



# ETL Azure Data Factory Project on Covid-19 Reporting

**Date:** 27/09/2024

**Time:** 08:00 AM

**Venue:** Thanh Pho Ho Chi Minh

## Concept of the Project

Dự án này là về việc thu thập một vài Bộ dữ liệu Covid-19 từ trang web ECDC, chuyển đổi chúng bằng nhiều thành phần ADF khác nhau, sau đó thực hiện chuyển đổi bằng cách sử dụng ADF, HDInsight và Databricks, sau đó tải chúng vào SQL Datawarehouse để nhóm Analytics có thể rút ra những hiểu biết hữu ích và có thể hành động được từ các bộ dữ liệu này. Mục tiêu chính là hiểu toàn diện về ảnh hưởng của COVID-19 đối với toàn bộ Khu vực Châu Âu trong suốt năm 2020.

## Taks

Nhiệm vụ của dự án này là thu thập dữ liệu từ nhiều nguồn dữ liệu, dọn dẹp và chuyển đổi dữ liệu để dữ liệu mạnh mẽ hơn và phù hợp hơn với mục tiêu. Sau đó, dữ liệu đã dọn dẹp sẽ được tải vào kho lưu trữ trung tâm, như Kho dữ liệu hoặc Hồ dữ liệu để nhóm phân tích có thể sử dụng dữ liệu đó bằng các công cụ BI của họ như Power BI. Kho dữ liệu sẽ bao gồm thông tin chi tiết về các trường hợp đã xác nhận, tỷ lệ tử vong đáng tiếc, các trường hợp nhập viện và ICU từ số liệu thống kê hồ dữ liệu hàng tuần của chúng tôi và số lượng xét nghiệm. Ngoài ra, chúng tôi có thể chạy Mô hình ML sử dụng dữ liệu này để dự đoán sự lây lan của COVID-19 ở khu vực Châu Âu.

## Source Data

ECDC ( <https://www.ecdc.europa.eu/en/covid-19> )

