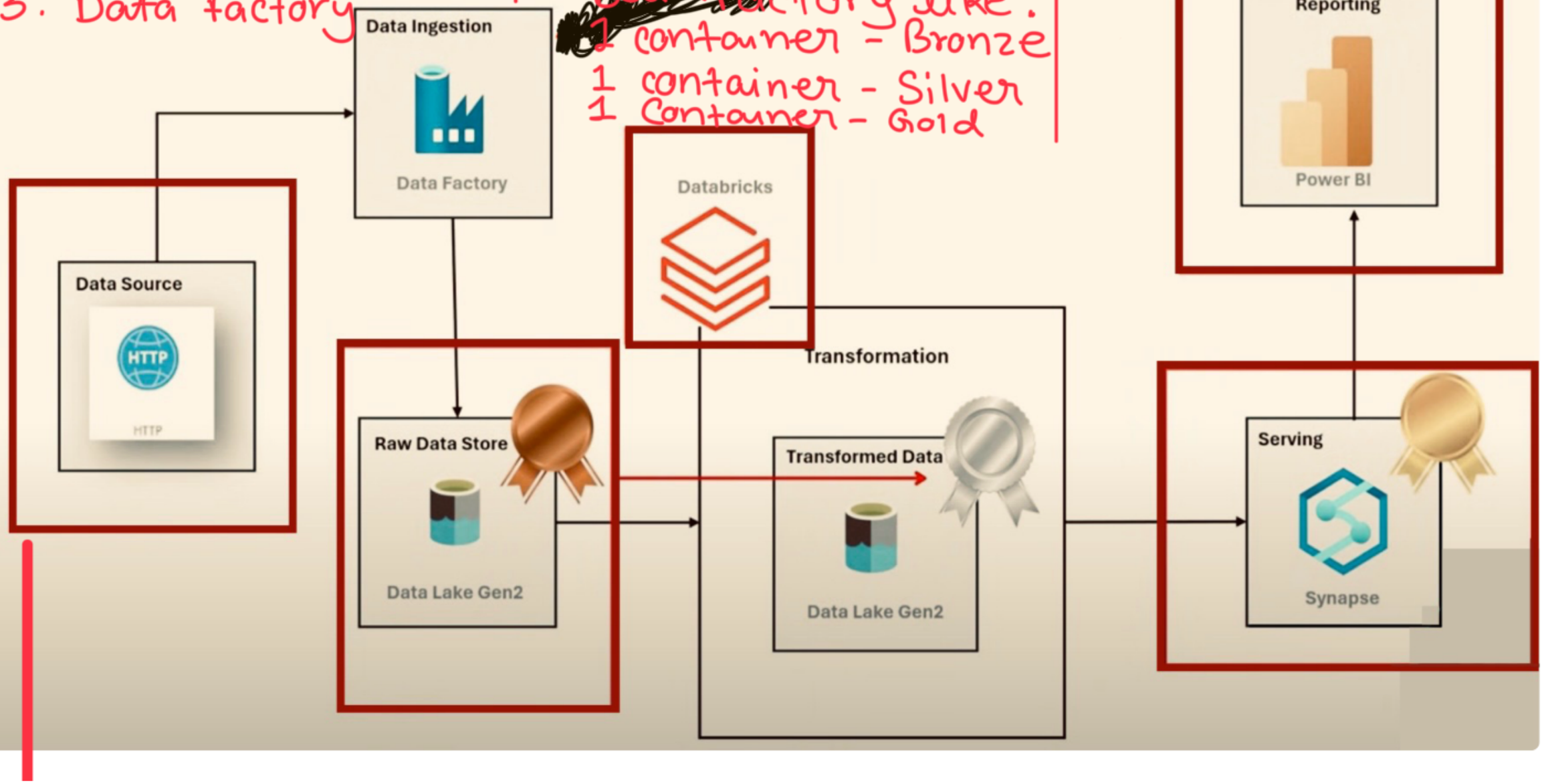
Azure End to End Data Enigneering Project

Data Architecture of the Project

1. Resource groups
2. Storage account
3. Data factory

4. Now we will create containers inside Data factory like:
   1 container - Bronze
   1 container - Silver
   1 Container - Gold



We will be pulling data directly out of APIS.

3 layers of data

☆ Bronze layer —— Keep the data as it is available in source. No transformation will be applied on it.

☆ Silver layer —— Then we will clean and transform the date present in Bronze layer and push it to Silver layer using spark.

☆ Gold layer —— We will build date warehouse and the most popular one right now is Azure Synapse Analytics

