

Azure Data Factory

Azure Data Factory (ADF) Detailed Notes

◆ What is ADF?

Azure Data Factory (ADF) is a cloud-based ETL (Extract, Transform, Load) and data integration service that allows you to:

- Orchestrate and automate data movement and transformation.
- Build complex pipelines with data from multiple sources (e.g., REST APIs, databases, file systems).
- Perform data ingestion from on-premise or cloud sources and push to storage or data warehouses.

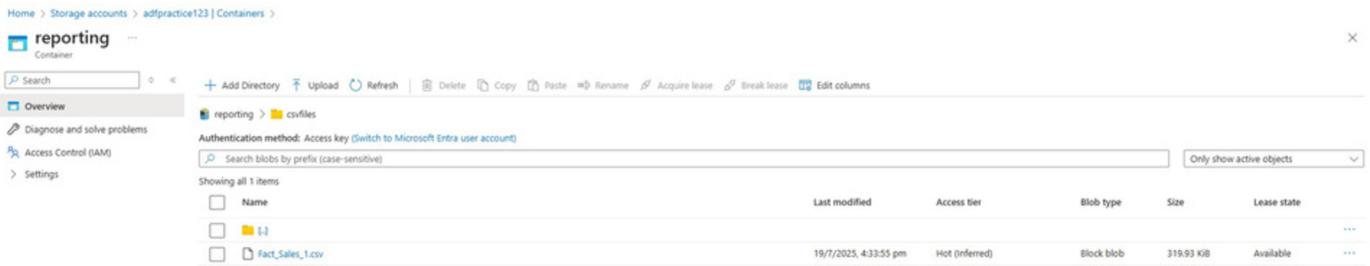
◆ Azure Resources Setup

1. Create a Resource Group

A logical container that holds related Azure resources.

2. Create a Storage Account

- Choose **Azure Data Lake Storage Gen2 (ADLS Gen2)**.
- Enable **Hierarchical Namespace** to support folder structures.



The screenshot shows the Azure Storage Explorer interface. On the left, there's a navigation tree with 'Home > Storage accounts > adfpractice123 | Containers > reporting'. The main area shows a table of blobs in the 'reporting' container. The table has columns: Name, Last modified, Access tier, Blob type, Size, and Lease state. One blob is listed: 'Fact_Sales_1.csv' (Last modified: 19/7/2025, 4:33:55 pm, Access tier: Hot (inferred), Blob type: Block blob, Size: 319.93 KB, Lease state: Available). There are also buttons for 'Add Directory', 'Upload', 'Refresh', 'Delete', 'Copy', 'Paste', 'Rename', 'Acquire lease', 'Break lease', and 'Edit columns'.

- Create containers/folders inside the storage (e.g., **source**, **reporting**).

Home > Storage accounts > adfpractice123 | Containers >

reporting Container

» + Add Directory ⚡ Upload ⏪ Refresh | 🗑 Delete 🕔 Copy 🕔 Paste 🗃 Rename ⚙️ Acquire lease ⚙️ Break lease

reporting

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

🔍 Search blobs by prefix (case-sensitive)

Showing all 0 items

<input type="checkbox"/>	Name	Last modified	Access tier	Blob type	Size	Lease state	...
No items found							

Home > Storage accounts > adfpractice123 | Containers >

destination Container

+ Add Directory ⚡ Upload ⏪ Refresh | 🗑 Delete 🕔 Copy 🕔 Paste 🗃 Rename ⚙️ Acquire lease ⚙️ Break lease Edit columns

destination > csv_files

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

🔍 Search blobs by prefix (case-sensitive)

Showing all 1 items

<input type="checkbox"/>	Name	Last modified	Access tier	Blob type	Size	Lease state	...
<input type="checkbox"/>	Fact_Sales_1.csv	19/7/2025, 3:40:47 pm	Hot (inferred)	Block blob	319.93 kB	Available	...

