

## 1. What is ADF?

- a) Azure Data Factory is **Azure's cloud ETL service** for scale-out server-less data integration and data transformation.
- b) It offers a **code-free UI** for intuitive authoring & single-pane-of-glass monitoring & management.
- c) You can also **lift and shift existing SQL Server Integration Service packages** to Azure and run them with full compatibility in Azure Data Factory.
- d) [https://www.techbrothersit.com/2021/10/azure-data-factory-tutorial-step-by-step\\_23.html](https://www.techbrothersit.com/2021/10/azure-data-factory-tutorial-step-by-step_23.html)

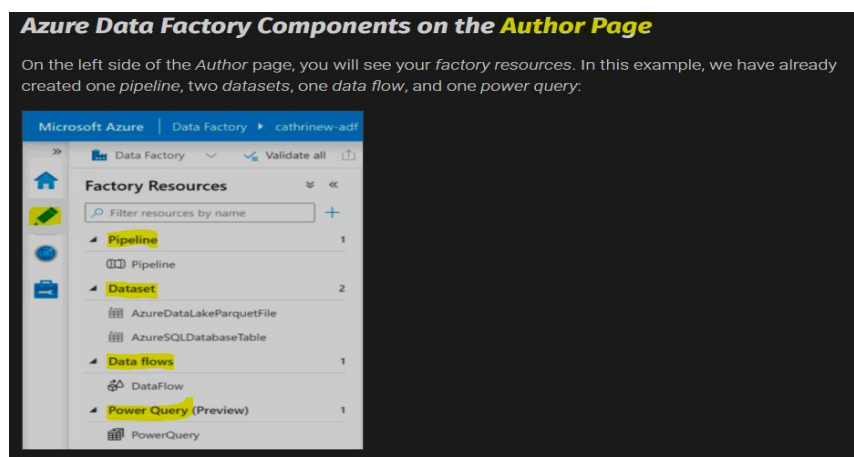
## 2. Components of ADF:

- a) <https://www.liktorius.com/wp-content/uploads/2020/01/Azure-Data-Factory-for-Beginners.pdf>



i.

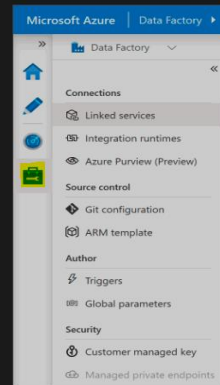
- b) <https://www.cathrinewilhelmsen.net/overview-azure-data-factory-components/>



i.

## Azure Data Factory Components on the Management Page

On the left side of the *Management* page, you will see components and services you can create and configure. We will focus on two of the core components in this post: *linked services* and *triggers*.



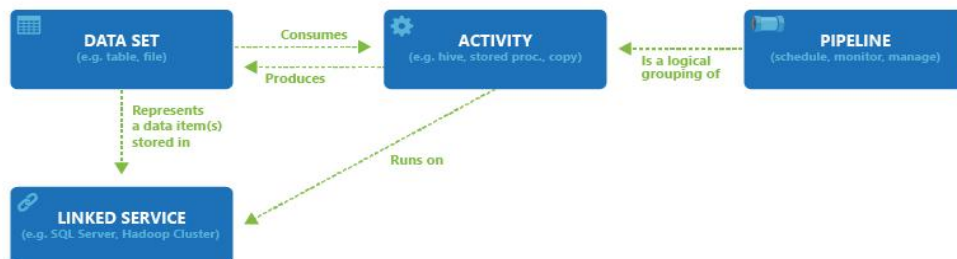
ii.

## Azure Data Factory Components

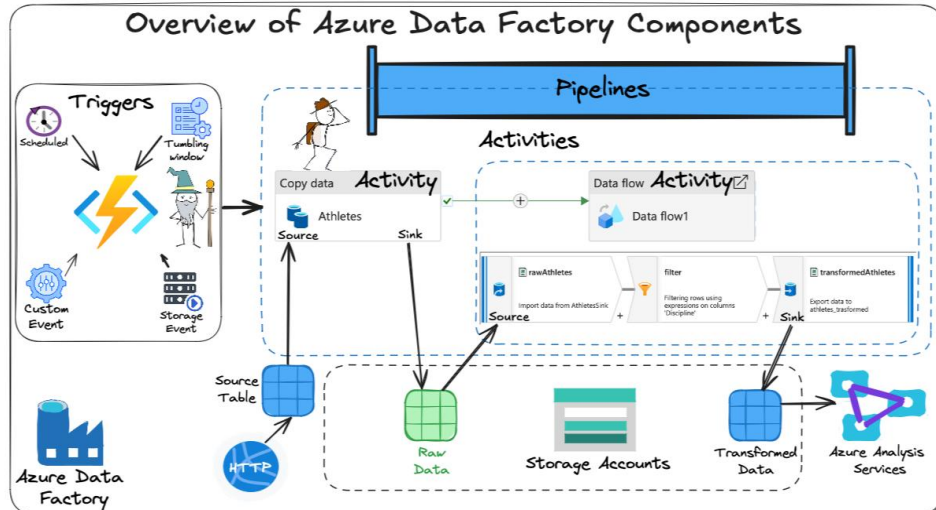
ADF has key components that work together to provide the platform on which you can compose data-driven workflows with steps to move and transform data:

- **Pipelines:** A data factory can have one or more pipelines, which are a logical grouping of activities performed by a work unit. This allows activities to be managed as a set instead of managing each one individually, and they can be chained to work sequentially or independently.
- **Data flow mapping:** allows you to create and manage transformation logic graphs that can be used to transform data of any size. You can also create a reusable library of transformation routines and run those processes in a scalable manner from the service pipelines in an automated fashion.
- **Activity:** represents a processing step in a pipeline and three types are supported: data movement activities, data transformation activities, and control activities.
- **Data sets:** represent data structures within data stores that point to or reference data to be used in activities.
- **Linked services:** define the connection information required for ADF to connect to external resources. They are often used to represent a data store or represent a computer course that can host the execution of an activity.
- **Triggers:** represent the processing unit that determines when to start the execution of a process.
- **Parameters:** are defined in the pipeline and are passed during the execution created by a trigger or a manually executed pipeline. Activities within the pipeline consume parameter values or a set of data also represent a parameter.
- **Control flow:** is an orchestration of pipeline activities that includes chaining activities in a sequence, branching, defining parameters at the pipeline level, and passing arguments while invoking the pipeline or from a trigger.
- **Variables:** can be used within pipelines to store temporary values or to be used together with other parameters to allow passing values between pipelines, data flows, and other activities.

c)



d)



e)

### 3. Notes on Azure Storage Account:

- An Azure storage account contains all of your Azure Storage data objects: blobs, files, queues, and tables.
- The storage account provides a unique namespace for your Azure Storage data that's accessible from anywhere in the world over HTTP or HTTPS.
- Data in your storage account is durable and highly available, secure, and massively scalable.
- Blob Storage vs Data Lake:
  - Azure Blob Storage** is one of the most common Azure storage types. It's an **object storage** service for workloads that need high-capacity storage.
  - Azure Data Lake** is intended primarily for **big data analytics** workloads.
  - Blob** -- which is shorthand for binary large object -- is ideal for large amounts of unstructured data, such as text, videos, photos, application back-end data and backup data. It's a general-purpose object store for unstructured data in a single hierarchy and a flat namespace.
    - Common uses for Azure Blob Storage include the following:
      - Storing files for distributed access, such as installation or upgrades.
      - Streaming video and audio.
      - Storing backups for DR and archiving.
      - Storing binary data, such as application back-end files & general-purpose data.
  - Azure Data Lake storage is currently separated into Gen1 and Gen2 options.** Microsoft will retire Data Lake Gen1 storage in February 2024, and all customers using it must migrate to Gen2 before this date.

1. Azure Data Lake Gen1 is a storage service that's optimized for big data analytics workloads. Its hierarchical file system can store machine learning data, including log files, as well as interactive streaming analytics. It is performance-tuned to run large-scale analytics systems that require massive throughput and bandwidth to query and analyze large amounts of data.
2. Azure Data Lake Gen2 converges the features and capabilities of Data Lake Gen1 with Blob Storage. It inherits the file system semantics, file-level security and scaling features of Gen1 and builds them on Blob Storage. This results in a low-cost, tiered-access, high-security and high availability big data storage option.

#### Azure Blob Storage

##### General Purpose Object Storage

- Global scale – All Azure regions
- Full BCDR capabilities
- Tiered - Hot/Cool/Archive
- Cost Efficient
- Large partner ecosystem

#### Azure Data Lake Store

##### Optimized for Big Data analytics

- Built for Hadoop
- Hierarchical namespace
- ACLs, AAD and RBAC
- Performance tuned for big data
- Very high scale capacity and throughput

#### Azure Data Lake Storage Gen2

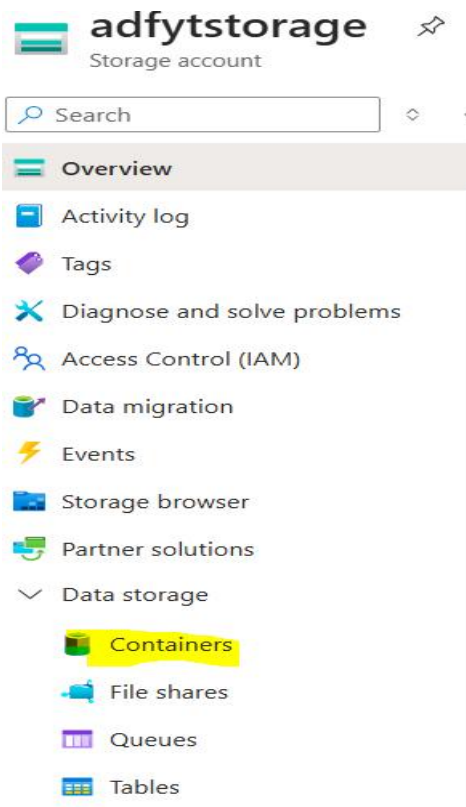
The best of Blobs and ADLS

v.

