Project

In this we will ingest data from ECDC (European Centre for Disease prevention and Control) website from where we will take data regarding Covid – 19 and then store it in Azure Data Lake Gen 2 using ADF pipelines.

We will ingest the following files:

- Covid – 19 new cases and deaths by country
- Covid – 19 Hospital Admissions & ICU cases
- Covid – 19 Testing Numbers
- Country Response to Covid – 19

These files are available in one of the Git Repo and the link to these files are:

https://github.com/loveleenverma/covid19/blob/main/ecdc_data/cases_deaths.csv

https://github.com/loveleenverma/covid19/blob/main/ecdc_data/hospital_admissions.csv

https://github.com/loveleenverma/covid19/blob/main/ecdc_data/testing.csv

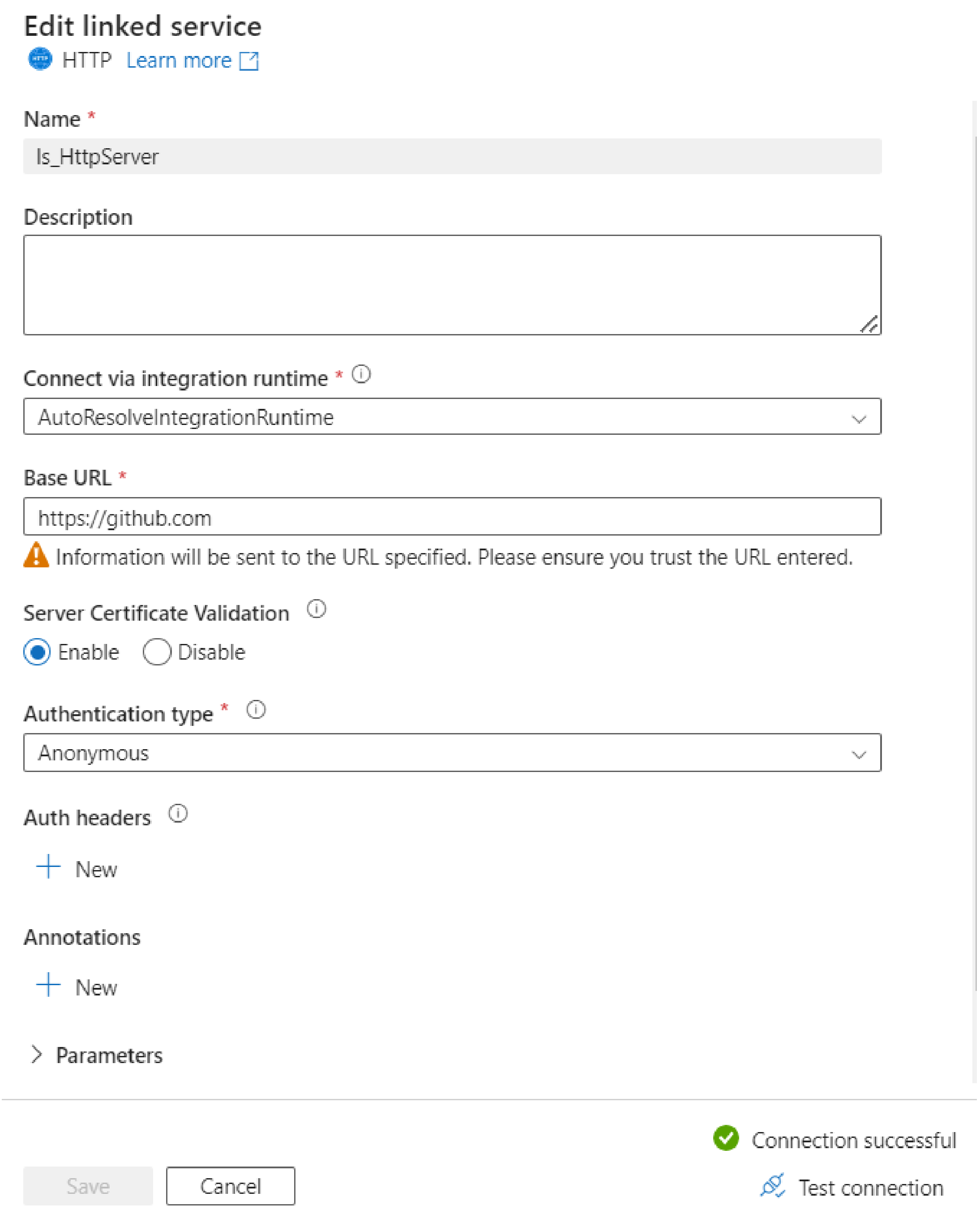
https://github.com/loveleenverma/covid19/blob/main/ecdc_data/country_response.csv

Our task is to build a pipeline that will ingest data from these links in regular intervals and store it in data lake after validation.