Home > Storage accounts > adfpractice123 | Containers >

destination Container

+ Add Directory ⚡ Upload ⏪ Refresh | 🗑 Delete 🕔 Copy 🕔 Paste 🗃 Rename ⚙️ Acquire lease ⚙️ Break lease Edit columns

destination > csv_files > git > Azure-Data-Factory > refs > heads > main > Raw Data

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

🔍 Search blobs by prefix (case-sensitive)

Showing all 1 items

<input type="checkbox"/>	Name	Last modified	Access tier	Blob type	Size	Lease state	...
<input type="checkbox"/>	Fact_Sales_2.csv	19/7/2025, 4:00:07 pm	Hot (inferred)	Block blob	2.8 kB	Available	...

The screenshot shows the 'sinkgit_csv' configuration page in Azure Data Factory. At the top, there's a preview section showing a CSV file icon and the text 'DelimitedText' and 'sinkgit_csv'. Below this, there are three tabs: 'Connection', 'Schema', and 'Parameters', with 'Connection' being the active tab. Under 'Connection', the 'Linked service' dropdown is set to 'linkedServiceDL', and the 'Test connection' button shows a green checkmark with the message 'Connection successful'. The 'File path' field contains 'destination / csv_file / File name' with a 'Browse' button. The 'Compression type' is set to 'No compression', 'Column delimiter' is 'Comma (,),' and 'Row delimiter' is 'Default (\r,\n, or \n\r)'. There are also 'Preview data' and 'Detect format' buttons.

3. Create Azure Data Factory Instance

- Use it to build pipelines for data movement and transformation.

◆ Linked Services in ADF

Linked services act as **connection strings** for ADF to interact with data sources and sinks.

- Go to: **Manage > Linked Services**.
- Create two linked services:
 - One for the **source** (e.g., HTTP or ADLS).
 - One for the **destination** (e.g., ADLS Gen2, Blob Storage).

Http dest file mentioned, can mention any filename. will pick correct file

Example:

- Source Linked Service → HTTP
- Sink Linked Service → ADLS Gen2

◆ Datasets

- Represents the **data structure** within the linked service.
- Used to define input and output data (e.g., CSV, JSON, Parquet).

Examples:

- **Fact_Sales_1.csv** dataset in the **source** folder.
- Output dataset in the **destination** (e.g.,**reporting/Fact_Sales_1.csv**)

Name	Last modified	Access tier	Blob type	Size	Lease state
file2.csv	19/7/2025, 4:04:14 pm	Hot (inferred)	Block blob	2.8 KiB	Available
Fact_Sales_1.csv	19/7/2025, 3:40:47 pm	Hot (inferred)	Block blob	319.93 KiB	Available

◆ Containers in ADLS = Data Lake Zones

- Logical partitions of storage.
 - Example container: **reporting** for storing cleansed and transformed facts data.
-

◆ Copy Activity

Used to **copy data from a source to a sink**.

Two examples:

1. From **HTTP API (source)** to **ADLS Gen2 folder (destination)**.

Preview data

Linked service: Linked_git

Object: anshlambagitz/Azure-Data-Factory/refs/heads/main/Raw%20Data/Fact_Sales_2.csv

#	transaction_id	transactional_date	product_id	customer_id	payment	credit_card	loya
1	4411	5/1/2022 1:02	P0305	6		30271790522719	F
2	4412	5/1/2022 4:00	P0242	7	visa	4041591026711540	F
3	4413	5/1/2022 5:49	P0529	2	mastercard	5048373491517664	F
4	4414	5/1/2022 6:58	P0336	7	mastercard	5048377130633352	T
5	4415	5/1/2022 8:47	P0399	6	visa	4041595611872	F
6	4416	5/1/2022 12:06	P0097	4	visa	4041591008610	F
7	4417	5/1/2022 19:42	P0644	7	mastercard	5048374238650248	T
8	4418	5/1/2022 20:33	P0370	4	americanexpress	374288720448306	T

Preview data

Linked service: linkedServiceDL

Object: Fact_Sales_1.csv

	transaction_id	transactional_date	product_id	customer_id	payment	credit_card	loy
1	1	2021-05-04 02:00:00	P0494	4	visa	4041593010498829	F
2	2	2021-05-04 03:04:00	P0221	5	visa	4041596151234556	F
3	3	2021-05-04 03:56:00	P0625	5	visa	4041594885335898	F
4	4	2021-05-04 05:20:00	P0431	8	mastercard	5108753677552345	F
5	5	2021-05-04 05:45:00	P0058	5	mastercard	5108752372298261	T
6	6	2021-05-04 06:58:00	P0385	6	americanexpress	374288563442549	F
-	-	2021-05-04	-----	-	-	-----	-

2. From ADLS source folder to ADLS destination folder.

Will data be duplicated if you re-run the pipeline?