Difference between blob storage and datalake

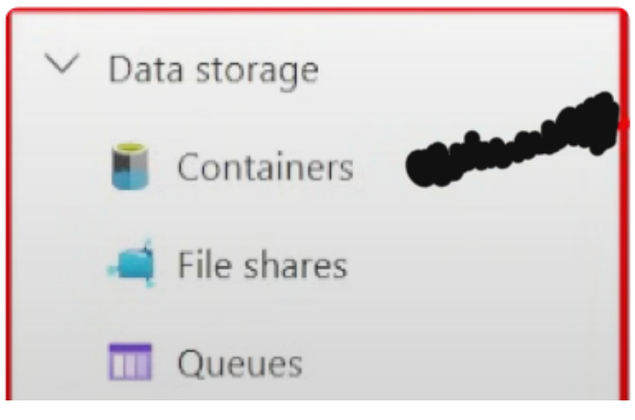
Blob

Datalake

and then file

we can not create folder in blob

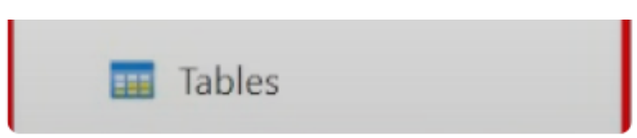
we can create Folders in Datalake



Also Known as Data lake

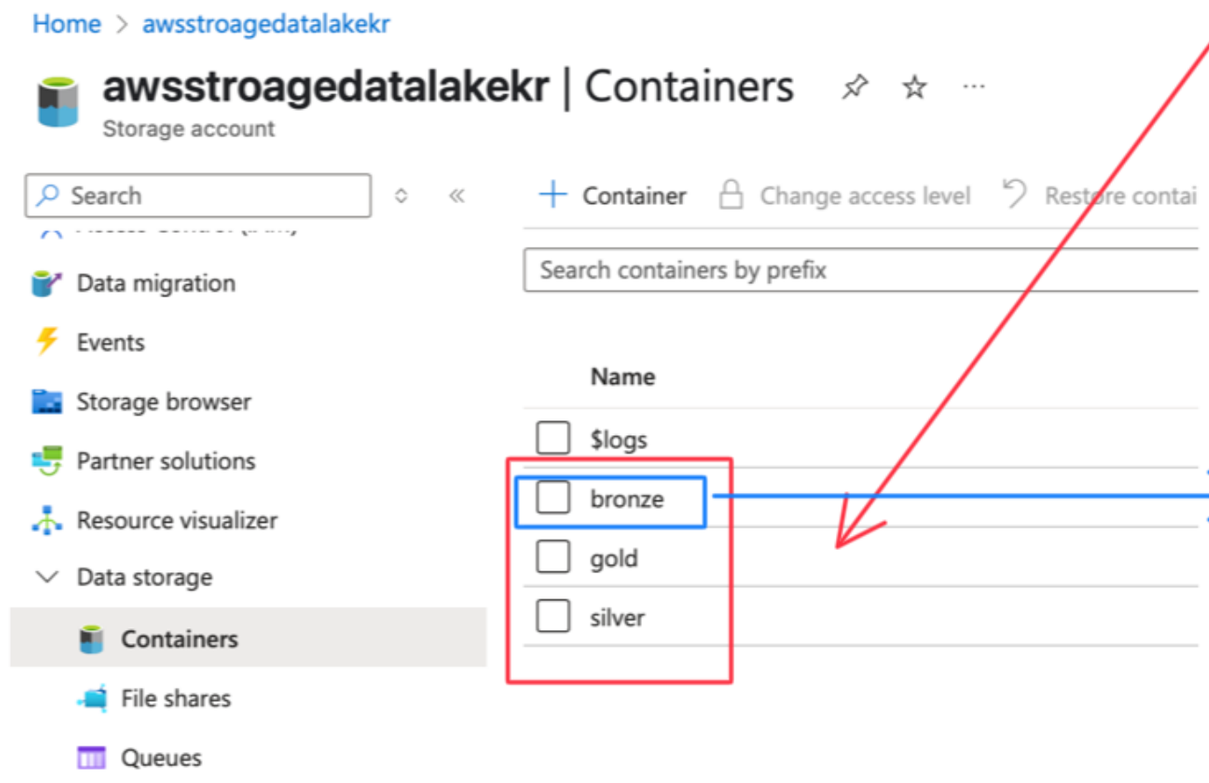
To share all the files throughout the organization

can be ,pson data

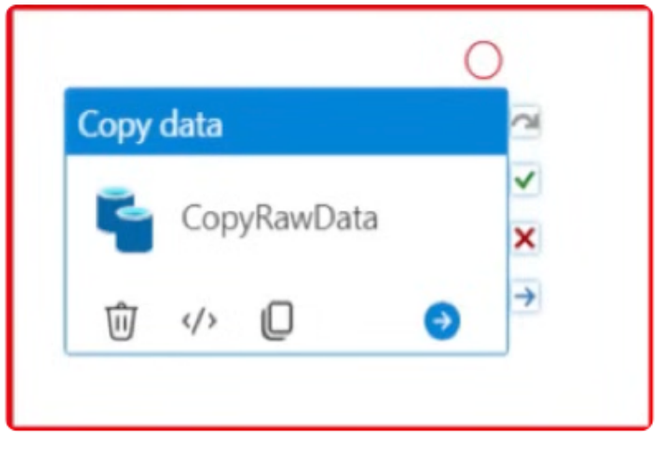


NOSQL databases

Now, we will create 3 containers

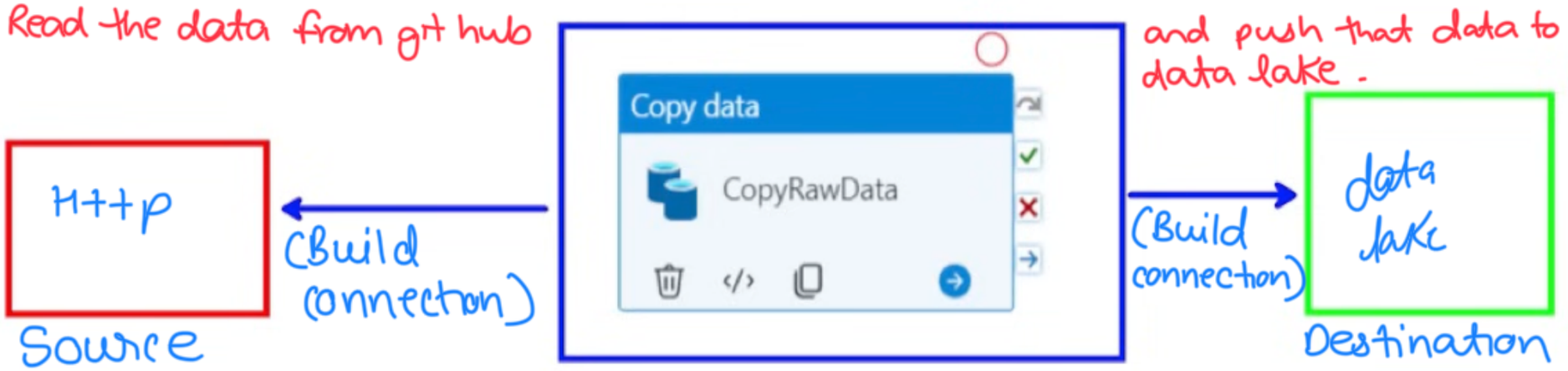


After creating 3 containers. Now, we will perform data loading in bronze layer



This will just load the data from the source and push the data to the destination.

Now, we want to create a linked service for connection.



Linked services

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

+ New

linked service has been created

Filter by name

Annotations : **Any**

Showing 1 - 2 of 2 items

Name	Type	Related	Annotations
httplinkedservice	HTTP	0	
storagedatalake	Azure Data Lake Storage Gen2	0	

Now , we will create dataset

[redacted] Source Sink [redacted]

Source dataset * ds_http Open New Preview data

Request method * GET

Additional headers

Request body

Request timeout

Max concurrent connections

Skip line count

Additional columns New

[redacted] Source Sink [redacted]

Sink dataset * ds_raw Open New

Copy behavior Select...