Process

As we need to ingest data from a HTTP endpoint into Data Lake so we have to follow the following Process:

- Create linked Service to establish a connection with both Data Lake and HTTP endpoint. For HTTP linked service as the data is there in github so we will provide HTTP://github.com as the base URL and as no authentication is there so give anonymous as authentication type while creating linked service.



Edit linked service

HTTP Learn more

Name *

ls_HttpServer

Description

Connect via integration runtime *

AutoResolveIntegrationRuntime

Base URL *

https://github.com

Information will be sent to the URL specified. Please ensure you trust the URL entered.

Server Certificate Validation

● Enable ○ Disable

Authentication type *

Anonymous

Auth headers

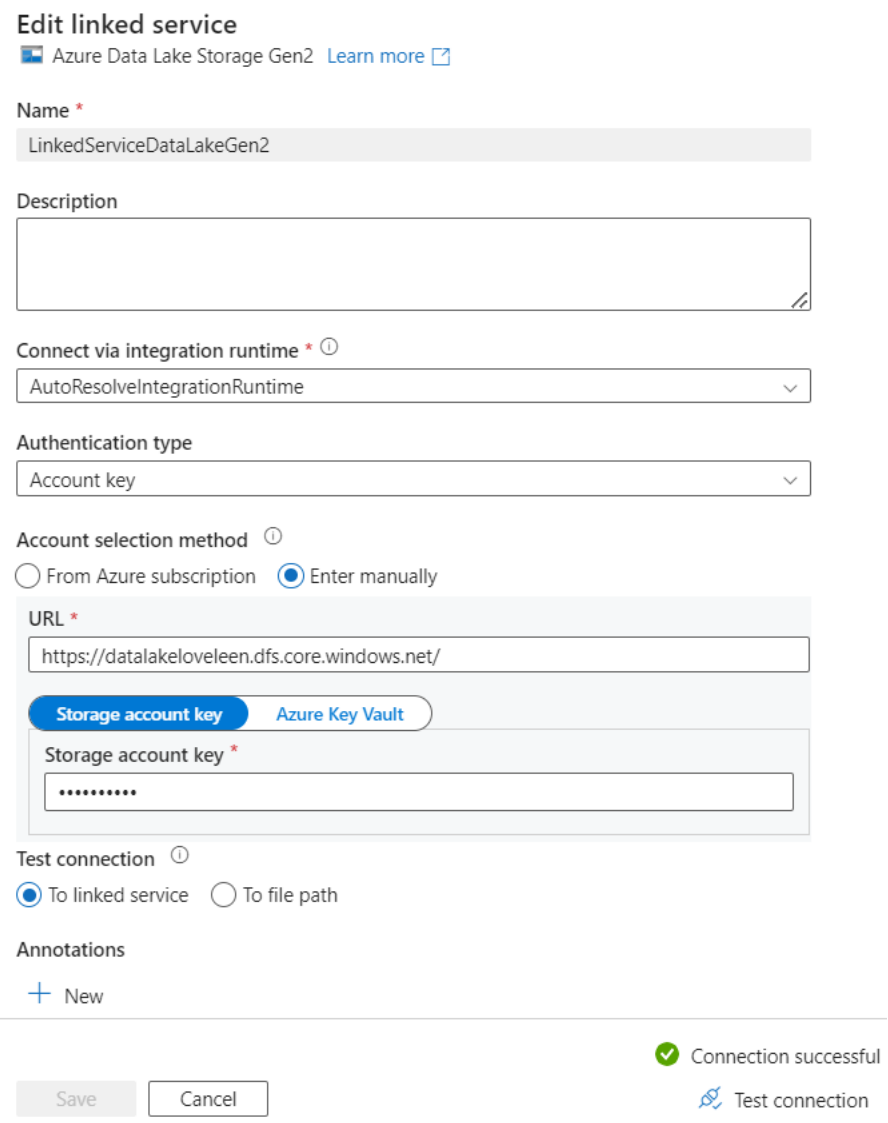
+ New

Annotations

+ New

> Parameters

Connection successful



Edit linked service

Azure Data Lake Storage Gen2

Name *: LinkedServiceDataLakeGen2

Description:

Connect via integration runtime *: AutoResolveIntegrationRuntime

Authentication type: Account key

Account selection method: ○ From Azure subscription ● Enter manually

URL *: https://datalakeloveleen.dfs.core.windows.net/

Storage account key *: ••••••••••

Test connection: ● To linked service ○ To file path

Annotations

Connection successful

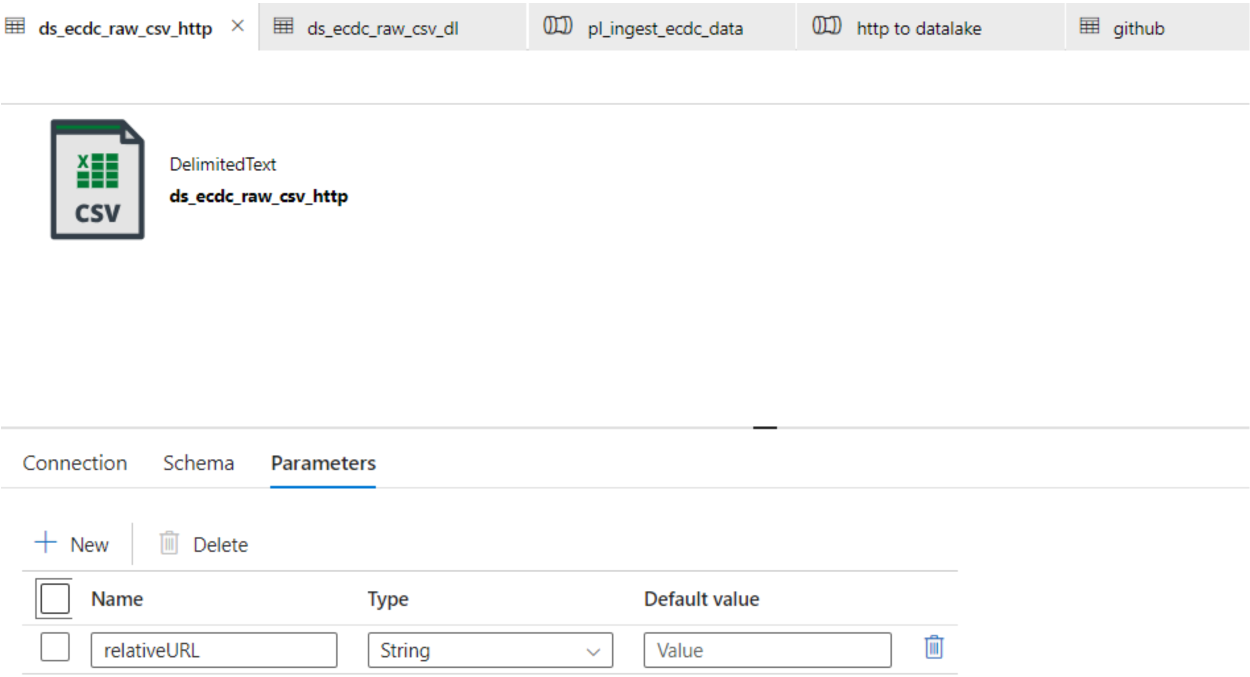
- Now create datasets that will refer to the exact locations of files and the container where we have to put the files in.
- We create the dataset for first file i.e., cases_deaths.csv
- Then we will create one copy data activity using those datasets and data movement is done.
- But if we want to copy every file listed above then in this scenario, we must create 8 datasets. 4 representing the http source and 4 representing the data lake container where the files will reside.
- So, for making our pipeline dynamic we can make use of parameters and variables.

- We will create two parameters at pipeline level – sourceRelativeURL and sinkFileName.
- We will create the same parameters at dataset level as well and when we use this dataset in any activity there we have to pass the value for that and here we will pass the pipeline level parameters.
- Now whenever we create a trigger we will pass the value for sourceRelativeURL and sinkFileName.
- These values will be passed to the dataset used in copy data activity like sourceRelativeURL in source side of copy data activity and sinkFileName to sink side.
- But this approach also needed manual efforts to enter the URL and File Name for every file individually.
- To automate everything,

  We can create a JSON:

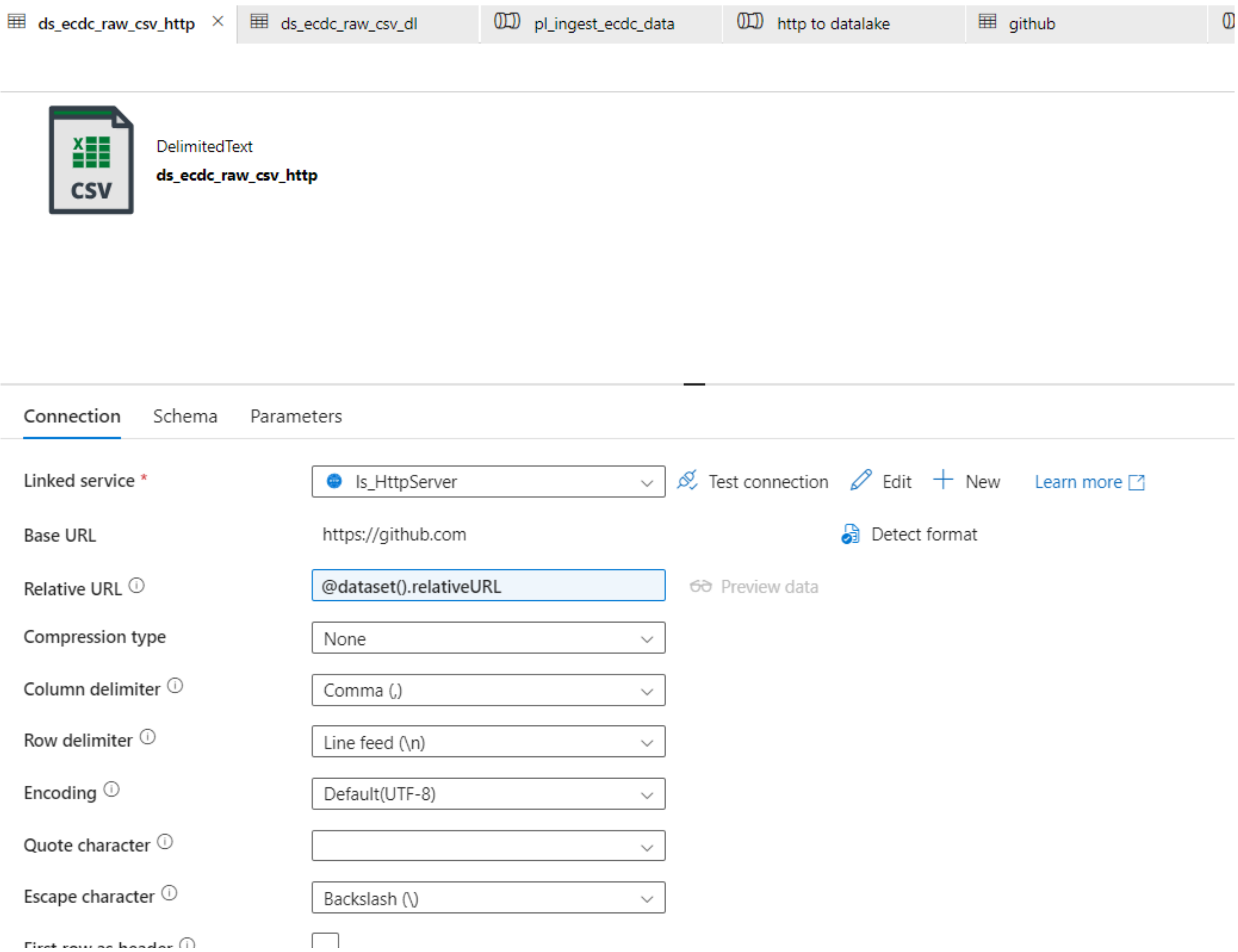
```
[
{
"SourceRelativeURL":"/loveleenverma/covid19/raw/main/ecdc_data/cases_deaths.csv",
"SinkFileName":"cases_deaths.csv"
},
{
"SourceRelativeURL":"/loveleenverma/covid19/raw/main/ecdc_data/hospital_admissions.csv",
"SinkFileName":"hospital_admissions.csv"
},
{
"SourceRelativeURL":"/loveleenverma/covid19/raw/main/ecdc_data/testing.csv",
"SinkFileName":"testing.csv"
},
{
"SourceRelativeURL":"/loveleenverma/covid19/raw/main/ecdc_data/country_response.csv",
"SinkFileName":"country_response.csv"
}
]
```