Dữ liệu dân số từ Azure Blob Storage

## Destination

Azure Data Lake Gen2 Storage

## Tools

Data Integration/Ingestion

ADF Data Flows within the Data Factory

## Transformation

Data Flows within the Data Factory

DataBricks

## Data Warehouse Solution

Azure SQL Database

## Visualization

Power BI Desktop

## Environment

Azure Subscription

Data Factory

Azure Blob Storage Account

Data Lake Storage Gen2

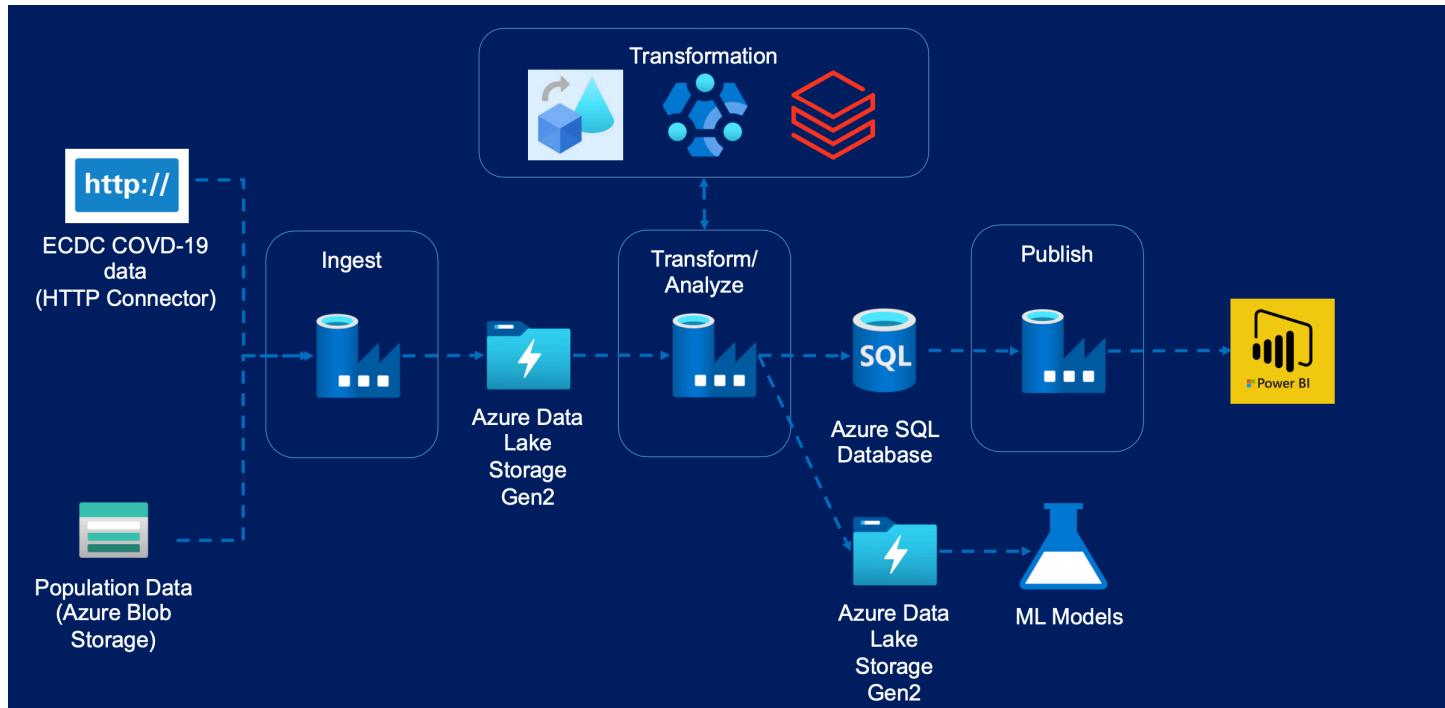
Azure SQL Database

Azure Databricks Cluster

HD Insight Cluster

## Người thực hiện dự án

# Solution Architecture Overview



## Data Extraction/Data Ingestion

Bốn tập dữ liệu khác nhau đã được nhập từ cả trang web ECDC và kho lưu trữ blob Azure vào Datalake Gen2. Chúng là:

Cases and Deaths Data

Hospital Admissions Data

Population Data

Test Conducted Data

Tôi đã sử dụng nhiều thành phần khác nhau của hoạt động ADF Pipeline để thu thập dữ liệu từ cả HTTP Data Source và Azure Storage Account vào Azure DataLake. Một số hoạt động đó là:

Validation Activity

## Get Metadata Activity

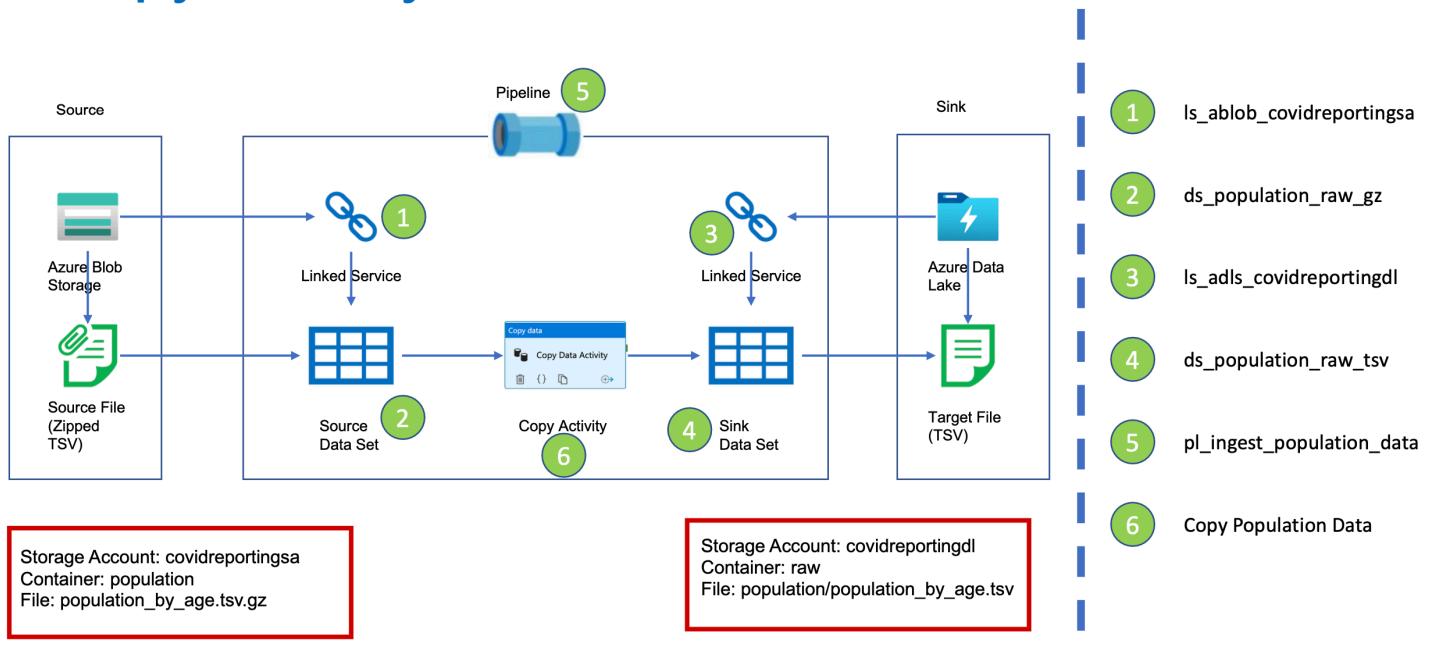
## Copy Activity

### **Population**

Dữ liệu dân số Nhập dữ liệu "dân số theo độ tuổi" của tất cả các quốc gia EU vào Data Lake để hỗ trợ các máy tính học tập dự kiến sẽ có tỷ lệ tử vong tăng lên do COVID-19

**Giải pháp:**

## Copy Activity



### Các bước thực hiện:

Create a Linked Service To Azure Blob Storage

Create a Source Data Set

Create a Linked Service To Azure Data Lake storage (GEN2)

Create a Sink Data set

Create a Pipeline:

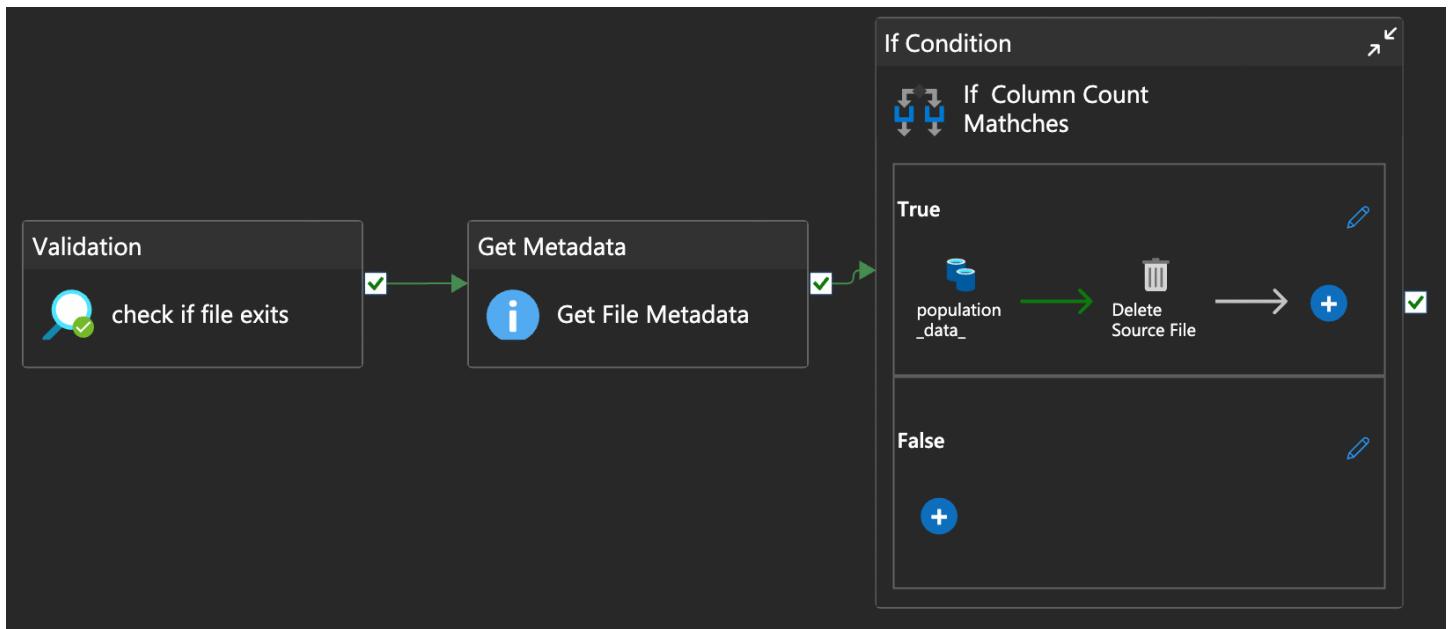
Execute Copy activity when the file becomes available

