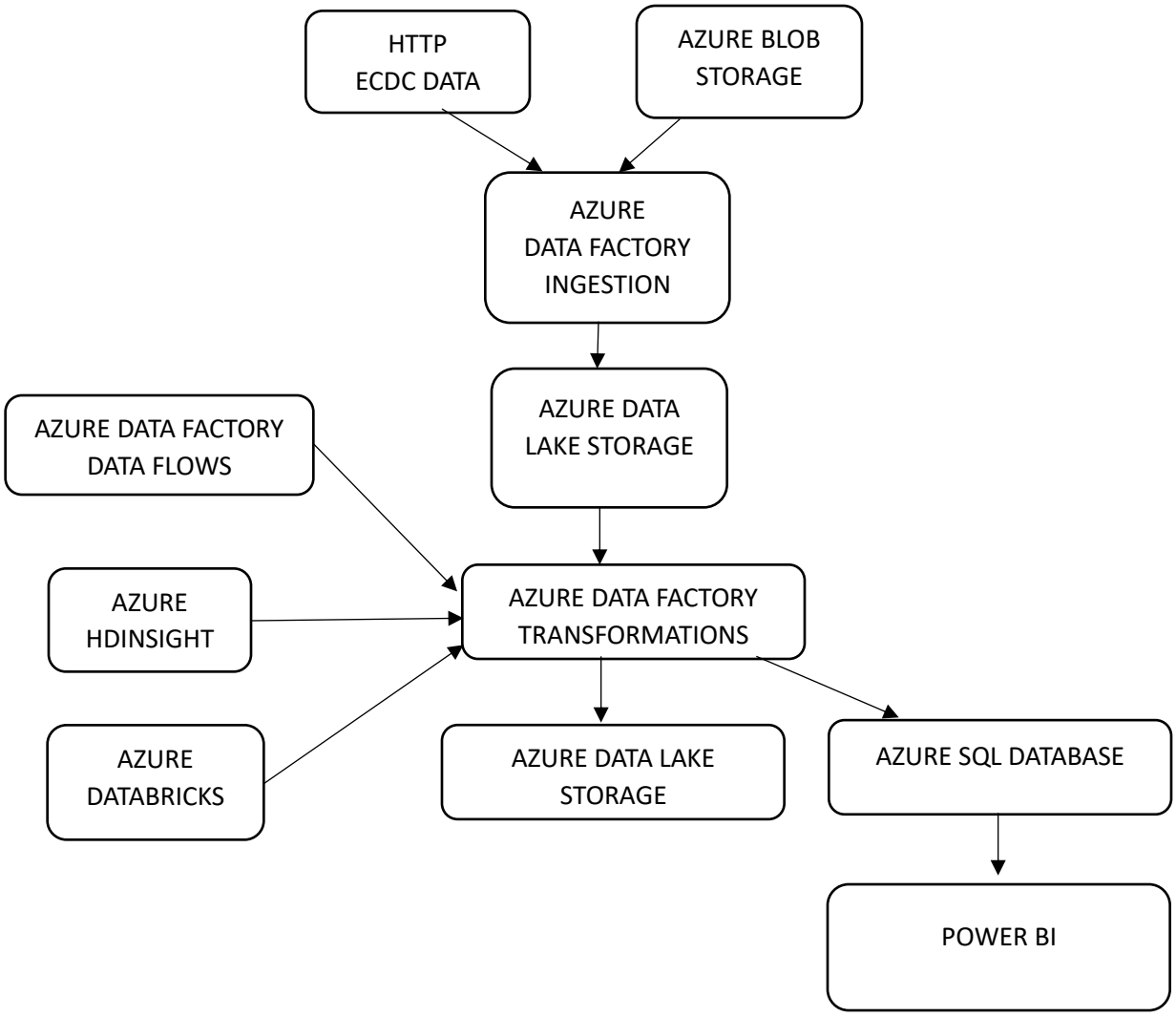
COVID – 19 REPORTING

Solution Architecture:



Data Sources:

- ECDC Website: Confirmed cases, Mortality, Hospitalization and ICU cases, Testing numbers
- Eurostat Website: Population by Age

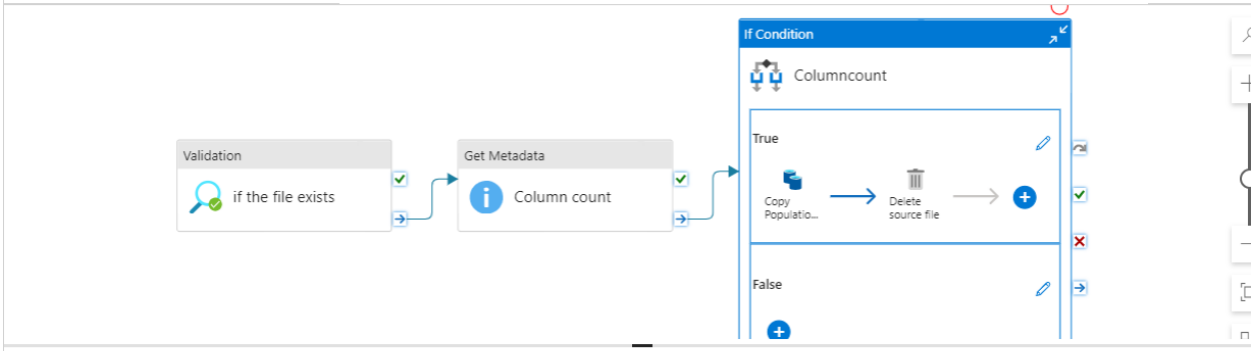
Data Extraction and Data Ingestion:

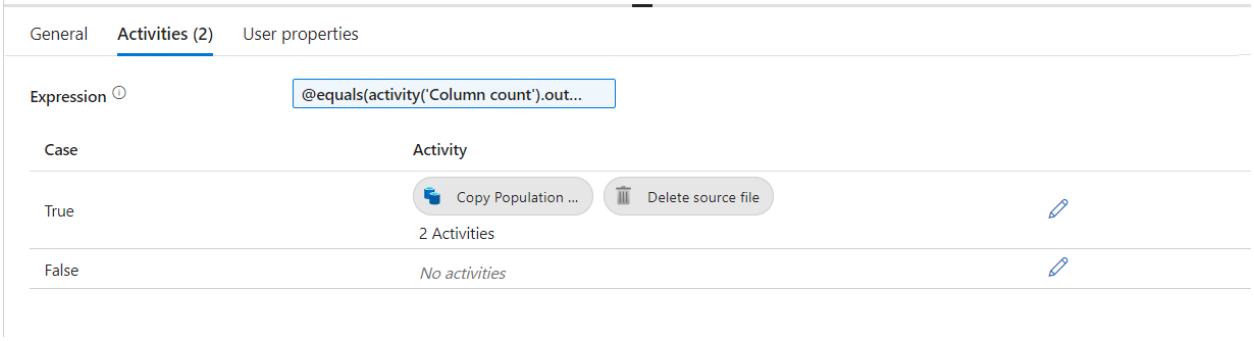
- The population dataset is a zipped tsv file in the Azure blob storage. This file is moved from the blob storage to the Azure data lake storage using the ingestion pipeline in the Azure data factory.
- Linked services for the Azure blob storage and Azure data lake storage are created and datasets are created for the zipped tsv file and the sink file (.tsv file in the Azure data lake storage)
- Ingestion pipeline is created with 3 activities –