Max concurrent connections

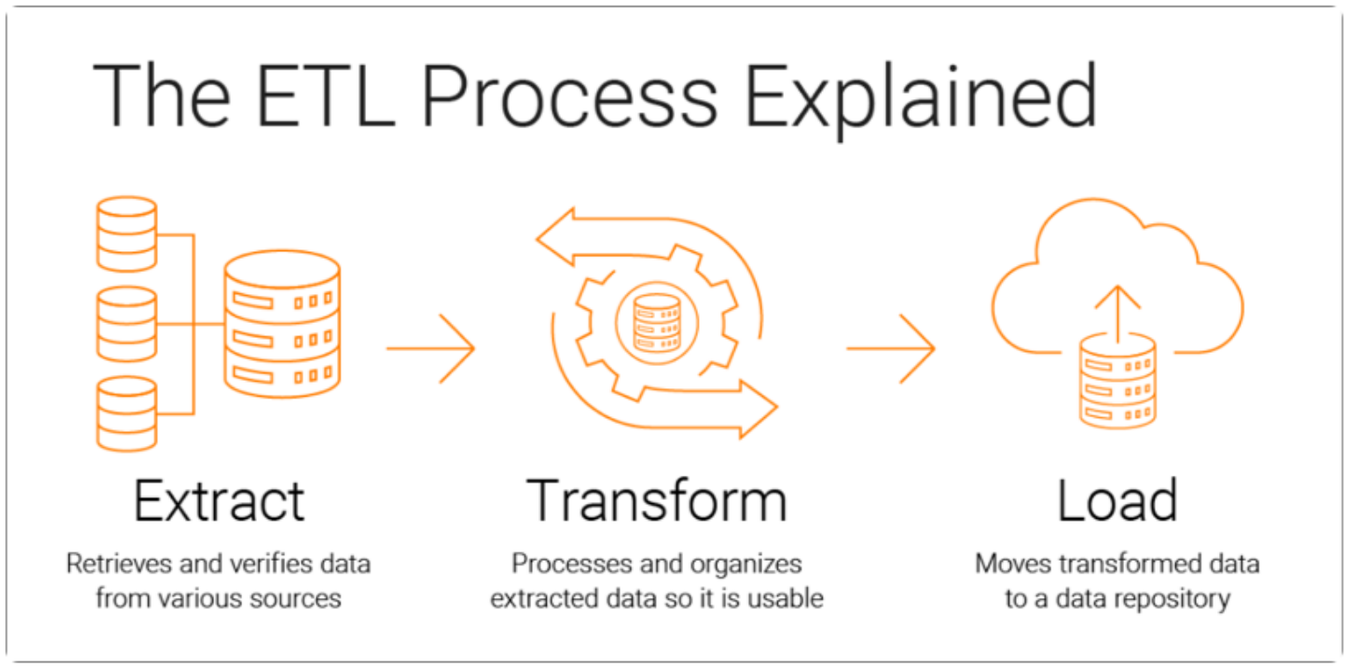
Block size (MB)

Metadata New

Quote all text

File extension .txt

Max rows per file



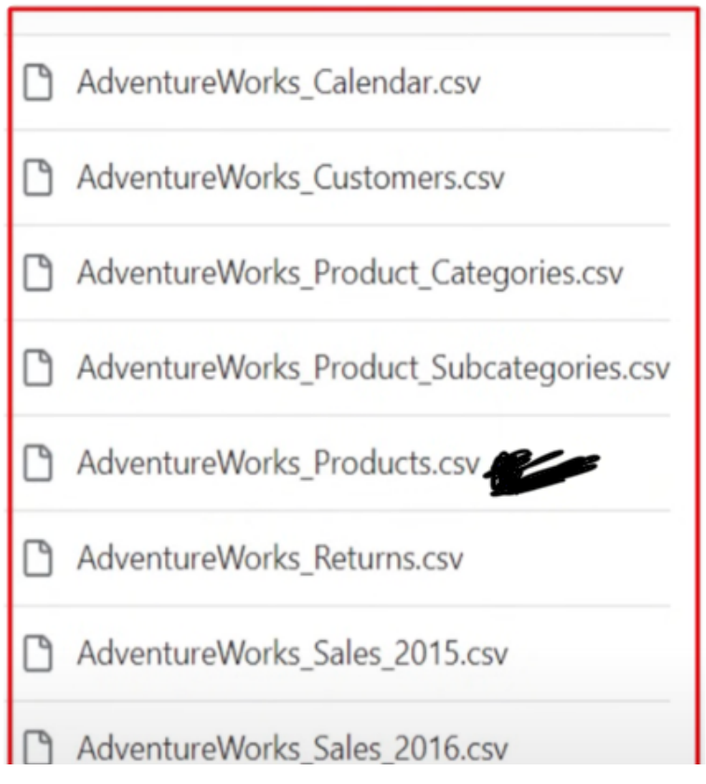
Static Pipeline & dynamic Pipeline

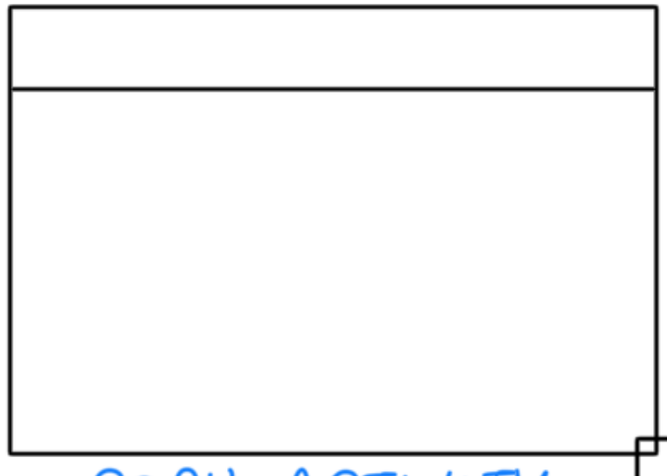
Static Pipeline :

A static pipeline has a fixed structure - its sequence of steps is predefined and does not change at runtime.

Dynamic Pipeline :

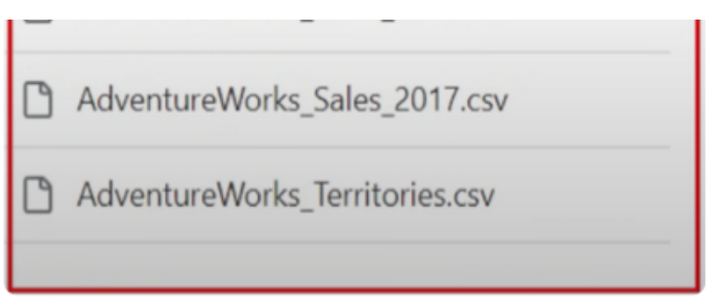
A dynamic Pipeline can change its structure during execution, often based on conditions or the characteristics of the input data.





[relative Url]
(folder)
(file)

