Azure Data Factory

ADF is a fully managed, serverless data integration solution for ingesting, Preparing and transforming all your data at scale.

**The data problem: -**

Traditional data integration solutions struggle to handle structured, semi structured and unstructured data due to lack of connectors.

MS added ADF tool

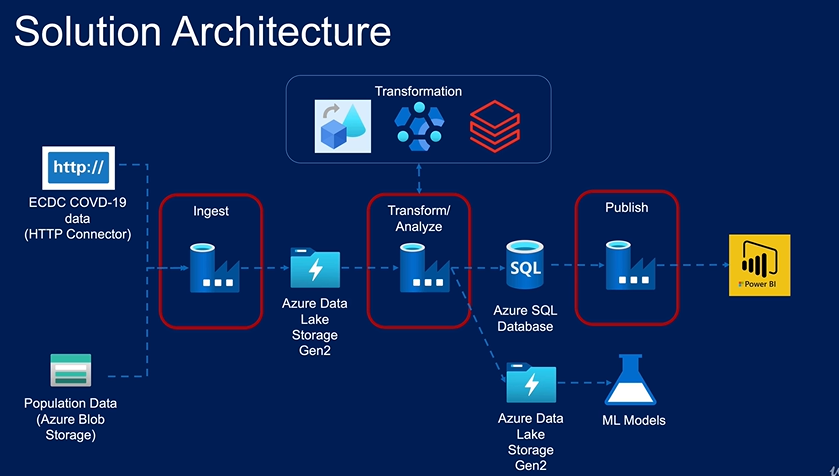
The ADF tool connects 90+ sources for ingesting the data, we can transform data and create reporting models for visualization.

1. Fully managed service
2. Serverless infrastructure
3. Data Integration service
4. Data transformation service
5. Data Orchestration service

**What azure data factory NOT**

1. ADF is not a Data Migration tool
2. ADF is not a data streaming service.
3. ADF not suitable for complex transformations (you can use Databricks)
4. ADF does not store any data (Not a Data storage service)

**Project Architecture Diagram: -**



Data Transformation Tool: - We can use Dataflows, HDInsight and Databricks

Data Storage Tool: - Azure Blob storage, Azure data lake storage Gen2 and Azure SQL DB

Reproting Tool: - Power BI

Azure Storage Account

1. Blob Storage
2. File storage
3. Disk Storage
4. Table Storage
5. Queue storage

Azure Data Lake Gen 2: -

It is designed to build an enterprise solution on Azure.

1. Enhance Performance
2. Better security
3. Enhance Management

Azure Cosmos DB: -

Cosmos DB is a fully managed NoSQL database for modern app development. It supports the number of API for accessing data such as SQL API, Cassandra, Mongo DB, Gramline API.

It’s used for web mobile gaming or IOT applications.

1. Globally Distributed
2. Multi Model.
3. High Throughput

**Storage solutions used for this project**

Azure SQL Database

Azure Blob storage

Azure data lake storage Gen 2

***Useful Links & Resources***

*Section: Overviews*

*Lecture: Project Overview*

*ECDC Website for Covid-19 Data -*[*https://www.ecdc.europa.eu/en/covid-19/data*](https://www.ecdc.europa.eu/en/covid-19/data)

*Euro Stat Website for Population Data -*[*https://ec.europa.eu/eurostat/estat-navtree-portlet-prod/BulkDownloadListing?file=data/tps00010.tsv.gz*](https://ec.europa.eu/eurostat/estat-navtree-portlet-prod/BulkDownloadListing?file=data/tps00010.tsv.gz)

*Lecture: Azure Storage Solutions*

*Introduction to Azure Storage services -*[*https://docs.microsoft.com/en-us/azure/storage/common/storage-introduction*](https://docs.microsoft.com/en-us/azure/storage/common/storage-introduction)

*Azure SQL Database -*[*https://docs.microsoft.com/en-us/azure/azure-sql/database/sql-database-paas-overview*](https://docs.microsoft.com/en-us/azure/azure-sql/database/sql-database-paas-overview)

*Azure Synapse Analytics -*[*https://docs.microsoft.com/en-us/azure/synapse-analytics/overview-what-is*](https://docs.microsoft.com/en-us/azure/synapse-analytics/overview-what-is)

*Azure Cosmos DB -*[*https://docs.microsoft.com/en-us/azure/cosmos-db/introduction*](https://docs.microsoft.com/en-us/azure/cosmos-db/introduction)

*Azure Data Lake Storage Gen2 -*[*https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-introduction*](https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-introduction)

Environment Setup: -

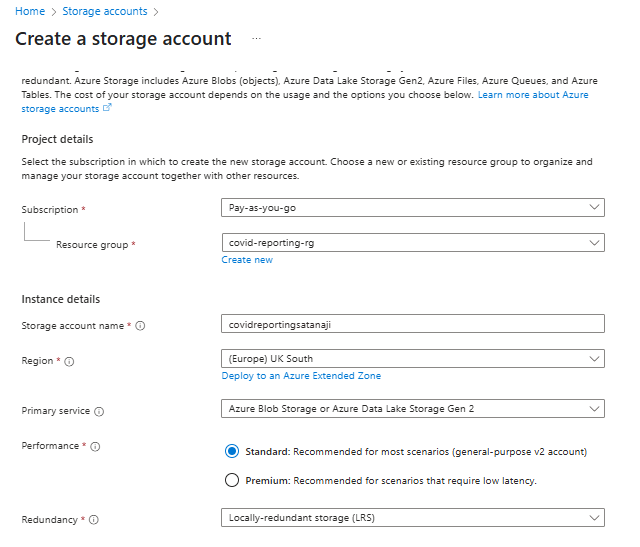
1. Azure subscription
2. Data Factory
3. Blob storage Account
4. Azure data Lake Gen 2
5. Azure SQL Database
6. Azure Databricks cluster
7. HD insight cluster

Note: - During setting up ADF account you can pin those resources in specific Dashboard to see specific user resources.

**Create Resources: -**

1. ADF: - you can create as it is for this

Storage account (blob storage): - for storage account you can choose below details and rest of the details as it is for this



1. Create Blob storage, ADLS Gen 2 account in same resource group
2. Azure SQL server account

In Azure SQL configuration we need to check: -

compute + storage

workload environment: - Development

Backup storage redundancy (LRS)

Connectivity Method (Public)

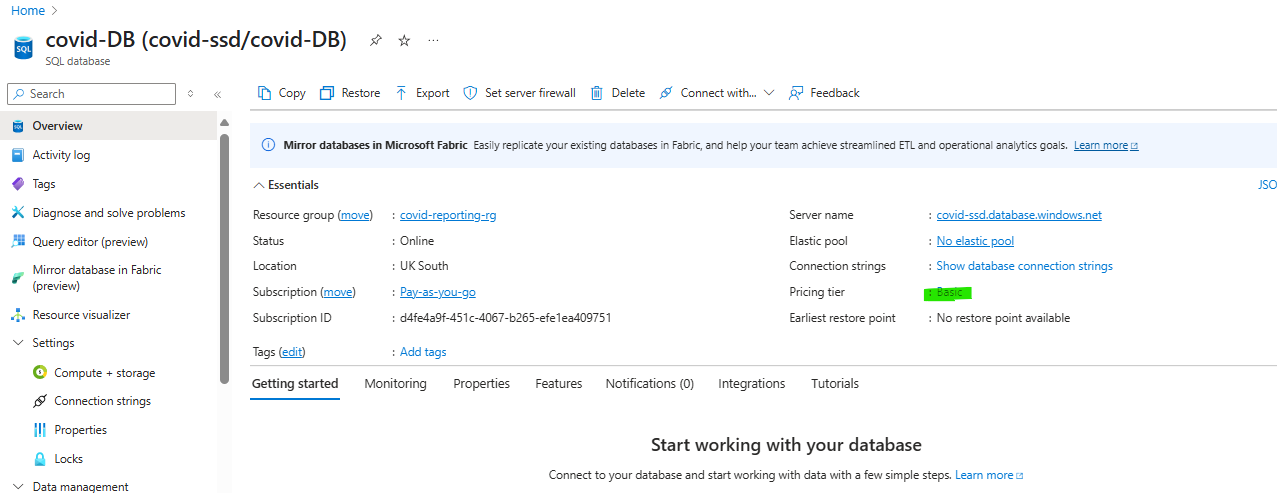
Allow Azure services and resources to access this server (Yes)

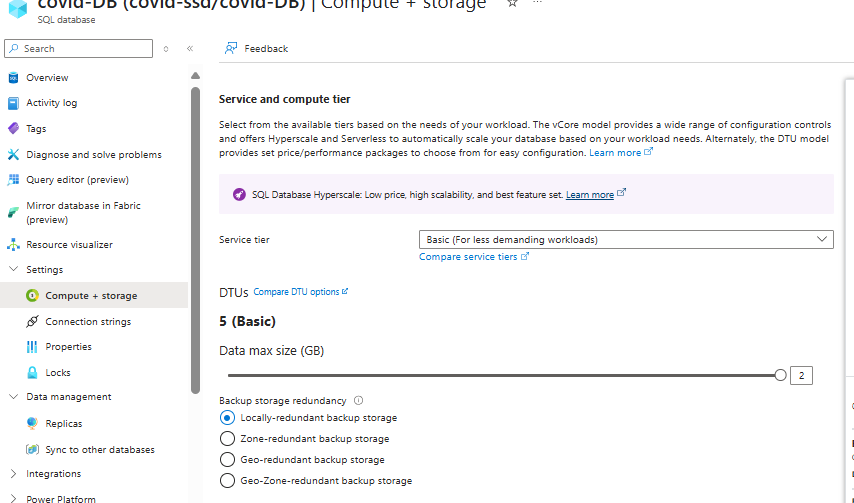
Add current client IP address (Yes)

DTU- Database Transaction unit for data read and write

Once the database is created, we can change pricing tier **basic** to another

Click on Basic 🡪 Service Tier / Backup redundancy





**Important Note: -**

* If need an external azure tool for Blob account install Azure storage explorer
* If you need an external azure tool for database, then install Azure Data Studio

**Model 1: -**

**Task1:-**

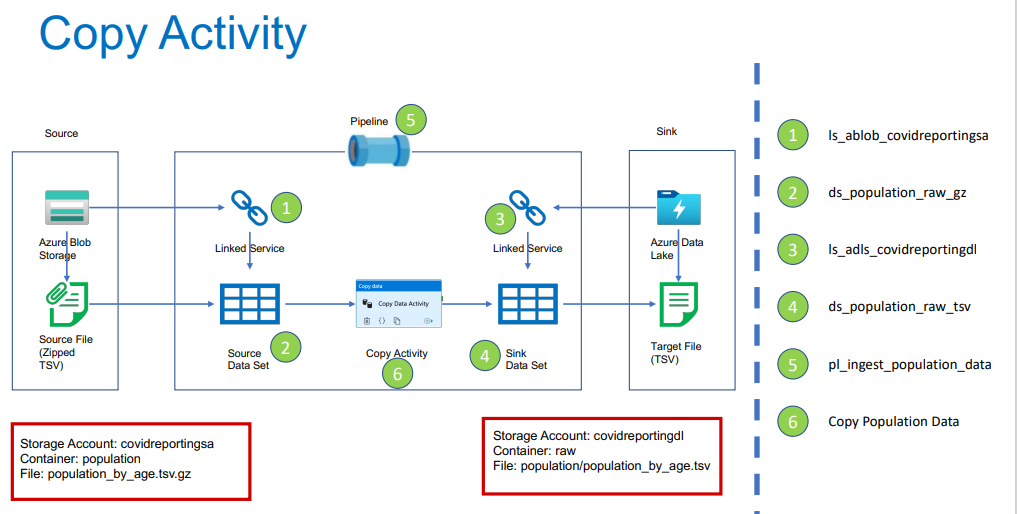
1. Ingesting Data from Blob storage to ADLS gen2 using copy activity

A diagram of a computer

AI-generated content may be incorrect.