	Data Lake	Blob Storage	Data Lake Gen 2
Hot/Cold Storage Tiers	NO	YES	<u>YES</u>
Redundant Storage	NO	YES	<u>YES</u>
AD Security	YES	NO	<u>YES</u>
HDFS Compatible	YES	NO	<u>YES</u>

vi.

4. Create a ADLS gen2 storage
  - a) Enable Hierarchical namespace\*\*\* >> "Containers" are Data lakes.



b)

## 5. Create ADF

**Create Data Factory** ...

**Basics** | Git configuration | Networking | Advanced | Tags | Review + create

One-click to create data factory with sample pipeline and datasets. Try it

**Project details**

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

Subscription \* ⓘ Azure subscription 1 ▼

Resource group \* ⓘ RG-ADFCourse ▼  
[Create new](#)

**Instance details**

Name \* ⓘ adf-course-ansh ✓

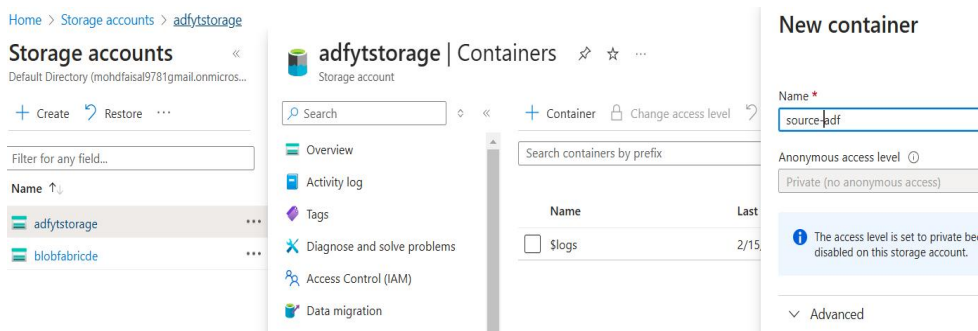
Region \* ⓘ East US ▼

Version \* ⓘ V2 ▼

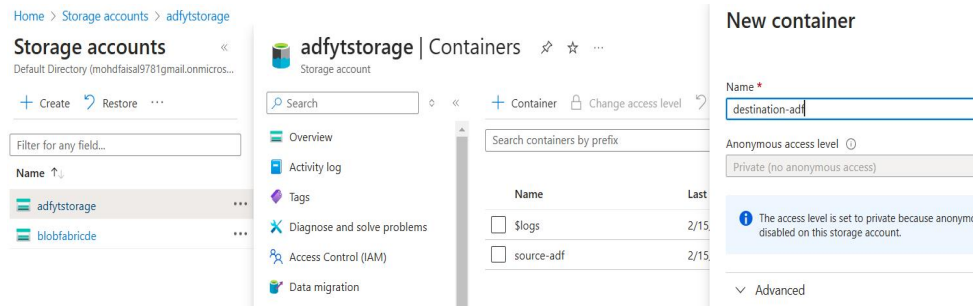
a)

## 6. Create new Container(s) in Data Lake storage account

a)

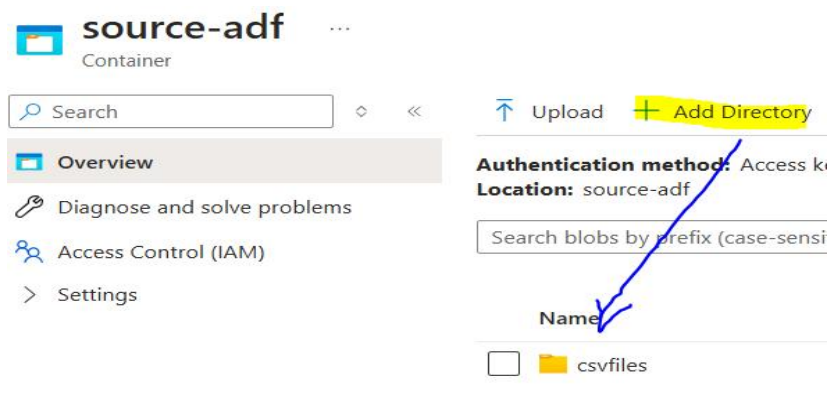


b)

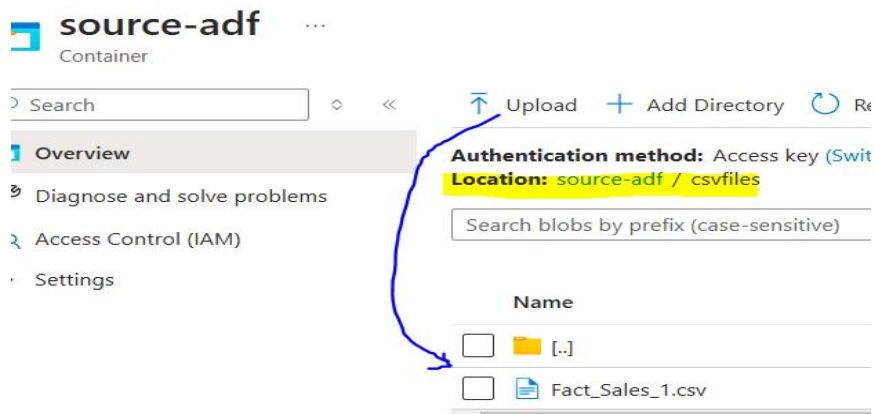


## 7. Upload source file 'fact\_sales\_1.csv' to 'csvfiles' directory of source container

a)

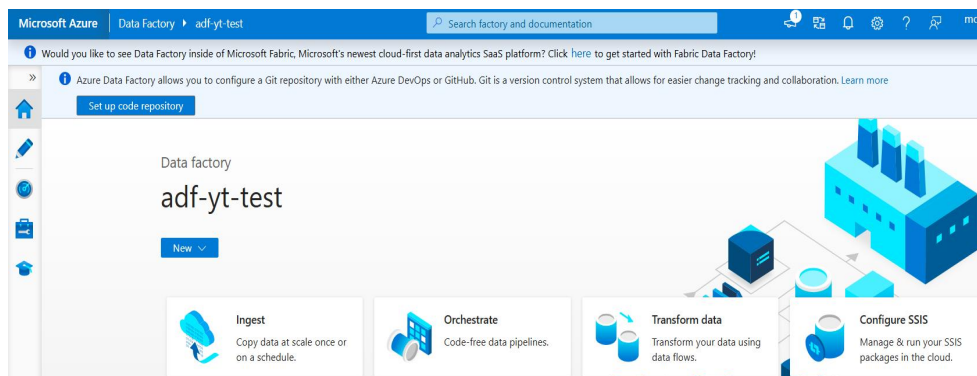




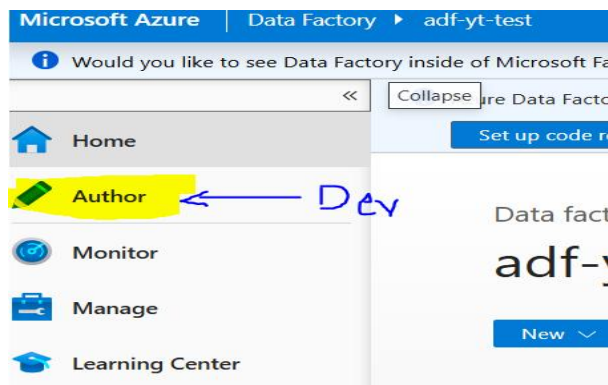


b)

## 8. ADF > Launch studio from resource we created in step-5



a)



b)

c) Create a connection to source

i. ADF > Manage > Linked Service > New >

### New linked service

Azure Data Lake Storage Gen2 [Learn more](#)

Name

LinkedService\_Source\_DL

Description

Connect via integration runtime \*

AutoResolveIntegrationRuntime

Authentication type

Account key

Account selection method

☒ From Azure subscription ☐ Enter manually

Azure subscription

Azure subscription - Pay as you go (2023-01-01)

Storage account name \*

adfytstorage

Test connection

☒ To linked service ☐ To file path

Annotations

New

Parameters

Connection successful

Test connection

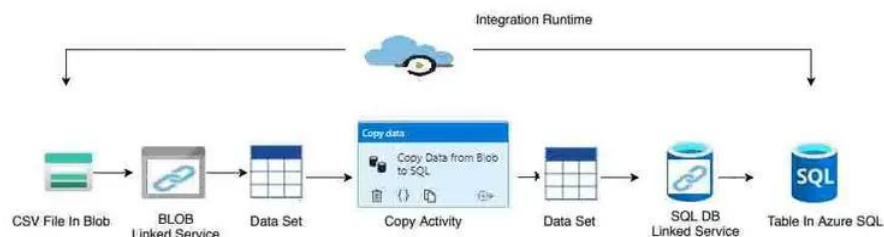
1.

