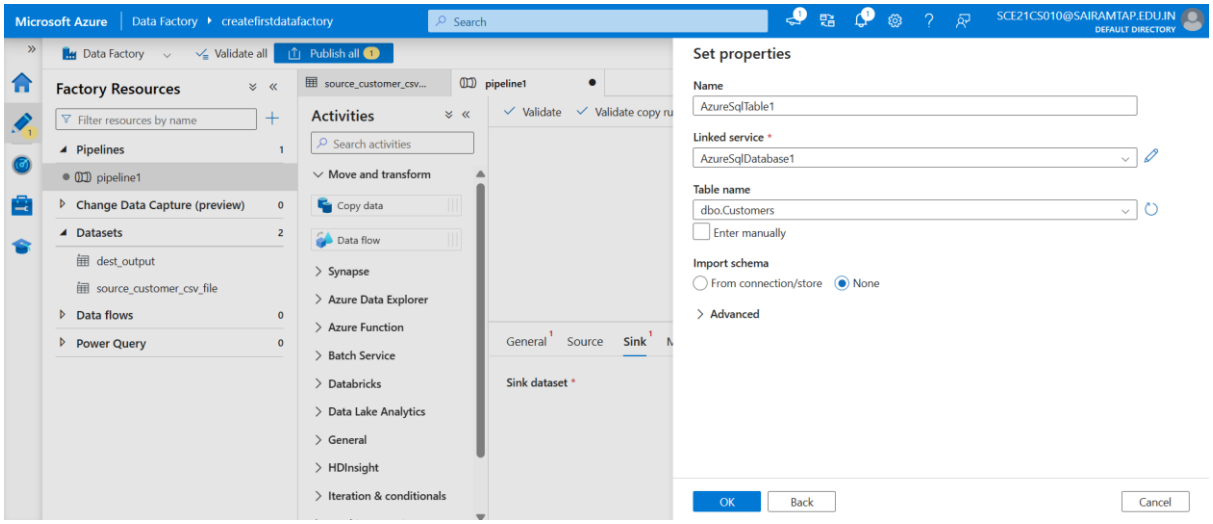
SQL authentication

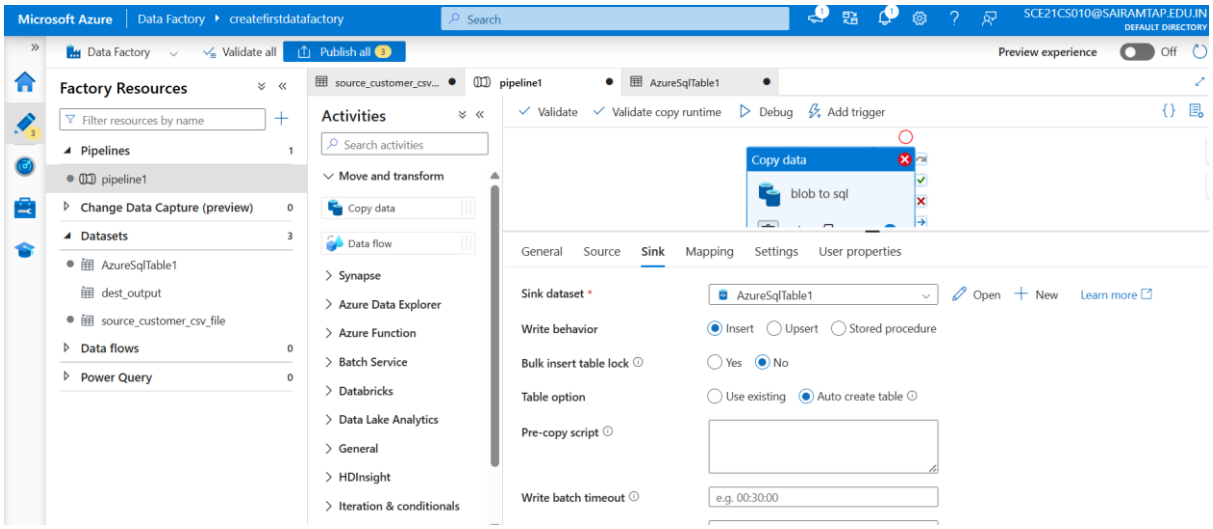
User name *

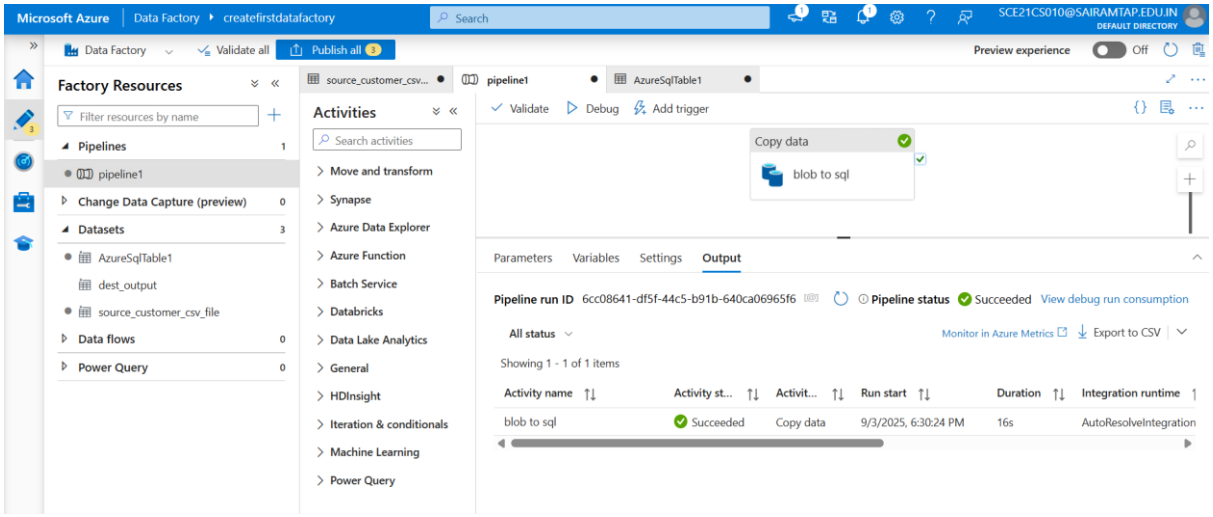
Create Cancel Test connection

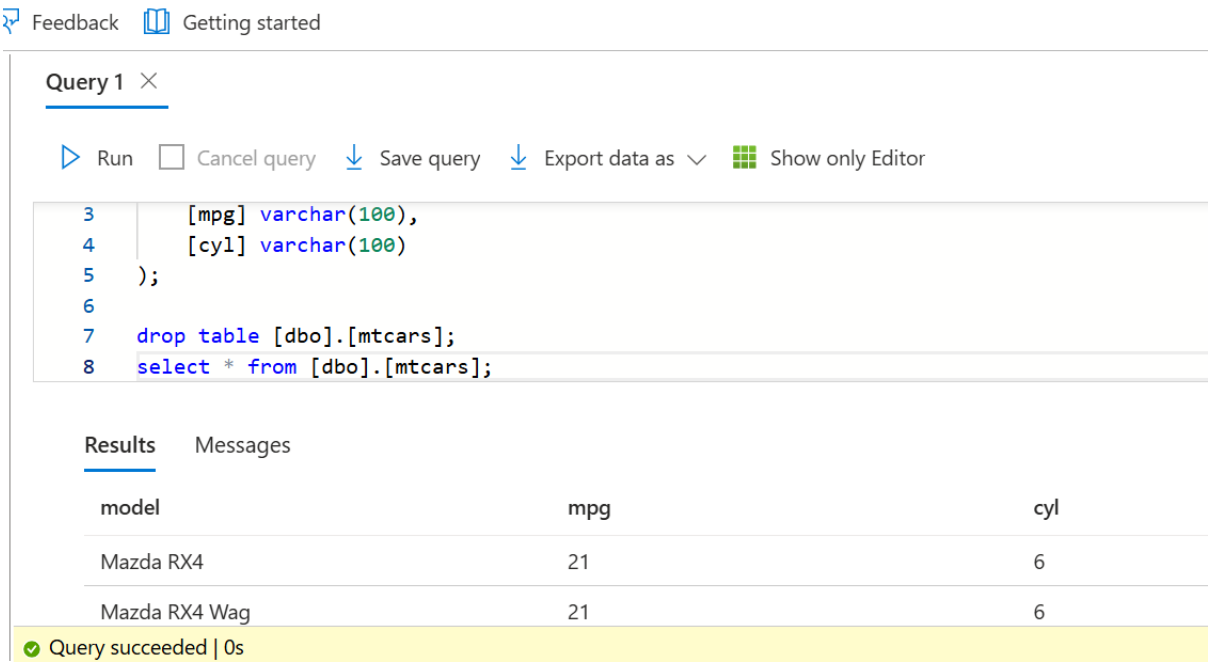





Microsoft Azure
Data Factory
createfirstdatafactory
Search
SCE21CS010@SAIRAMTAP.EDU.IN
DEFAULT DIRECTORY
Validate all
Publish all 1
Factory Resources
Filter resources by name
Pipelines 1
pipeline1
Change Data Capture (preview) 0
Datasets 2
dest_output
source_customer_csv_file
Data flows 0
Power Query 0
source_customer_csv...
pipeline1
Activities
Search activities
Move and transform
Copy data
Data flow
Synapse
Azure Data Explorer
Azure Function
Batch Service
Databricks
Data Lake Analytics
General
HDInsight
Iteration & conditionals
Validate
Validate copy ru
General
Source
Sink
Sink dataset *
Set properties
Name
AzureSqlTable1
Linked service *
AzureSqlDatabase1
Table name
dbo.Customers
Enter manually
Import schema
From connection/store
None
Advanced
OK
Back
Cancel


Microsoft Azure
Data Factory
createfirstdatafactory
Search
SCE21CS010@SAIRAMTAP.EDU.IN
DEFAULT DIRECTORY
Data Factory
Validate all
Publish all 3
Preview experience
Off
Factory Resources
Filter resources by name
Pipelines 1
pipeline1
Change Data Capture (preview) 0
Datasets 3
AzureSqlTable1
dest_output
source_customer_csv_file
Data flows 0
Power Query 0
source_customer_csv...
pipeline1
AzureSqlTable1
Activities
Search activities
Move and transform
Copy data
Data flow
Synapse
Azure Data Explorer
Azure Function
Batch Service
Databricks
Data Lake Analytics
General
HDInsight
Iteration & conditionals
Validate
Validate copy runtime
Debug
Add trigger
Copy data
blob to sql
General
Source
Sink
Mapping
Settings
User properties
Sink dataset *
AzureSqlTable1
Open
New
Learn more
Write behavior
Insert
Upsert
Stored procedure
Bulk insert table lock
Yes
No
Table option
Use existing
Auto create table
Pre-copy script
Write batch timeout
e.g. 00:30:00


Microsoft Azure
Data Factory
createfirstdatafactory
Search
SCE21CS010@SAIRAMTAP.EDU.IN
DEFAULT DIRECTORY
Data Factory
Validate all
Publish all 3
Preview experience
Off
Factory Resources
Filter resources by name
Pipelines 1
pipeline1
Change Data Capture (preview) 0
Datasets 3
AzureSqlTable1
dest_output
source_customer_csv_file
Data flows 0
Power Query 0
source_customer_csv...
pipeline1
AzureSqlTable1
Activities
Search activities
Move and transform
Synapse
Azure Data Explorer
Azure Function
Batch Service
Databricks
Data Lake Analytics
General
HDInsight
Iteration & conditionals
Machine Learning
Power Query
Validate
Debug
Add trigger
Copy data
blob to sql
Parameters
Variables
Settings
Output
Pipeline run ID 6cc08641-df5f-44c5-b91b-640ca06965f6
Pipeline status
Succeeded
View debug run consumption
All status
Monitor in Azure Metrics
Export to CSV
Showing 1 - 1 of 1 items
Activity name
Activity st...
Activit...
Run start
Duration
Integration runtime
blob to sql
Succeeded
Copy data
9/3/2025, 6:30:24 PM
16s
AutoResolveIntegration



---

1) Azure DevOps

Azure DevOps is Microsoft's cloud-based platform that provides tools for collaboration, version control, build, testing, and deployment of applications. It is widely used for managing the end-to-end software development lifecycle (SDLC).

Key Components:

- Azure Repos → Git repositories for source code management.
- Azure Pipelines → Continuous Integration & Continuous Deployment (CI/CD).
- Azure Boards → Agile planning (work items, user stories, sprint planning, Kanban boards).
- Azure Artifacts → Package management (NuGet, npm, Maven, etc.).
- Azure Test Plans → Testing and quality assurance.

---

2) CI/CD Integration

CI/CD stands for Continuous Integration and Continuous Deployment/Delivery. It is a DevOps practice that automates building, testing, and deploying applications to ensure faster and more reliable delivery.