1. Create Container name is **population**, and upload tsv.gz file in in this blob container.
2. Create container in ADLS gen 2 name is raw

Use the naming convention below for all the details



Create 2 Linked service as per the above diagram naming convention point 1 and 3

Create Dataset for source but our file in .gzip(.gz) format we need to select below option in dataset

A screenshot of a computer

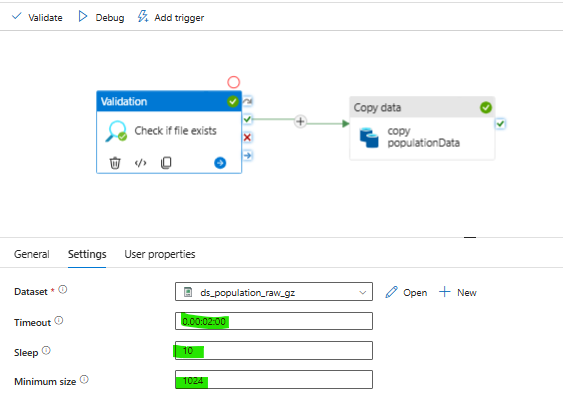
AI-generated content may be incorrect.

Create pipeline with proper source and sink as per the above created.

**Task2: - Control flow 1**

**Validation Activity: -** Execute copy activity when the file becomes available

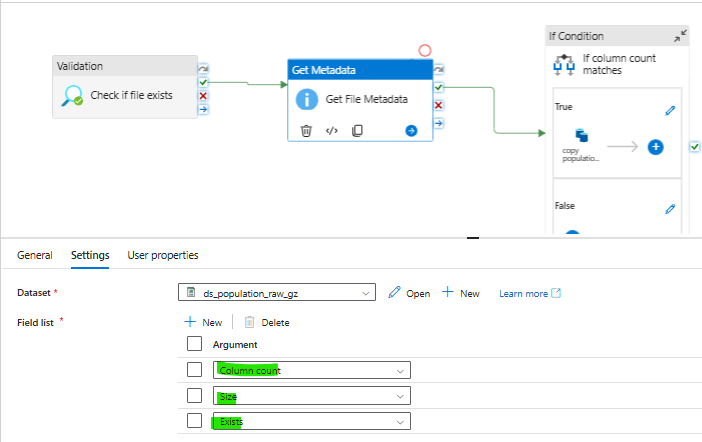
Validation activity keeps looking at the files in storage location once file will available validation activity execute successfully otherwise timeout as per our configuration



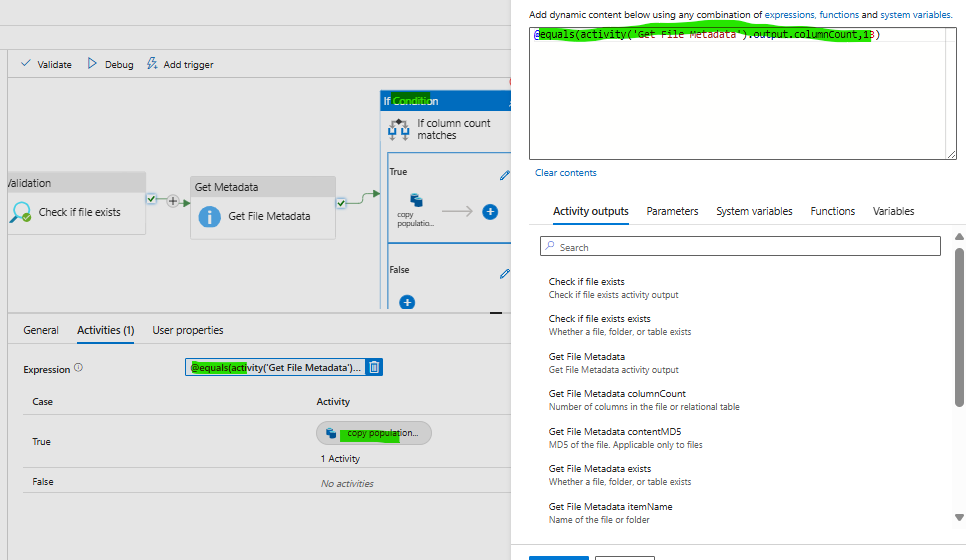
**Control flow 2: -**

**Get Metadata: -**We need to check if the source file column count is 13 then our copy activity is executed otherwise does not execute.

In metadata activity we need to add arguments such as (column count, size, exists)



**IF condition: -** add if condition and check metadata activity column count is 13 then execute True condition i.e. copy activity otherwise execute False condition



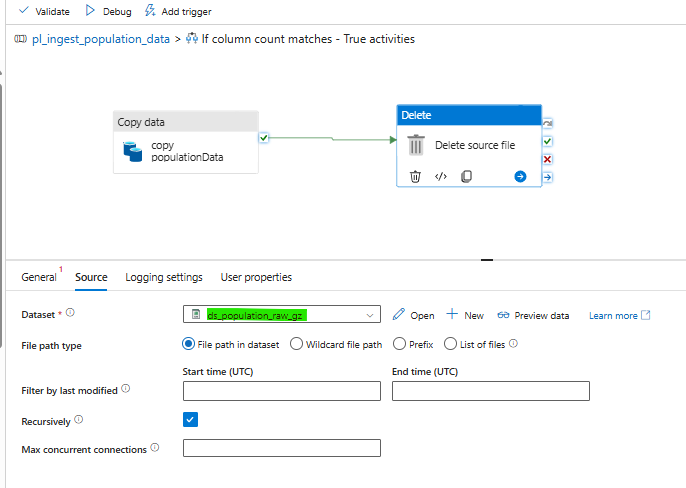
True condition: - add copy activity

False condition: - add web activity use dummy URL and add dummy body

**Copy Activity: -** add copy activity and add source and sink in this

**Delete Activity: -** once file is copied then delete source file from storage account (Blob storage)

Add Delete activity in if condition after Copy activity.



**Triggers: -**

1. Schedule trigger: -

* It’s a run of calendar / clock
* Supports periodic and specific times
* Triggers to pipeline is many to many relationship (we can attach all our ingestion pipelines in one trigger OR you can invoke same pipeline more than one trigger
* Can only be scheduled for future time to start

1. Tumbling window trigger: -

* Runs at periodic intervals.
* Windows are fixed sized, non-overlapping
* Can be scheduled for the past window / slices
* Trigger to pipeline one to one

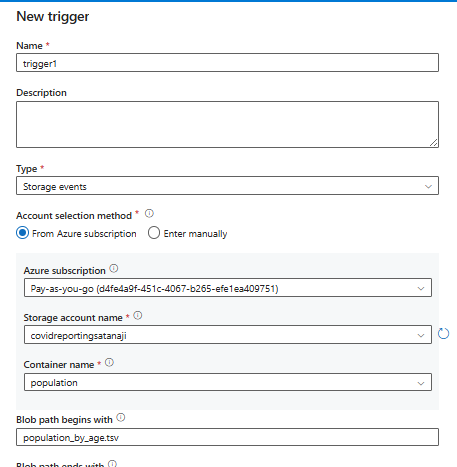
1. Event trigger: -

* Runs in response to events
* Events can be creation or deletion of Blobs/ Files
* Trigger to Pipeline many to many

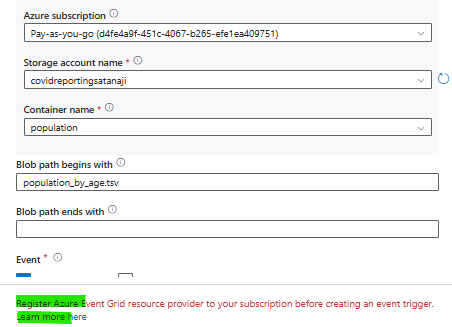
Note: - Event trigger has now been renamed to **storage events trigger**

Also, Microsoft has introduced a new trigger called Custom events trigger (This is in preview)

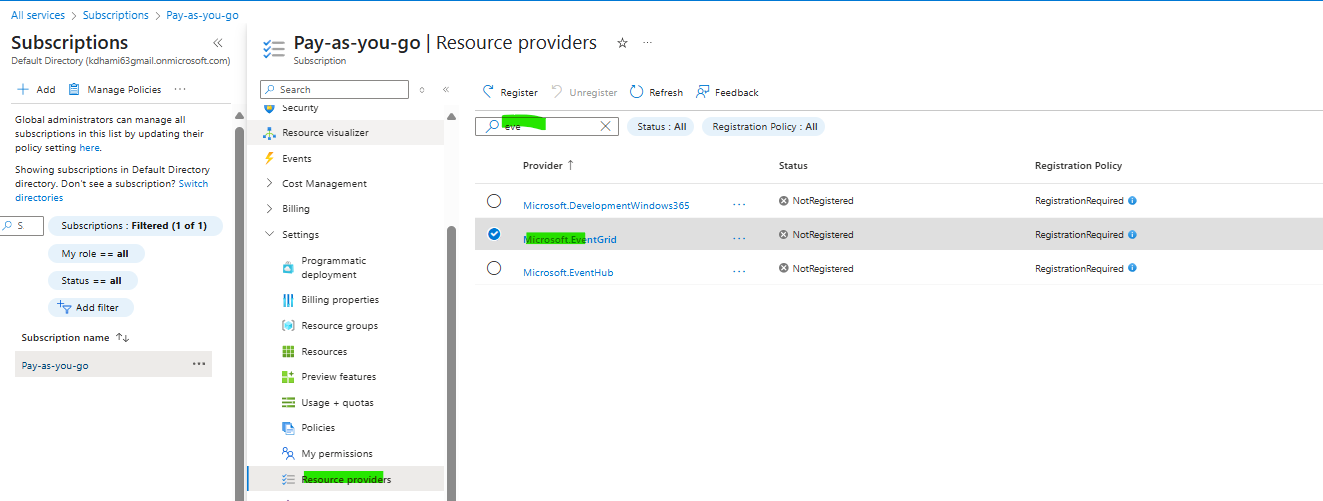
We need to create Event based trigger as per the below steps



Once you click ok, we got one error



Go to subscription 🡪 Resource Provider 🡪 Search event Grid 🡪 Click on radio button 🡪 Register it



Rename the name of trigger and attached it pipeline and publish the changes. If you add a file in source blob container trigger automatically execute and take file from source and put into destination server.