Check metadata counts before loading the data using the IF Condition

Finally, Load Data into our destination

# ScheduleTrigger

## Pipeline Overview:



## ECDC Data

### Bốn tệp CSV:

[Case & Deaths Data.csv](#)

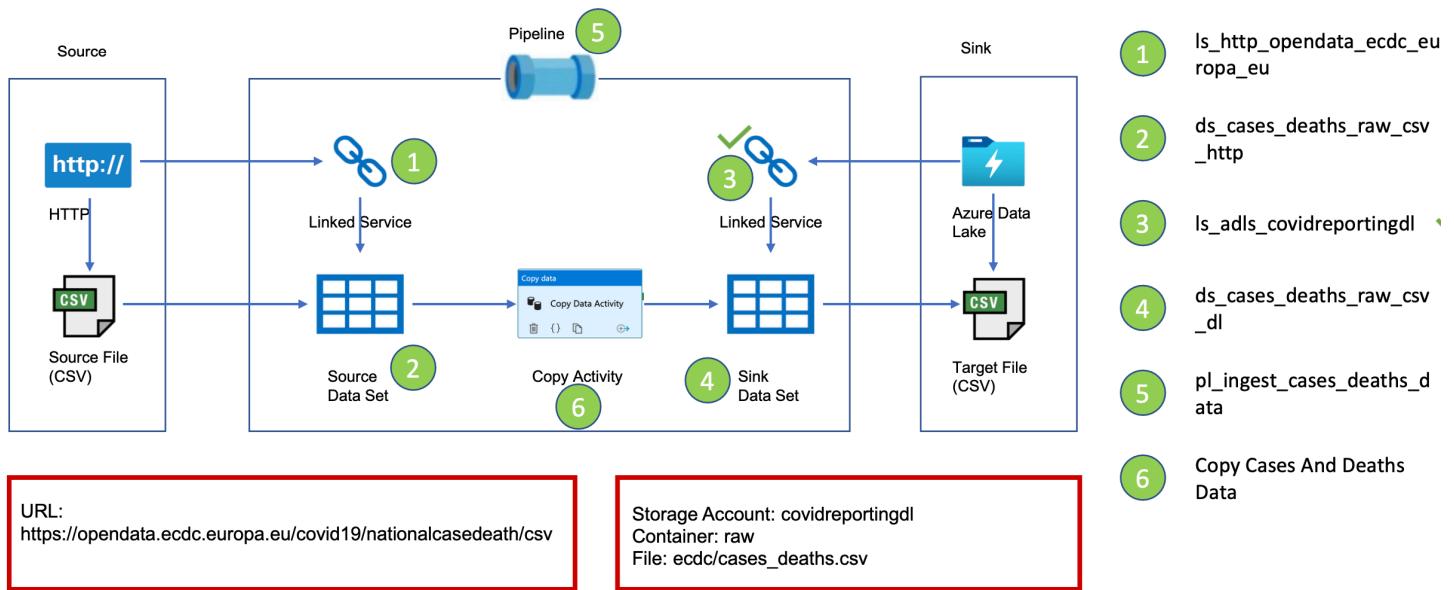
[Hospital Admission Data.csv](#)

[testing.csv](#)

[country\\_response.csv](#)

### Giải pháp:

# Copy Activity – Case & Deaths Data



## Các bước thực hiện:

Create a Linked Service using an HTTP connector

Create a Source Data Set

Create a Linked Service To Azure Data Lake storage (GEN2)

Create a Sink Data set

Create a Pipeline With Parameters & Variables

Lookup to get all the parameters from json file, then pass it to ForEach ECDC DATA as shown below