Breakdown:

- **CI (Continuous Integration):**
  - Developers push code frequently to a shared repo (e.g., Azure Repos, GitHub).
  - Code is automatically built and tested using pipelines.

- Ensures issues are caught early.

- **CD (Continuous Delivery/Deployment):**
  - Automates the release of validated code to different environments (Dev, QA, UAT, Prod).
  - Continuous Delivery → requires manual approval before production.
  - Continuous Deployment → fully automated deployment into production.

Azure DevOps CI/CD Example Flow:

1. Developer pushes code to Azure Repos (Git).
2. Pipeline (CI) triggers → builds app, runs unit tests.
3. Release pipeline (CD) → deploys to staging → approval → production.
4. Monitoring & feedback loop via dashboards/logs.

In short:

- Azure DevOps = Platform with tools for DevOps.
- CI/CD Integration = The practice of automating build/test/deployment pipelines, often implemented in Azure Pipelines.

Key Features and Services – Global Data Centers

- **Global Presence** – Azure has **data centers across the world** (regions).
- **Regions** – A geographical area containing one or more data centers (e.g., East US, Central India).
- **Availability Zones (AZs)** – Physically separate data centers within a region to provide **high availability** and **fault tolerance**.
- **Paired Regions** – Each region is paired with another (for disaster recovery).
- **Global Network** – Low-latency, high-speed connectivity between regions.
- **Compliance & Security** – Meets local laws (GDPR, HIPAA, etc.) by placing data in the right region.

In short: Azure's global data centers ensure **availability, redundancy, disaster recovery, and compliance** worldwide.

---

2) Key Features and Services – Azure Subscription and Resource Groups

- **Azure Subscription**:
  - Acts as a **container for resources** (VMs, databases, storage, etc.).
  - Defines **billing boundary** (you get billed per subscription).
  - Can have multiple subscriptions for **separation of environments** (Dev, Test, Prod).
  - Controls **access management** via Azure RBAC.

- **Resource Group (RG)**:

- A **logical container** inside a subscription.
- Used to group related resources (VM, Storage, Database for one app).
- Supports **lifecycle management** (deploy, update, delete as a unit).
- Helps in **organization, access control, and cost tracking**.

---

1) You have a pipeline that ingests files from an on-premises SQL Server every hour. Sometimes the files arrive late. How would you design the pipeline to handle late-arriving files without failing? – tumbling window triggers

Trigger (hourly with 15-min offset)
↓
Get Metadata (check file)
↓
Until Activity (wait & re-check)
↓
Copy Activity (SQL → Data Lake)
↓
Validation / Transform

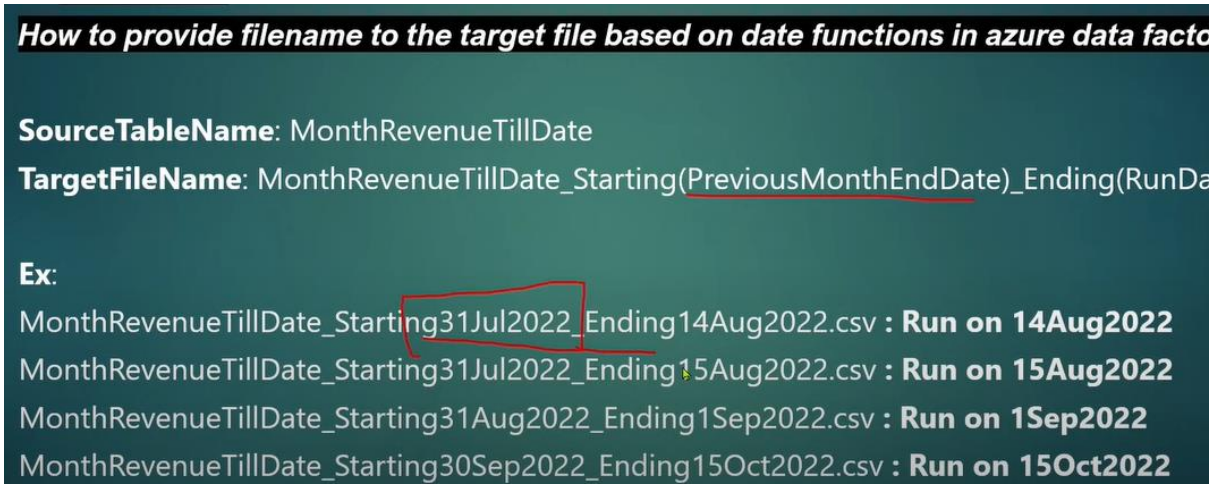
You are using **Copy Activity** to move JSON files from Blob to SQL Database, but some rows fail because of schema drift (extra columns added in source). How will you handle these error rows?

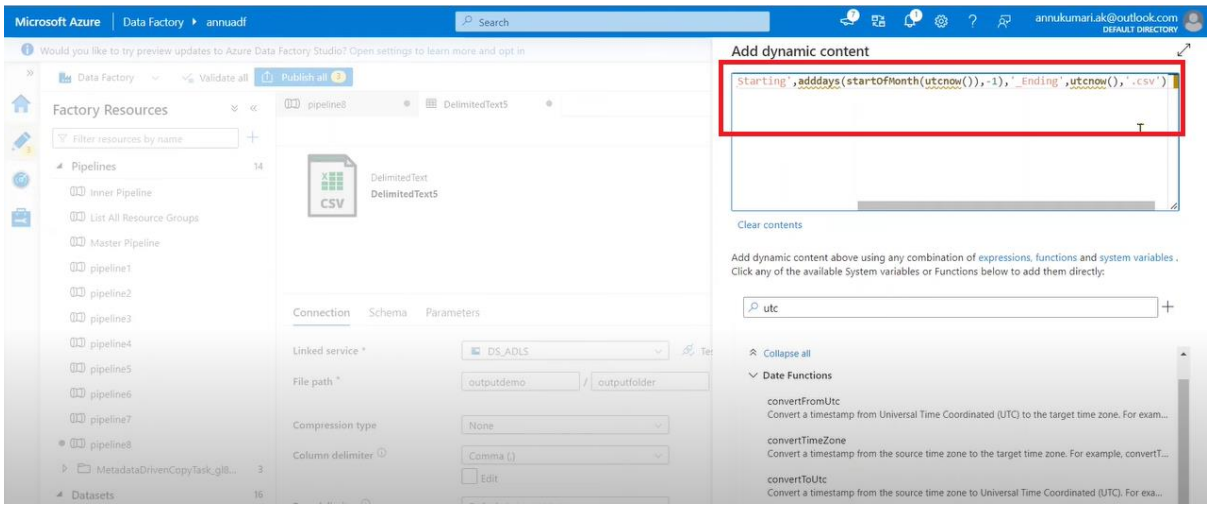
Key Options in Copy Activity → **Fault tolerance** Tab

Option	What It Does
Skip incompatible rows	Ignores rows with errors; loads only valid rows.
Redirect rows to	Sends bad rows to another sink (Blob/ADLS/SQL) with an error report.
Maximum error rows	Limits how many bad rows to skip/redirect before failing the activity.

---

1. How to provide filename to the target file based on date function






```
ys(startOfMonth(utcnow()),-1,'ddMMMyyyy'),'_Ending',utcnow(),'.csv')
```

@concat('monthStart', adddays('startofMonth', (utcnow()),'Ending', utcnow())

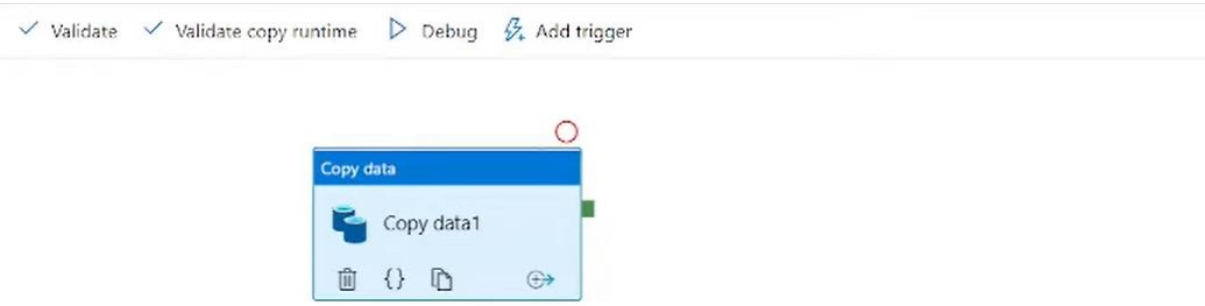
---

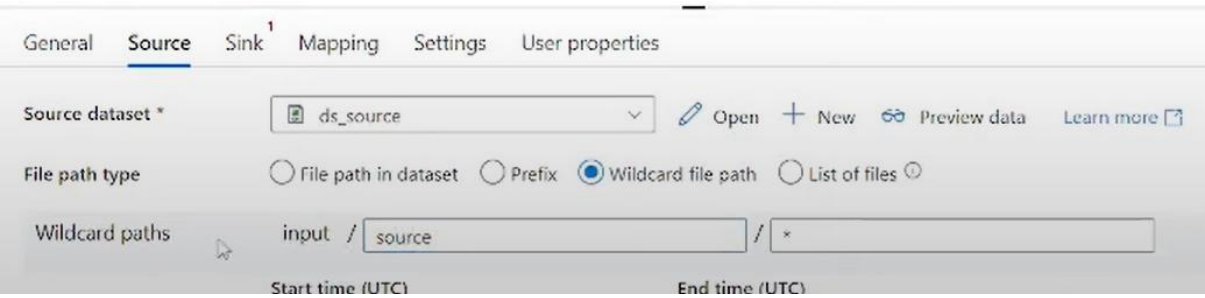
2. How to copy all the file present in folder to another folder using ADF

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
England.Batsman.csv	8/14/2022, 6:23:53 PM	Hot (Inferred)		Block blob	115 B	Available
England.Bowlers.csv	8/14/2022, 6:23:53 PM	Hot (Inferred)		Block blob	115 B	Available
India.Batsman.csv	8/14/2022, 6:24:33 PM	Hot (Inferred)		Block blob	105 B	Available
India.Bowlers.csv	8/14/2022, 6:24:33 PM	Hot (Inferred)		Block blob	105 B	Available

Method – 1

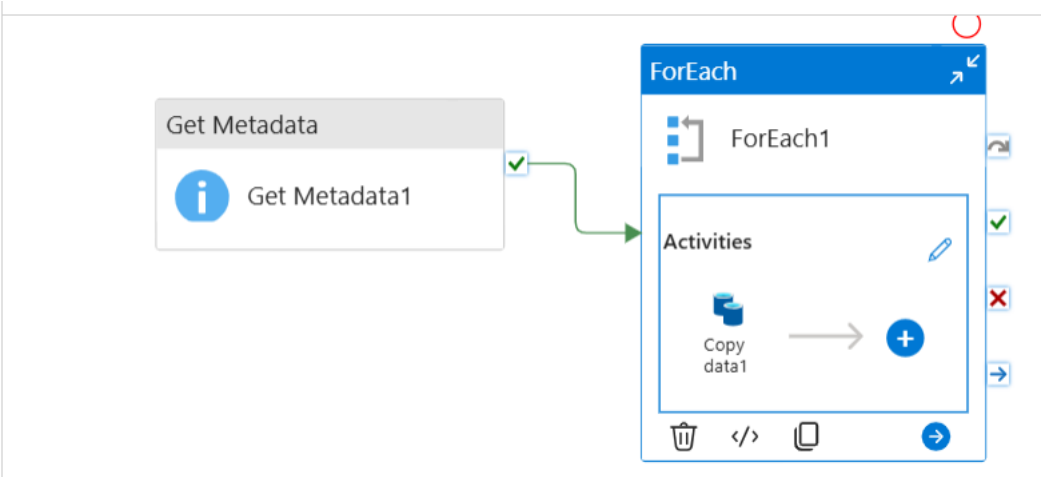
- **Configure Source Dataset Create a dataset for your source (e.g., Azure Blob Storage, ADLS). Use the Wildcard File Path option in the source settings to specify multiple files or folders. For example: Folder Path: demo/source/ Wildcard File Name: *.csv This will select all .csv files in the specified folder.**