**Model 2 (ECDC data): -** Data ingestion from HTTP connector load to ADLS Gen 2

**EDCD website: -** https://www.ecdc.europa.eu/en/covid-19/data

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a test

AI-generated content may be incorrect.

You can open each link and copy csv file link to use the connection for ADF

**Task 1: -** Copy data for death cases from URL and load CSV file in ADLS Gen 2

Note: - We have a destination link service for ADLS gen 2 account so no need to create for this

A diagram of a data flow

AI-generated content may be incorrect.

1. Create Link service for HTTP connector for source data

Base URL (Linked Service): - https://opendata.ecdc.europa.eu

RelativeUrl (DATASET): - covid19/nationalcasedeath/csv

A screenshot of a computer

AI-generated content may be incorrect.

1. Create Datasets for source and Destination

Source: -

A screenshot of a computer

AI-generated content may be incorrect.

Destination: -

We need to load the data into ECDC folder in ADLS gen 2 account.

A screenshot of a computer

AI-generated content may be incorrect.

1. Create Pipeline

Create Pipeline and add copy activity into it and provide source and sink details and debug it.

A screenshot of a computer

AI-generated content may be incorrect.

**Imp Note**: - Above task is the single file copy data from HTTP and load into ADLS Gen 2, but on website data not updated properly so that we can use GITHUB connector for fetching the data.

**Example: -** for GITHUB connector we need to change only below 2 URLs in above Pipeline

Base URL (Linked Service): - <https://raw.githubusercontent.com>

RelativeUrl (DATASET): - cloudboxacademy/covid19/main/ecdc\_data/cases\_deaths.csv

**All URL’s: -**

**GITHUB: -**

**BaseUrl** - https://raw.githubusercontent.com

**RelativeUrl** for each of the files will be

cloudboxacademy/covid19/main/ecdc\_data/cases\_deaths.csv

cloudboxacademy/covid19/main/ecdc\_data/hospital\_admissions.csv

cloudboxacademy/covid19/main/ecdc\_data/testing.csv

cloudboxacademy/covid19/main/ecdc\_data/country\_response.csv

**Private Links by Ramesh: -**

**BaseUrl** - https://ecdc.blob.core.windows.net

**RelativeUrl**for each of the files will be

covid19/cases\_deaths.csv

covid19/country\_response.csv

covid19/hospital\_admissions.csv

covid19/testing.csv

covid19/case\_deaths\_uk\_ind\_only.csv

**Task 2: - (static copy the data from HTTP to ADLS)**

Using Parameter and Variable we can copy multiple files using HTTP connections and load into ADLS Gen 2

We can create parameters on dataset and pass this parameter within the dataset.

Create variables on pipelines and pass the required values and map this on pipeline source and sink copy activity.

Flow: -

Dataset for source

A screenshot of a computer

AI-generated content may be incorrect.

Dataset for sink

A screenshot of a computer program

AI-generated content may be incorrect.

Pipeline

A screenshot of a computer

AI-generated content may be incorrect.

Copy activity

Souce:-

A screenshot of a computer

AI-generated content may be incorrect.

Sink: -

A screenshot of a computer

AI-generated content may be incorrect.

So now delete the variable on pipeline level and create same name parameters on pipeline level. And debug the pipeline and the window pop up to provide values for pipeline level parameter.

Then create a schedule trigger for this pipeline and pass those parameter values on trigger level.

So, we have 4 http source files and at that time we need to create 4 triggers to fetch the data but it’s not suitable.

Suppose we have 100 files then this task is not possible overcome this problem please check below task

**Task 3: - (Dynamically copy the data from HTTP to ADLS using JSON)**

Create JSON file with details as below

[

{

"sourceRelativeURL":"cloudboxacademy/covid19/raw/main/ecdc\_data/cases\_deaths.csv",

"sinkFileName":"cases\_deaths.csv"

},

{

"sourceRelativeURL":"cloudboxacademy/covid19/raw/main/ecdc\_data/hospital\_admissions.csv",

"sinkFileName":"hospital\_admissions.csv"

}

]

Save the above file in json format and upload JSON file in blob storage in **config** folder.

A screenshot of a computer

AI-generated content may be incorrect.

Dataset: - create dataset for Json connector use linked service for Blob storage.

A screenshot of a computer

AI-generated content may be incorrect.

Create one test pipeline for capturing all the details from JSON file

1. Lookup activity: - connect above dataset and disable First Row Only

A screenshot of a computer

AI-generated content may be incorrect.

1. Foreach Activity: - Need value coming from lookup activity

A screenshot of a computer

AI-generated content may be incorrect.

1. Set Variable Activity: - Create one variable on pipeline level

A screenshot of a computer

AI-generated content may be incorrect.

Pass this variable name and in value provide output value from lookup its

Item().sourceRelativeURL

A screenshot of a computer

AI-generated content may be incorrect.

**Task4(Final) -** Copy all 4 files dynamically, using JSON, HTTP to ADLS Gen 2

1. Create JSON file as per the below code and save as a JSON and load file in to blob storage in configs folder.

[

{

"sourceBaseURL":"https://github.com",

"sourceRelativeURL":"cloudboxacademy/covid19/raw/main/ecdc\_data/cases\_deaths.csv",

"sinkFileName":"cases\_deaths.csv"

},

{

"sourceBaseURL":"https://github.com",

"sourceRelativeURL":"cloudboxacademy/covid19/raw/main/ecdc\_data/hospital\_admissions.csv",

"sinkFileName":"hospital\_admissions.csv"

},

{

"sourceBaseURL":"https://github.com",

"sourceRelativeURL":"cloudboxacademy/covid19/raw/main/ecdc\_data/testing.csv",

"sinkFileName":"testing.csv"

},

{

"sourceBaseURL":"https://github.com",

"sourceRelativeURL":"cloudboxacademy/covid19/raw/main/ecdc\_data/country\_response.csv",

"sinkFileName":"country\_response.csv"

}

]

1. Linked service: - create parameter and pass this as Base URL.

**A screenshot of a computer

AI-generated content may be incorrect.**

1. Dataset: -
   1. Change the existing dataset file in file path for JSON connector which is used for lookup activity.

A screenshot of a computer

AI-generated content may be incorrect.

* 1. Source dataset for copy activity: -

Create 2 parameters for **relativeurl** and **baseURL**

A screenshot of a computer

AI-generated content may be incorrect.

* 1. Sink dataset for Copy activity: -

Create parameter name is **filename**

A screenshot of a computer

AI-generated content may be incorrect.

1. Lookup activity: - connect the above data set to lookup activity (as per the image below)

**A screenshot of a computer

AI-generated content may be incorrect.**

1. Foreach activity: -

A screenshot of a computer

AI-generated content may be incorrect.

1. Copy Activity: -
   1. Source: -

A screenshot of a computer

AI-generated content may be incorrect.

* 1. Sink: -

A screenshot of a computer

AI-generated content may be incorrect.

Debug the pipeline and all 4 files are copied from GITHUB / website and load into storage account ADLS Gen 2

1. Trigger (Scheduled trigger)

Create Scheduled trigger and attached pipeline for this

A screenshot of a computer

AI-generated content may be incorrect.

**Model 3: -** (**Cases and Deaths data** transformation using dataflows)

Dataflows: - Please find the transformation diagram below for this dataset

Transformations: - (Cases and Death Data)

1. Filter all the data only for Europe
2. Rename the name country\_Code to country\_code\_3\_digit
3. Create cases\_count and deaths\_count using indicator and daily\_count column.
4. Rename date column to reported\_date.
5. Remove continent and rate\_14\_day columns in data

Filter only Europe data

Raw file from ECDC Transformed File

|  |
| --- |
| **Columns in raw file (ECDC)** |
| country |
| country\_code |
| ***~~continent~~*** |
| population |
| indicator |
| daily\_count |
| date |
| ***~~rate\_14\_day~~*** |
| source |

|  |
| --- |
| **Transformed Columns in file** |
| country |
| country\_code\_2\_digit**(Lookup**) |
| country\_code\_3\_digit |
| population |
| cases\_count |
| deaths\_count |
| reported\_date **(Rename**) |
| source |

We have all country lookup file upload it on ADLS gen 2 🡪 under lookup container 🡪 for fetch 2 and 3-digit country codes.

A screenshot of a computer

AI-generated content may be incorrect.

Also, we need only UK and India country data for testing so filter the **case\_deaths.csv** file and only keep this 2-country data and load into **ADLS Gen 2🡪 ECDC 🡪 case\_deaths\_uk\_ind\_only.csv** because of we need to pass this file as a sample file during dataflow debug no need to debug whole file during debug.

A screenshot of a computer

AI-generated content may be incorrect.

Create Dataflow: -

Add source in this dataflow 🡪 Source setting add dataset for this 🡪 go to projection and change data type **string** to **integer** for daily\_count column

A screenshot of a computer

AI-generated content may be incorrect.

Filter Transformation: - add filter on “**continent=="Europe" && not(isNull(country\_code))**”

Select Transformation: - select required columns please find the below image.

A screenshot of a computer

AI-generated content may be incorrect.

Pivot transformation: - use pivot on below column with pivot key and grouping condition.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Lookup Transformation: - we have one country lookup file we need to use this file with pivot output details and fetch all the columns.

A screenshot of a computer

AI-generated content may be incorrect.

Select Transformation: - select required column for output.

A screenshot of a computer

AI-generated content may be incorrect.

Sink Transformation: - Load these details in ADLS gen 2

A screenshot of a computer

AI-generated content may be incorrect.

Create pipeline and call above dataflow in pipeline and trigger now this pipeline

We need a set compute (4 worker node and 4 diver node (basic))

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Pipeline: - Create pipeline and add dataflow activity and call above dataflow in pipeline and trigger now this pipeline.