Create

Back

## d) COPY ACTIVITY:

- i. Copy activity is basically used for ETL purpose or lift and shift where you want to move the data from one data source to the other data source. **While you copy the data you can also do the transformation.**
- ii. <https://azurelib.com/azure-data-factory-copy-activity/>

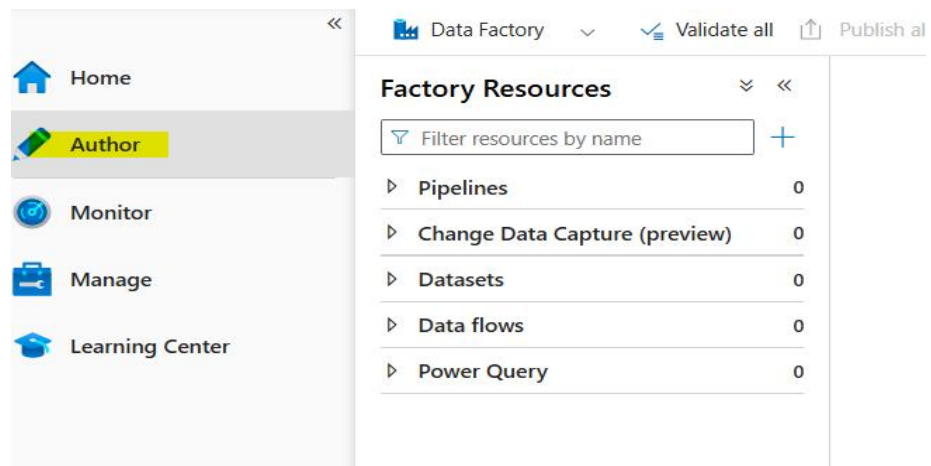


Use case: Move CSV data from Azure Blob to Azure SQL DB using Data Factory

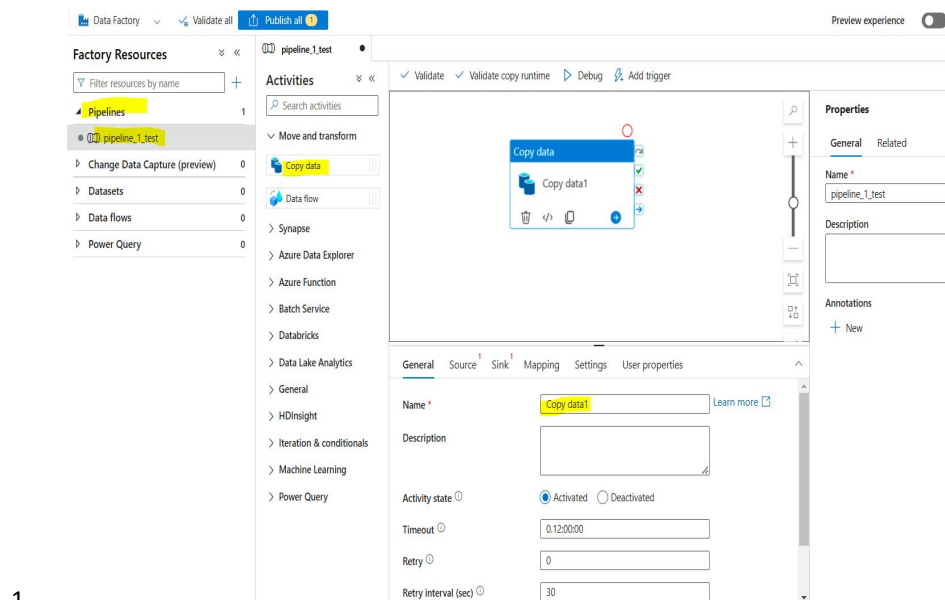
iii.



iv. Go to Author >

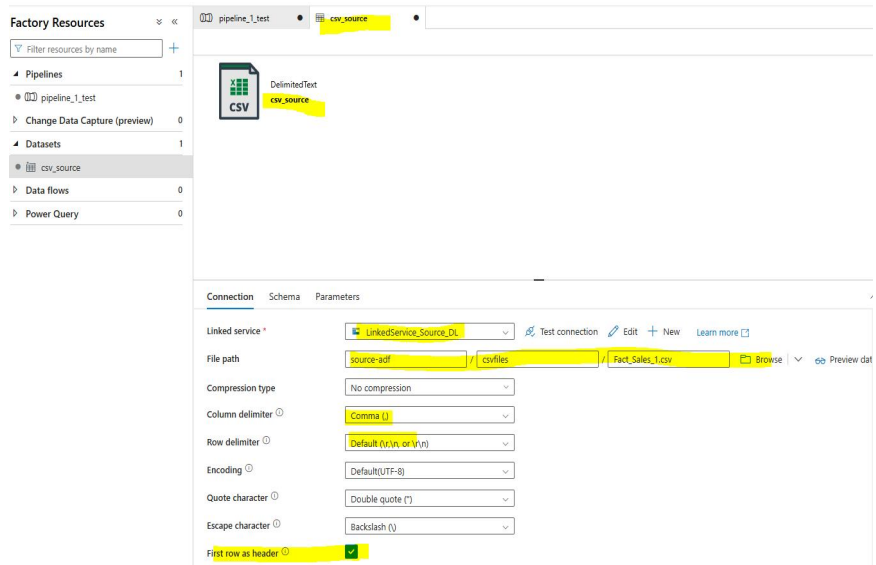


v. Create Pipeline > Add Copy Activity > Specify source > Sink >

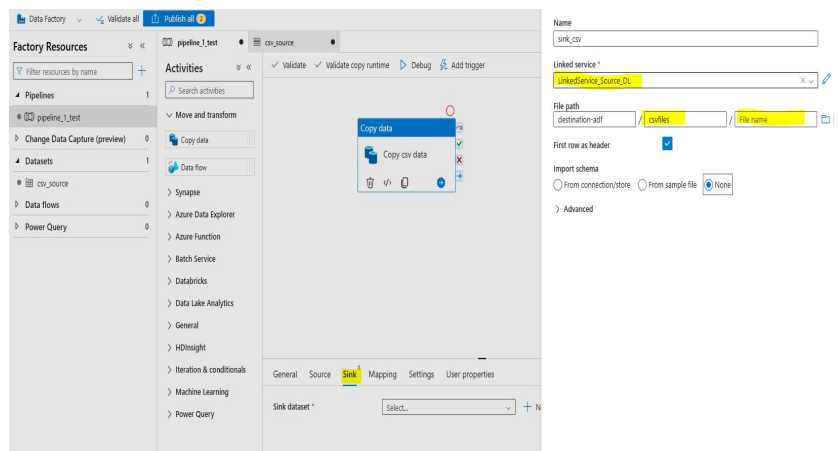


1.

2.

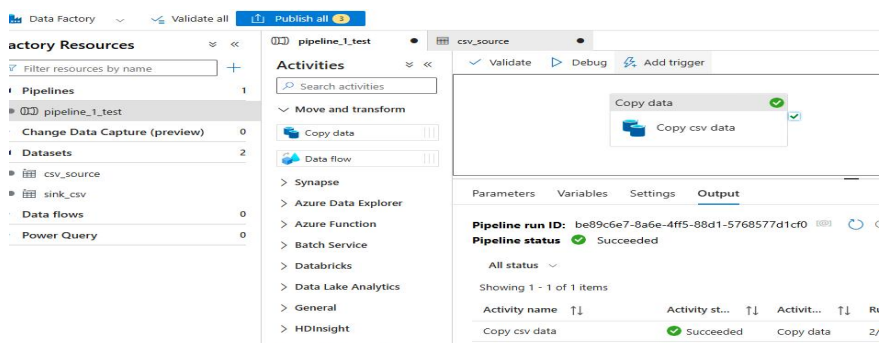


3.

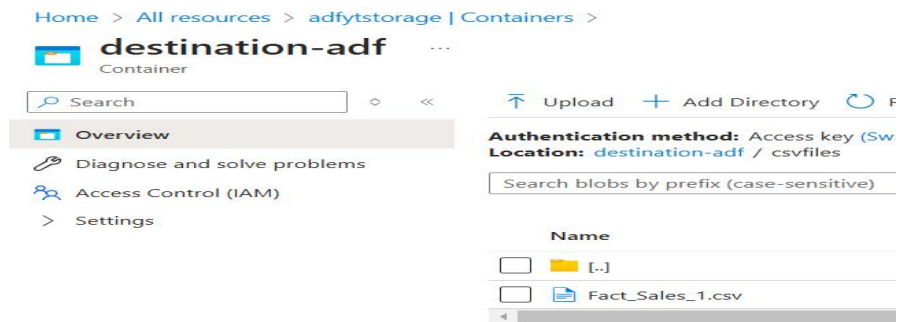


Note: Here we are using same Linked service as target is ADLS G2.. creating new directory in sink config... and file name will be set be ADF..