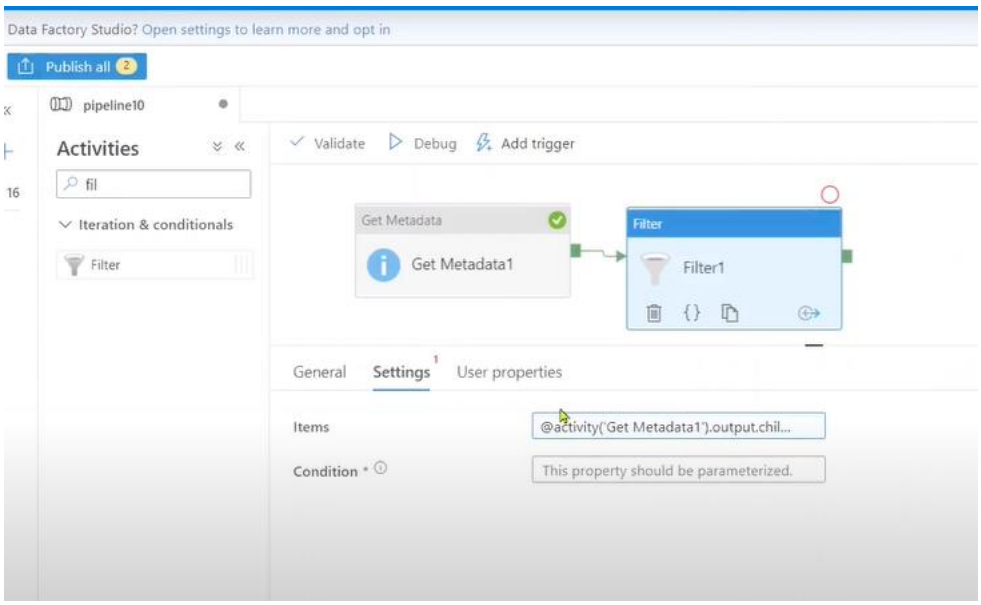
Method – 2 (dynamic copying)

- **Use ForEach Activity for Dynamic Copying If you need to copy files from multiple folders dynamically: Use a get metadata Activity to fetch folder/file paths. Pass the output of the get metadata (child item) to a ForEach Activity. Inside the ForEach loop, configure the Copy Activity with dynamic parameters for source and sink paths: Source Path: @item().SourcePath Sink Path: @item().DestinationPath**

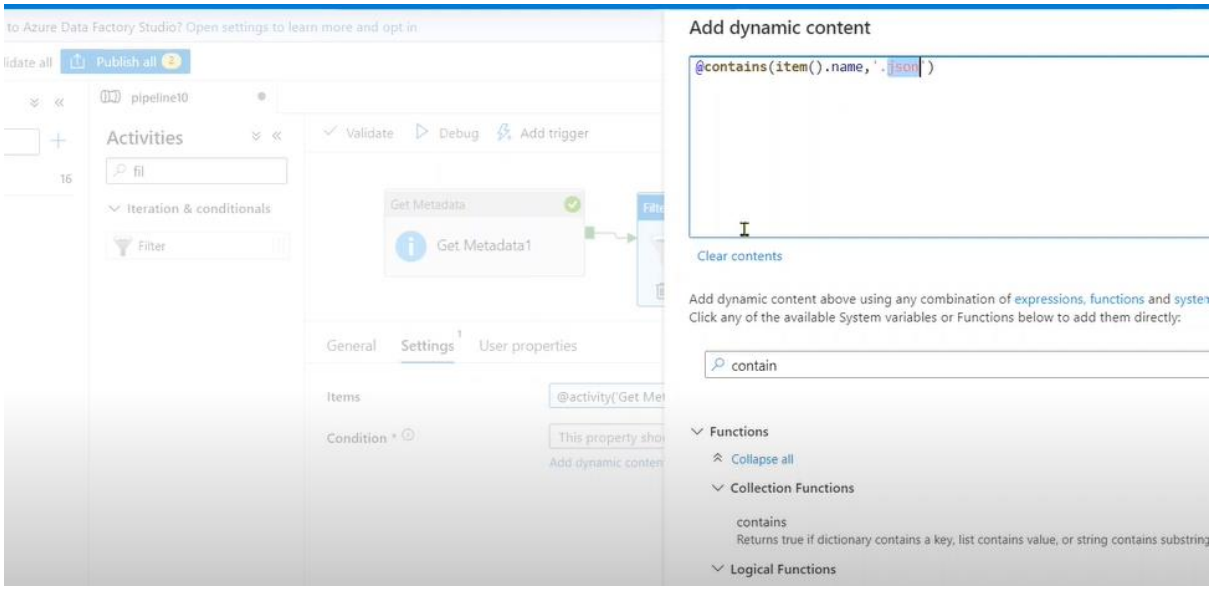


3. **How to copy files of specific format from one ADLS folder to another using ADF pipeline**

Step-1: get all file information using **Get Metadata activity** and the output to the **filter activity**

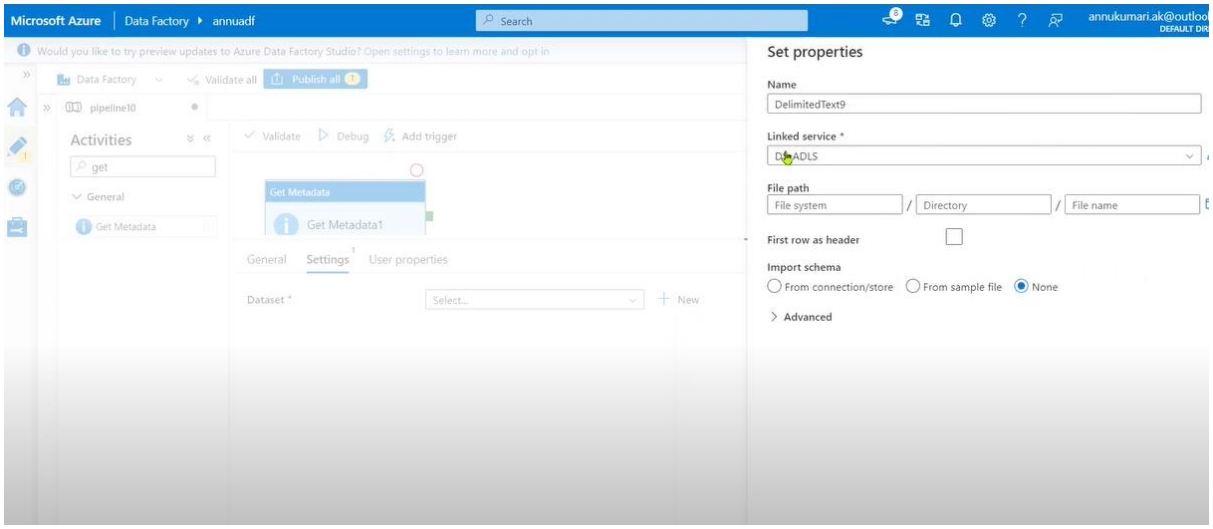


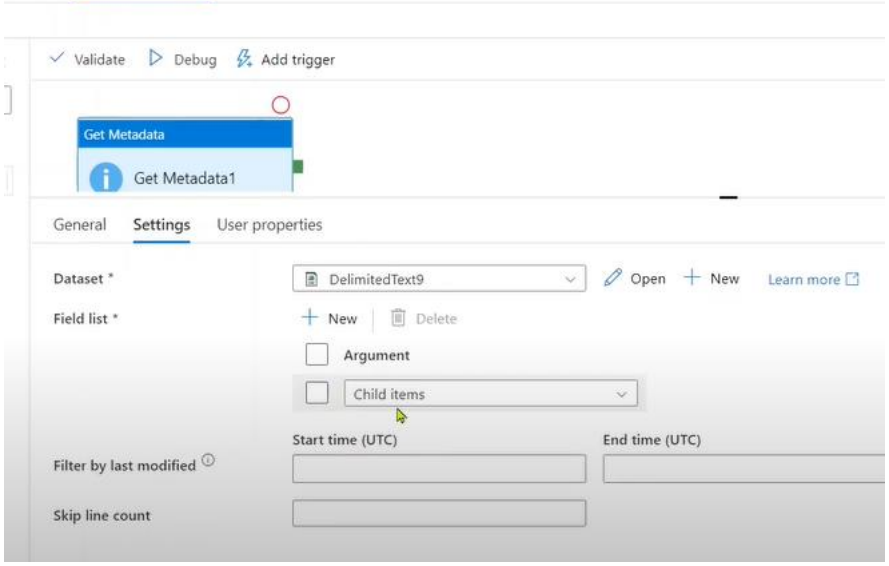
We can give dynamic file type also



4. **How to load all the ADLS foldername and count of files inside each folder into SQL table.**

Not given any file path



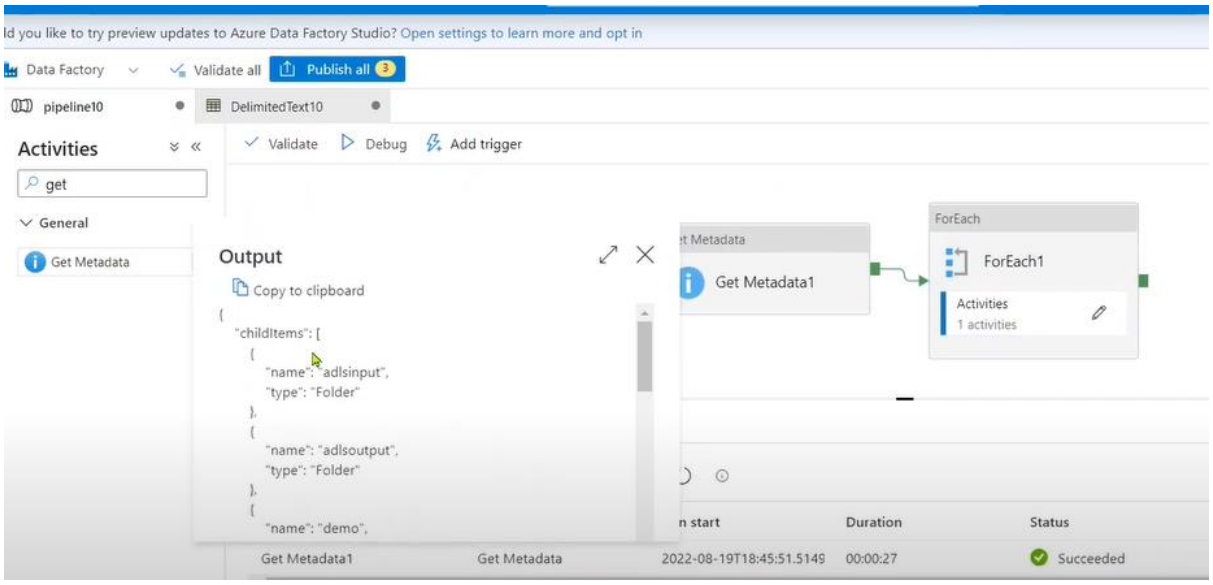


Output

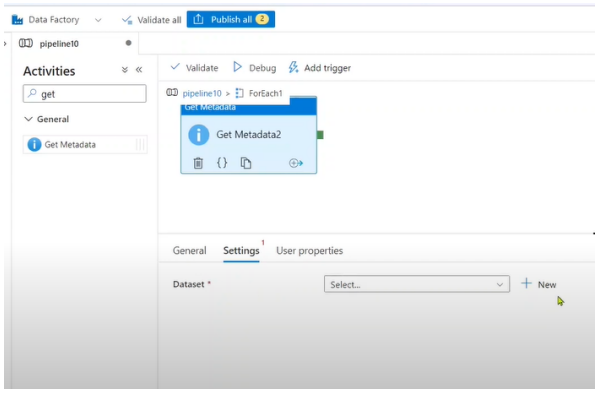
Copy to clipboard

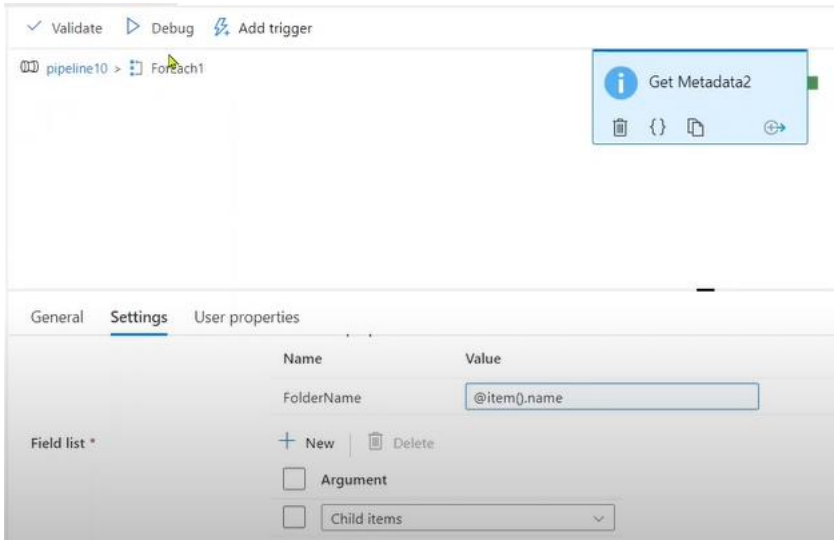

```
{
  "childItems": [
    {
      "name": "adlsinput",
      "type": "Folder"
    },
    {
      "name": "adlsoutput",
      "type": "Folder"
    },
    {
      "name": "demo",
      "type": "Folder"
    },
    {
      "name": "demofol",
      "type": "Folder"
    },
    {
      "name": "dev",
      "type": "Folder"
    },
    {
      "name": "test",
      "type": "Folder"
    }
  ],
  "effectiveIntegrationRuntime": "AutoResolveIntegrationRuntime (East US)",
  "executionDuration": 0,
  "durationInQueue": {
```