**Model 4: -** (**Hospital Admission** data transformation using dataflows)

**Transformed Daily File**

Raw file from ECDC

|  |
| --- |
| **Columns in raw file (ECDC)** |
| country |
| Indicator |
| date |
| Year\_week |
| Value |
| source |
| ***~~url~~*** |

|  |
| --- |
| **Column Name** |
| country |
| country\_code\_2\_digit**(Lookup**) |
| country\_code\_3\_digit**(Lookup**) |
| population **(Lookup**) |
| reported\_date |
| hospital\_occupancy\_count |
| icu\_occupancy\_count |
| source |

**Transformed Weekly File**

|  |
| --- |
| **Column Name** |
| country |
| country\_code\_2\_digit**(Lookup**) |
| country\_code\_3\_digit**(Lookup**) |
| population **(Lookup**) |
| reported\_year\_week**(transformed**) |
| reported\_week\_start\_date**(lookup**) |
| reported\_week\_end\_date **(lookup**) |
| new\_hospital\_occupancy\_count |
| new\_icu\_occupancy\_count |
| source |

|  |
| --- |
| **Columns in raw file (ECDC)** |
| country |
| Indicator |
| date |
| Year\_week |
| Value |
| source |
| ***~~url~~*** |

Create Dataflows: -

1. Source Transformation: - Add source and do the configuration below. Add dataset for this

A screenshot of a computer

AI-generated content may be incorrect.

Dataset:-

A screenshot of a computer

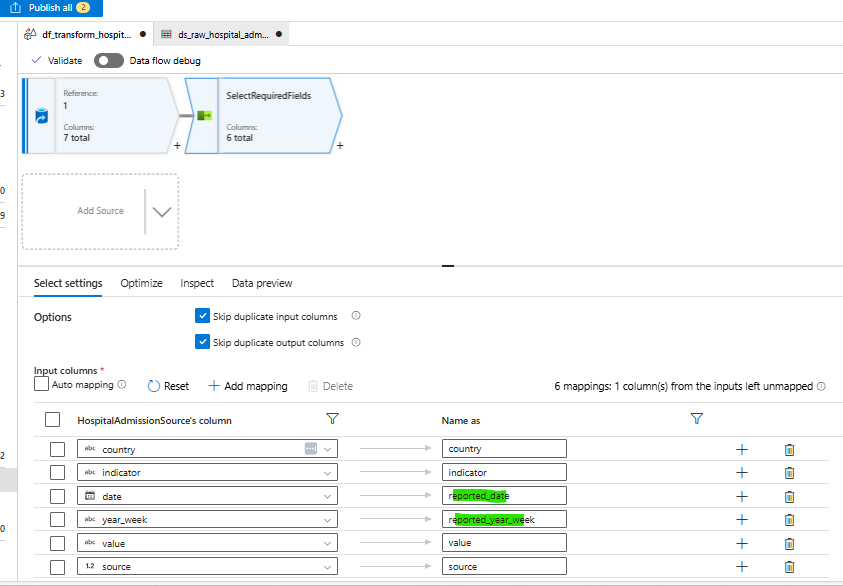
AI-generated content may be incorrect.

1. Select Transformation: - As per the requirements we need to change below points

Remove url column

Rename date column to reported\_date

Rename year\_week column to reported\_year\_week



1. Lookup Transformation: - In this transformation we need to create 2 and 3 digit country code using **country\_lookup.csv** file with country key word. This is like as join condition in sql .

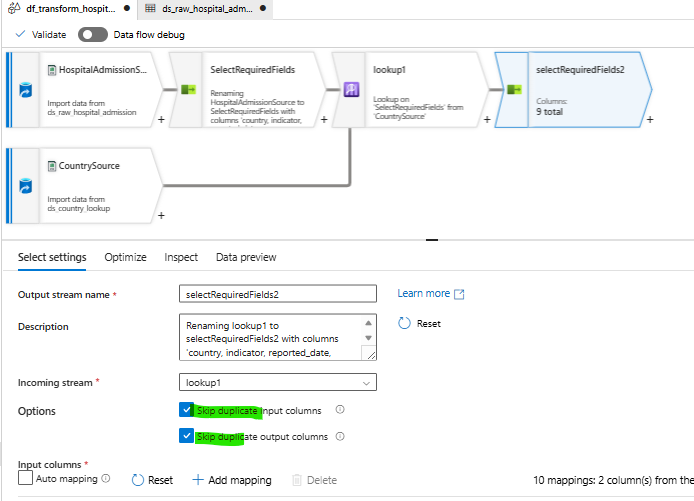
So now we have all the columns from hospital administrations and country lookup files we need to select only required columns using select transformation.

A screenshot of a computer

AI-generated content may be incorrect.

1. Select Transformation: - By using skip on duplicate input and output columns, duplicated columns are automatically ignored

We need to delete the continent column which is not required.



1. Conditional Split Transformation: - So, we need to split this file with 2 outputs because we need to segregate the Daily and weekly data using **Indicator** column.

A screenshot of a computer

AI-generated content may be incorrect.

1. Source Transformation: - we have one dim\_date.csv file we need to fetch weeks start and end date, year and weeks with use of dim\_date file.

Upload dim\_date file in ADLS gen 2 🡪lookup folder.

Create a new dataset for this dim\_date file in source Transformation.

A screenshot of a computer

AI-generated content may be incorrect.

1. Derived column Transformation: - By using derived column we need year and week in column, so we need to use below logic in expression

A screenshot of a computer

AI-generated content may be incorrect.

1. Aggregate Transformation: - Add this transformation and in Aggregate we can add week start and week end date and in Group option you can add below logic.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Now we have added the same logic as per group by in DerivedECDCYearWeek, so we can delete this Transformation.

1. Join Transformation: - Add Join Transformation for week split column Transformation and AggDimDate

A screenshot of a computer

AI-generated content may be incorrect.

1. Pivot Transformation: - add below group by fields in this Transformation.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Save pivot transformation you need to use in Daily conditional split transformation.

1. Sort Transformation: - Add sort transformation for both daily and week above transformation and add below fields in this.



A screenshot of a computer

AI-generated content may be incorrect.

1. Select transformation:- select required column and change the name as per the requirement

**Weekly:-**

A screenshot of a computer

AI-generated content may be incorrect.

**Daily:-**

A screenshot of a computer

AI-generated content may be incorrect.

1. Sink Transformation:- Add this files in ADLS gen 2 in below dataset location

**Weekly dataset:-**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Daily Dataset: -**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Pipeline: -** create pipeline and add dataflow activity and map the above dataflow into this. Trigger this pipeline.

A screenshot of a computer

AI-generated content may be incorrect.

So, in Processed container🡪 under ecdc folder created 2 folders and, in this folder, created files but we need only one file with specific names. So, we need to change something in sink and sort Transformation

Sink Transformation 🡪Clear folder click, File name option (Output to single file), optimize (single partitioning) for Daily and weekly sink

Sort Transfomarion🡪 Single Partitioning

Note:- Dataflows executes transformation in spark and spark works on distribution computing, similar like HDInsight

Pipeline: - **pl\_ingest\_ecdc\_data** 🡪 we have changed json file with adding the folder name as per the below image. and replace this json with original one and rerun the pipeline. So all the files are ingested into specific folder

A screenshot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

So, we need to change the datasets for fetching the data from specific directory we need to add the same like as below image for 4 datasets

A screenshot of a computer

AI-generated content may be incorrect.

Dataflows execute the transformation in spark

**Module- Azure HDInsight Service [Testing File]**

**Managed Identity: -** It is an access account only for data lake storage Gen2 via managed identity.

* HDInsight only allows access to datalake storage Gen 2 (ADLS Gen2) account via managed Identity.

Azure HDInsight is a managed cluster platform that makes it easy to run big data frameworks like Apache Spark, Apache Hive, LLAP, Apache Kafka, Apache Hadoop, and others in your Azure environment. It's designed to handle large volumes of data with high speed and efficiency.

**User Assigned Managed Identity: -** Create User Assigned Managed Identity service for accessing Azure HDInsight

Provide access for ADLS Gen 2 account to this user defined managed identity.

Go to ADLS Gen2 service 🡪 Go to Access Control (IAM) 🡪 Add 🡪 Add role assignment🡪 Storage blob owner 🡪search Managed identity Name and provide access 🡪then okay

**Azure HDInsight Service**: - Create Azure HDInsight service, during this creation got error subscription does not register this HDInsight service, then you need to go subscription and click on Resource provider and search HDInsight and register it and recreate Azure HDInsight service.

Provide all the details (login password etc..) and select primary and secondary ADLS storage

A screenshot of a computer

AI-generated content may be incorrect.

**Note: -** Region should be same for ADLS, HDInsight and Managed Identity.

Azure HDInsight cluster billing starts once you are created so you can create do your work and delete this as early otherwise billing will increase.

Create HDL queries to execute on Hive Ambari inbuilt tool in Azure HDInsight

Other tables available to connect JDBC connect (SQuirreL SQL and dbvisualizar)

We can use Ambari tool for checking the scripts and tables

Query:-

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Important \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Make sure you change the storage account names to your storage account name

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Important \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CREATE DATABASE IF NOT EXISTS covid\_reporting\_lookup;

CREATE DATABASE IF NOT EXISTS covid\_reporting\_raw;

CREATE DATABASE IF NOT EXISTS covid\_reporting\_processed;

DROP TABLE IF EXISTS covid\_reporting\_lookup.dim\_date;

CREATE EXTERNAL TABLE IF NOT EXISTS covid\_reporting\_lookup.dim\_date (

date\_key INT,

the\_date DATE ,

the\_year INT,

the\_month INT,

the\_day INT,

day\_name STRING,

day\_of\_year BIGINT,

week\_of\_month BIGINT,

week\_of\_year BIGINT,

month\_name STRING,

year\_month INT,

year\_week INT

)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION 'abfs://lookup@covidreportingdl.dfs.core.windows.net/dim\_date/'

TBLPROPERTIES ("skip.header.line.count"="1");

DROP TABLE IF EXISTS covid\_reporting\_lookup.dim\_country;

CREATE EXTERNAL TABLE IF NOT EXISTS covid\_reporting\_lookup.dim\_country (

country STRING,

country\_code\_2\_digit STRING ,

country\_code\_3\_digit STRING

)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION 'abfs://lookup@covidreportingdl.dfs.core.windows.net/dim\_country/'

TBLPROPERTIES ("skip.header.line.count"="1");

DROP TABLE IF EXISTS covid\_reporting\_raw.testing;

CREATE EXTERNAL TABLE IF NOT EXISTS covid\_reporting\_raw.testing (

country STRING,

country\_code STRING,

year\_week STRING,

new\_cases BIGINT,

tests\_done BIGINT,

population BIGINT,

testing\_rate DOUBLE,

positivity\_rate DOUBLE,

testing\_data\_source STRING

)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION 'abfs://raw@covidreportingdl.dfs.core.windows.net/ecdc/testing/'

TBLPROPERTIES ("skip.header.line.count"="1");

DROP TABLE IF EXISTS covid\_reporting\_processed.testing;

CREATE TABLE IF NOT EXISTS covid\_reporting\_processed.testing (

country STRING,

country\_code\_2\_digit STRING,

country\_code\_3\_digit STRING,

year\_week STRING,

week\_start\_date STRING,

week\_end\_date STRING,

new\_cases BIGINT,

tests\_done BIGINT,

population BIGINT,

testing\_rate DOUBLE,

positivity\_rate DOUBLE,

testing\_data\_source STRING

)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION 'abfs://processed@covidreportingdl.dfs.core.windows.net/ecdc/testing'