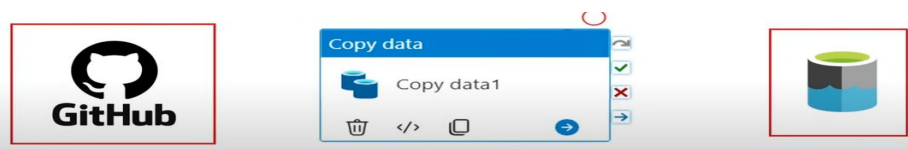
4. Debug the pipeline



5. Validate in destination container



e) COPY DATA using REST API:

- i. 
- ii. Create new pipeline
- iii. Create new Copy Activity

1. Source: HTTP > New Linked Service

Note: For Linked Service, base url is **parsed** "Raw" button url of file on git hub

**New linked service**  
HTTP [Learn more](#)

Name \*

Description

Connect via integration runtime \* ⓘ  
☒ AutoResolveIntegrationRuntime

Base URL \*  
  
⚠ Information will be sent to the URL specified. Please ensure you trust the URL entered.

Server certificate validation ⓘ  
☒ Enable ☐ Disable

Authentication type \* ⓘ

Auth headers ⓘ  
[+ New](#)

Annotations  
[+ New](#)

> Parameters  
> Advanced ⓘ

Connection successful

Then in the source properties, it'll ask the **Relative URL**, get same from **parsed**

“Raw” button url of file on git hub

### Set properties

Name  
csv\_source\_git

Linked service \*  
http\_linked\_service\_git

Relative URL  
anshlambagit/Azure-Data-Factory/refs/heads/main/Raw%20Data/Fact\_Sales\_2.csv

## 2. Sink >

### Set properties

Name  
sink\_git\_csv

Linked service \*  
LinkService\_Source\_DL

File path  
destination-adf / csvfiles / File name

First row as header ☒

Import schema  
☐ From connection/store ☐ From sample file ☒ None

> Advanced


## iv. Debug & Validate

pipeline\_git\_to\_dat... •

✓ Validate ▶ Debug ⚡ Add trigger

Copy data ✓  
Copy data git to DL ✓

Parameters Variables Settings **Output**

**Pipeline run ID:** 7db08203-db8d-4b14-a46b-7a1b4c463495 ⓘ  ⓘ

**Pipeline status** ✓ Succeeded

All status ▾ [Monitor in](#)

Showing 1 - 1 of 1 items

Activity name	Activity st...	Activit...	Run start
Copy data git to DL	✓ Succeeded	Copy data	2/15/2025, 3:41:39 PM

1.

2. **Note:** As we didn't specify the file name in Sink properties, ADF has taken full path from git-hub

[All resources](#) > [adfytstorage | Containers](#) >

## destination-adf

Container

Upload Add Directory Refresh Rename Delete Change tier Acquire lease

Overview

Diagnose and solve problems

Access Control (IAM)

Settings

**Authentication method:** Access key ([Switch to Microsoft Entra user account](#))  
**Location:** destination-adf / csvfiles / anshlambagit / Azure-Data-Factory / refs / heads / main / Raw Data

Search blobs by prefix (case-sensitive)

	Name	Modified	Access tier	Ar
<input type="checkbox"/>	Fact_Sales_2.csv	2/15/2025, 3:41:50 PM	Hot (Inferred)	

3. To fix this File path, edit sink properties and re-run & validate



DelimitedText  
sink\_git\_csv

Connection Schema Parameters

Linked service \* LinkedService\_Source\_DL Test connection Edit + New Learn more

File path destination-adf / csvfiles / Fact\_sales\_2.csv Add dynamic content [Alt+Shift+D]

[Home](#) > [All resources](#) > [adfytstorage | Containers](#) >

destination-adf

Container

Search

Upload Add Directory Refresh Rename

Overview

Diagnose and solve problems

Access Control (IAM)

Settings

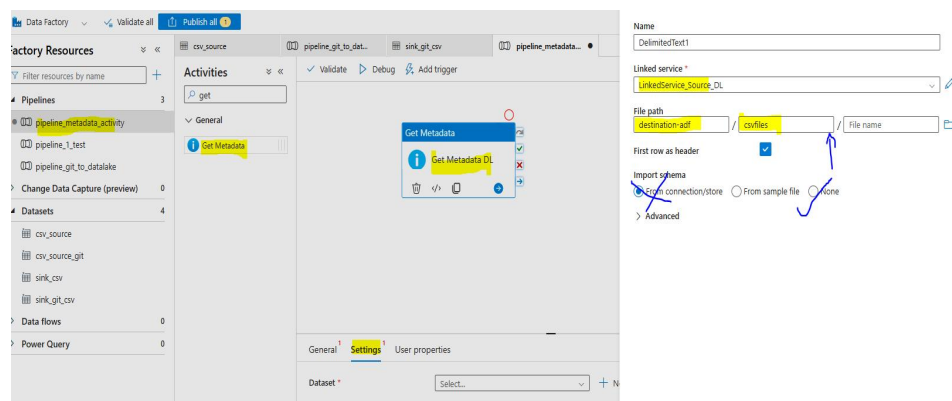
**Authentication method:** Access key ([Switch to Microsoft Entra user account](#))  
**Location:** destination-adf / csvfiles

Search blobs by prefix (case-sensitive)

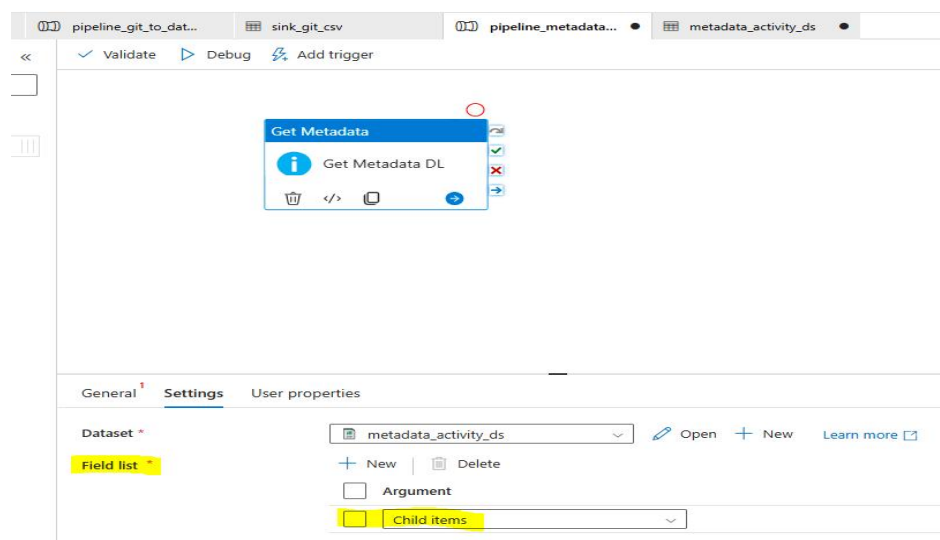
	Name
<input type="checkbox"/>	Fact_Sales_1.csv
<input type="checkbox"/>	Fact_sales_2.csv

## f) Get Metadata Activity:

- i. Suppose we have a data governance policy where only 'Fact\_sales\_1.csv' (out of multiple files in source folder) has to be ingested / copied to destination.
- ii. While working in Azure Data Factory, **sometimes we need to retrieve metadata information, like the file name, file size, file existence, etc.** We can use the Get Metadata activity to retrieve metadata information from the data set and then we can use that metadata information in subsequent activities. Refer <https://www.sqlservercentral.com/articles/working-with-get-metadata-activity-in-azure-data-factory> for more details.
- iii. <https://www.mssqltips.com/sqlservertip/6246/azure-data-factory-get-metadata-example/>
- iv. New Pipeline > Add 'Get Metadata' activity > Set properties > Debug >> This will give array of files in folder..



Note: We need metadata of folder.. so file name is not given..