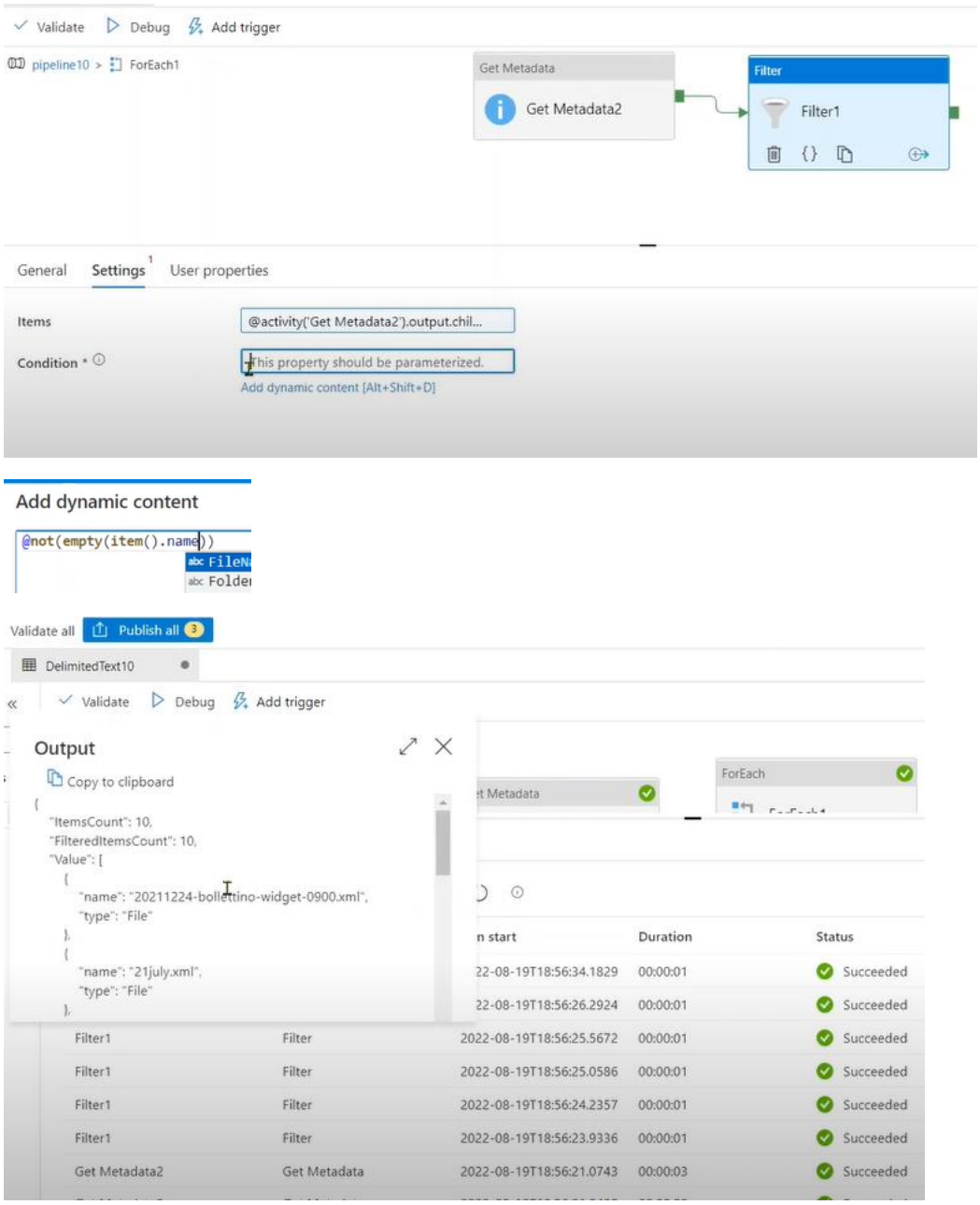
The ouput of get metadata 1 is given to For each activity



Inside that another get metadata is called

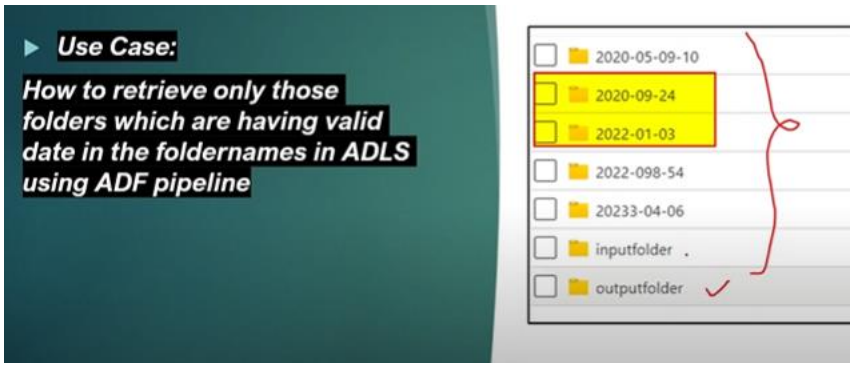






---

5. How to retrieve folder having valid date in the FolderName



1. Use Get Metadata on the parent container.

- Field list → Child items.

2. In a ForEach loop over @activity('Get Metadata1').output.childItems.
3. Inside ForEach add If Condition with an expression that checks if the folder name is a valid date.

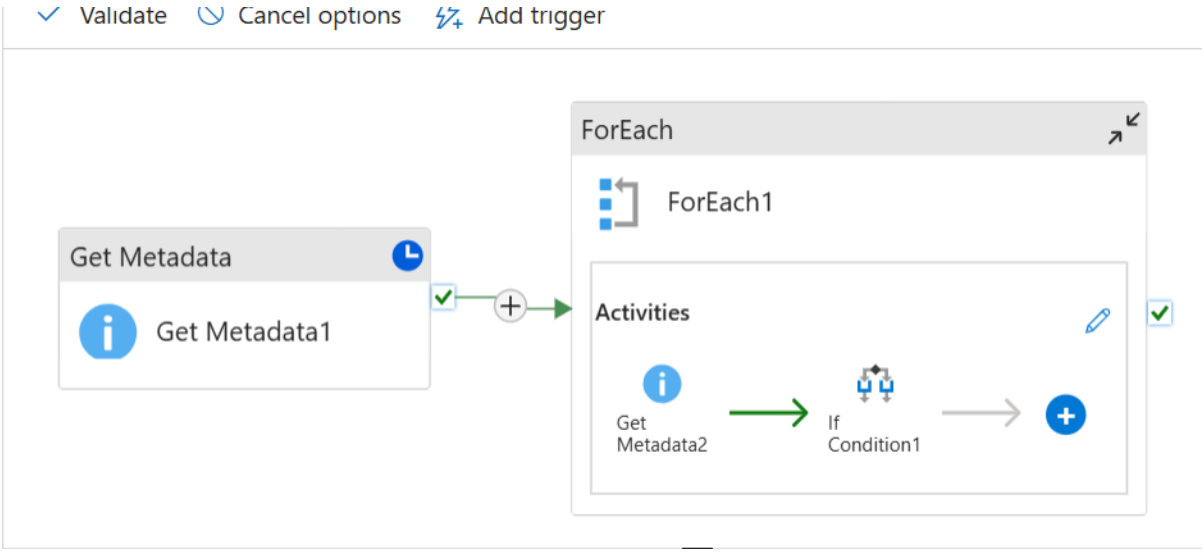
If your folders are yyyyMMdd: @not(empty(tryParseDateTime(item().name, 'yyyyMMdd')))

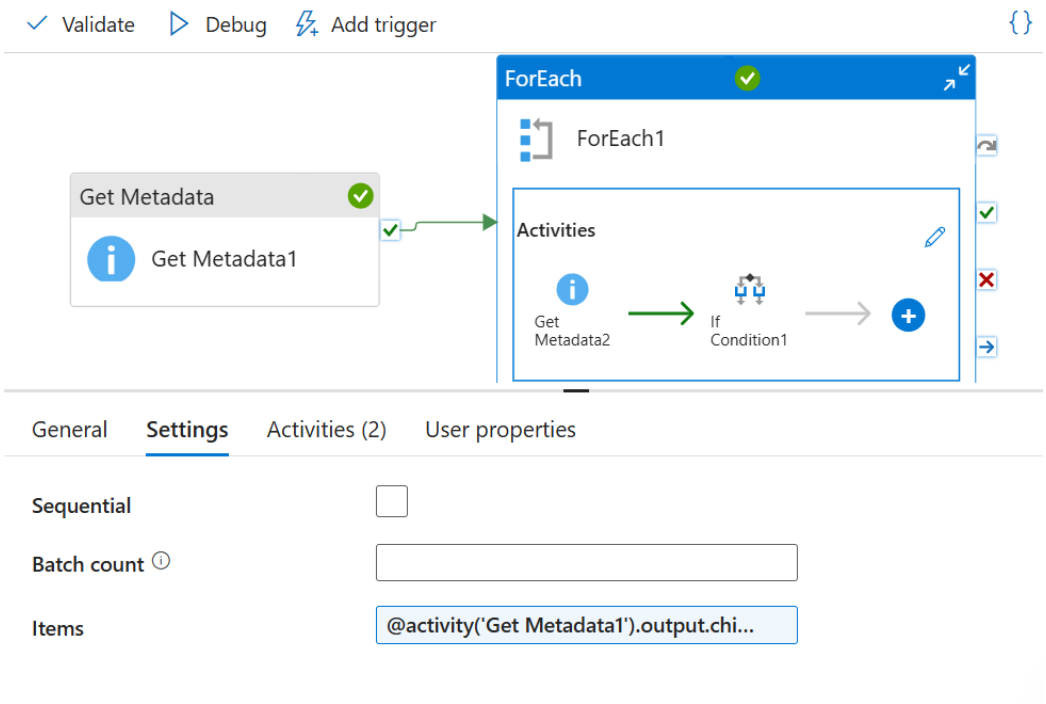
If your folders are yyyy-MM-dd: @not(empty(tryParseDateTime(item().name, 'yyyy-MM-dd')))

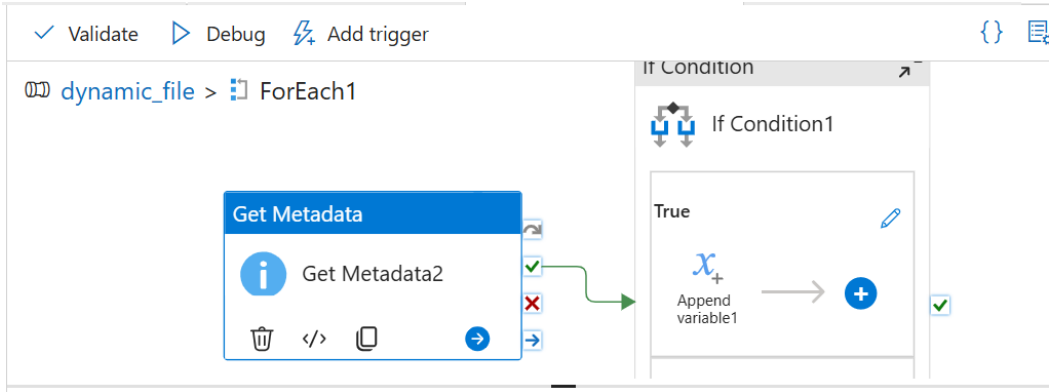
---

6. **How to get the list of empty and non-empty ADLS folder.**

Get metadata → for each → get metadata (directory) → IF (True) → append variable(file name) IF (false) → append variable(file name)