- Now First we will create two datasets for source (HTTP named ds_ecdc_raw_csv_http) and sink (Data Lake named ds_ecdc_raw_csv_dl). In these datasets we will create two parameters, SourceRelativeURL and SinkFileName. Please refer to below screenshot:



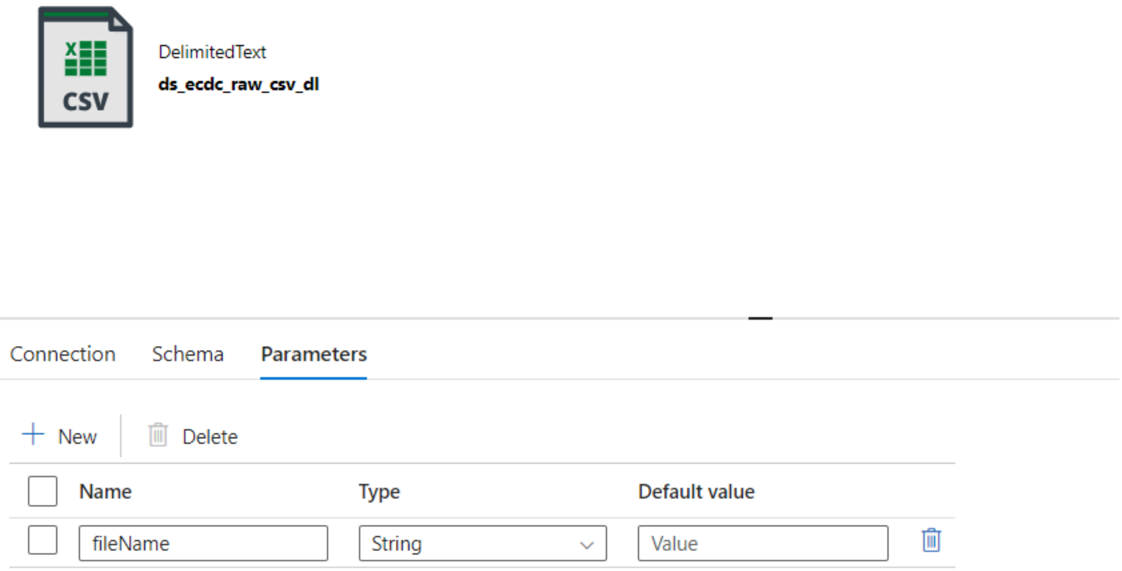
DelimitedText
ds_ecdc_raw_csv_http

□	Name	Type	Default value
□	relativeURL	String	Value



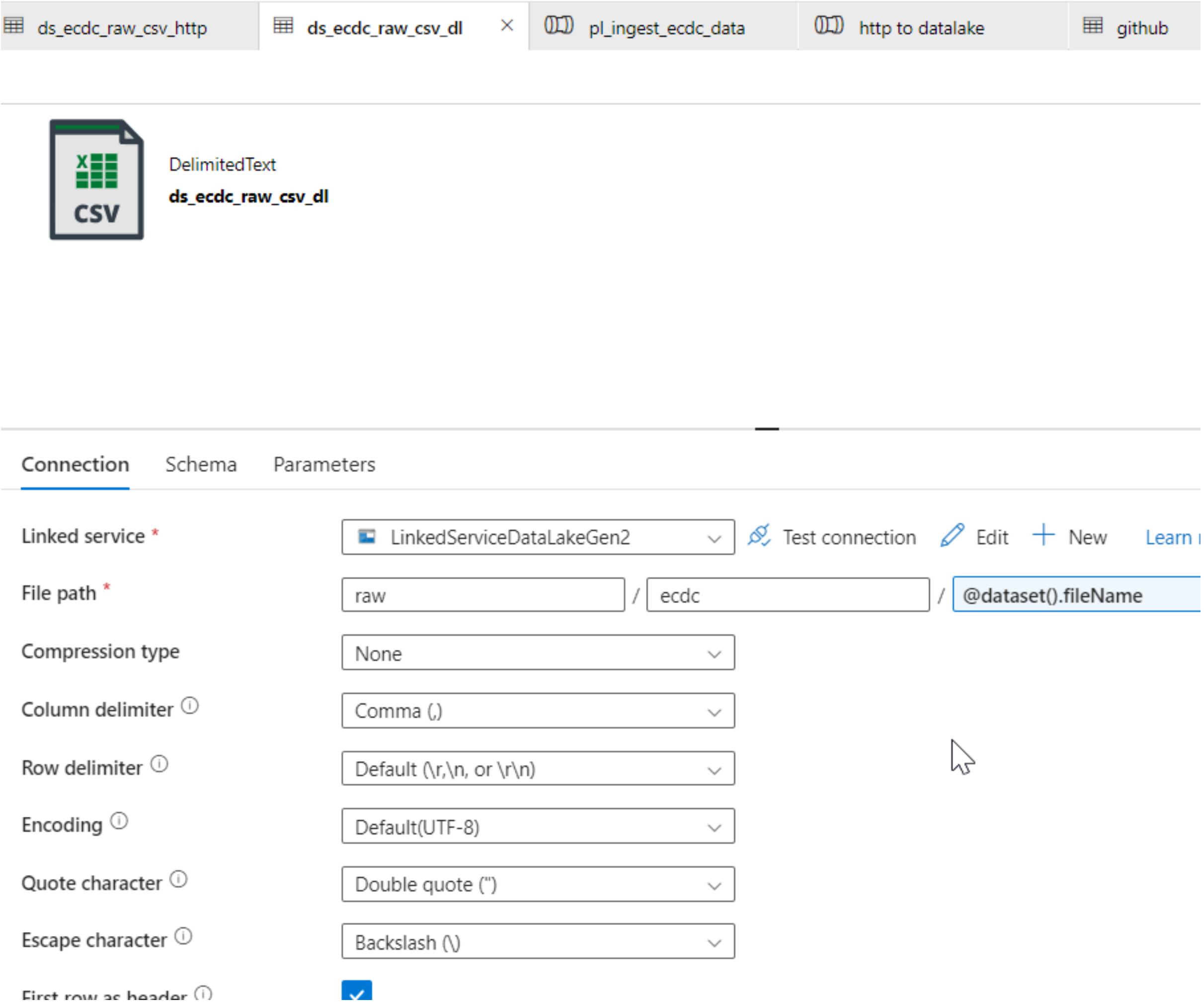
DelimitedText
ds_ecdc_raw_csv_http

Linked service *	ls_HttpServer
Base URL	https://github.com
Relative URL ⓘ	@dataset().relativeURL
Compression type	None
Column delimiter ⓘ	Comma (,)