- It will copy again unless there's logic to check for existing files or overwrite settings are adjusted.

The screenshot shows the Azure Data Factory pipeline editor interface. The top navigation bar includes 'Data Factory', 'Validate all', 'Publish all (11)', 'Preview experience (Off)', and other settings. The main workspace displays a pipeline named 'pipelineGit'. The 'Activities' pane on the left lists 'cop' (selected), 'Move and transform', and 'Copy data'. The pipeline steps are: 'onlySelectedFiles' (selected) > 'ForEachCSV' > 'IfFileMatches' (True) > 'activities'. The 'activities' step is expanded, showing a 'Copy data' activity with the name 'Copy Fact data'. The 'Sink' tab is selected in the 'Copy data' activity configuration pane. Under 'Sink dataset', the 'reporting_sink' dataset is chosen. In the 'Dataset properties' section, the 'Name' is 'p_file_name' and the 'Value' is '@item().name', with a 'Type' of 'string'. The 'Copy behavior' dropdown is set to 'Select...'. The overall interface is clean and modern, typical of cloud-based development tools.

◆ URL Concepts in HTTP Linked Service

- **Base URL:** `https://site.com`
 - **Relative URL:** Path after the base URL
 - ADF combines them to make the complete endpoint.
-

◆ Data Quality & Governance

- Store meaningful, validated data in the destination.
 - Instead of just naming files like `file.csv`, use descriptive names like `facts.csv` under a governed container like `reporting`.
-

◆ Metadata Activity

- Returns **metadata about files** or folders in ADLS.
- Commonly used to:
 - Check for the existence of files
 - List all files in a folder (child items)

Validate Debug Add trigger

Get Metadata

i Get Metadata1

General Settings User properties

Field list * + New | Delete

Argument
Child items

Start time (UTC) End time (UTC)

Filter by last modified ⓘ

Skip line count

```
graph LR; GetMetadata[Get Metadata] --> ForEach[ForEach]
```