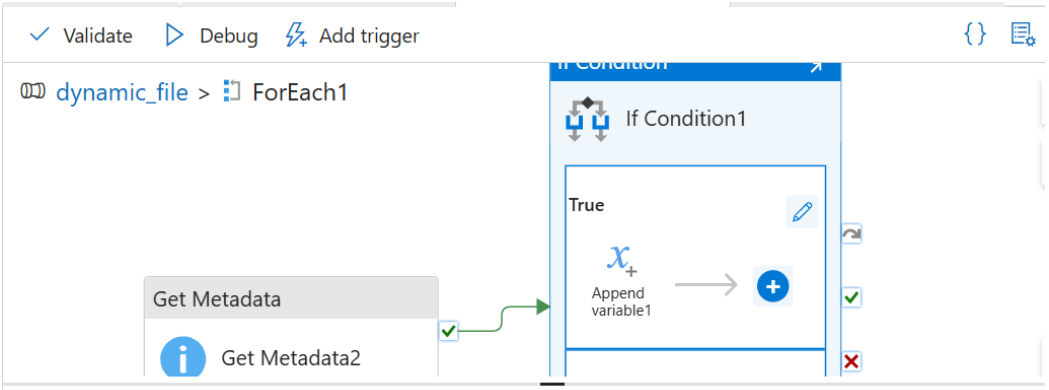
General Settings User properties

Dataset * filecontainer

Open New Learn more

Dataset properties

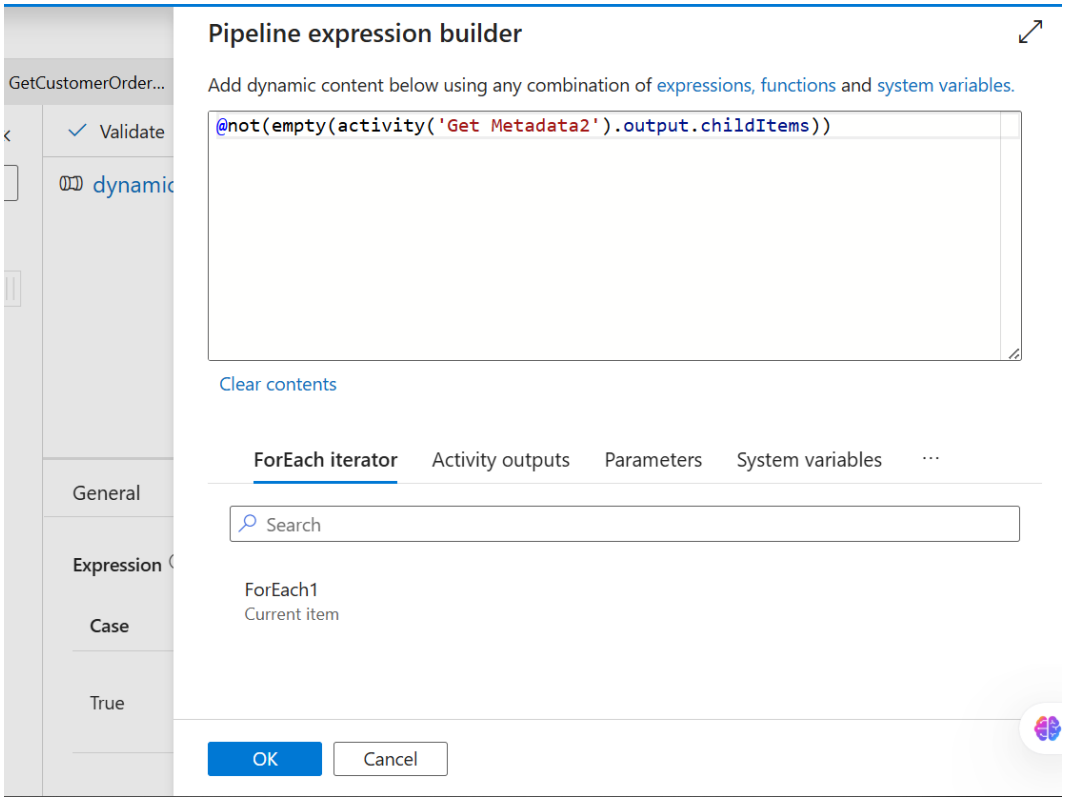
Name	Value	Type
directory_name	@item().name	String

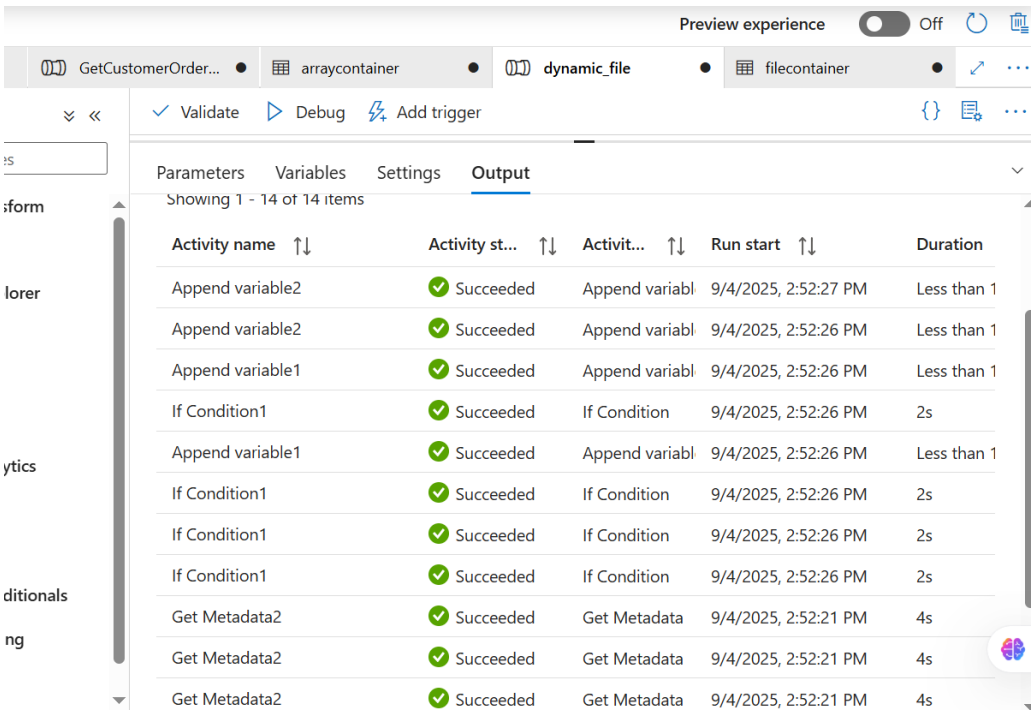


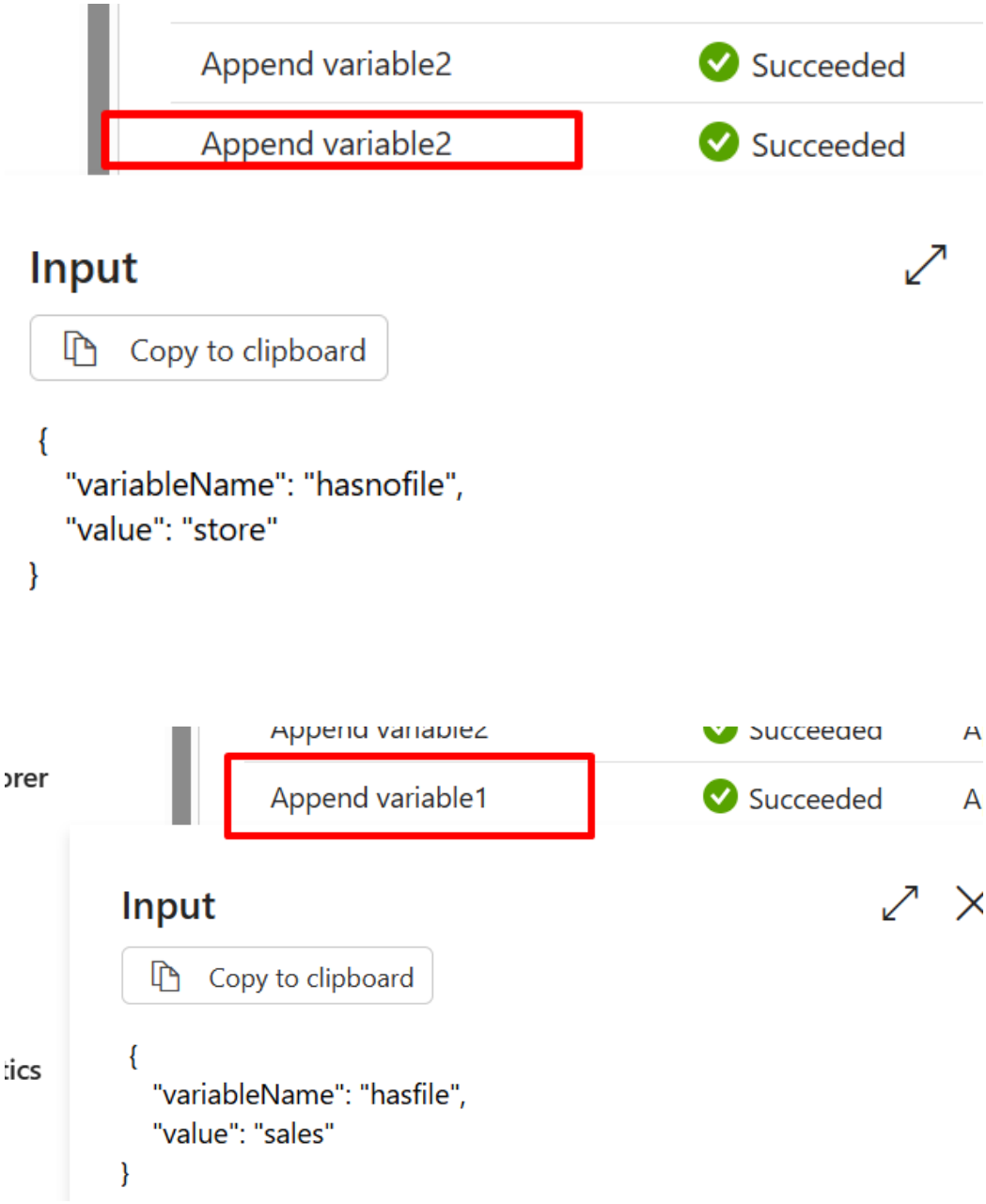
General Activities (2) User properties

Expression @not(empty(activity('Get Metadata2'...

Case	Activity
True	Append variable1 1 Activity
	Append variable2


Pipeline expression builder
GetCustomerOrder...
Validate
dynamic
Add dynamic content below using any combination of expressions, functions and system variables.
@not(empty(activity('Get Metadata2').output.childItems))
Clear contents
ForEach iterator
Activity outputs
Parameters
System variables
General
Search
Expression
ForEach1
Case
Current item
True
OK
Cancel

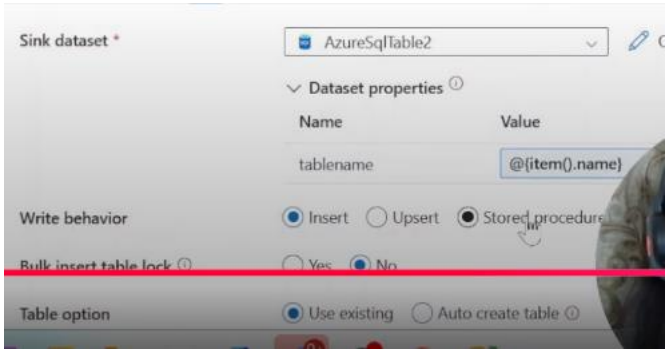

Preview experience
Off
GetCustomerOrder...
arraycontainer
dynamic_file
filecontainer
Validate
Debug
Add trigger
Parameters
Variables
Settings
Output
Showing 1 - 14 of 14 items
Activity name
Activity st...
Activit...
Run start
Duration
Append variable2 Succeeded Append variabl 9/4/2025, 2:52:27 PM Less than 1
Append variable2 Succeeded Append variabl 9/4/2025, 2:52:26 PM Less than 1
Append variable1 Succeeded Append variabl 9/4/2025, 2:52:26 PM Less than 1
If Condition1 Succeeded If Condition 9/4/2025, 2:52:26 PM 2s
Append variable1 Succeeded Append variabl 9/4/2025, 2:52:26 PM Less than 1
If Condition1 Succeeded If Condition 9/4/2025, 2:52:26 PM 2s
If Condition1 Succeeded If Condition 9/4/2025, 2:52:26 PM 2s
If Condition1 Succeeded If Condition 9/4/2025, 2:52:26 PM 2s
Get Metadata2 Succeeded Get Metadata 9/4/2025, 2:52:21 PM 4s
Get Metadata2 Succeeded Get Metadata 9/4/2025, 2:52:21 PM 4s
Get Metadata2 Succeeded Get Metadata 9/4/2025, 2:52:21 PM 4s



How to dynamically copy only matching files from ADLS to SQL?