Below is snip of successful run > see output >



Publish all

pipeline\_metadata... metadata\_activity\_ds

Activities

get

General

Get Metadata

Get Metadata DL

Parameters Variables Settings Output

Pipeline run ID: cd20d7e0-73b7-48c7-b44f-76441b8ae861

Pipeline status: Succeeded

All status

Showing 1 - 1 of 1 items

Activity name	Activity st...	Activit...	Run start
Get Metadata DL	Succeeded	Get Metadata	2/15/2025, 6:39:48 PM

Output

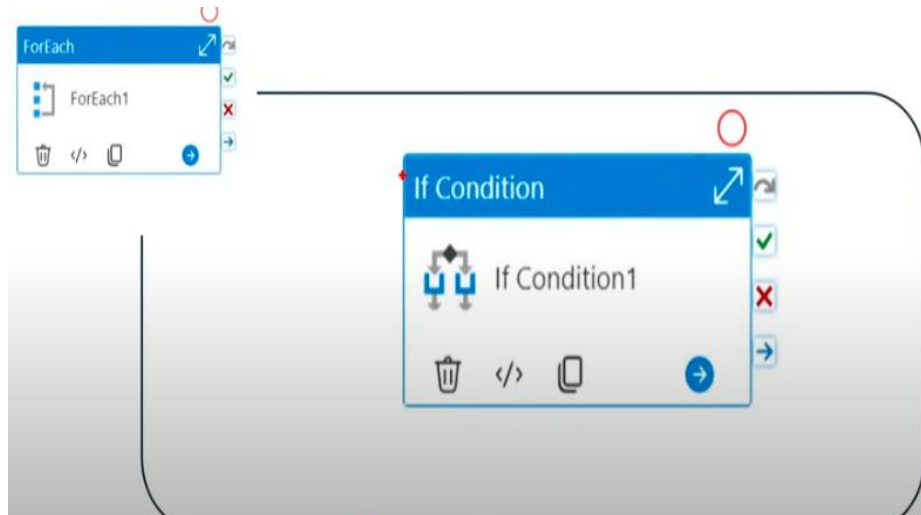
This'll be the output >

### Output

Copy to clipboard

```
{
  "childItems": [
    {
      "name": "Fact_Sales_1.csv",
      "type": "File"
    },
    {
      "name": "Fact_sales_2.csv",
      "type": "File"
    }
  ],
  "effectiveIntegrationRuntime": "AutoResolveIntegrationRuntime (Canada Central)",
  "executionDuration": 2,
  "durationInQueue": {
    "integrationRuntimeQueue": 0
  },
  "billingReference": {
    "activityType": "PipelineActivity",
    "billableDuration": [
      {
        "meterType": "AzureIR",
        "duration": 0.016666666666666666,
        "unit": "Hours"
      }
    ]
  }
}
```

- v. ForEach file (using **ForEach activity** as we have array of filenames above) > Now give the filename to **IF activity** >> i.e if the file\_name matches > load to target



ud-first data analytics SaaS platform? Click [here](#) to get started with Fabric Data Factory!

Publish all

pipeline\_metadata... metadata\_activity\_ds

Activities

for

Move and transform

Copy data

Data flow

Iteration & conditionals

ForEach

Get Metadata

Get Metadata DL

ForEach

ForEach\_met activity

Activities

No activities

General Settings Activities (0) User properties

Sequential ☒

Items \*

This property should be parameterized.

Add dynamic content [Alt+Shift+D]