Row delimiter ⓘ	Line feed (\n)
Encoding ⓘ	Default(UTF-8)
Quote character ⓘ	
Escape character ⓘ	Backslash (\)
First row as header ⓘ	□



DelimitedText
ds_ecdc_raw_csv_dl

□	Name	Type	Default value
□	fileName	String	Value



DelimitedText

ds_ecdc_raw_csv_dl

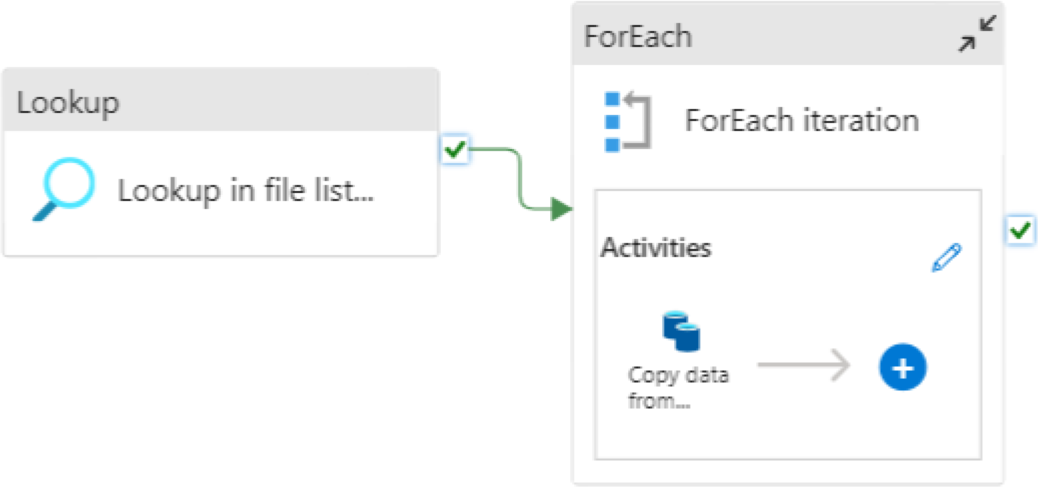
Linked service *	LinkedServiceDataLakeGen2
File path *	raw / ecdc / @dataset().fileName
Compression type	None
Column delimiter	Comma (,)
Row delimiter	Default (\r,\n, or \r\n)
Encoding	Default(UTF-8)
Quote character	Double quote (")
Escape character	Backslash (\)
First row as header	☑