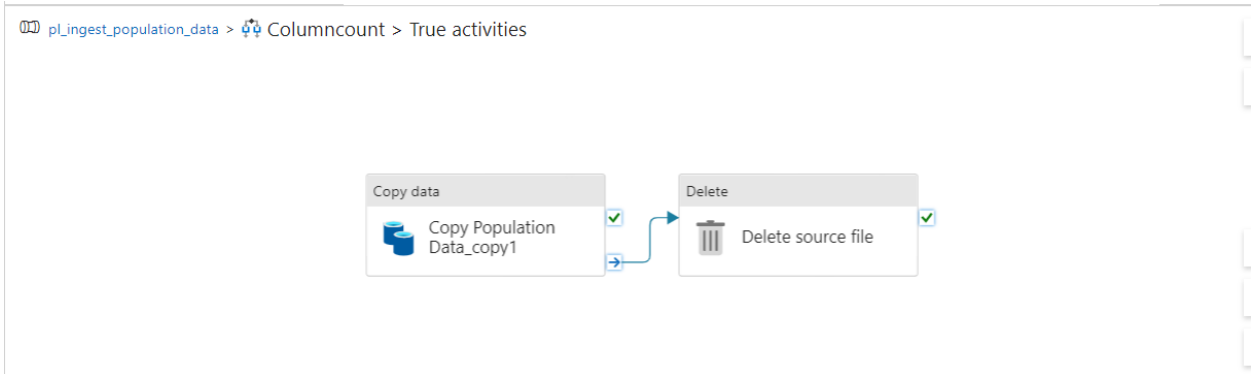
  1. Validation activity - to check if the file exists or not.

  2. Get Metadata activity - to check if the file contains the correct number of columns.

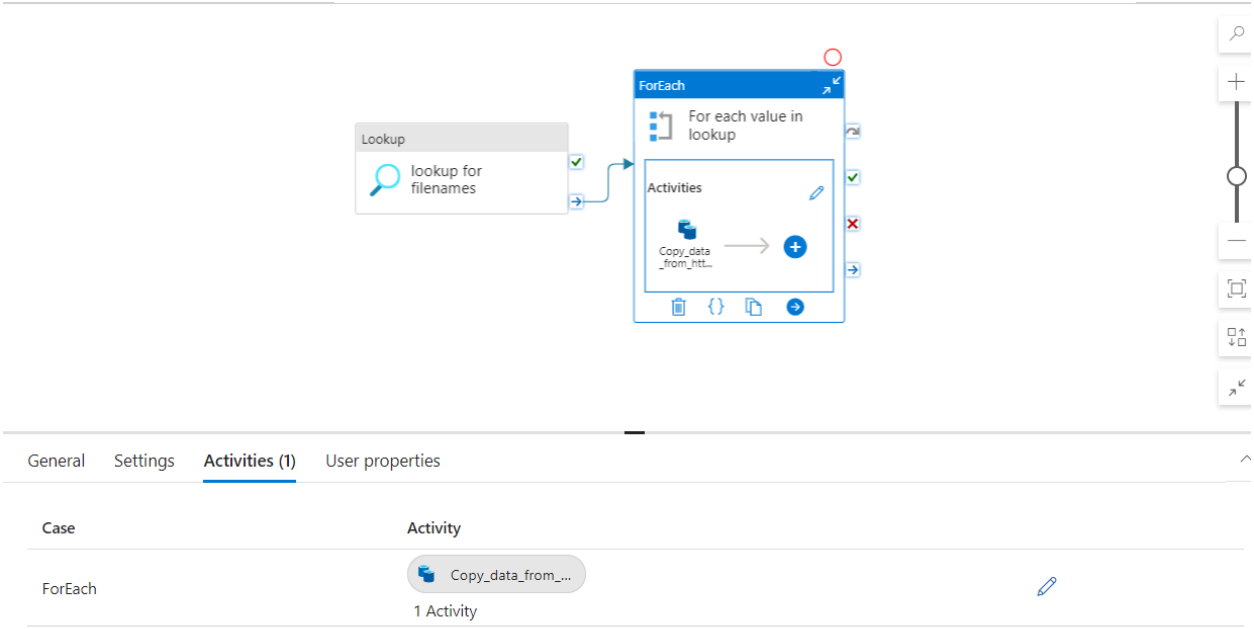
  3. If Condition activity - to check if the output from the Get Metadata activity is true or not. If it is true then a Copy activity is created, the file is copied to data lake and it is deleted from the source, otherwise an email is sent indicating that the pipeline has failed.
- An event trigger is used to invoke the pipeline whenever a file with a particular name comes into the blob storage.