Just these 3 values will be changing for this

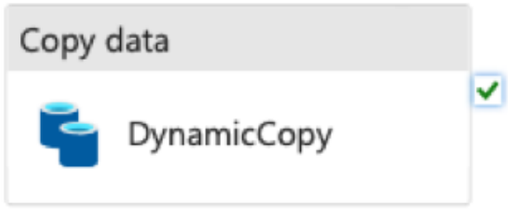


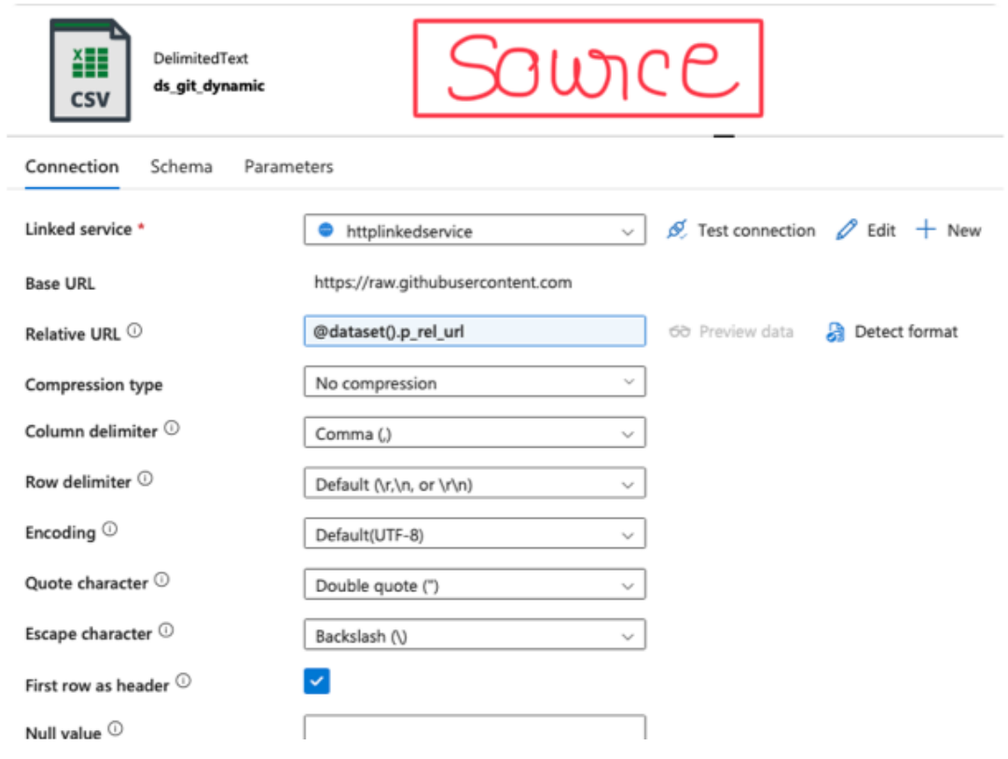
(Total 10 iterations)

we want to pull this data. one way of doing is repeating the copy activity again and again. But that will be static approach which we do not want to do that.
So, we will use for loop.

Copy activity.

So, instead of changing those 3 values everytme. we will create 3 parameters, and keep on changing the values of these 3 parameters, So everytme each iteration will have different set of parameters. and for that we will use For Each Activity.





DelimitedText
ds_sink_dynamic

Sink

Connection | Schema | Parameters

Linked service * : storagedatalake | Test connection | Edit | New | Learn more

File path : bronze / @dataset().p_sink_folder / @dataset().p_file_name

Compression type : No compression

Column delimiter : Comma (,)

Row delimiter : Default (\r,\n, or \r\n)

Encoding : Default(UTF-8)

Quote character : Double quote (")

Escape character : Backslash (\)

First row as header : ☑

Null value :

General | **Source** | Sink | Mapping | Settings | User properties

Source dataset * : ds_git_dynamic | Open | New | Preview data | Learn more

Dataset properties

Name	Value	Type
p_rel_url	Value	String

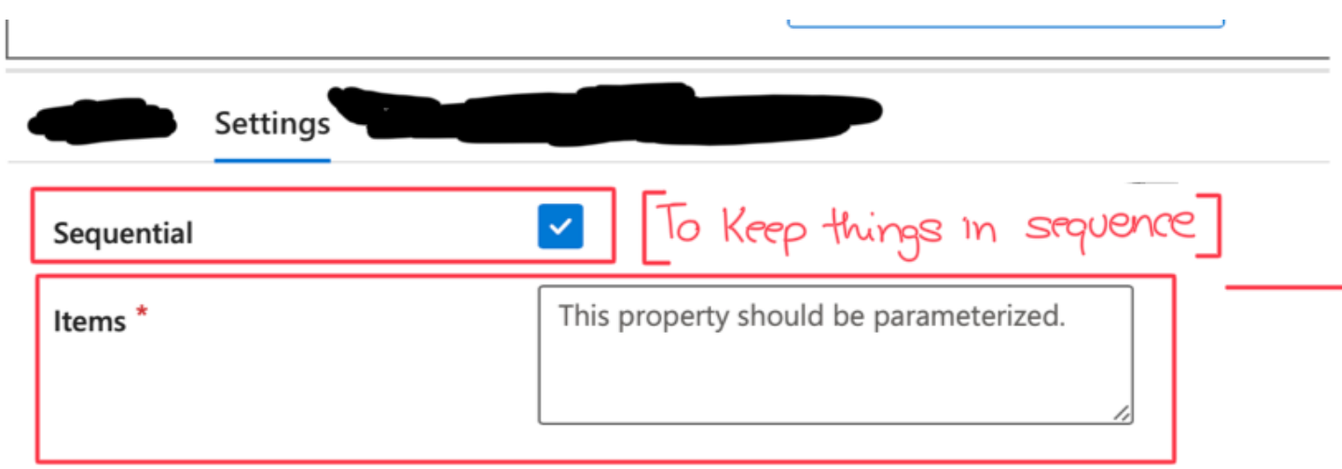
General | Source | **Sink** | Mapping | Settings | User properties

Sink dataset * : ds_sink_dynamic | Open | New | Learn more

Dataset properties

Name	Value	Type
p_sink_folder	Value	String
p_file_name	Value	String

we need to provide values for 3 parameters



we need to give something to run a loop. we will create a json file with arrays and Parameters inside.