TBLPROPERTIES ("skip.header.line.count"="1");

INSERT OVERWRITE TABLE covid\_reporting\_processed.testing

SELECT t.country ,

c.country\_code\_2\_digit ,

c.country\_code\_3\_digit ,

t.year\_week ,

MIN(d.the\_date) AS week\_start\_date,

MAX(d.the\_date) AS week\_end\_date ,

t.new\_cases ,

t.tests\_done ,

t.population ,

t.testing\_rate ,

t.positivity\_rate ,

t.testing\_data\_source

FROM covid\_reporting\_raw.testing t

JOIN covid\_reporting\_lookup.dim\_date d

ON (t.year\_week = CONCAT(CONCAT(d.the\_year, '-W'), LPAD(d.week\_of\_year, 2, '0')))

JOIN covid\_reporting\_lookup.dim\_country c

ON (t.country\_code = c.country\_code\_2\_digit)

GROUP BY t.country ,

c.country\_code\_2\_digit ,

c.country\_code\_3\_digit ,

t.year\_week ,

t.new\_cases ,

t.tests\_done ,

t.population ,

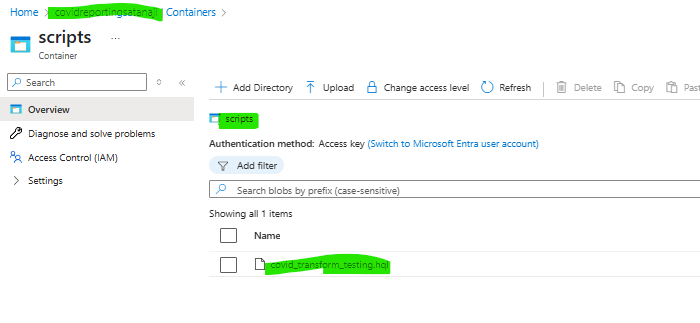
t.testing\_rate ,

t.positivity\_rate ,

t.testing\_data\_source ;

We can create External table for creating tables in specific folder using above hql query.

Save the above query in file (covid\_transform\_testing.hql) and upload it on below path new container as [scripts]



Go to ADF to create new pipeline (**pl\_process\_testing\_data**) and add HDInsight section and add Hive activity.

Add link service (**ls\_hdi\_covid\_cluster**) during creation of linked service provide all the details like hdi cluster, user of cluster and password.

Go to script option in hive activity and linked service for blob storage & select script path of blob storage and debug the pipeline

Go to ADLS gen 2, Processed container and testing folder created and into this folder file is created.

Go to Ambari Hive view and fire query

Select \* from covid\_reporting\_processed.testing LIMIT 10;

Once all the above steps have been completed then delete the Azure HDInsight

**Module: - Databricks Activity [Population File]**

Go to Azure portal and create service is Azure Databricks and launch this service with same login id single sign in.

What is the Cluster / Compute?

Cluster is a computation resource in databricks.

Types of clusters

* 1. **All-purpose / interactive cluster: -** it collaborates with other team member, its manually created and manually terminated and restart.
  2. **Job cluster: -** used to run a databricks job and automatically terminate once job is run.

A diagram of a cluster

AI-generated content may be incorrect.

A blue squares with white text

AI-generated content may be incorrect.

Create compute cluster in Azure databricks as per the below options.

A screenshot of a computer

AI-generated content may be incorrect.

* Mounting Azure Data Lake storage in Azure databricks

1. **Create Service principle: -** go to Microsoft entra 🡪 App regration and create the app.

create secret of this app and note down this information

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Application (client) ID :- cb90403c-bfcf-472d-b759-a5a0d2f0fbc7

Directory (tenant) ID :- dbfe462b-24d7-4bb6-8244-557b82a15734

Secret ID :- 629b0b86-1886-493c-a96f-89f9530168ac

Value :- EDu8Q~8AqI8Qn3v2oretOIcBO~JKzzsmeHQVzaDK

1. Granet accesses this service principle to ADLS gen2 account

Role:- storage blob contributor

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Mount storage account to Azure databricks :- [Create Notebook on databricks]

Notebook Name :- [mount\_storage.py]

**Mount 3 containers in ADLS gen 2 in Azure data bricks [*Raw, Processed and Lookup*]**

%md

### Mount the following data lake storage gen2 containers

1. raw

2. processed

3. lookup

%md

#### Set-up the configs

##### Please update the following

- application-id

- service-credential

- directory-id

configs = {"fs.azure.account.auth.type": "OAuth",

           "fs.azure.account.oauth.provider.type": "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider",

           "fs.azure.account.oauth2.client.id": "cb90403c-bfcf-472d-b759-a5a0d2f0fbc7",

           "fs.azure.account.oauth2.client.secret": "EDu8Q~8AqI8Qn3v2oretOIcBO~JKzzsmeHQVzaDK",

           "fs.azure.account.oauth2.client.endpoint": "https://login.microsoftonline.com/dbfe462b-24d7-4bb6-8244-557b82a15734/oauth2/token"}

%md

#### Mount the raw container

##### Update the storage account name before executing

dbutils.fs.mount(

  source = "abfss://raw@covidreportingdltanaji.dfs.core.windows.net/",

  mount\_point = "/mnt/covidreportingdltanaji/raw",

  extra\_configs = configs)

%md

#### Mount the processed container

##### Update the storage account name before executing

dbutils.fs.mount(

  source = "abfss://processed@covidreportingdltanaji.dfs.core.windows.net/",

  mount\_point = "/mnt/covidreportingdltanaji/processed",

  extra\_configs = configs)

%md

#### Mount the lookup container

##### Update the storage account name before executing

dbutils.fs.mount(

  source = "abfss://lookup@covidreportingdltanaji.dfs.core.windows.net/",

  mount\_point = "/mnt/covidreportingdltanaji/lookup",

  extra\_configs = configs)

Check the subfolder in tis container use the command below: -

dbutils.fs.ls("/mnt/covidreportingdltanaji/processed")

OR

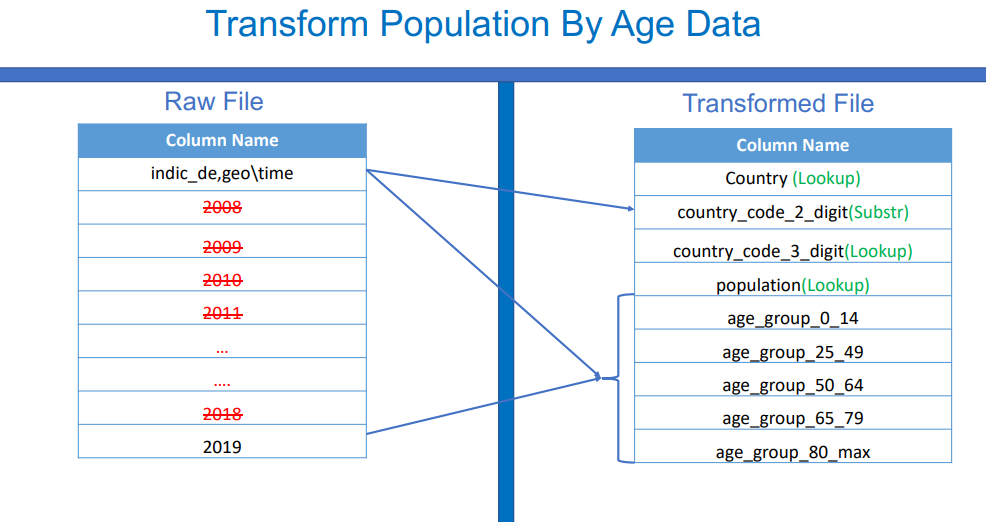
display(dbutils.fs.mounts())

If you need to unmount storage account then use below command

Example :- dbutils.fs.unmount("/mnt/covidreportingdltanaji/processed")

So, we do need this cluster because we are executing the notebooks in Azure Data Factory.

**Transform population by Age data using databricks:-**



As per the above image we need to do transformations in Raw file. [transform\_population\_data.py]

%md

#### Transform Population By Age data by performing the transformations below

######-----------------------------------------------------------------------

1. Split the country code & age group

2. Exclude all data other than 2019

3. Remove non numeric data from percentage

4. Pivot the data by age group

5. Join to dim\_country to get the country, 3 digit country code and the total population.

######-----------------------------------------------------------------------

%md

##### Replace \*\*storage account name\*\* with your storage account name before executing.

from pyspark.sql.functions import \*

%md

##### Read the population data & create a temp view

df\_raw\_population = df\_raw\_population.withColumn('age\_group', regexp\_replace(split(df\_raw\_population['indic\_de,geo\\time'], ',')[0], 'PC\_', '')).withColumn('country\_code', split(df\_raw\_population['indic\_de,geo\\time'], ',')[1])

df\_raw\_population = df\_raw\_population.select(col("country\_code").alias("country\_code"),

                                             col("age\_group").alias("age\_group"),

                                             col("2019 ").alias("percentage\_2019"))

df\_raw\_population.createOrReplaceTempView("raw\_population")

%md

##### Pivot the data by age group

# Create a data frame with pivoted percentages

df\_raw\_population\_pivot = spark.sql("SELECT country\_code, age\_group, cast(regexp\_replace(percentage\_2019, '[a-z]', '') AS decimal(4,2)) AS percentage\_2019 FROM raw\_population WHERE length(country\_code) = 2").groupBy("country\_code").pivot("age\_group").sum("percentage\_2019").orderBy("country\_code")

df\_raw\_population\_pivot.createOrReplaceTempView("raw\_population\_pivot")

%md

##### Read the country lookup

# Create a data frame for the country lookup

df\_dim\_country = spark.read.csv("/mnt/covidreportingdltanaji/lookup/dim\_country/", sep=r',', header=True)

df\_dim\_country.createOrReplaceTempView("dim\_country")

%md

##### Join population data with country lookup

df\_processed\_population = spark.sql("""SELECT c.country,

       c.country\_code\_2\_digit,

       c.country\_code\_3\_digit,

       c.population,

       p.Y0\_14  AS age\_group\_0\_14,

       p.Y15\_24 AS age\_group\_15\_24,

       p.Y25\_49 AS age\_group\_25\_49,

       p.Y50\_64 AS age\_group\_50\_64,

       p.Y65\_79 AS age\_group\_65\_79,

       p.Y80\_MAX AS age\_group\_80\_max

  FROM raw\_population\_pivot p

  JOIN dim\_country c ON p.country\_code = country\_code\_2\_digit

 ORDER BY country""")

%md

##### Write output to the processed mount point

df\_processed\_population.write.format("com.databricks.spark.csv").option("header","true").option("delimiter", ",").mode("overwrite").save("/mnt/covidreportingdltanaji/processed/population")

%md

##### display data from above data frame

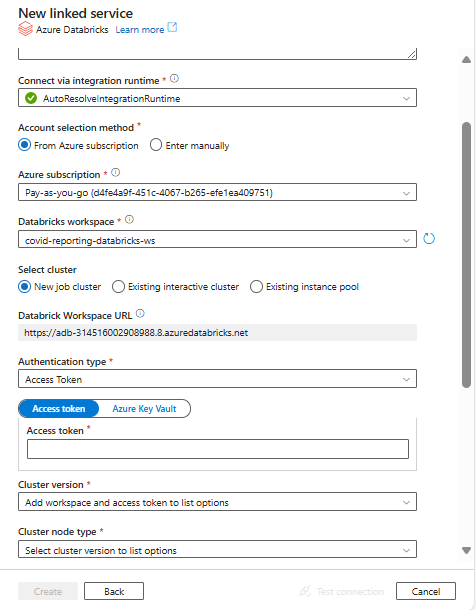
display(df\_processed\_population)

**Create Pipeline to call Databricks Notebook: -**

Create linked service for Azure databricks , this is under Compute option in linked service window.