Data Factory Validate all Publish all 11 Preview experience Off

Activities <<

cop Move and transform Copy data

Validate Debug Add trigger

Get Metadata

i Get Metadata1

General Settings User properties

Dataset * metadataDS Open New Learn more

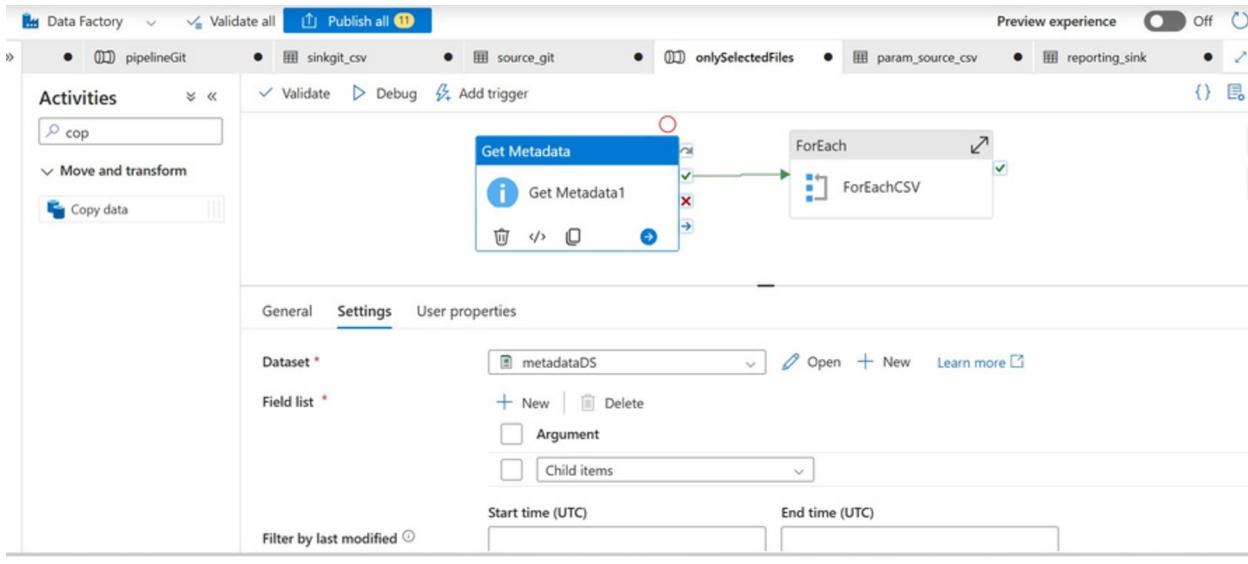
Field list * + New | Delete

Argument
Child items

Start time (UTC) End time (UTC)

Filter by last modified ⓘ

```
graph LR; GetMetadata[Get Metadata] --> ForEach[ForEach]
```

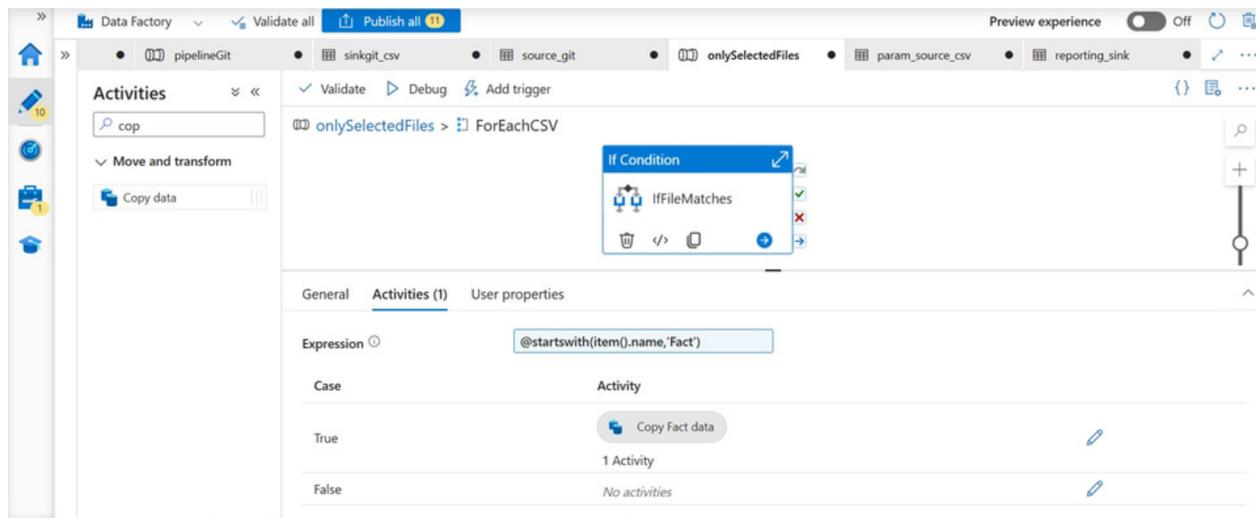


◆ Conditional Copy: If Condition Activity

- Use when you want to **copy only specific files**.