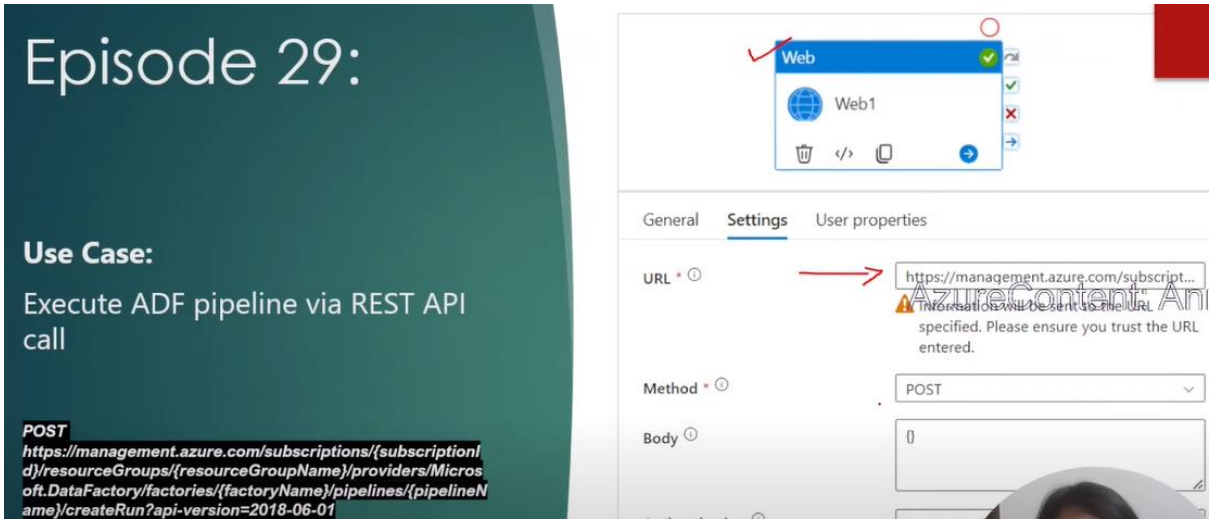
Lookup activity (to get the location from file location.csv) → for each → executive pipeline

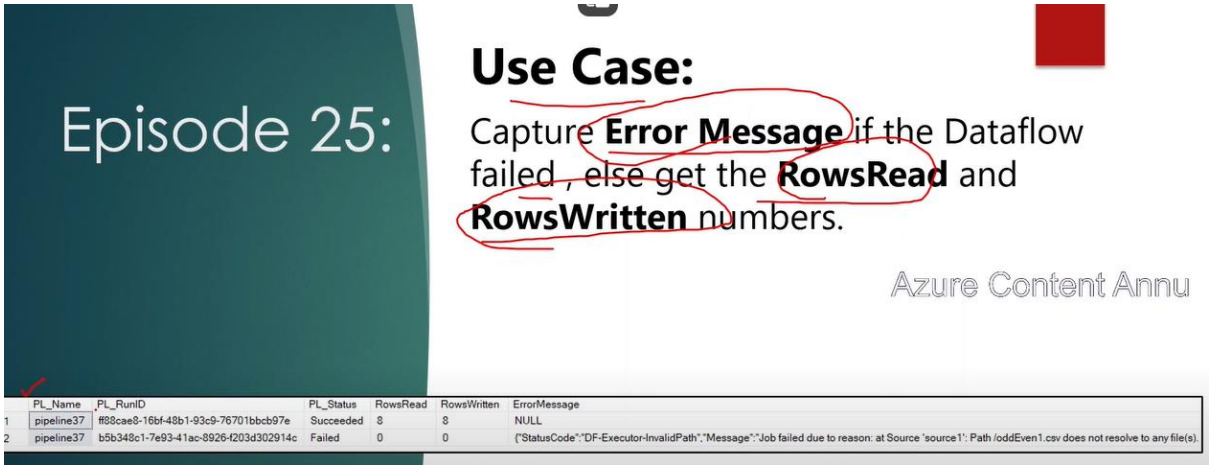
Get meta-activity (to get the file name like abc_argentina, abc_london, location.csv,….) → for each → if (@contain(item().name, location) → copy activity



auto create table (to create a new table)

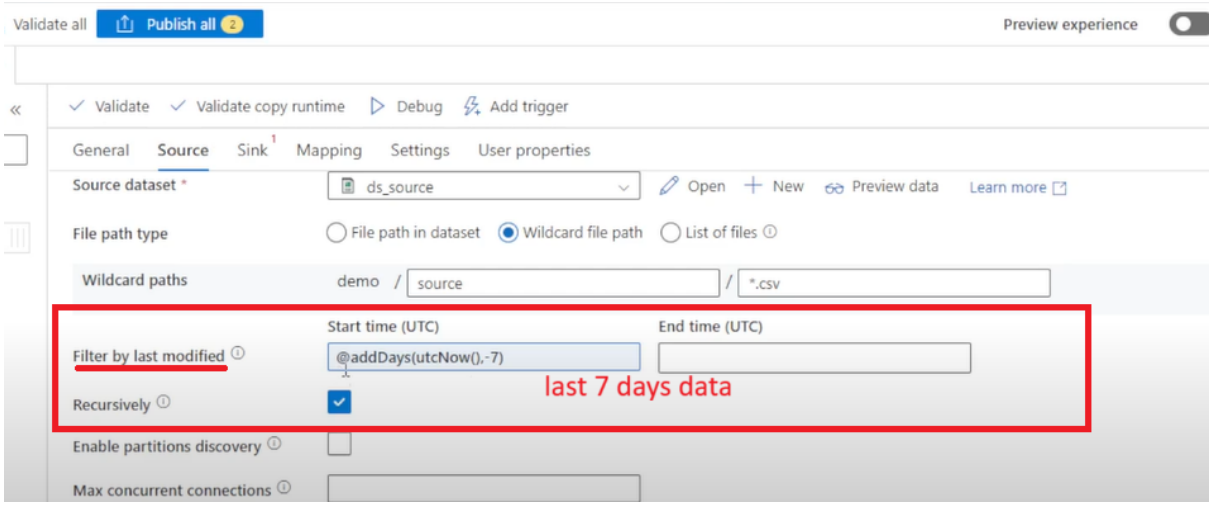
How to execute one ADF pipeline from other ADF pipeline via rest API??



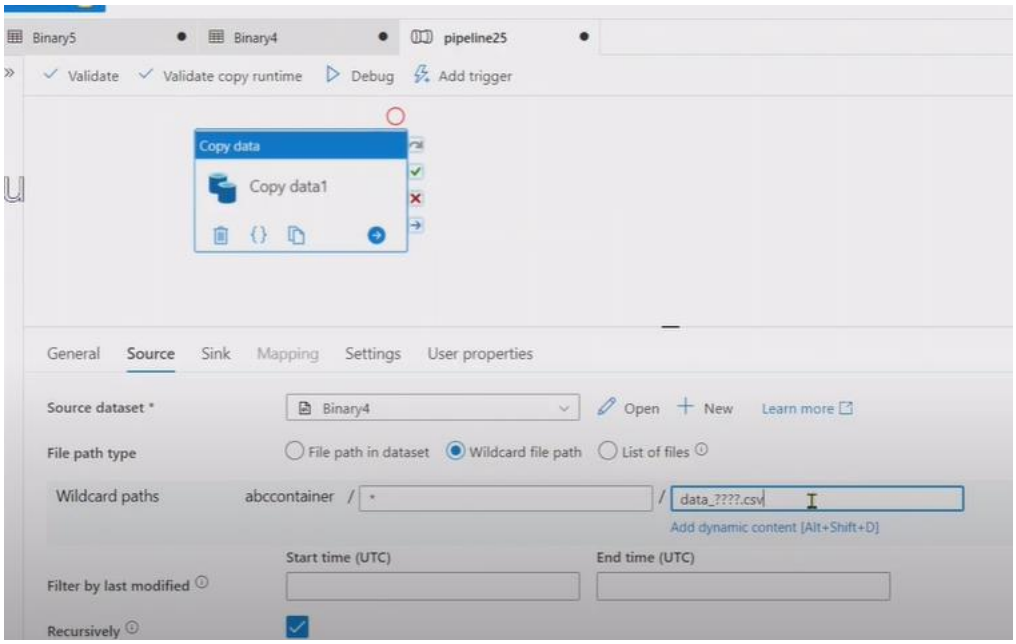


Capture error message if the dataflow failed, else (if succeed) get the RowsRead and RowsWritten numbers (in SQL table).

Copy files uploaded three days back from ADLS Gen2. → @addDays(utcNow(),-7) end-@utcnow()

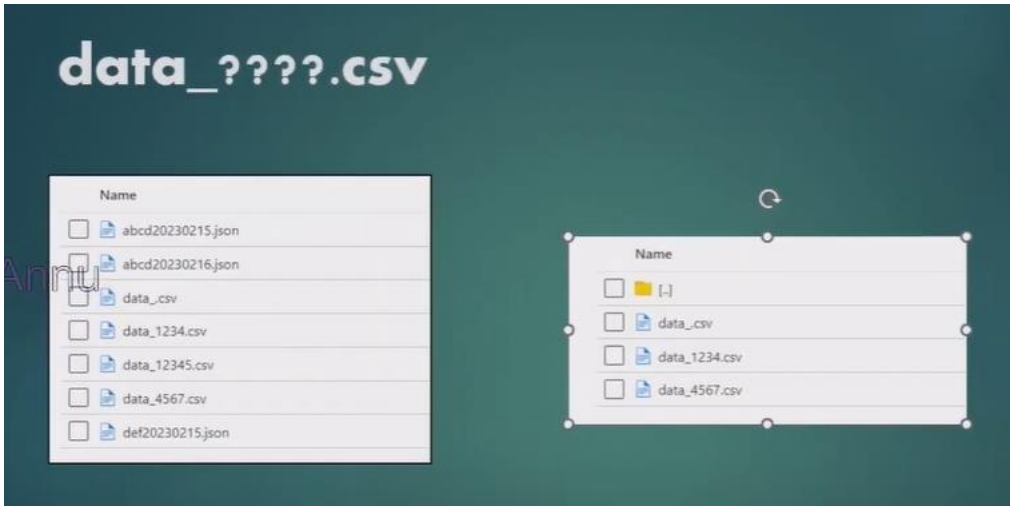


Wildcard

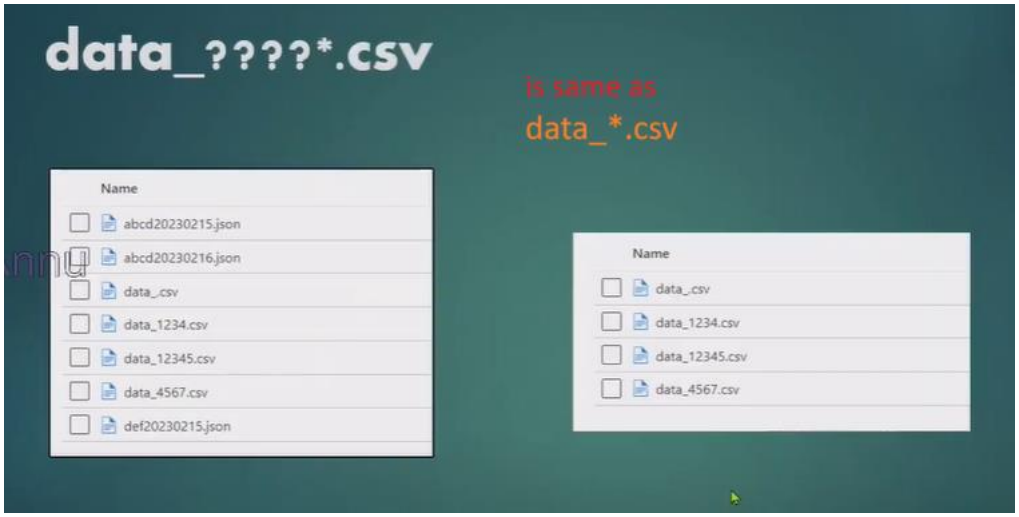


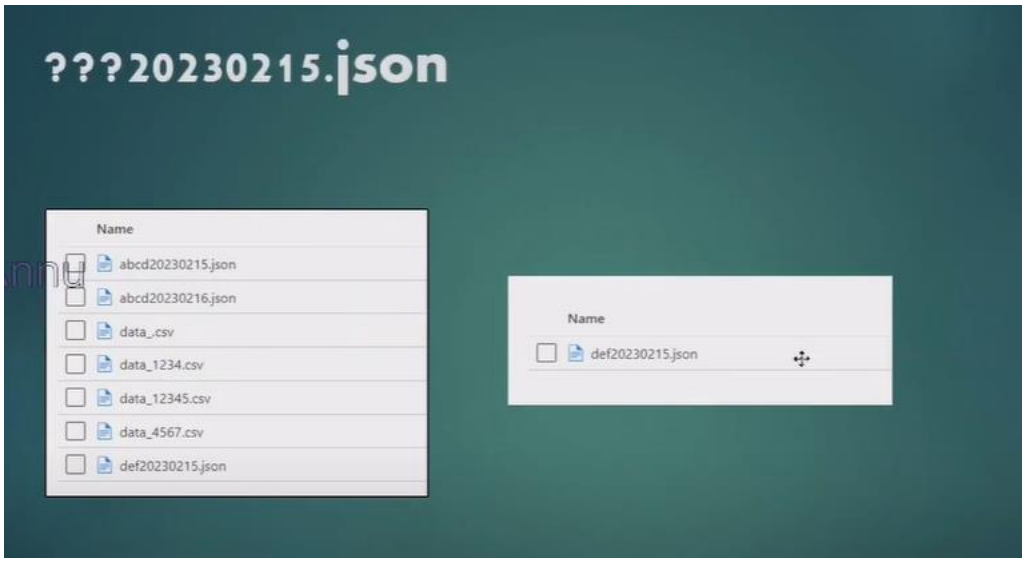
data_????.csv → matches data_.csv,data_12.csv, data_123.csv, data_1234.csv, data_tttt.csv,