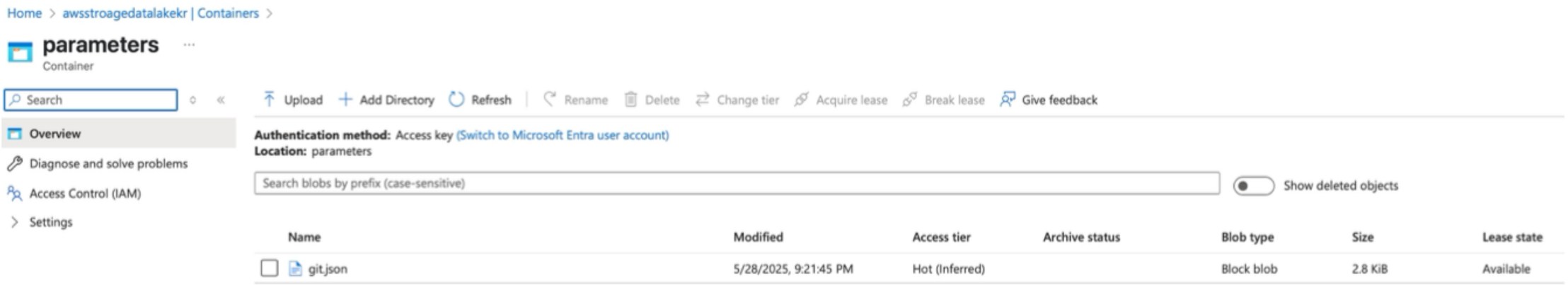
Json file

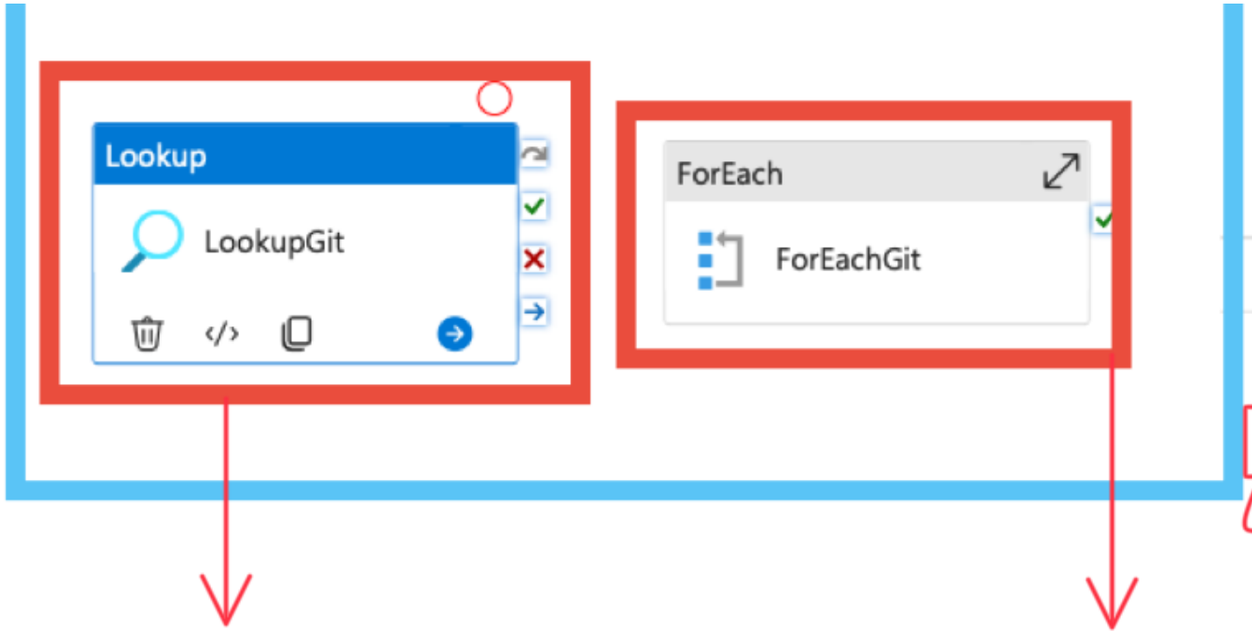
git.json

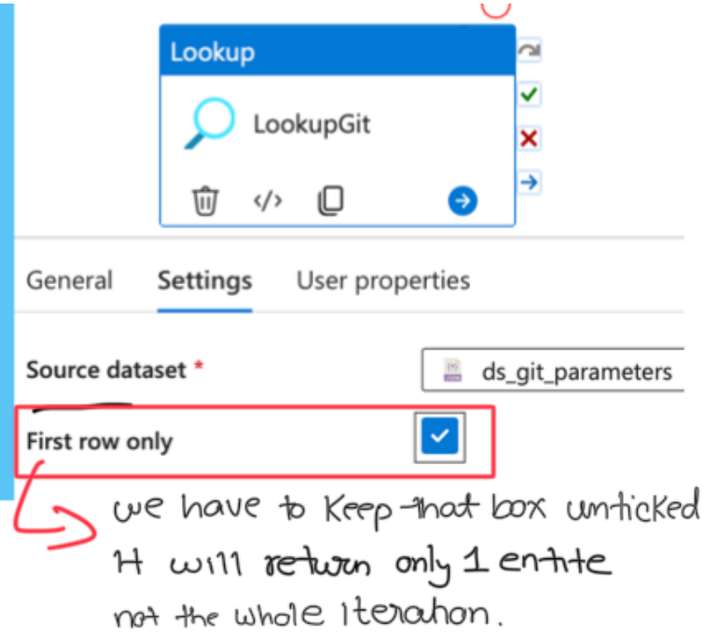
```
[
    {
        "p_rel_url" : "KumarVaibhav27/Azure-End-to-End-Data-Engineering-Project/refs/heads/main/Data/AdventureWorks_Calendar.csv",
        "p_sink_folder" : "AdventureWorks_Calendar",
        "p_file_name" : "AdventureWorks_Calendar.csv"
    },
    {
        "p_rel_url" : "KumarVaibhav27/Azure-End-to-End-Data-Engineering-Project/refs/heads/main/Data/AdventureWorks_Customers.csv",
        "p_sink_folder" : "AdventureWorks_Customers",
        "p_file_name" : "AdventureWorks_Customers.csv"
    },
    {
        "p_rel_url" : "KumarVaibhav27/Azure-End-to-End-Data-Engineering-Project/refs/heads/main/Data/AdventureWorks_Product_Categories.csv",
        "p_sink_folder" : "AdventureWorks_Product_Categories",
        "p_file_name" : "AdventureWorks_Product_Categories.csv"
    },
    {
        "p_rel_url" : "KumarVaibhav27/Azure-End-to-End-Data-Engineering-Project/refs/heads/main/Data/AdventureWorks_Product_Subcategories.csv",
        "p_sink_folder" : "AdventureWorks_Product_Subcategories",
        "p_file_name" : "AdventureWorks_Product_Subcategories.csv"
    },
    {
        "p_rel_url" : "KumarVaibhav27/Azure-End-to-End-Data-Engineering-Project/refs/heads/main/Data/AdventureWorks_Products.csv",
        "p_sink_folder" : "AdventureWorks_Products",
        "p_file_name" : "AdventureWorks_Products.csv"
    },
    {
        "p_rel_url" : "KumarVaibhav27/Azure-End-to-End-Data-Engineering-Project/refs/heads/main/Data/AdventureWorks_Returns.csv",
        "p_sink_folder" : "AdventureWorks_Returns",
        "p_file_name" : "AdventureWorks_Returns.csv"
    },
```

```
    {
        "p_rel_url" : "KumarVaibhav27/Azure-End-to-End-Data-Engineering-Project/refs/heads/main/Data/AdventureWorks_Sales_2015.csv",
        "p_sink_folder" : "AdventureWorks_Sales_2015",
        "p_file_name" : "AdventureWorks_Sales_2015.csv"
    },
    {
        "p_rel_url" : "KumarVaibhav27/Azure-End-to-End-Data-Engineering-Project/refs/heads/main/Data/AdventureWorks_Sales_2015.csv",
        "p_sink_folder" : "AdventureWorks_Sales_2015",
        "p_file_name" : "AdventureWorks_Sales_2015.csv"
    },
    {
        "p_rel_url" : "KumarVaibhav27/Azure-End-to-End-Data-Engineering-Project/refs/heads/main/Data/AdventureWorks_Sales_2016.csv",
        "p_sink_folder" : "AdventureWorks_Sales_2016",
        "p_file_name" : "AdventureWorks_Sales_2016.csv"
    },
    {
        "p_rel_url" : "KumarVaibhav27/Azure-End-to-End-Data-Engineering-Project/refs/heads/main/Data/AdventureWorks_Sales_2017.csv",
        "p_sink_folder" : "AdventureWorks_Sales_2017",
        "p_file_name" : "AdventureWorks_Sales_2017.csv"
    },
    {
        "p_rel_url" : "KumarVaibhav27/Azure-End-to-End-Data-Engineering-Project/refs/heads/main/Data/AdventureWorks_Territories.csv",
        "p_sink_folder" : "AdventureWorks_Territories",
        "p_file_name" : "AdventureWorks_Territories.csv"
    }
]
```

It will be feeding these 3 parameters to that activity. and dynamically pushing the data to Azure.