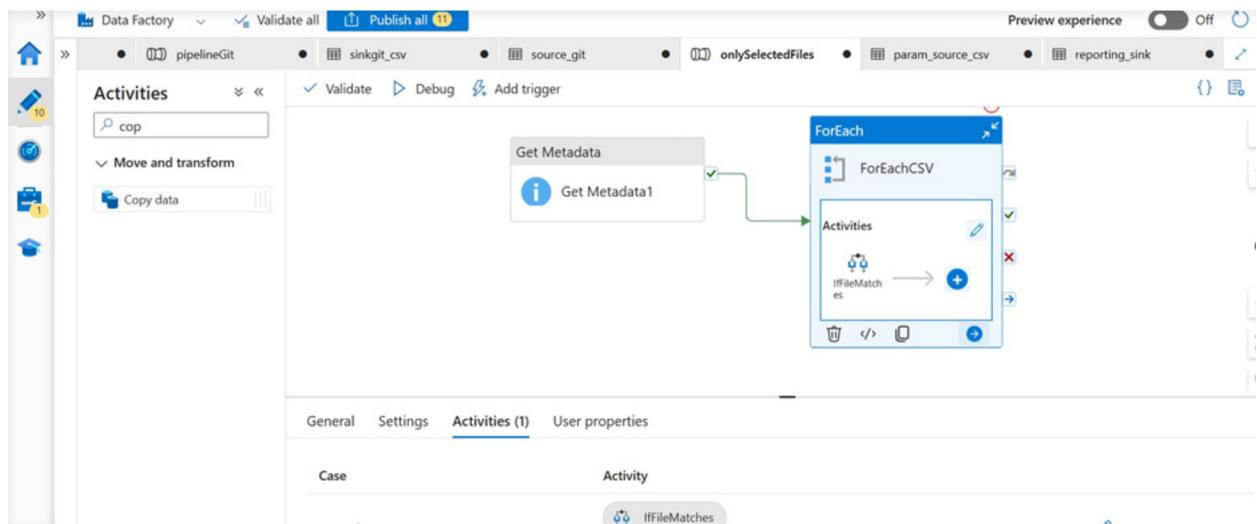
Example Use Case:

- Copy only if file **name starts with "Fact"**.
- Else → **Do nothing**.



Steps:

1. Use **Get Metadata** to get child items.
2. Use **For Each** activity to iterate through the files.



3. Inside the loop, use **If Condition** to filter files by name.
4. If true → Perform **Copy Activity**.

The screenshot shows the Azure Data Factory pipeline designer interface. An 'If Condition' activity is nested within a 'ForEachCSV' activity. The 'If Condition' activity has an 'IfFileMatches' condition. The 'Activities' tab of the 'If Condition' activity shows a single case labeled 'True' with the value 'No activities'. To the right, a 'Pipeline expression builder' window is open, displaying the expression '@startswith(item().name, 'Fact')'. The pipeline navigation bar at the top includes tabs for Validate all, Publish all, and various triggers and datasets.

◆ Control Flow: Activity Outcomes

Each activity in ADF can branch based on outcomes:

- **On Success:** Next step if the activity succeeds.
- **On Failure:** If the activity fails.
- **On Completion:** Executes regardless of result.
- **On Skip:** Executes if the activity is skipped.