Schedule Trigger

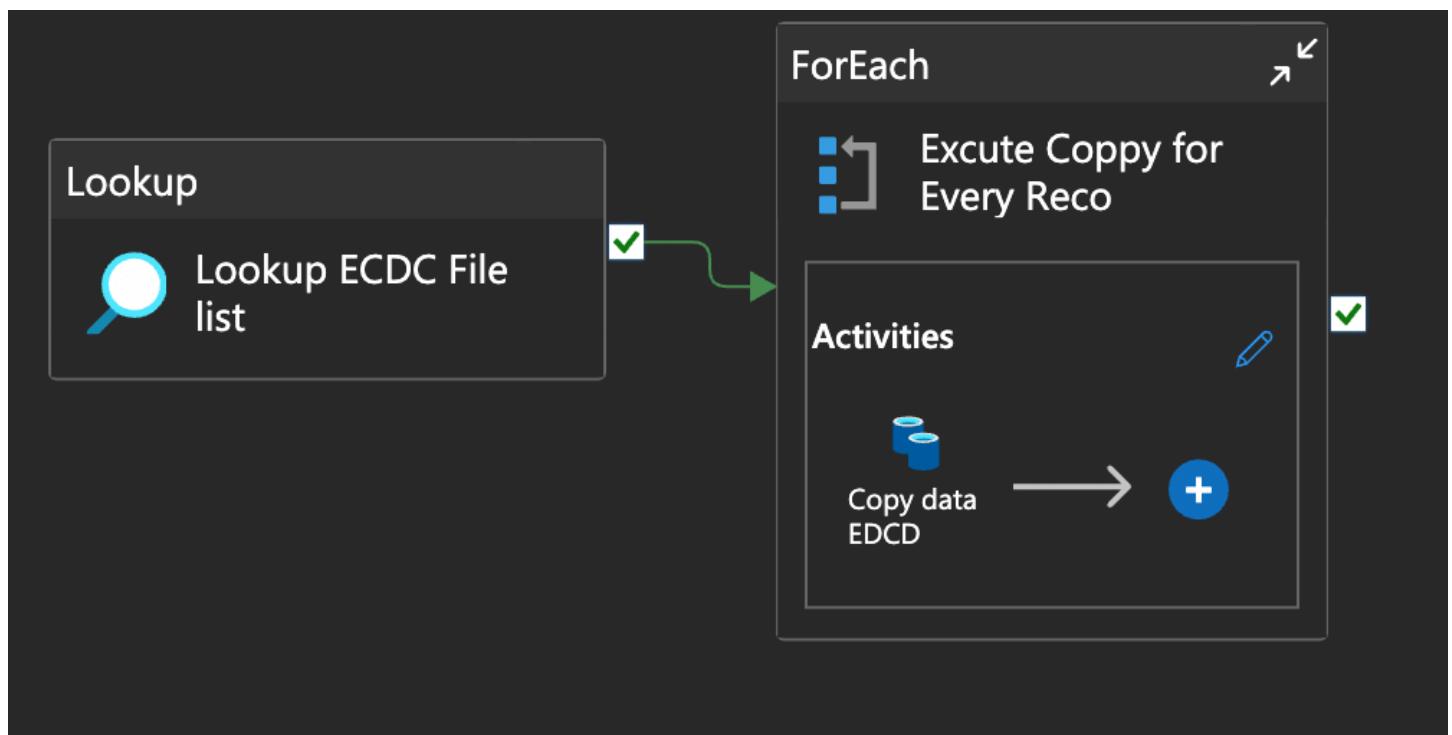
## File Json

```

[
  {
    "sourceBaseURL": "https://github.com",
    "sourceRelativeURL": "cloubboxacademy/covid19/raw/main/ecdc_data/cases_deaths.csv",
    "sinkFileName": "cases_deaths/cases_deaths.csv"
  },
  {
    "sourceBaseURL": "https://github.com",
    "sourceRelativeURL": "cloubboxacademy/covid19/raw/main/ecdc_data/hospital_admissions.csv",
    "sinkFileName": "hospital_admissions/hospital_admissions.csv"
  },
  {
    "sourceBaseURL": "https://github.com",
    "sourceRelativeURL": "cloubboxacademy/covid19/raw/main/ecdc_data/testing.csv",
    "sinkFileName": "testing/testing.csv"
  },
  {
    "sourceBaseURL": "https://github.com",
    "sourceRelativeURL": "cloubboxacademy/covid19/raw/main/ecdc_data/country_response.csv",
    "sinkFileName": "country_response/country_response.csv"
  }
]

```

## Pipeline Design



## Data Transformation

Dữ liệu về các ca bệnh và tử vong cùng với dữ liệu nhập viện đã được chuyển đổi bằng cách sử dụng ADF Data Flows. Chuyển đổi luồng dữ liệu được sử dụng trên cả

hai tập dữ liệu bao gồm;