About Lookup Activity: Fetching Dynamic Data

The Lookup activity is designed to retrieve data from various sources, such as databases, files, or query execution results.
The output can range from a single value to an array of objects or records.

Here are a few use cases:

Dynamically gather configuration data based on pipeline execution.
Obtain a list of file names for subsequent processing.

ForEach Loop: Iterating Through Data

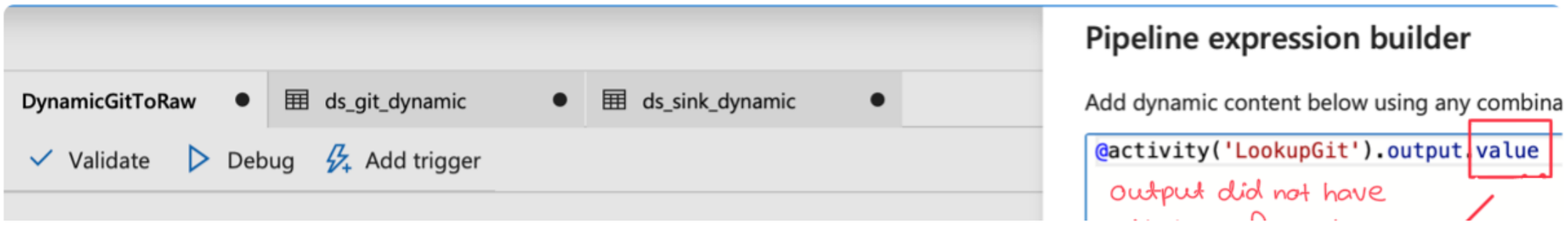
The ForEach loop iterates through a collection of data, typically an array, to perform repeated tasks on each item.
It executes a set of activities within the loop for each item in the collection.

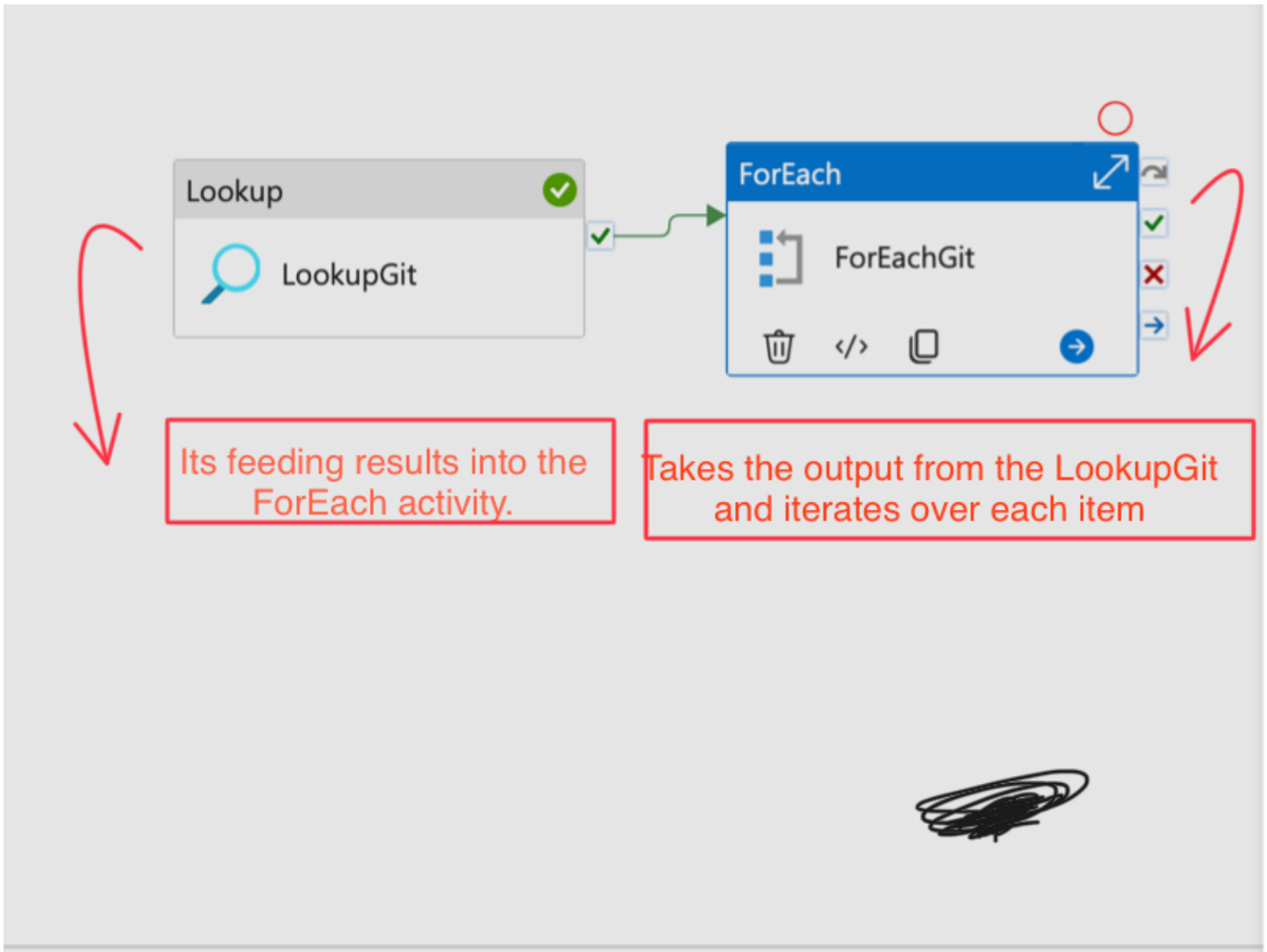
Here are a few use cases:

Handle each record from a Lookup activity's output.
Copy each file in a list to a different location.
Execute separate activities for each ID retrieved.

Benefits of Lookup and ForEach activities

- The combination of Lookup and ForEach activities allows for flexible data processing based on the retrieved data.
- Efficiently handles repetitive operations on multiple records, making data workflows more streamlined.
- Breaks down complex pipelines into smaller, reusable activities, enhancing maintainability and scalability.





all the information. because we just want to pass the array. and Array was stored in value key.