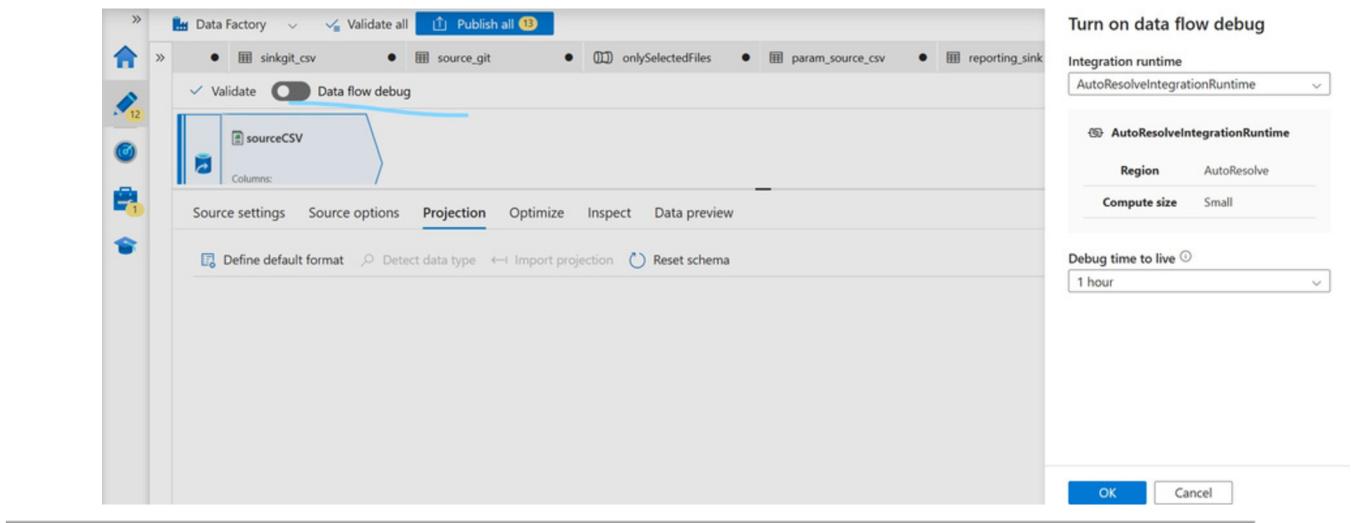
◆ Parameters

- Help make pipelines **dynamic and reusable**.
- Example: Pass source file name, destination folder name, or filter condition.

The screenshot shows the Azure Data Factory pipeline editor. On the left, there's a dataset configuration for a 'DelimitedText' type. The 'File path' field is set to 'destination / csv_files / @dataset().p_file_name'. To the right of the pipeline, a 'Pipeline expression builder' window is open. It shows the expression '@dataset().p_file_name' in the 'Parameters' tab. The 'OK' button is visible at the bottom of the builder window.

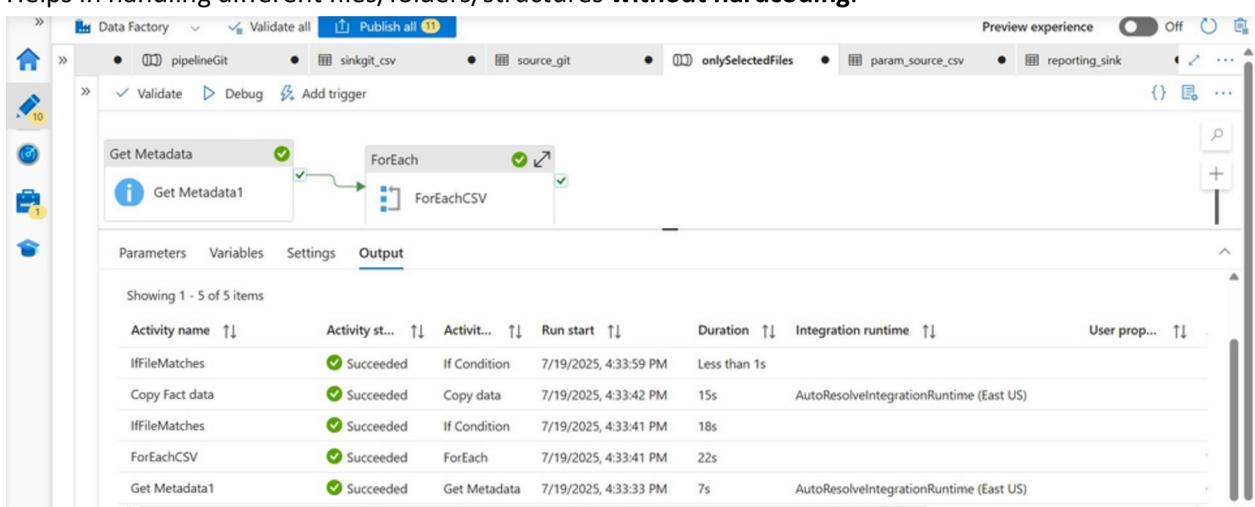
◆ Data Flow (Mapping Data Flow)

- Visual interface for **data transformation** logic.
- More like SSIS with features like joins, filters, derived columns, etc.
- Used for **row-level transformation**.



◆ Dynamic Pipelines

- Use expressions, parameters, system variables to **make pipelines dynamic**.
- Helps in handling different files/folders/structures **without hardcoding**.



Wildcard Path:

Used in datasets to read multiple files matching a pattern (e.g., Fact_*.csv). Helps automate ingestion of similar structured files.