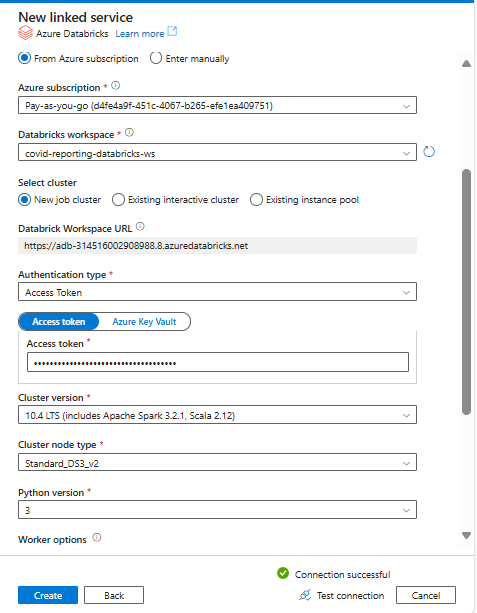
In this linked service option, we need to share token details for accessing databricks notebook from databricks.

Go to Azure databricks profile 🡪 click setting option 🡪 go to user 🡪 Developer 🡪 Manage token 🡪 Manage 🡪 and generate token.



A screenshot of a computer

AI-generated content may be incorrect.



* Create pipeline using above linked service and provide path for notebook which is created on Azure databricks portal. And trigger this pipeline and monitor this in monitor option.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**Module- Copy data to Azure SQL**

***Copy Activity – Data Lake to SQL***

1. **Cases and Deaths Data copy into SQL Server: -** [Download SQL script and create same table in your database]

**DDL SQL scripts: -**

CREATE SCHEMA covid\_reporting

GO

CREATE TABLE covid\_reporting.cases\_and\_deaths

(

country VARCHAR(100),

country\_code\_2\_digit VARCHAR(2),

country\_code\_3\_digit VARCHAR(3),

population BIGINT,

cases\_count BIGINT,

deaths\_count BIGINT,

reported\_date DATE,

source VARCHAR(500)

)

GO

CREATE TABLE covid\_reporting.hospital\_admissions\_daily

(

country VARCHAR(100),

country\_code\_2\_digit VARCHAR(2),

country\_code\_3\_digit VARCHAR(3),

population BIGINT,

reported\_date DATE,

hospital\_occupancy\_count BIGINT,

icu\_occupancy\_count BIGINT,

source VARCHAR(500)

)

GO

CREATE TABLE covid\_reporting.testing

(

country VARCHAR(100),

country\_code\_2\_digit VARCHAR(2),

country\_code\_3\_digit VARCHAR(3),

year\_week VARCHAR(8),

week\_start\_date DATE,

week\_end\_date DATE,

new\_cases BIGINT,

tests\_done BIGINT,

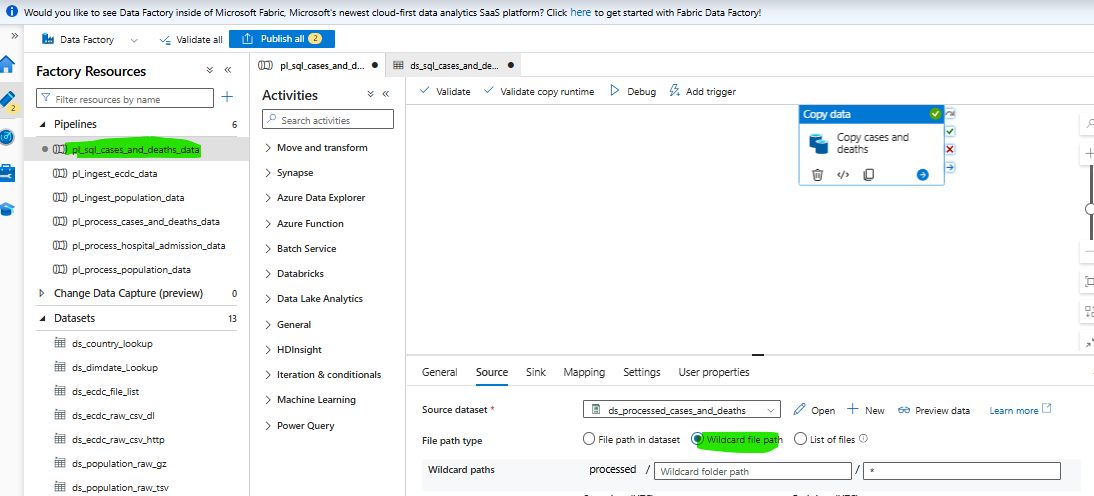
population BIGINT,

testing\_data\_source VARCHAR(500)

)

GO

* Create a pipeline and use copy activity and use existing dataset for source [**ds\_processed\_cases\_and\_deaths**] only select wildcard file path



* Sink dataset: - we need to create sink dataset as well as linked service as well because we do not have a connection for Azure SQL server and database.

**Linked service for Azure SQL: -**

**A screenshot of a computer screen

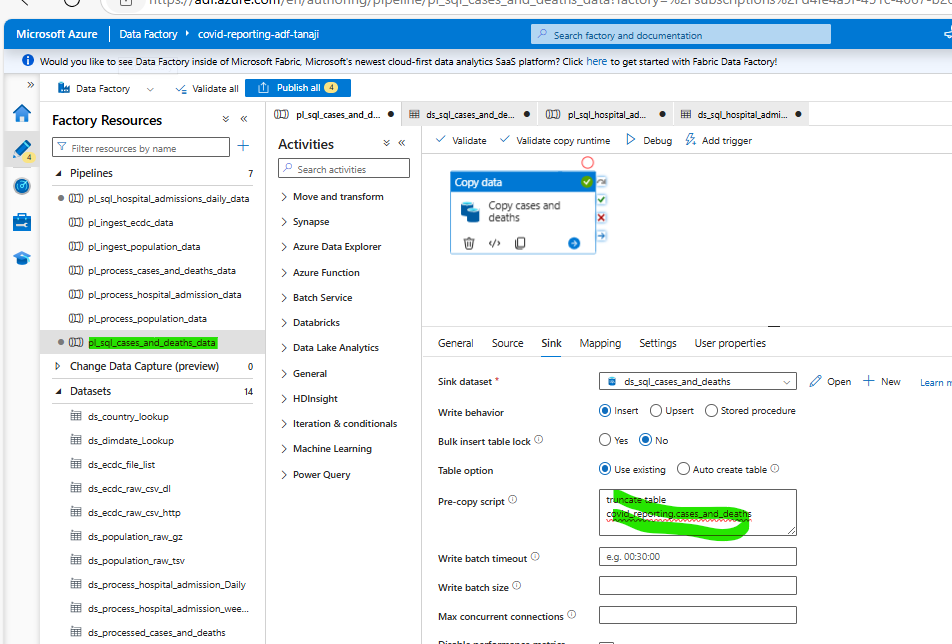
AI-generated content may be incorrect.**

**Dataset:-**

**A screenshot of a computer

AI-generated content may be incorrect.**

Pipeline: - (Add pre-copy script in sink option)



Debug the pipeline and check database table with the command below

**A screenshot of a computer

AI-generated content may be incorrect.**

1. **Hospital Admissions Daily Data copy into SQL Server: -** [we have already created the table using DDL command

Create new pipeline as use copy activity name is [Copy hospital admission Data]

Source dataset: - use existing source data and select only wildcard file path option

**A screenshot of a computer

AI-generated content may be incorrect.**

**Sink Dataset & Linked service: -** we need to use existing Azure SQL linked service but need to create new dataset for [**covid\_reporting.hospital\_admissions\_daily**] table

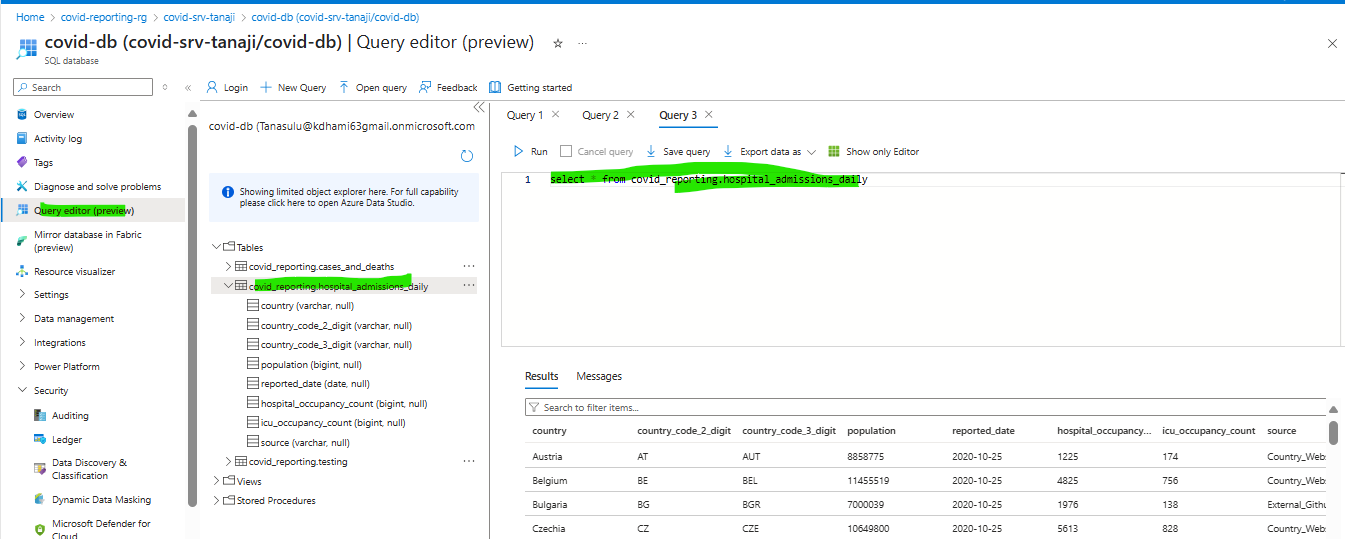
A screenshot of a computer

AI-generated content may be incorrect.