Pipeline expression builder

Add dynamic content below using any combination of expressions, fu

@activity('Get Metadata DL').output.childItems

Clear contents

Activity outputs Parameters System variables Fun

Search

Get Metadata DL

Get Metadata DL activity output

Get Metadata DL childItems

List of subfolders and files in the given folder

Get Metadata DL exists

Whether a file, folder, or tabl

List of subfolders and files in the

Get Metadata DL itemName

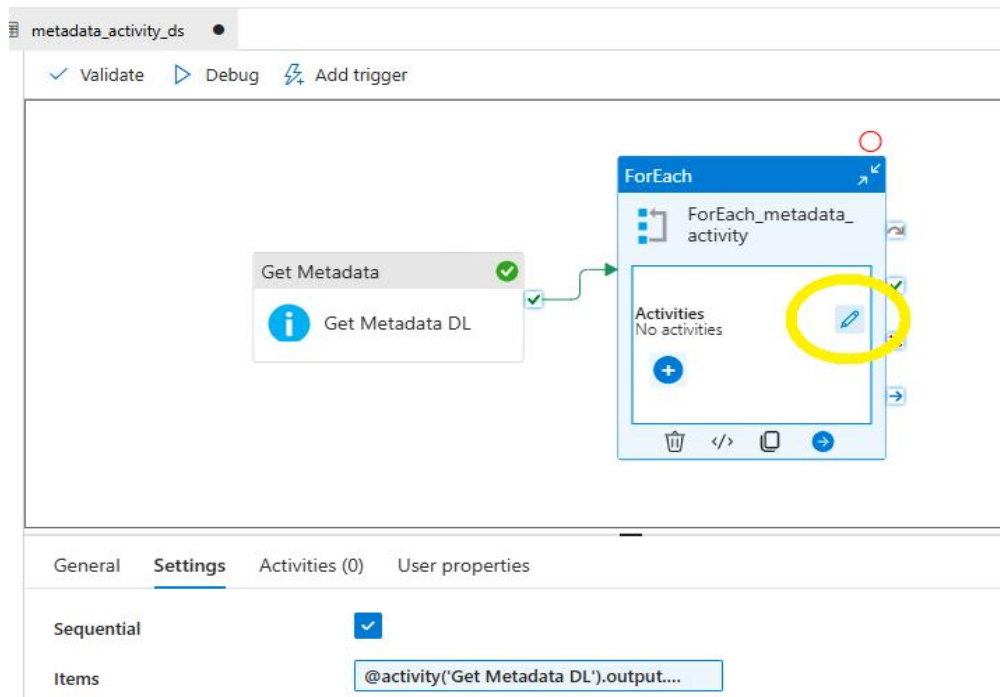
Name of the file or folder

Get Metadata DL itemType

Type of the file or folder. Returned value is File or Folder

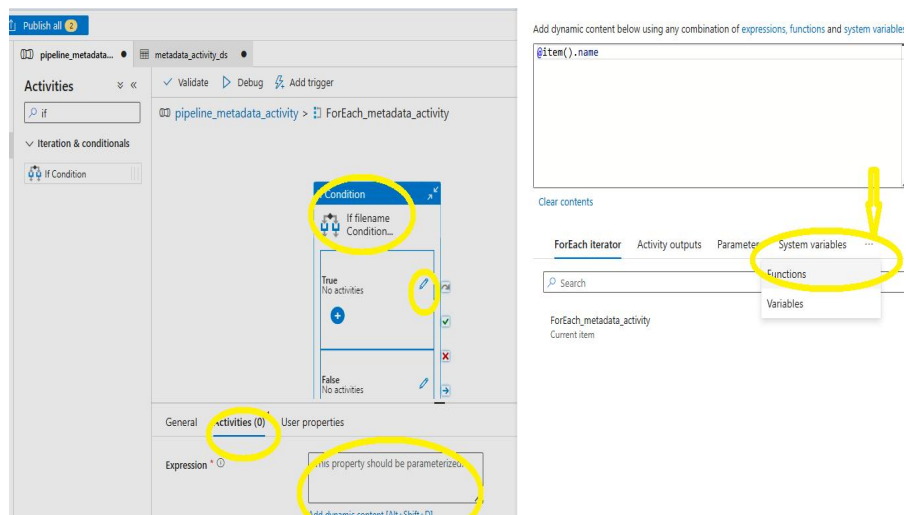
Get Metadata DL lastModified

Then go Inside the FOREACH >>



Add IF Activity >>

Syntax we refer as per output above



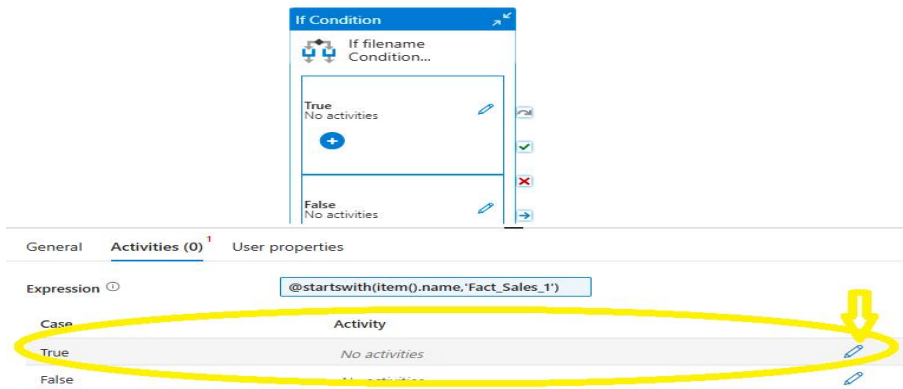
>

## Pipeline expression builder

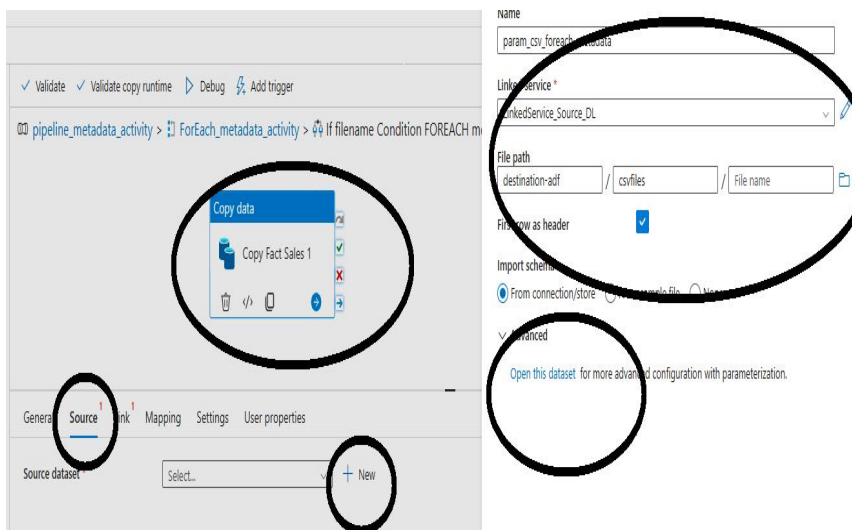
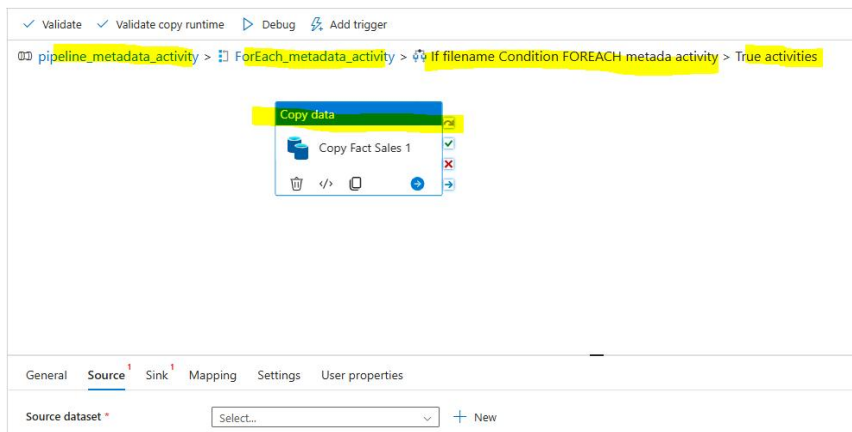
Add dynamic content below using any combination of [expressions](#), [functions](#)

```
@startswith(item().name, 'Fact_Sales_1')
```

> If above expression is TRUE > Copy output file > Edit TRUE condition >



>> Edit TRUE condition > Add Copy Activity >> Here we'll add PARAMs as File name >>



>> On selecting “Open the dataset” from Advance > we get >>

>> Create parameter >

Name	Type	Default value
p_file_name	String	Value

> Use the parameter >



DelimitedText  
param\_csv\_foreach\_metadata

Connection Schema Parameters

Linked service \* LinkedService\_Source\_DL Test connection Edit New Learn more

File path destination-adf / csvfiles / File name Browse

Compression type No compression

Column delimiter Comma (,)

>

pipeline\_metadata... param\_csv\_foreach...

DelimitedText  
param\_csv\_foreach\_metadata

Connection Schema Parameters

Linked service \* LinkedService\_Source\_DL Test connection Edit New Learn more

File path destination-adf / csvfiles / File name Browse

Clear contents

Parameters Functions

@dataset().p\_file\_name

p\_file\_name

>> Now Go to Copy Activity >

pipeline\_metadata... param\_csv\_foreach...

Activities Validate Validate copy runtime Debug Add trigger

copy

Move and transform

Copy data

Copy data

Copy Fact Sales 1

General Source Sink Mapping Settings User properties

Source dataset \* param\_csv\_foreach\_metadata Open New Preview data Learn more

Dataset properties

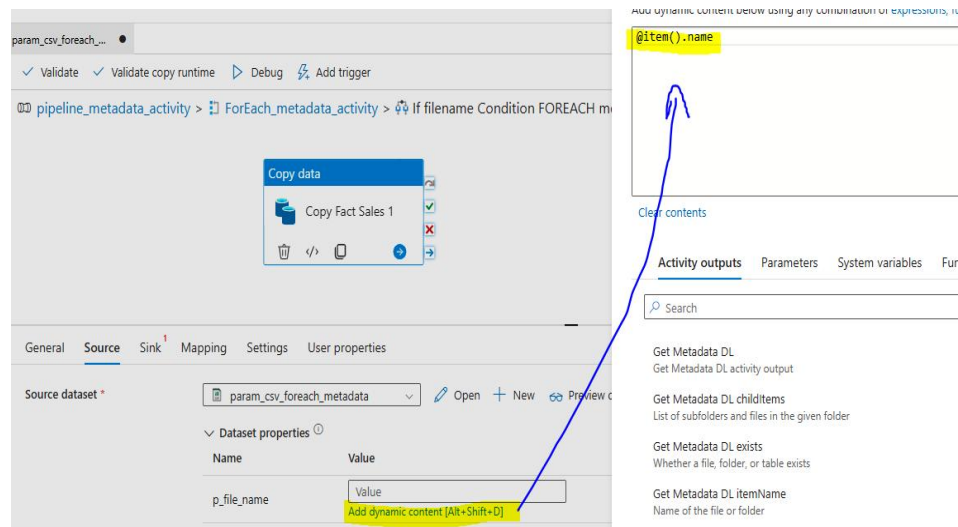
Name	Value
p_file_name	Value

File path type File path in dataset Wildcard file path List of files

>> Here we pass the value to Parameter >>

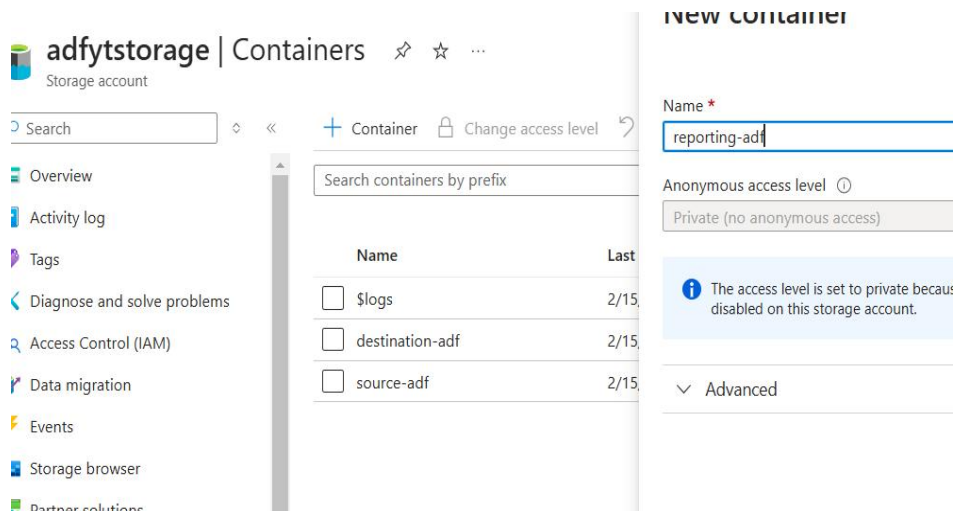


This Filename is coming from ForEach , i.e. >> Here it'll only COPY the file whose condition is satisfied in each iteration of ForEach >>

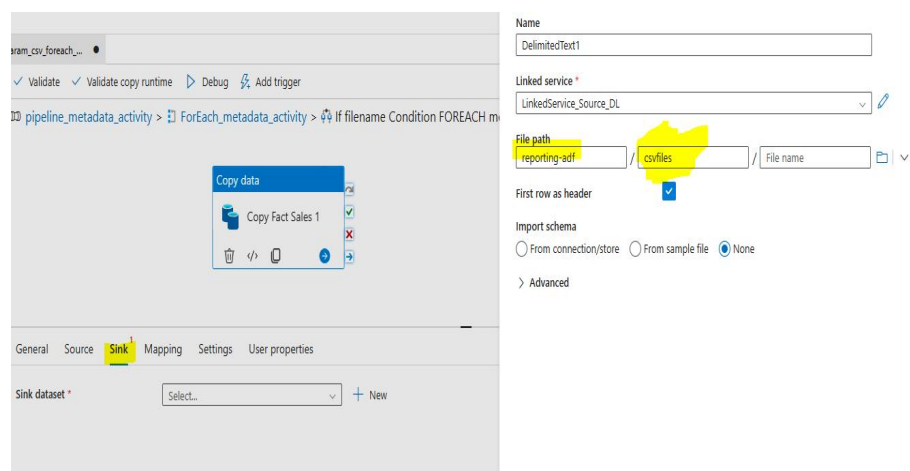


>>

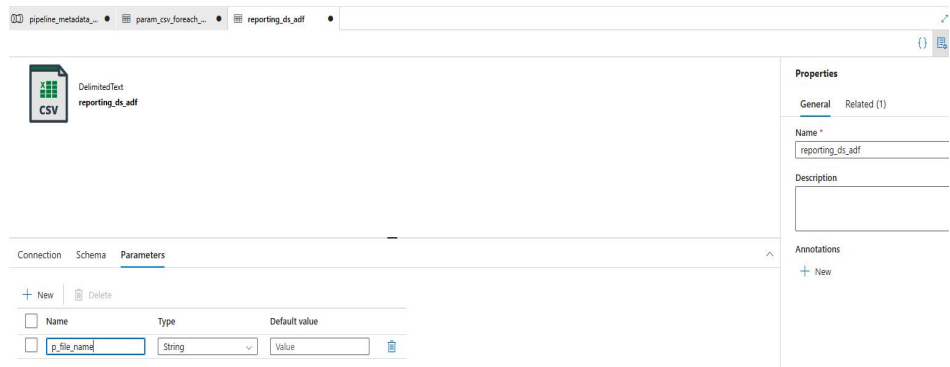
Create New container 'reporting-adf' > This will be used as sink >