The screenshot shows the 'Source options' tab for a dataset named 'sourceCSV'. The 'Wildcard paths' section is highlighted, showing an input field with a placeholder 'Add dynamic content [Alt+Shift+D]'. Below it are other configuration options: 'Partitions root path', 'Allow no files found', 'List of files', 'Multiline rows', and 'Maximum columns' set to 20480. On the left sidebar, there are sections for Pipelines, Datasets, and Change Data Capture (preview), with various datasets listed under Datasets.

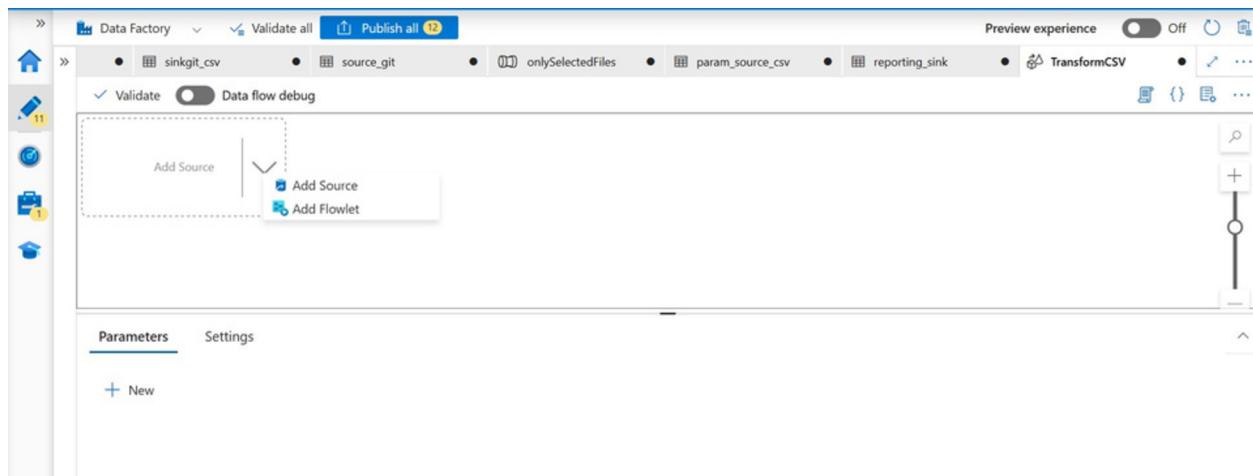
Importing Projection:

In **data flow**, this lets you import the schema from the source to define the structure and mapping of incoming data.

The screenshot shows the 'Projection' tab for the same 'sourceCSV' dataset. A tooltip message 'Successfully started importing the schema for sourceCSV (Source).' is visible in the top right corner. The 'Importing' status indicator is active. At the bottom of the tab, there are buttons for 'Define default format', 'Detect data type', 'Importing projection' (which is currently selected), and 'Reset schema'.

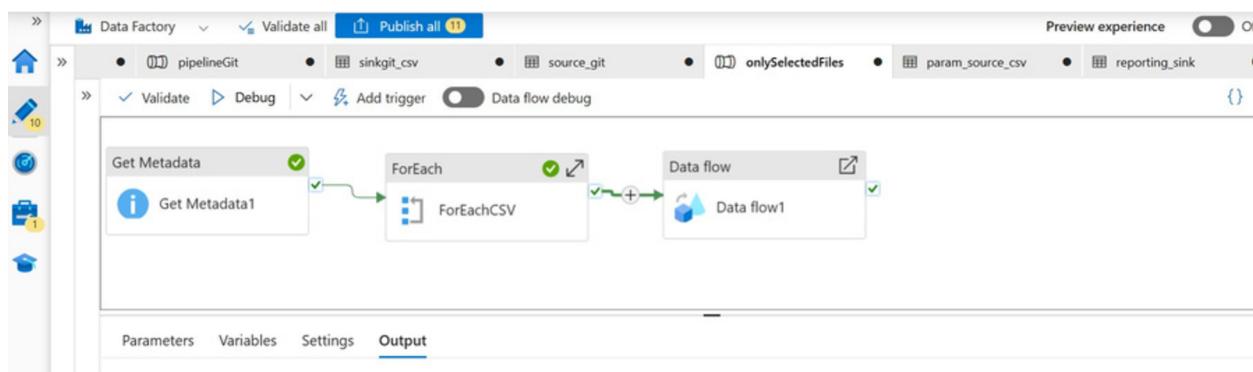
Source Select (Add Source):

In Data Flow, use Add Source to bring in input datasets for transformation. Multiple sources can be merged or joined based on logic.