**Pipeline: -** (Add pre-copy script in sink option) and debug the pipeline and check data in database table

A screenshot of a computer

AI-generated content may be incorrect.



1. **Testing Copy into SQL Server: -**

First go to storage account ADLS gen 2 🡪 processed 🡪 ecdc 🡪 testing

We have created this file using hive script 🡪 this file does not have headers.

As we have checked hive script and SQL database script, we do not have 2 columns in SQL table.

A screenshot of a computer

AI-generated content may be incorrect.

For cases & deaths and Hospital admission data we are fetching from dataflows, but for testing data we are using Azure HDinsight so this service did not need sink dataset.

So, we need to create it.

A screenshot of a computer

AI-generated content may be incorrect.

Remove file name in dataset

A screenshot of a computer

AI-generated content may be incorrect.

Sink Dataset: -

A screenshot of a computer

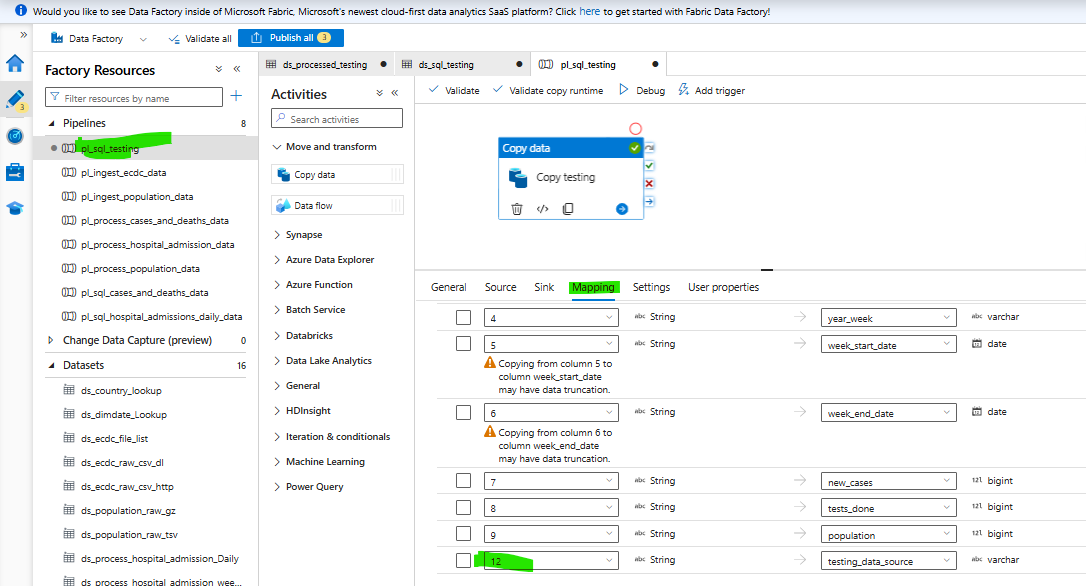
AI-generated content may be incorrect.

Pipeline: - create new pipeline and add source and sink dataset as per the created above and in sink dataset please add pre-copy script.

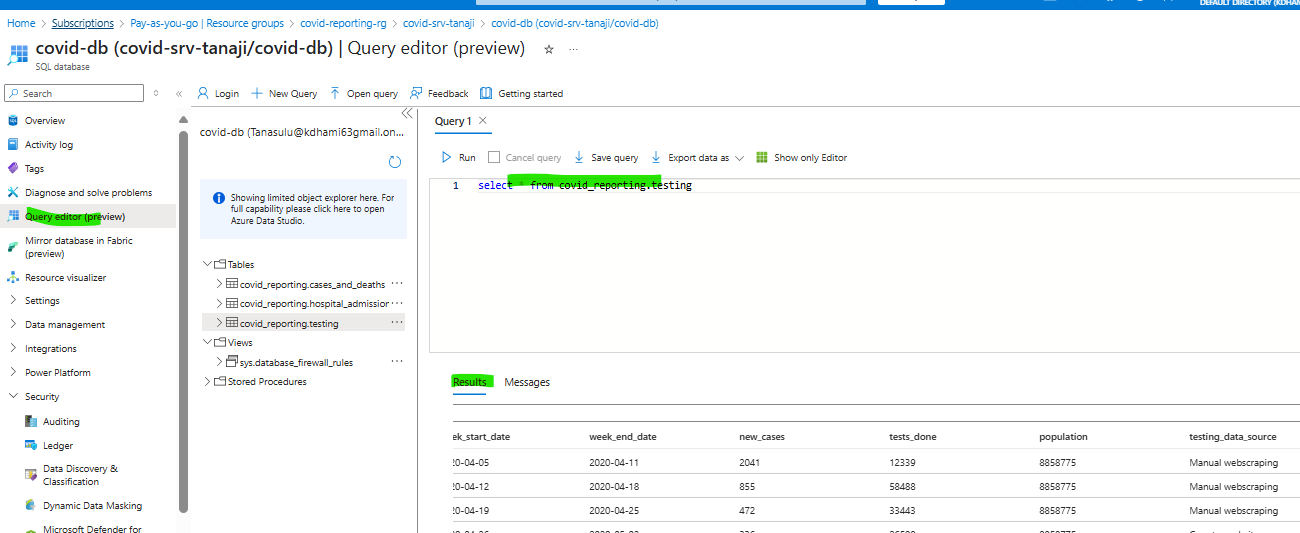
A screenshot of a computer

AI-generated content may be incorrect.

Go to mapping and skip these 2 extra columns which are not available in SQL but available in csv file in ADLS gen 2 storage account.



Check in SQL database table: -



**Module- Making Pipeline production ready**

Firstly, we need to create folders on pipeline& Dataset and map specific pipeline & dataset in this folder. As per the below image.

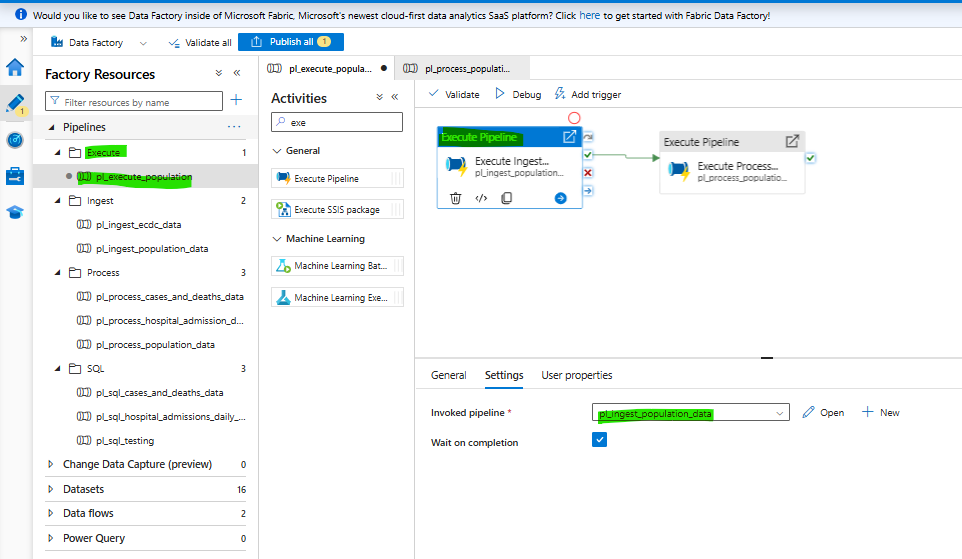
A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

Now we need to create one more folder to execute flow wise pipeline using execute pipeline activity.

* We have population data flow pipelines [ **pl\_ingest\_population\_data**] and [**pl\_process\_population\_data**] now we need create one more pipeline and use these 2 pipelines and using execute pipeline activity [Refer below image]



A screenshot of a computer

AI-generated content may be incorrect.

* Now in [**pl\_ingest\_population\_data**] pipeline we have configured trigger for execution we need to delete this trigger and recreate on [**pl\_execute\_population**] pipeline.

**Note:-** we will use existing trigger, but ADF don’t allows us to change the name for created triggers so that we are recreating the new trigger and delete existing one.

**Trigger1 [Event based trigger]: -** We are creating event trigger [as per the below image we will create new trigger and delete existing one as marked as Red color]

A screenshot of a computer

AI-generated content may be incorrect.

Go to the pipeline and attach new created trigger on parent pipeline i.e. [**pl\_execute\_population].**

* Now we can upload file in blob storage account [**covidreportingsatanaji**] in **population** container

****

Once file uploaded trigger will auto execute and process the file and delete file automatically as per the pipeline created.

**Trigger run: -**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Pipeline Runs: -** so based on trigger parent and child pipeline are successfully run

**A screenshot of a computer

AI-generated content may be incorrect.**

**Trigger2 [Tumbling window trigger]**: - In this trigger we can create 5 triggers and one trigger dependent on other trigger execution.

**Note: -** So now we do not have HDinsight cluster so we will skip those pipeline trigger. [We can skip testing pipelines it is coming from HDInsight cluster].

We can delete Scheduled trigger [**tr\_ingest\_ECDC\_data**] because of we can creating tumbling window trigger for that.

**Note: -** Delete this 🡪 publish ADF 🡪 and then create new trigger with same name.

A screenshot of a computer

AI-generated content may be incorrect.

1. Trigger for ingestion: - [**tr\_ingest\_ecdc\_data**] time should be one day before

A screenshot of a computer

AI-generated content may be incorrect.

1. Trigger for process (time should be one day before)

[**tr\_process\_cases\_and\_deaths\_data**] & [**tr\_process\_hospital\_admission\_data**] these 2 triggers are dependent on [**tr\_ingest\_ecdc\_data**]

Please check below images for this

A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

1. Trigger for SQL: - (time should be one day before)
   1. [**tr\_sql\_cases\_and\_deaths\_data**] trigger dependent on [**tr\_process\_cases\_and\_deaths\_data**] trigger.
   2. [**tr\_sql\_hospital\_admission\_data**] trigger dependent on [**tr\_process\_hospital\_admission\_data**] trigger

Please check below images.