Clear contents

Activity outputs | Parameters | Syst

Search

LookupGit
LookupGit activity output

LookupGit count
Count of the rows

LookupGit value array
Array of row data

General | Settings [1] | Activities (0) | User properties

Sequential

Items *

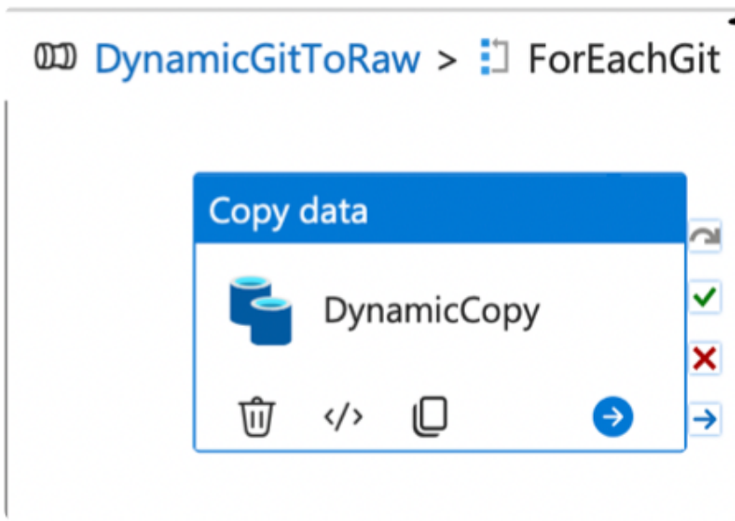
This property should be parameterized.

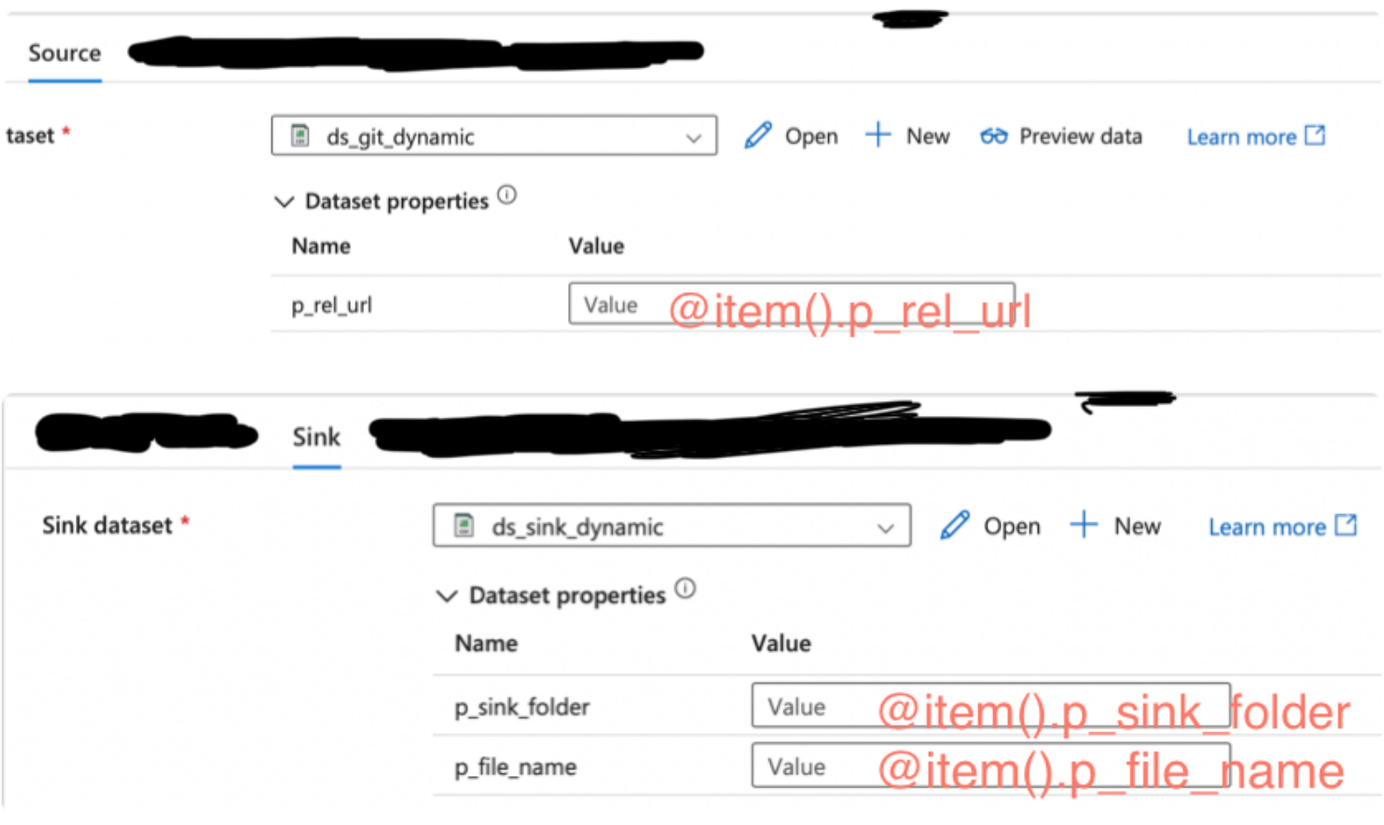
Add dynamic content [Alt+Shift+D]

@activity('LookupGit').output.value

OK | Cancel

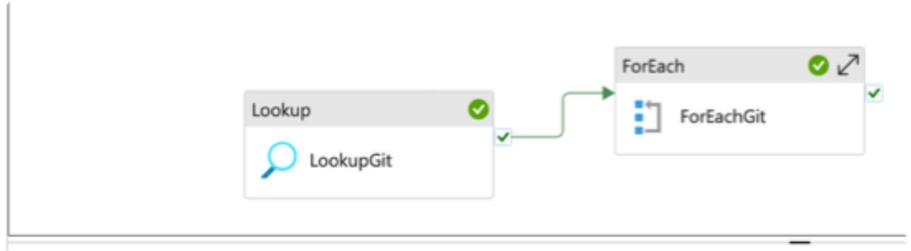
After that, we will cut the "Copy Data" block and paste it inside the "For Each Git" and then we will give value for:

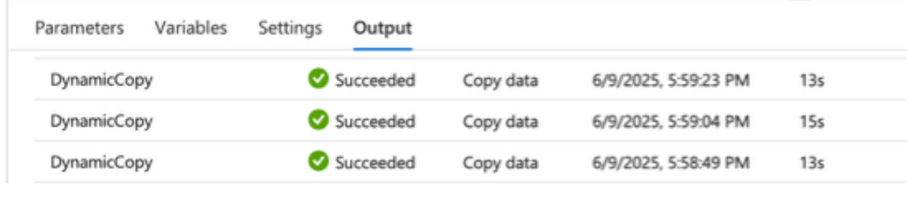




p_rel_url in Source

p_sink_folder
p_file_name
in Sink





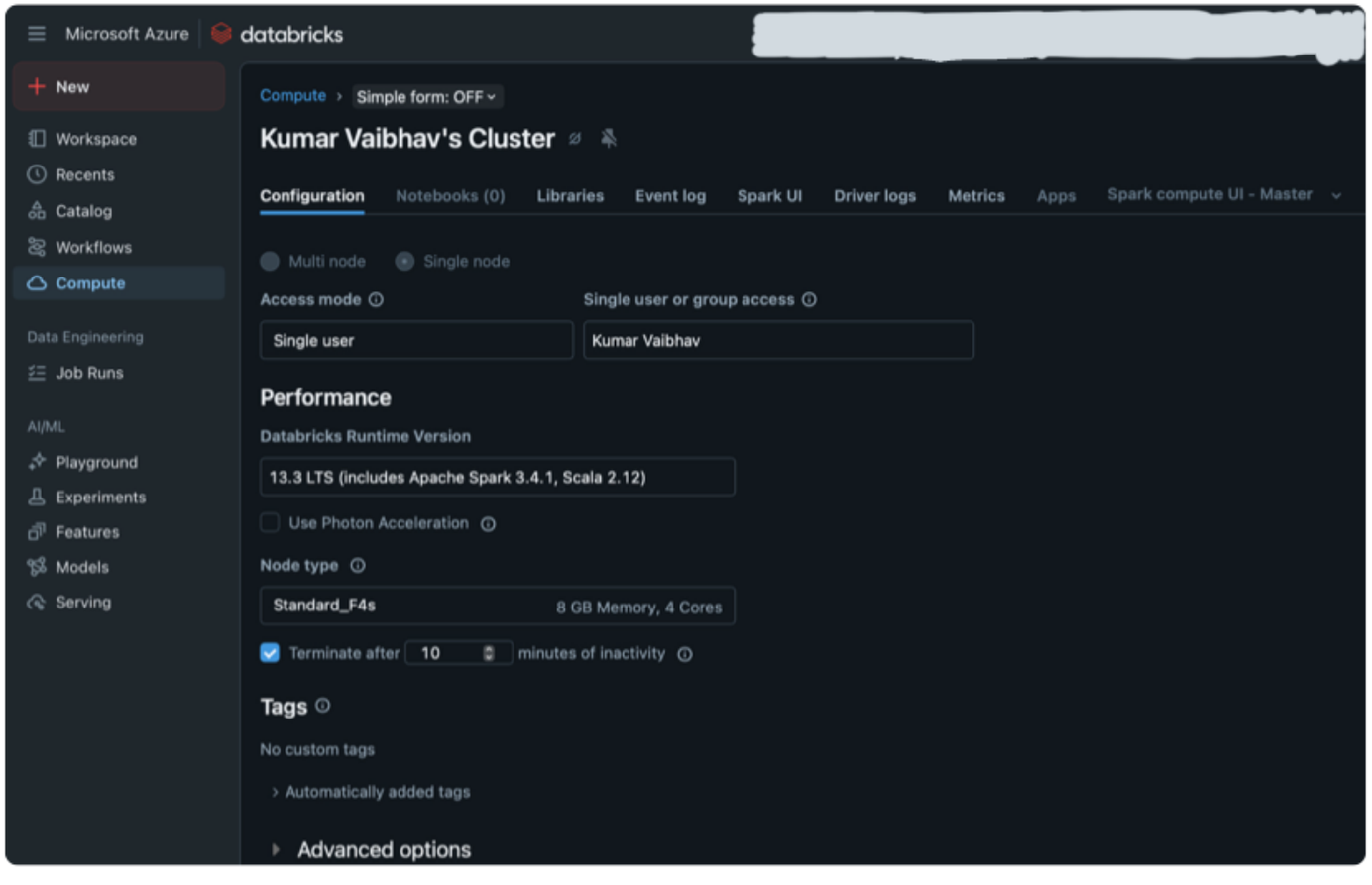
All the files are successfully Copy or pushed into the bronze folder.

DynamicCopy	Succeeded	Copy data	6/9/2025, 5:58:32 PM	15s
DynamicCopy	Succeeded	Copy data	6/9/2025, 5:58:17 PM	13s
DynamicCopy	Succeeded	Copy data	6/9/2025, 5:58:01 PM	14s
DynamicCopy	Succeeded	Copy data	6/9/2025, 5:57:46 PM	13s
DynamicCopy	Succeeded	Copy data	6/9/2025, 5:57:29 PM	16s
DynamicCopy	Succeeded	Copy data	6/9/2025, 5:57:12 PM	16s
DynamicCopy	Succeeded	Copy data	6/9/2025, 5:56:56 PM	14s
DynamicCopy	Succeeded	Copy data	6/9/2025, 5:56:43 PM	12s
ForEachGit	Succeeded	ForEach	6/9/2025, 5:56:42 PM	2m 56s
LookupGit	Succeeded	Lookup	6/9/2025, 5:56:23 PM	19s

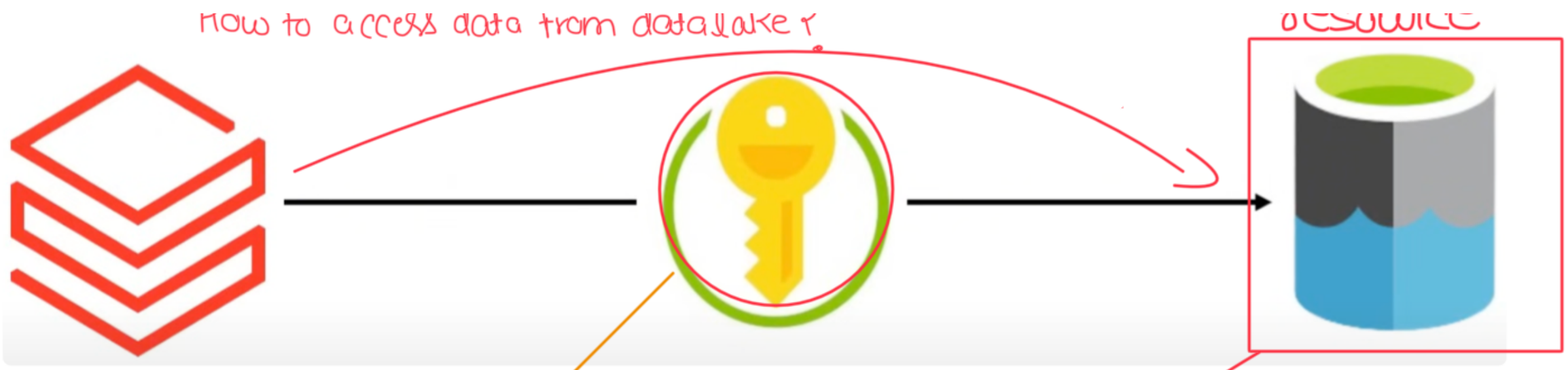
★ Now next Phase : Azure Databricks Resources

Starting with compute

Once our cluster is ready then we can begin using Databricks.



How to access data from datalake ?



Data access - (Stored in Datalake)
we will create an application,
known as service based application
and this application will have access
to azure datalake. And this
application will be used by databricks.

Data is in
Data lake.

Could not continue working because I didn't have the access to **MICROSOFT ENTRA ID** -> **APP REGISTRATION.**