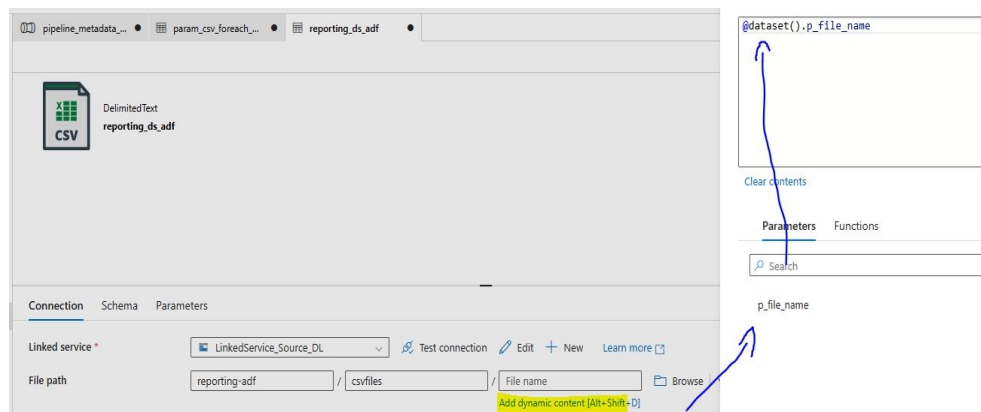
>>



>> Add PARAM to this sink dataset >

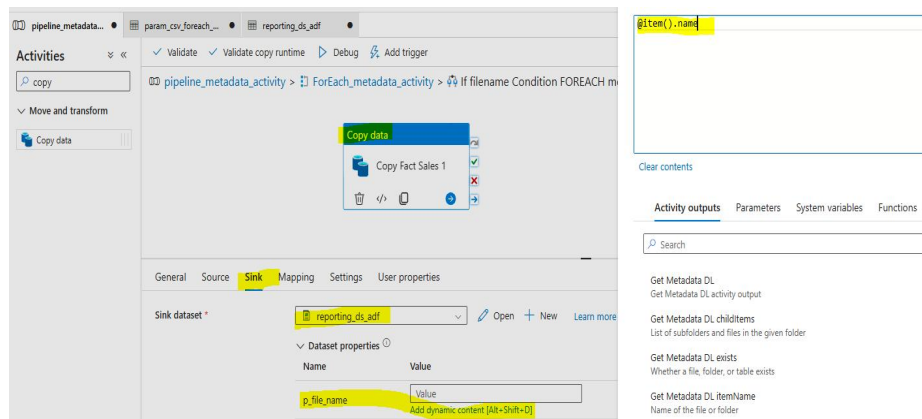


>>



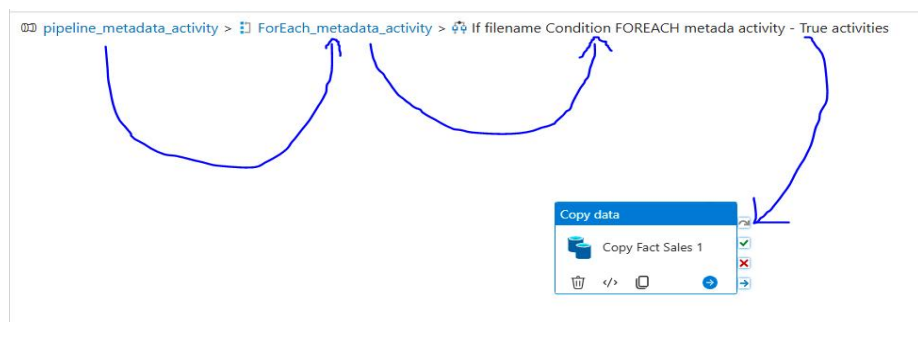
>>

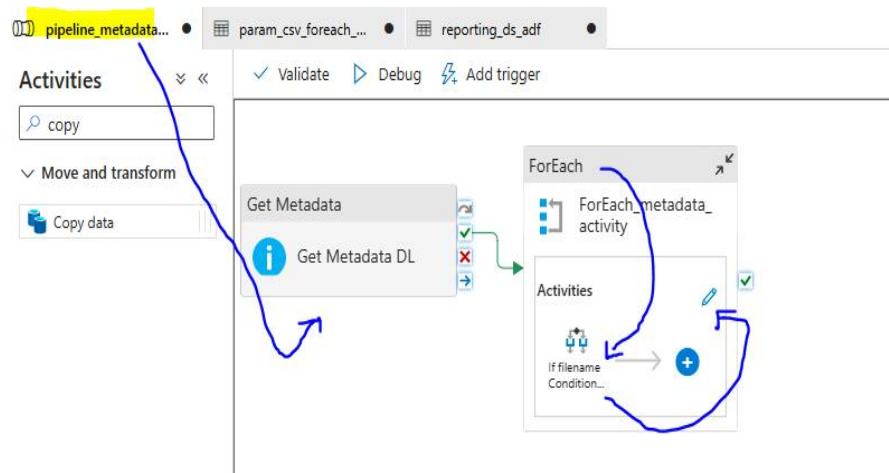
Use that PARAM in COPY sink properties >



>>

NOW THIS IS THE END to END view of this exercise >>





>>> DEBUG & VALIDATE >> ONLY FACT\_SALES\_1 file is copied!!!

pipeline\_metadata... x

Activities

copy

Move and transform

Copy data

Get Metadata

Get Metadata DL

ForEach

ForEach\_metadata\_activity

Activities

If filename Condition...

Parameters Variables Settings Output

Pipeline run ID: 72393c51-5b93-40b5-bc30-3b6bce18a5da

Pipeline status: In progress

All status List

Showing 1 - 4 of 4 items

Activity name	Activity status	Activity...	Run start	Duration	Integration runtime	User prop...	Activity run ID
Copy Fact Sales 1	Succeeded	Copy data	2/16/2025, 2:49:31 PM	16s	AutoResolveIntegrationRuntime (Canada Central)		f52886e5-c358-4398-a925-28a11969
If filename Condition FOREAC...	In progress	If Condition	2/16/2025, 2:49:31 PM	18s			b0e38f8a-4eab-4c06-aa25-0e9330ca
ForEach_metadata_activity	In progress	ForEach	2/16/2025, 2:49:30 PM	18s			a11ccbe8-ef26-4f32-8d29-797458a2
Get Metadata DL	Succeeded	Get Metadata	2/16/2025, 2:49:22 PM	7s	AutoResolveIntegrationRuntime (Canada Central)		f2bbe31a-b331-4e8c-917a-29fab637

Home > Storage accounts > adfytstorage | Containers >

reporting-adf

Container

Search

Upload Add Directory Refresh Rename Delete Change tier Acquire

Overview

Diagnose and solve problems

Access Control (IAM)

Settings

Authentication method: Access key (Switch to Microsoft Entra user account)

Location: reporting-adf / csvfiles

Search blobs by prefix (case-sensitive)

Name	Modified	Access tier
[...]		
Fact_Sales_1.csv	2/16/2025, 2:49:45 PM	Hot (Inferred)

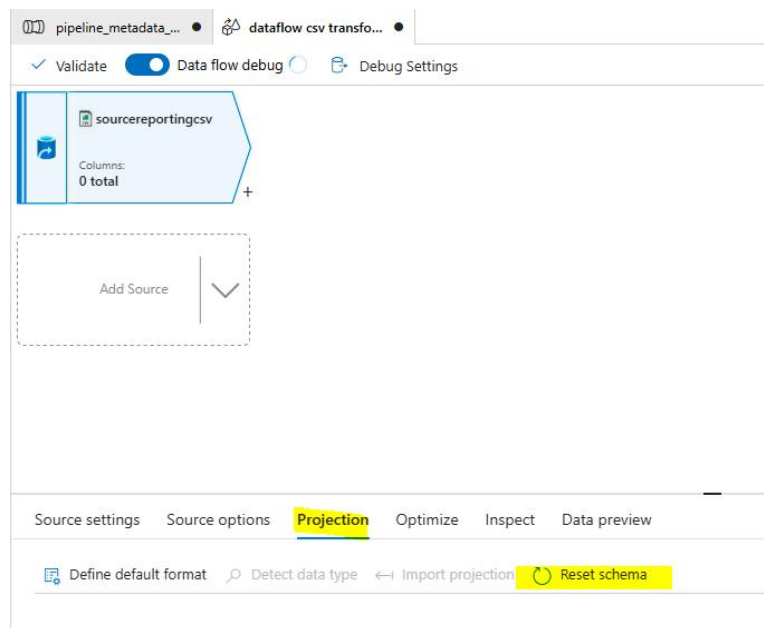
## g) DATAFLOWS in ADF:

- Once we have the data in reporting container, what if we want to **TRANSFORM** the data??
- Then we will use **DataFlows** (Refer [tutorial](#) on git-hub)
- <https://www.sqlshack.com/data-flow-transformations-in-azure-data-factory/>
- [Alter Row Transformation](#) is used for [UPSERTs](#) in data flow.