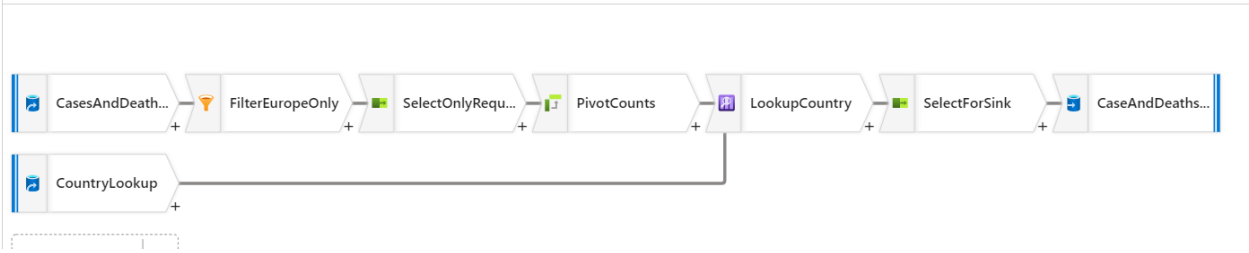
- From the ECDC website 3 files namely, cases and deaths, hospital admissions, and testing were extracted using the json file that contains the links to those files. The json file is uploaded into the Azure blob storage.
- A parameterized linked service is created to link the Azure data lake storage to the http source and a parameterized dataset is created resembling the data coming from the http.
- Ingestion pipeline is created which contained the following activities in it –
    1. Lookup activity - to read the json file in the Azure blob storage.
    2. Foreach activity – to evaluate each record from the output of the lookup activity and for each record, a Copy activity is executed to copy the files from the web resource to the Azure data lake storage (in csv format).



Data Transformation:

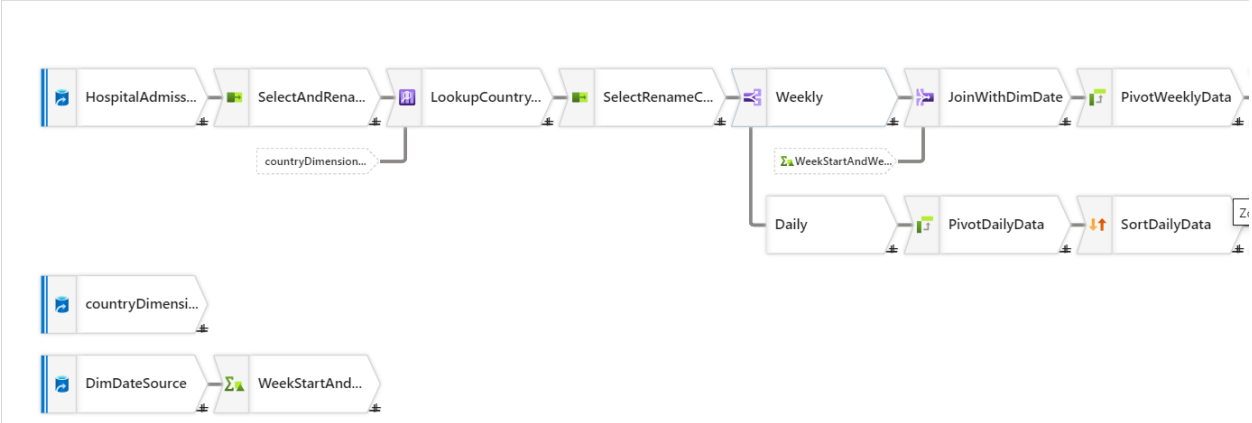
<u>Cases and deaths file:</u>

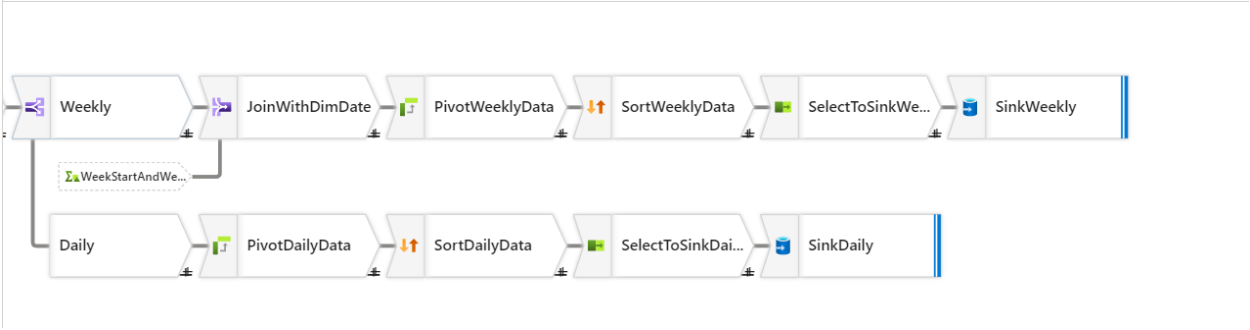
- Various transformations in the data flows in the Azure data factory are used to transform the cases and deaths file –
    1. Source transformation: To pick the dataset from the source.
    2. Filter transformation: To filter only the records which have the Europe continent and have no null values in the country code.
    3. Select transformation: It is used to rename a few columns.
    4. Pivot transformation: It is used to pivot a column grouping the other columns to create two new columns which show the confirmed cases and deaths.
    5. Lookup transformation: It is used to lookup another file based on a condition and combine the rows with the original file. It works similar to the left join in SQL.
    6. Sink transformation: It is used to copy the file to the sink, in this case it is Azure data lake storage.



Hospital admissions file:

- The below transformations are used to transform the hospital admissions file –
    1. Source transformation: To pick the dataset from the source.
    2. Select transformation: To select the columns that are needed and to rename a few columns.
    3. Lookup transformation: To lookup a file, get the columns from that file and merge with the original file.
    4. Conditional split transformation: This transformation is used to split the dataset into different datasets based on the specified conditions. In this case, the dataset is divided into weekly and daily datasets based on the indicator column.
    5. Aggregate transformation: To aggregate a column and create new columns based on the aggregated column. In this case, aggregated on the weekly data to get the week start date and weekend date.
    6. Join transformation: This transformation is used to join two datasets similar to SQL joins. In this case, the aggregated data and the original data are joined.
    7. Sort transformation: Sort transformation is used to sort the output from the previous transformation based on the specified columns.
    8. Sink transformation: To copy the output from the previous transformation to the sink datastore that is Azure data lake storage in this case.





Testing file:

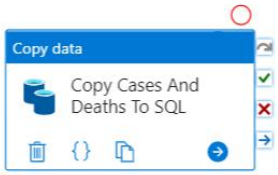
- HDInsight with a Hadoop cluster is used in this project to transform the testing file. The hive script is written and uploaded into the azure blob storage.
- Storage accounts are mounted onto the HDInsight cluster using the managed identity.
- A pipeline has been used to execute the hive activity which executed the hive script that made required transformations and uploaded the processed file to the Azure data lake storage.

Population file:

- Azure databricks is used to transform the population file in this project. Python notebook files are written with necessary transformations and uploaded into the databricks cluster.
- Storage accounts are mounted onto the databricks cluster using the service principal identity.
- A pipeline with the Notebook activity is executed to run the transformations and write the processed file to the Azure data lake storage.

Data Loading:

- Data is loaded into the Azure data lake storage when the transformations are executed. The processed data in the data lake can be used for building machine learning models.
- Data is loaded into the Azure SQL database via the pipelines which has the copy activity for creating reports in Power BI.



Reporting:

- In the Power BI tool, Azure SQL database is connected using the server's name and the database name. The tables in that database are loaded into the power BI.
- Reports are created using that data to show the confirmed cases and deaths by country and by month. Another one is created to show the tests done and the confirmed cases by country.