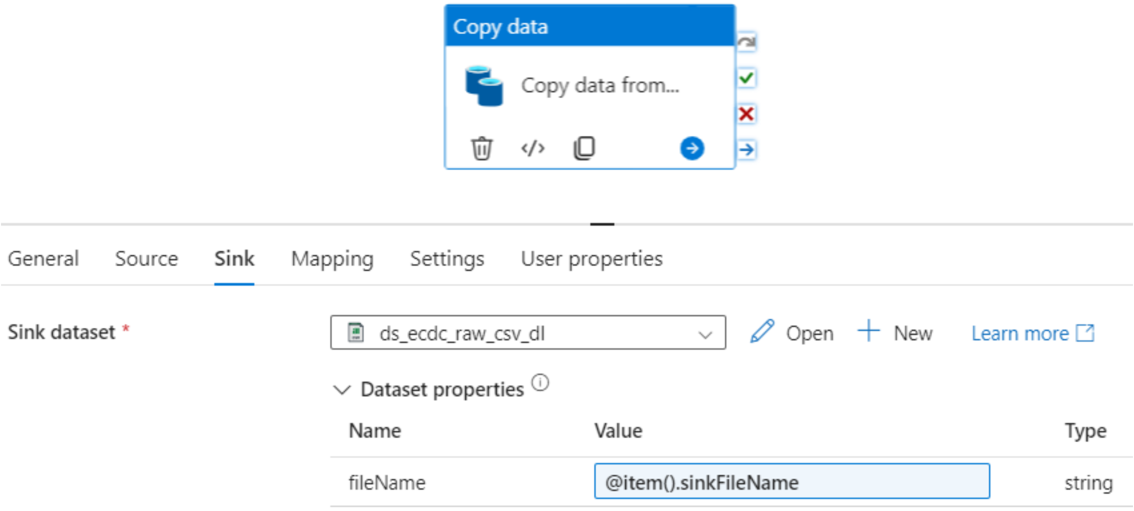
- Now we will create one pipeline named pl_ingest_ecdc_data.

  In this pipeline we will create one lookup activity, In that we will use the Azure Blob Storage dataset (can be created easily like above two only). This lookup activity will give following output:

```
{
        "count": 4,
        "value": [
                {
                        "SourceRelativeURL":
"/loveleenverma/covid19/raw/main/ecdc_data/cases_deaths.csv",
            "SinkFileName": "cases_deaths.csv"
        },
        {
            "SourceRelativeURL":
"/loveleenverma/covid19/raw/main/ecdc_data/hospital_admissions.csv",
            "SinkFileName": "hospital_admissions.csv"
        },
        {
            "SourceRelativeURL":
"/loveleenverma/covid19/raw/main/ecdc_data/testing.csv",
            "SinkFileName": "testing.csv"
        },
        {
            "SourceRelativeURL":
"/loveleenverma/covid19/raw/main/ecdc_data/country_response.csv",
            "SinkFileName": "country_response.csv"
        }
    ],
    "effectiveIntegrationRuntime": "AutoResolveIntegrationRuntime
(Central India)",
    "billingReference": {
        "activityType": "PipelineActivity",
        "billableDuration": [
            {
                "meterType": "AzureIR",
                "duration": 0.016666666666666666,
                "unit": "DIUHours"
            }
        ]
    },
    "durationInQueue": {
        "integrationRuntimeQueue": 0
    }
}
```

- Now we will use for each activity, This, will have iteration items as the values.
- So we have to pass the following as expression there: `@activity('Lookup in file list for data').output.value`
- Now for each iteration we will execute the copy activity where the source and sink datasets will require the value.
- Now we will pass these iteration values to them respectively. Please refer to below screenshots:

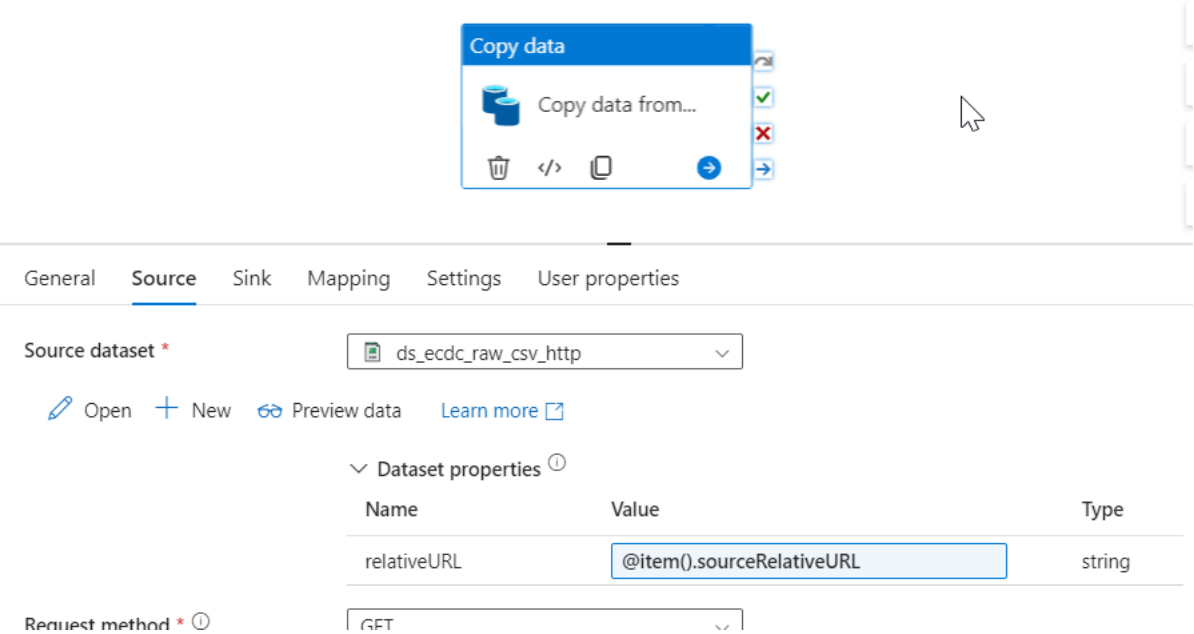




Sink dataset * ds_ecdc_raw_csv_dl

Dataset properties

Name	Value	Type
fileName	@item().sinkFileName	string



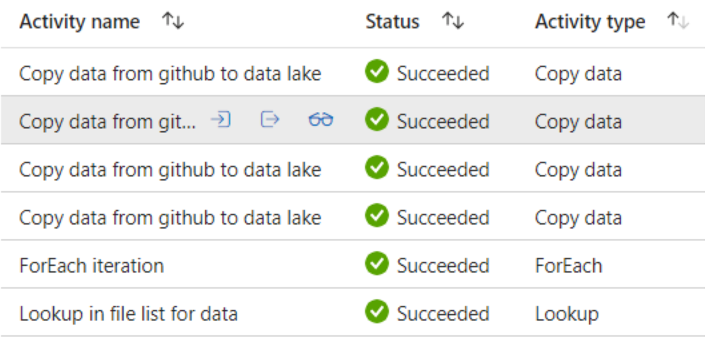
Source dataset * ds_ecdc_raw_csv_http

Dataset properties

Name	Value	Type
relativeURL	@item().sourceRelativeURL	string

Request method * GET

After executing the data will gets ingested and we are done.



Activity name ↑↓	Status ↑↓	Activity type ↑↓
Copy data from github to data lake	✓ Succeeded	Copy data
Copy data from git...	✓ Succeeded	Copy data
Copy data from github to data lake	✓ Succeeded	Copy data
Copy data from github to data lake	✓ Succeeded	Copy data
ForEach iteration	✓ Succeeded	ForEach
Lookup in file list for data	✓ Succeeded	Lookup

NOTE:

- Here the links that we are using might return the html page.
- To avoid that you must use raw in place of blob in the relative URL, Then you will get the access to files.

/loveleenverma/covid19/raw/main/ecdc_data/cases_deaths.csv

/loveleenverma/covid19/raw/main/ecdc_data/hospital_admissions.csv

/loveleenverma/covid19/raw/main/ecdc_data/testing.csv

/loveleenverma/covid19/raw/main/ecdc_data/country_response.csv

- If this also not worked use https://raw.githubusercontent.com as base URL while creating the linked service. Now you can just remove the blob from URL. No need to use raw also.

/loveleenverma/covid19/main/ecdc_data/cases_deaths.csv

/loveleenverma/covid19/main/ecdc_data/hospital_admissions.csv

/loveleenverma/covid19/main/ecdc_data/testing.csv

/loveleenverma/covid19/main/ecdc_data/country_response.csv