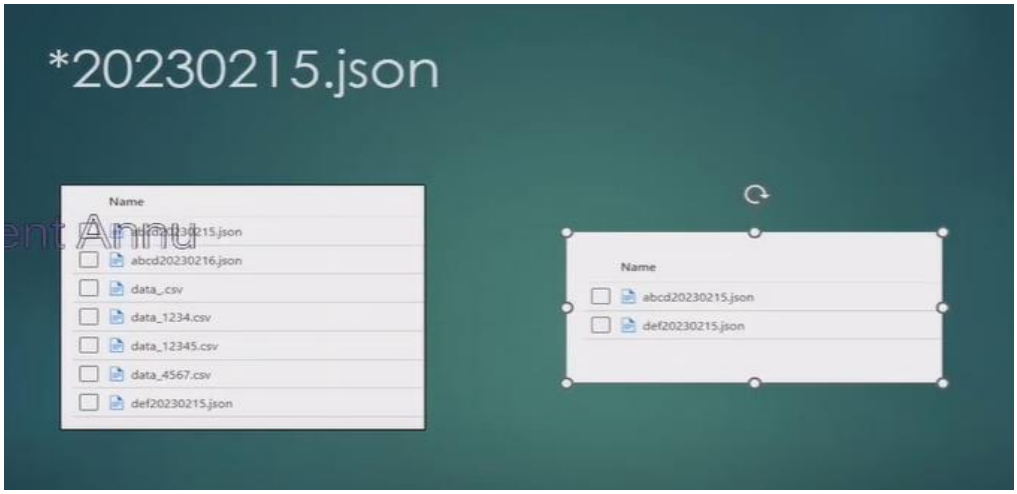
But not matches data_abced.csv



We can also use **data_.*** instead of file name

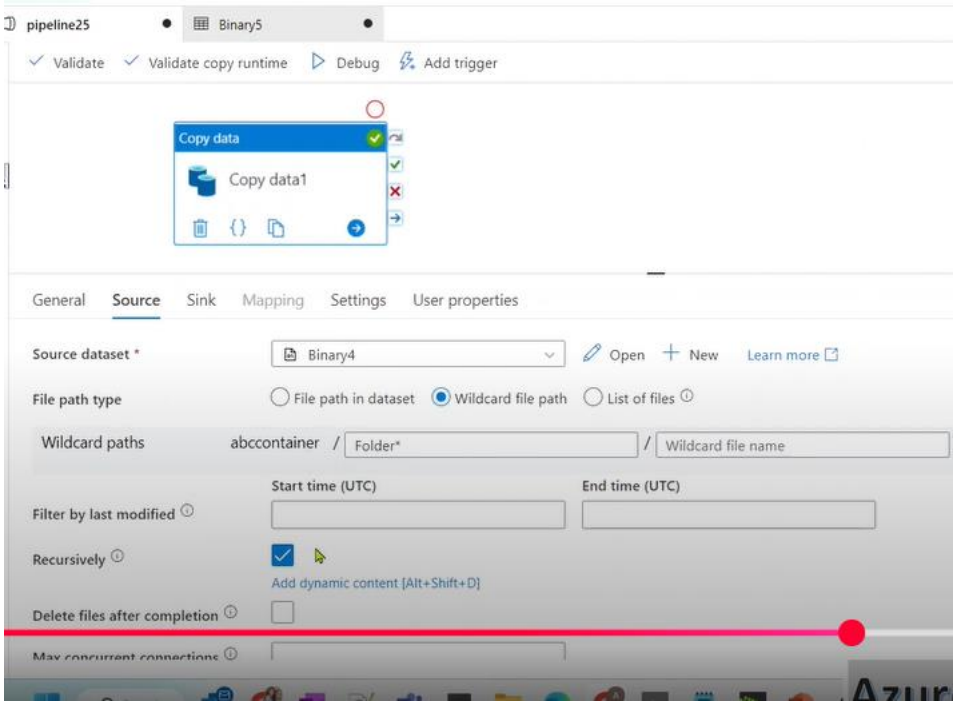

data_????*.csv
is same as
data_*.csv
Name
abcd20230215.json
abcd20230216.json
data_.csv
data_1234.csv
data_12345.csv
data_4567.csv
def20230215.json
Name
data_.csv
data_1234.csv
data_12345.csv
data_4567.csv


???20230215.json
Name
abcd20230215.json
abcd20230216.json
data_.csv
data_1234.csv
data_12345.csv
data_4567.csv
def20230215.json
Name
def20230215.json

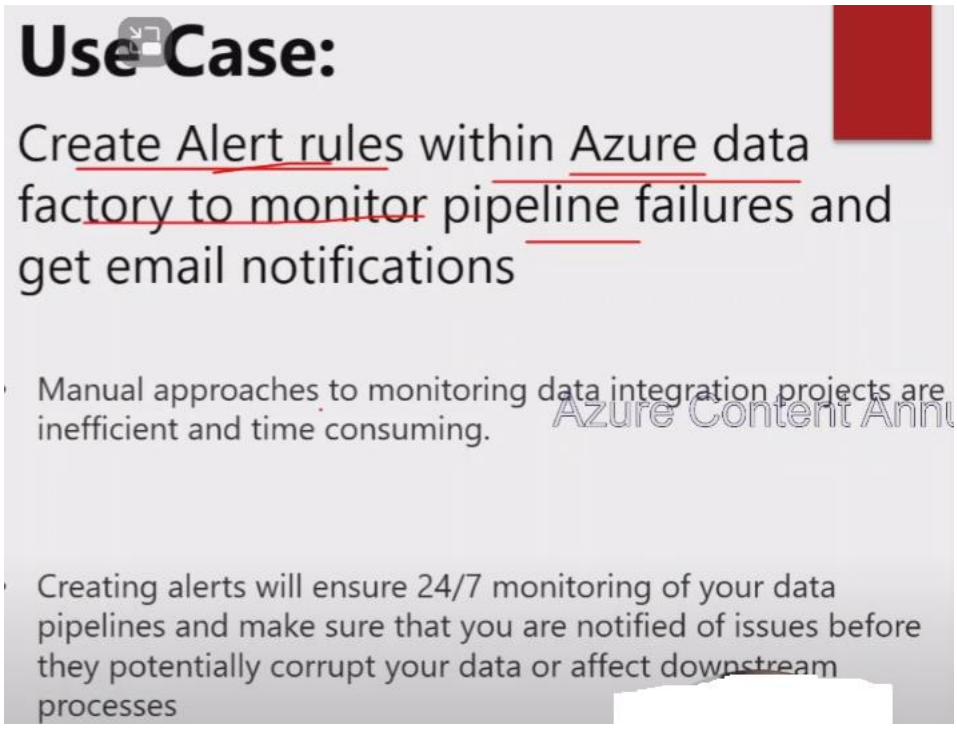

*20230215.json
Name
abcd20230215.json
abcd20230216.json
data_.csv
data_1234.csv
data_12345.csv
data_4567.csv
def20230215.json
Name
abcd20230215.json
def20230215.json

Wildcard

folderPath	fileName	recursive	Source folder structure and filter result (files in **bold** are retrieved)
Folder*	(Empty, use default)	false	FolderA **File1.csv** **File2.json** Subfolder1 File3.csv File4.json File5.csv AnotherFolderB File6.csv
Folder*	(Empty, use default)	true	FolderA **File1.csv** **File2.json** Subfolder1 **File3.csv** **File4.json** **File5.csv** AnotherFolderB File6.csv
Folder*	*.csv	false	FolderA **File1.csv** File2.json Subfolder1 File3.csv File4.json File5.csv AnotherFolderB File6.csv
Folder*	*.csv	true	FolderA **File1.csv** File2.json Subfolder1 **File3.csv** File4.json **File5.csv** AnotherFolderB File6.csv



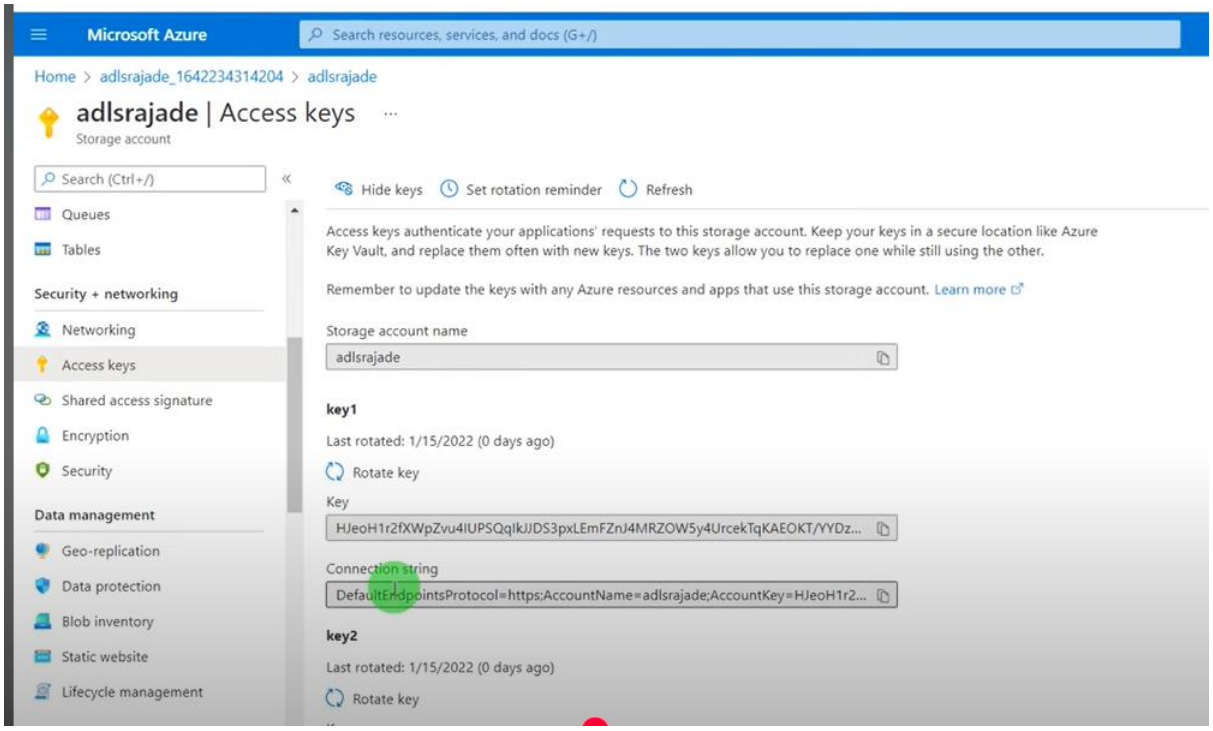
How to get email notification for pipeline failure by creating alert rules within ADF?