- v.
- vi. **DATAFLOW uses SPARK** for transformations.
- vii. Now add dataflow > add source > turn on Data Flow Debug (spark comoute and Time to Live) >

>>>  
 >>> Now IMPORT PROJECTION (i.e. Schema) >



>>> Add some transformations (SELECT & FILTER for example) >>

	transaction_id	transactional_date	product_id	customer_id	payment	cost	quantity	price
1	2021-05-04 02:00:00.000	P0494	4	visa	17.33	2	18.29	
2	2021-05-04 03:04:00.000	P0221	5	visa	0.59	1	1.49	
3	2021-05-04 03:56:00.000	P0625	5	visa	5.15	3	5.89	
4	2021-05-04 05:20:00.000	P0431	8	mastercard	10.67	2	11.59	
5	2021-05-04 05:45:00.000	P0058	5	mastercard	11.38	2	12.39	
6	2021-05-04 06:58:00.000	P0385	6	americanexpress	13.22	1	14.69	
7	2021-05-04 07:03:00.000	P0575	4	visa	2.81	1	3.99	
8	2021-05-04 07:45:00.000	P0187	5	americanexpress	4.17	1	4.89	

>> Add sink to data flow >>

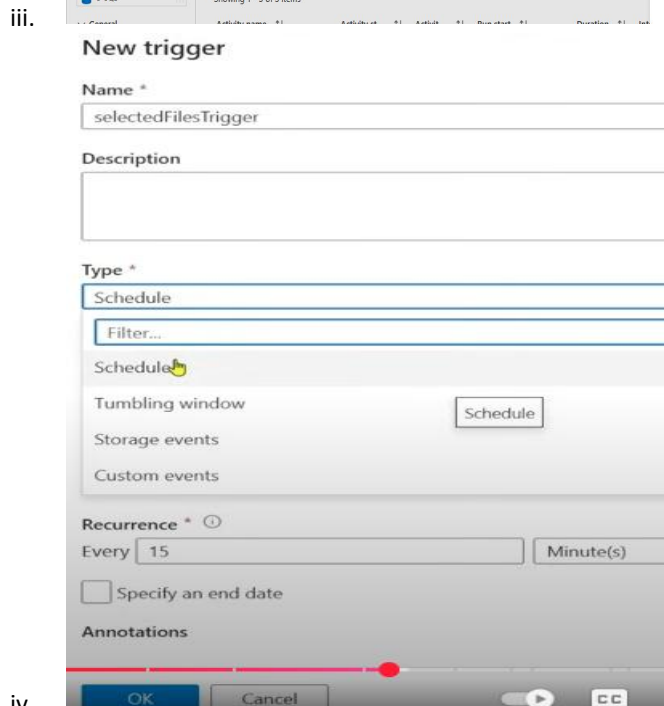
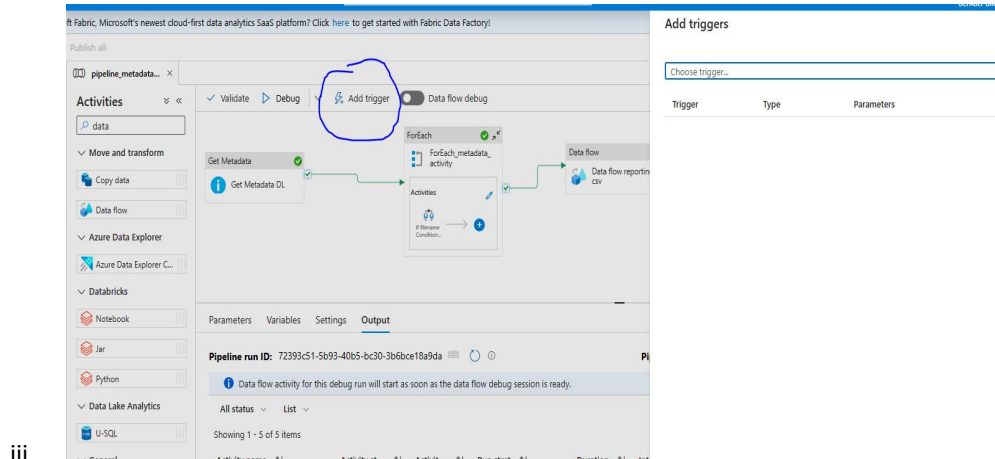
**Sink configuration details:**

- Name: DelimitedText2
- Linked service: LinkedService\_Source\_DL
- File path: reporting-adj / dataflow-results
- File name: [empty]
- First row as header: ☒
- Import schema: ☐ From connection/store ☐ From sample file ☒ None

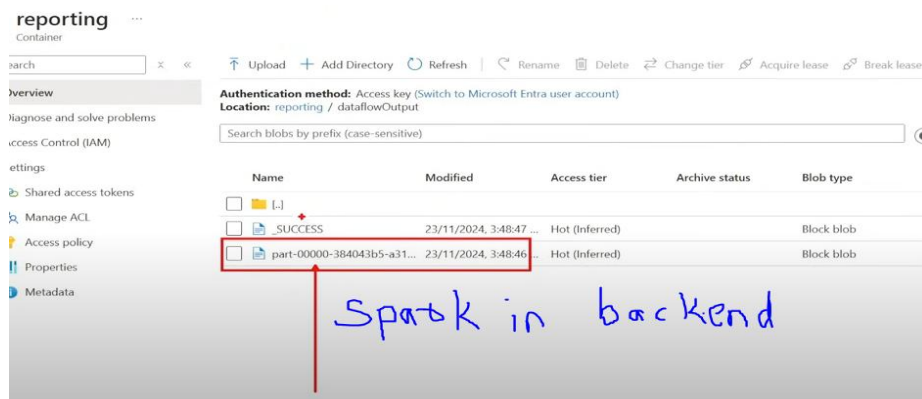
Note: The format of output file is shown in Triggers below.

## h) Schedule Trigger in ADF:

- i. <https://aspinfo.medium.com/what-are-the-triggers-in-adf-14ec208fdbdc>
- ii. <https://medium.com/@hasninemirza/azure-data-factory-triggers-cdb193142c9e>



- v. When Trigger is executed > Go to Monitor > Also validate >





## i) Set Variable Activity:

- i. "Set variable" Activity is used to set the value of an existing variable of type String, Bool, or, Array defined in a Data Factory Pipeline.
- ii. <https://medium.com/codex/introduction-to-set-variable-activity-73bcddcf0318>
- iii. **Set Variable vs Pipeline Variable:** <https://medium.com/@rganesh0203/set-variable-vs-pipeline-variable-in-adf-21bf2eaec44b>
- iv. **Set Pipeline Return Value:**
  1. pass values between two ADF pipelines
  2. <https://www.techbrothersit.com/2023/05/set-pipeline-return-value-in-azure-data.html>

## j) Storage Events Trigger:

- i. Similar to event driven in AWS S3
- ii. <https://pragmaticworks.com/blog/azure-data-factory-event-triggers>
- iii. <https://www.skynorthsoftware.com/blog/posts/copy-blob-event-trigger/>

## k) Execute Pipeline Activity:

- i. to call a pipeline from another pipeline or to pass parameter from one pipeline to another pipeline.
- ii. <https://azurelib.com/execute-pipeline-activity-in-adf/>

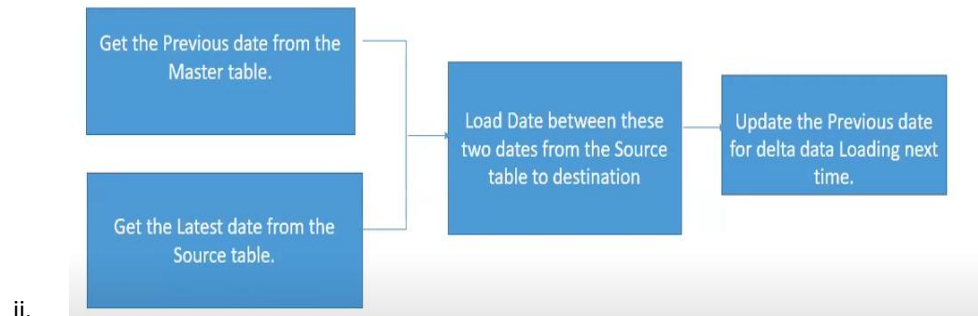
**What could be the business use cases or scenarios where you need to call a pipeline from another pipeline**

- You wanted to chain two different pipeline flow together. For example one pipeline is pulling the master data and other pipeline is pulling the entity data and you want to run both of them in sequence.
  - You wanted to split the existing pipeline into two or more pipeline, because keeping all the logic in one pipeline making it very big to handle and maintain.
  - There could be scenario where you wanted to do the **nesting of foreach activity**, but **unfortunately it isn't allowed in the Azure data factory** to have nested **foreach activity**. Hence to solve this problem what you can do is you split the pipeline into two. Both the pipeline keeps one **foreach activity**. Now call the second pipeline from the first pipeline using the execute pipeline activity.
- iii.

## l) Real time- Incremental Load:

- i. <https://www.youtube.com/watch?v=z5frQ3RyFmY>

## Incremental Load Flow



## Pipeline Flow

Create two Lookup activities.

- Use the first Lookup activity to retrieve the Previous date.
- Use the second Lookup activity to retrieve the Latest Date. These Date values are passed to the Copy activity in the Flow.

iii.

>

Create a Copy activity that copies rows from the source data store with the value of the Lastmodify column greater than the Previous date value and less than the Latest Date value. Then, it copies the delta data from the source data store to Blob storage as a new file.

>

Create a StoredProcedure activity that updates the Previous date value for the pipeline that runs next time.

>