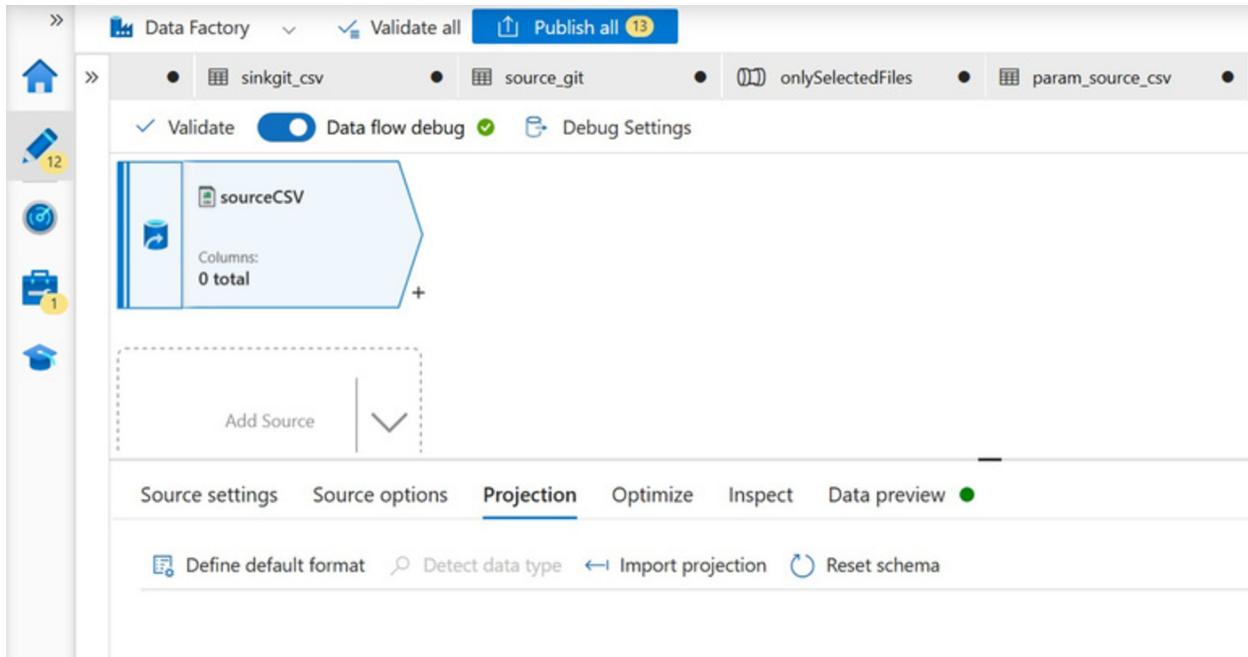
Data Flow for Spark:

ADF's Data Flow runs on **Apache Spark** clusters in the background, allowing **distributed data transformation at scale**.



Data Flow Debug Mode:

Once **debug mode** is enabled in a Data Flow, a green check mark appears. This allows you to preview data at each transformation stage before publishing or running the pipeline.



◆ Common ADF Best Practices

- Use **naming conventions** for clarity.
- Parameterize wherever possible.
- Use **debug mode** to test pipelines.
- Organize **folders and pipelines** for modularity.

Answers to Key Questions

Q1: If data already copied from API to storage, and I run the copy command again, will it duplicate?

- Yes, unless overwrite or existence check logic is applied.
- Use 'If Condition' + Get Metadata to avoid copying duplicates.

Q2: If I make changes in pipeline activities, do I need to re-run the pipeline?

- Yes. If the pipeline has already run, any changes require a **new pipeline run** to reflect updated logic.