A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

Go to pipeline and attach all the triggers as we have created. [ Please find the below table for your refernce]

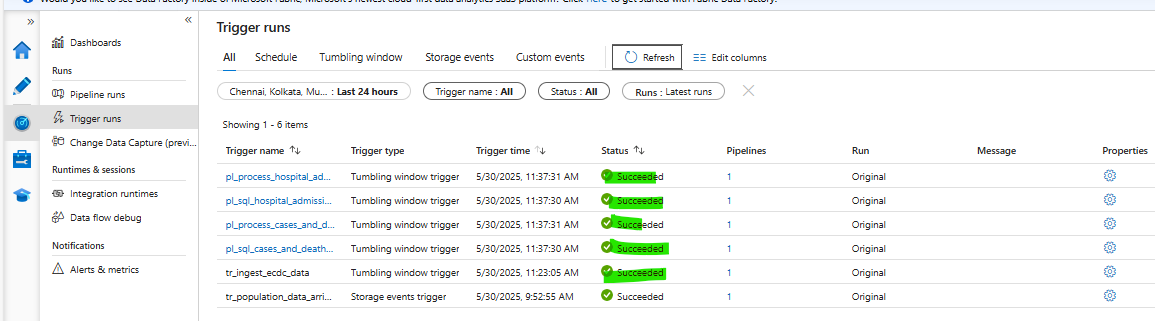
Flow:- go to pipeline 🡪 Trigger 🡪 New/Edit 🡪 choose trigger 🡪 Okay

|  |  |  |
| --- | --- | --- |
| **Pipeline Name** | **Trigger Name** | **Dependant on which Trigger** |
| pl\_ingest\_ecdc\_data | tr\_ingest\_ecdc\_data |  |
| pl\_process\_cases\_and\_deaths\_data | tr\_process\_cases\_and\_deaths\_data | tr\_ingest\_ecdc\_data |
| pl\_process\_hospital\_admission\_data | tr\_process\_hospital\_admission\_data | tr\_ingest\_ecdc\_data |
| pl\_sql\_cases\_and\_deaths\_data | tr\_sql\_cases\_and\_deaths\_data | tr\_process\_cases\_and\_deaths\_data |
| pl\_sql\_hospital\_admissions\_daily\_data | tr\_sql\_hospital\_admission\_data | tr\_process\_hospital\_admission\_data |

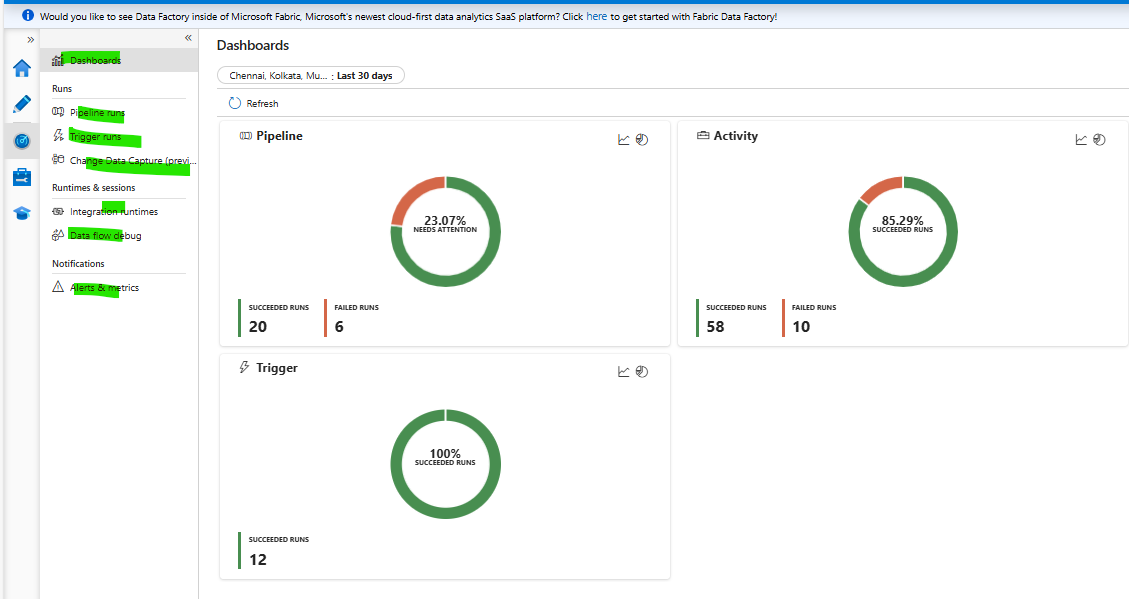
Once published then check in monitor, trigger runs and check all triggers are running.

A screenshot of a computer

AI-generated content may be incorrect.



**Module- Monitoring**

****

**Create Alert and Metrics: -** Create an alert on ecdc pipeline

Add Criteria: - Failed trigger runs Metrics.

A screenshot of a computer

AI-generated content may be incorrect.

Configure Notification: -

A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer screen

AI-generated content may be incorrect.

and add action group 🡪 create alert rule

so now you can change some URL in config folder in json file and save and reupload it and run the trigger. Trigger will fail and you got notification on mobile or email as per the above configuration.

**Re-Run failed activity :-** go to pipeline run in monitor tab and use this option. [PF Image]

A screenshot of a computer

AI-generated content may be incorrect.

**Reporting on metrics: -** Go to Alert & Metrics 🡪 Metrics 🡪 new monitor window opens 🡪 you can choose you appropriate filters 🡪 and save this page on Dashboard.

**Azure Monitor service**

go to diagnostic setting🡪 select storage account 🡪 give name of setting 🡪 and select categories 🡪 and save it 🡪 rerun the pipeline 🡪 log container will create in storage account with JSON log file.

**Log Analytics Workspace service: -** create this service for push logs from Azure monitor to log analytics workspace. it stored in tabular view and extract with KQL (Kusto query language).

Go to ADF 🡪 Diagnostic setting 🡪 edit setting 🡪 there is option enable 🡪 Send to log Analytics workspace 🡪 select check box 🡪 resource specific 🡪 rerun the pipeline 🡪 go to log analytics workspace 🡪 go to logs 🡪 Get started 🡪 the window pop up 🡪 write KQL query to extract data.

**Module: - continuous integration and continuous delivery CI/CD**

Development and operation team working collaboratively this is called DevOps [without communication Gap]

**Characteristics: -**

* 1. Collaboration, trust and transparency
  2. Agile Development approach
  3. Continuous integration/ delivery
  4. Automation
  5. Continuous improvement.

Option 1) Using ADF publish

In this scenario we are directly connected to ADF repo and we made the changes there, but in organization there is more than one developer, so this live mode not we do not have version controlling it direct overwrite existing changes to new changes.

A diagram of a software

AI-generated content may be incorrect.

So, we need to enable source control solution such as GIT as per the below image we need to enable GIT using Azure Devops repos and it ability to create feature branches to implement new changes and merge into master branch.

Also new publish branch auto created for ARM template for test and prod release.

A diagram of a diagram

AI-generated content may be incorrect.

Azure Devops :-

It is SAAS platform to implement devops project to end to end [CI/CD]

1. Boards: - Plan project work, Track work, collaborate team etc.
2. Repos: - manage code
3. Pipeline: - build and release pipelines.
4. Test Plans: - Testing plans configure, feedback from test holder and response to developers.
5. Artifacts: library service used store packages.

A screenshot of a computer program

AI-generated content may be incorrect.

Below URL to login Azure devops:-

<https://aex.dev.azure.com/>

provide login credentials and login with same directory as in ADF

A screenshot of a contact page

AI-generated content may be incorrect. 

As per the below TanajiSulagave is my orgnization Name and provide project name

A screenshot of a computer

AI-generated content may be incorrect.

Go to Organization setting check details of organization and Microsoft Entra

A screenshot of a computer

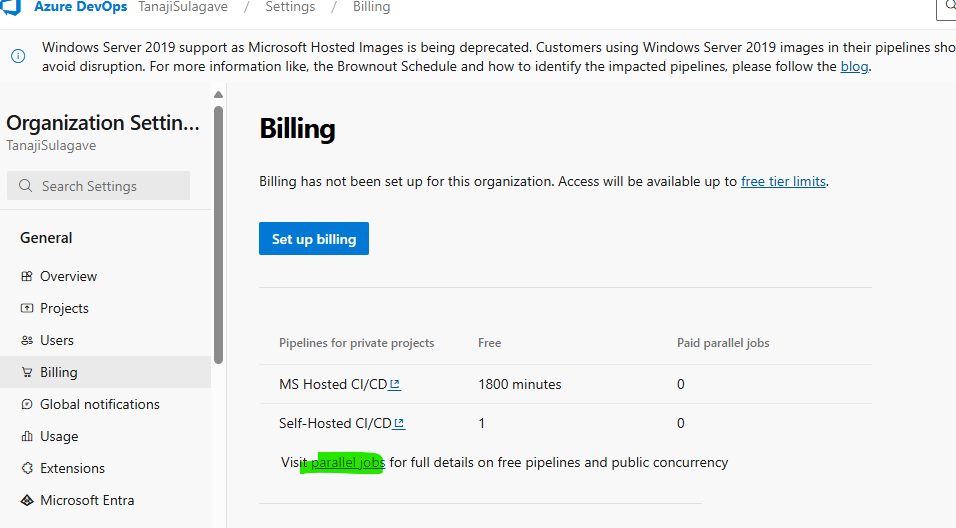
AI-generated content may be incorrect.

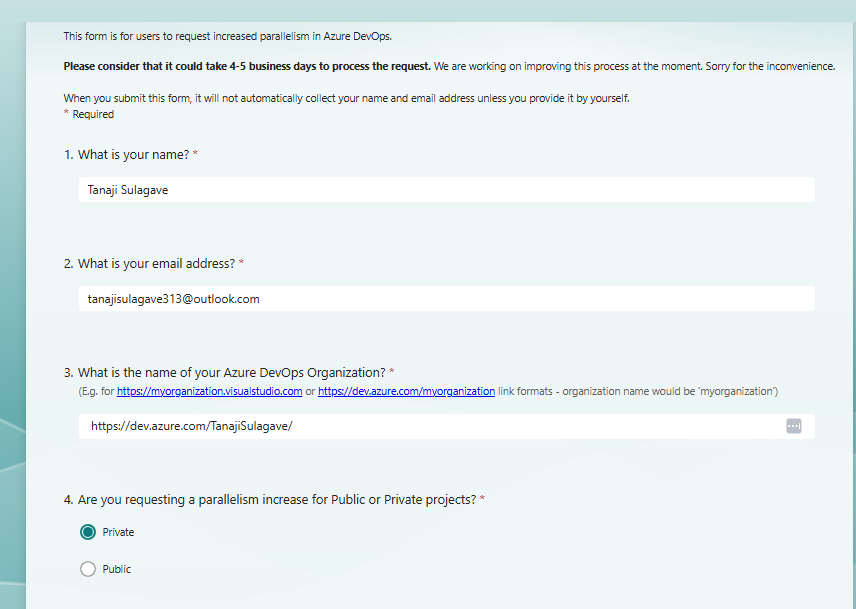
**Optional: -** If you enable parallelism job then submit one form its 2 days take for active this

Open new tab 🡪 <http://aka.ms/azpipelines-parallelism-request> 🡪 use this URL 🡪 fill all the details 🡪 Azure devops organization is URL is below image

A screenshot of a computer

AI-generated content may be incorrect.





Go to devOps and create project

A screenshot of a computer screen

AI-generated content may be incorrect.

So now Project is created and below is the page

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.