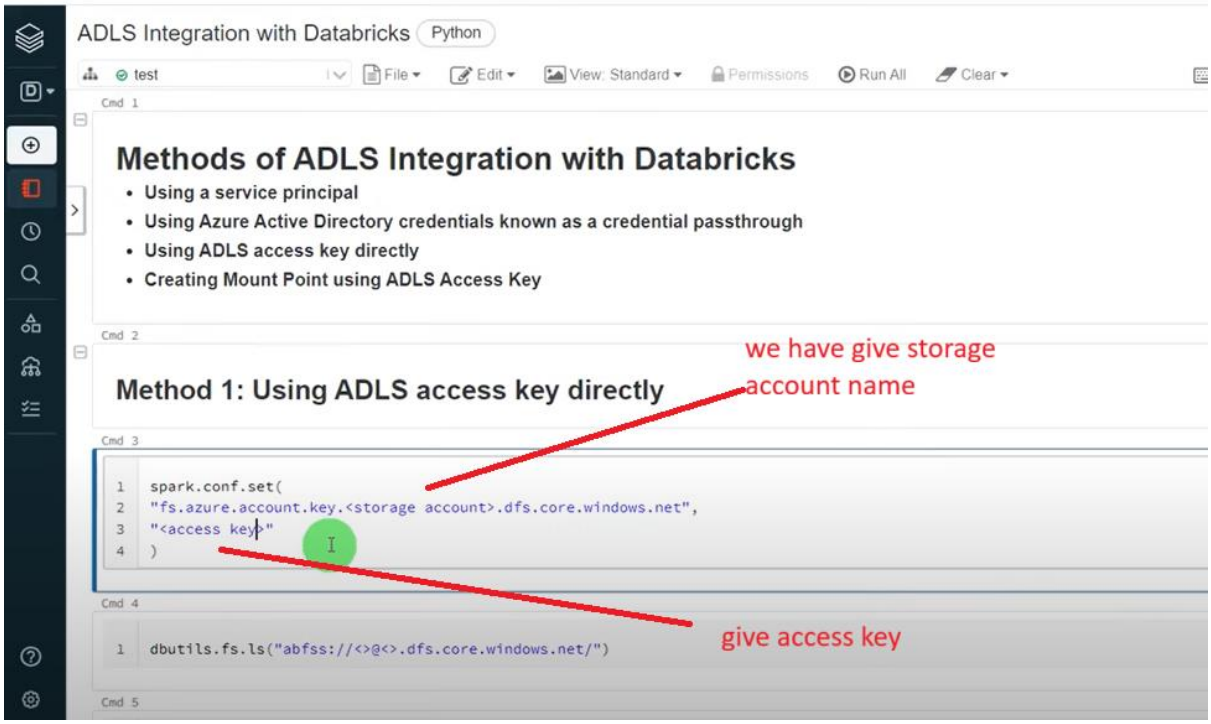


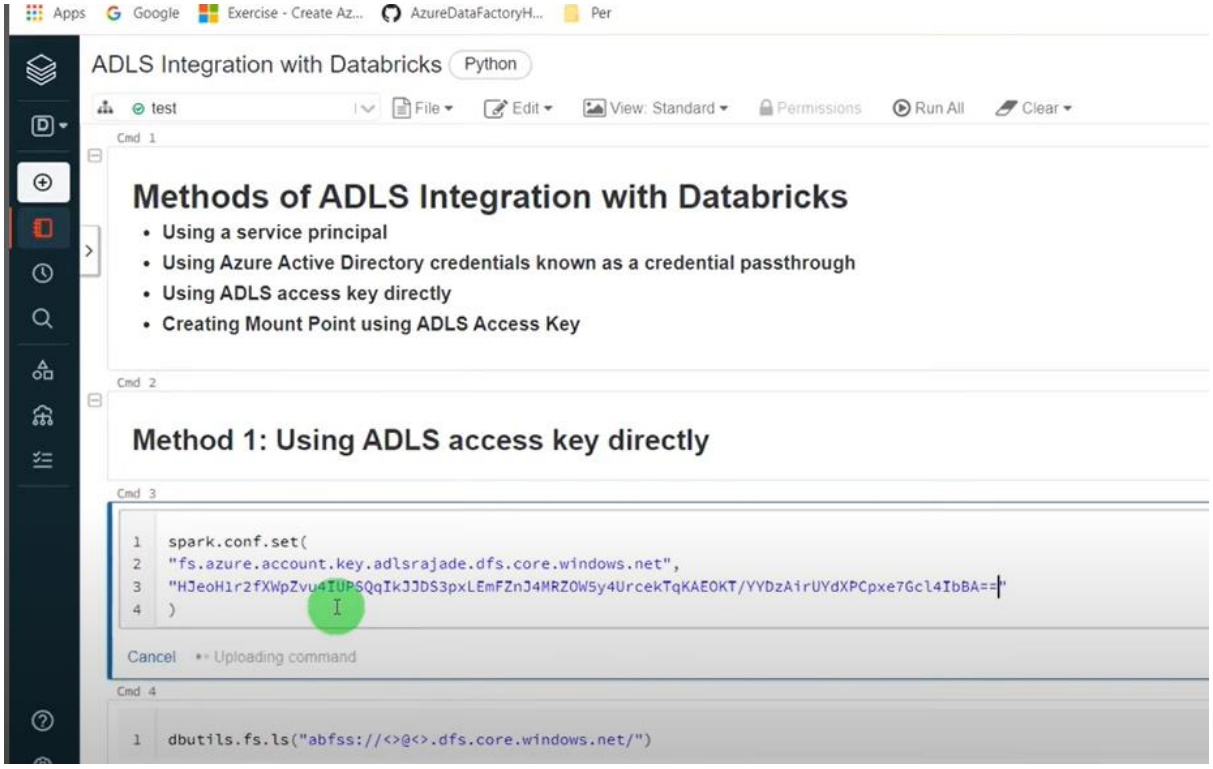
ADLS integration with Databricks

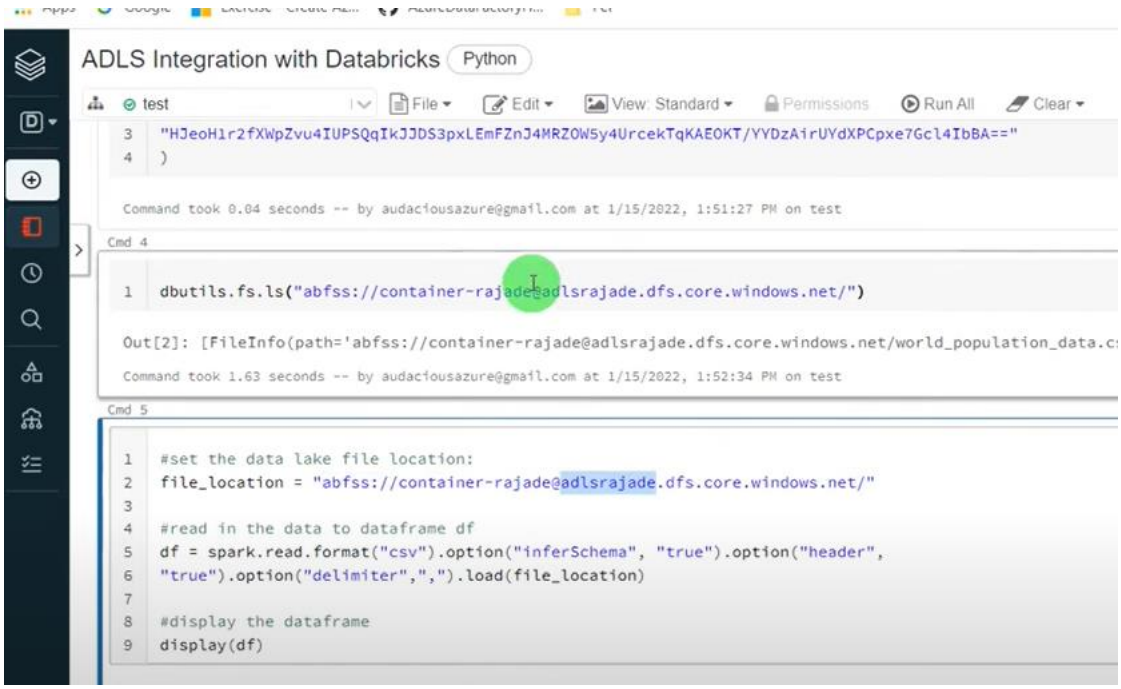
Method-3: using ADLS Access key directly

Once created ADLS storage account → get access key : “click tab on left menu Access Key”

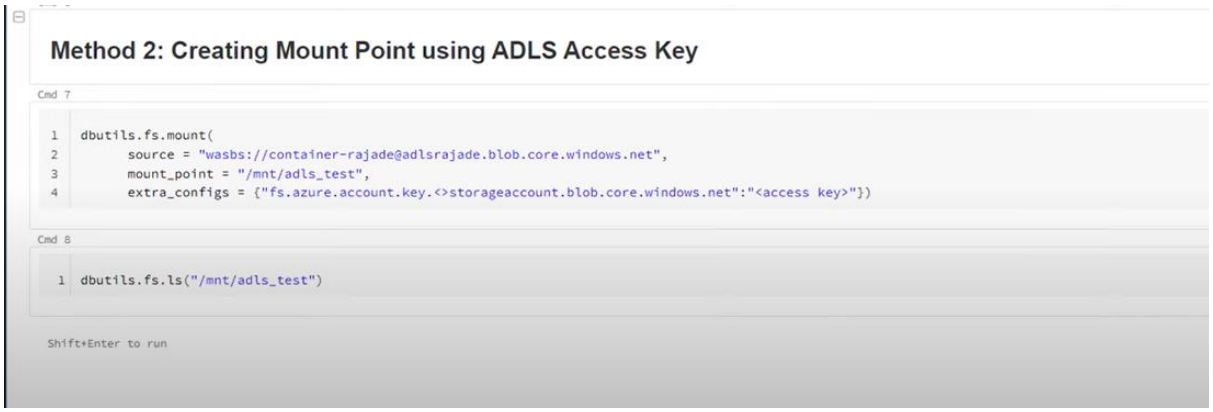

Microsoft Azure
Search resources, services, and docs (G+/)
Home > adlsrajade_1642234314204 > adlsrajade
adlsrajade | Access keys
Storage account
Search (Ctrl+/)
Queues
Tables
Security + networking
Networking
Access keys
Shared access signature
Encryption
Security
Data management
Geo-replication
Data protection
Blob inventory
Static website
Lifecycle management
Hide keys
Set rotation reminder
Refresh
Access keys authenticate your applications' requests to this storage account. Keep your keys in a secure location like Azure Key Vault, and replace them often with new keys. The two keys allow you to replace one while still using the other.
Remember to update the keys with any Azure resources and apps that use this storage account. Learn more
Storage account name
adlsrajade
key1
Last rotated: 1/15/2022 (0 days ago)
Rotate key
Key
HJeoH1r2fXWpZvu4IUPSQqIkJJDS3pxLEmFZnJ4MRZOW5y4UrcekTqKAEOKT/YYDz...
Connection string
DefaultEndpointsProtocol=https;AccountName=adlsrajade;AccountKey=HJeoH1r2...
key2
Last rotated: 1/15/2022 (0 days ago)
Rotate key


ADLS Integration with Databricks
Python
test
File
Edit
View: Standard
Permissions
Run All
Clear
Cmd 1
Methods of ADLS Integration with Databricks
Using a service principal
Using Azure Active Directory credentials known as a credential passthrough
Using ADLS access key directly
Creating Mount Point using ADLS Access Key
Cmd 2
Method 1: Using ADLS access key directly
we have give storage account name
Cmd 3
spark.conf.set(
"fs.azure.account.key.<storage account>.dfs.core.windows.net",
"<access key>"
)
Cmd 4
dbutils.fs.ls("abfss://<>@<>.dfs.core.windows.net/")
give access key
Cmd 5


Apps
Google
Exercise - Create Az...
AzureDataFactoryH...
Per
ADLS Integration with Databricks
Python
test
File
Edit
View: Standard
Permissions
Run All
Clear
Cmd 1
Methods of ADLS Integration with Databricks
Using a service principal
Using Azure Active Directory credentials known as a credential passthrough
Using ADLS access key directly
Creating Mount Point using ADLS Access Key
Cmd 2
Method 1: Using ADLS access key directly
Cmd 3
spark.conf.set(
"fs.azure.account.key.adlsrajade.dfs.core.windows.net",
"HJeoH1r2fXWpZvu4IUPSQqIkJJDS3pxLEmFZnJ4MRZOW5y4UrcekTqKAEOKT/YYDzAirUYdXPCpxe7Gcl4IbBA=="
)
Cancel
Uploading command
Cmd 4
dbutils.fs.ls("abfss://<>@<>.dfs.core.windows.net/")


ADLS Integration with Databricks
Python
test
File
Edit
View: Standard
Permissions
Run All
Clear
"HJeoH1r2fXWpZvu4IUPSQqIkJJDS3pxLEmFZnJ4MRZOW5y4UrcekTqKAEOKT/YYDzAirUYdXPCpxe7Gcl4IbBA=="
)
Command took 0.04 seconds -- by audaciousazure@gmail.com at 1/15/2022, 1:51:27 PM on test
Cmd 4
dbutils.fs.ls("abfss://container-rajade@adlsrajade.dfs.core.windows.net/")
Out[2]: [FileInfo(path='abfss://container-rajade@adlsrajade.dfs.core.windows.net/world_population_data.c
Command took 1.63 seconds -- by audaciousazure@gmail.com at 1/15/2022, 1:52:34 PM on test
Cmd 5
#set the data lake file location:
file_location = "abfss://container-rajade@adlsrajade.dfs.core.windows.net/"
#read in the data to dataframe df
df = spark.read.format("csv").option("inferSchema", "true").option("header",
"true").option("delimiter",",").load(file_location)
#display the dataframe
display(df)

Method – 4



In source give → “wasbs://<container-name>@<endpoint>”