Select transformation

Lookup transformation

Filter transformation

Join transformation

Sort transformation

Conditional split transformation

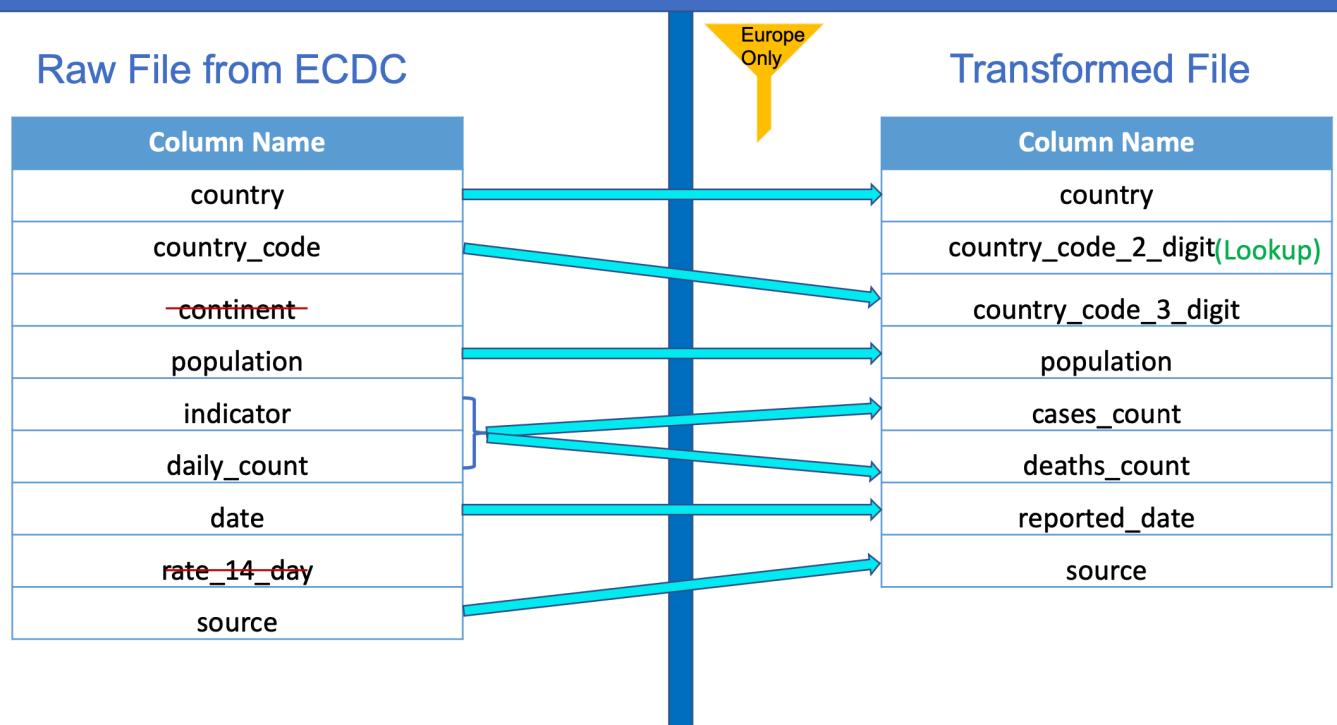
Derived columns transformation

Sink transformation

## Data Flows (1) Cases & Deaths Data:

Giải pháp:

### Transform Cases & Deaths Data



### Các bước thực hiện:

Cases And Deaths Source (Azure Data Lake Storage Gen2 )

Filter Europe-Only Data

Select only the required columns

PivotCounts using indicator Columns(confirmed cases, deaths) and get the sum of daily cases count

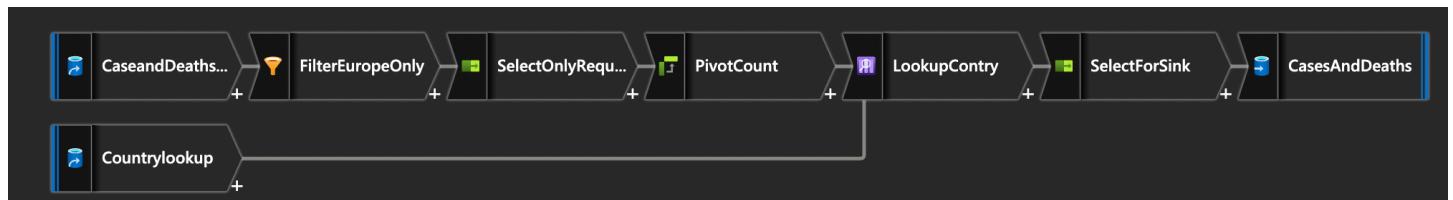
Lookup Country to get country\_code\_2\_digit,country\_code\_3\_digit columns

Select Only the required columns for the Sink

Create a Sink dataset (Azure Data Lake Storage Gen2)

Used Schedule Trigger

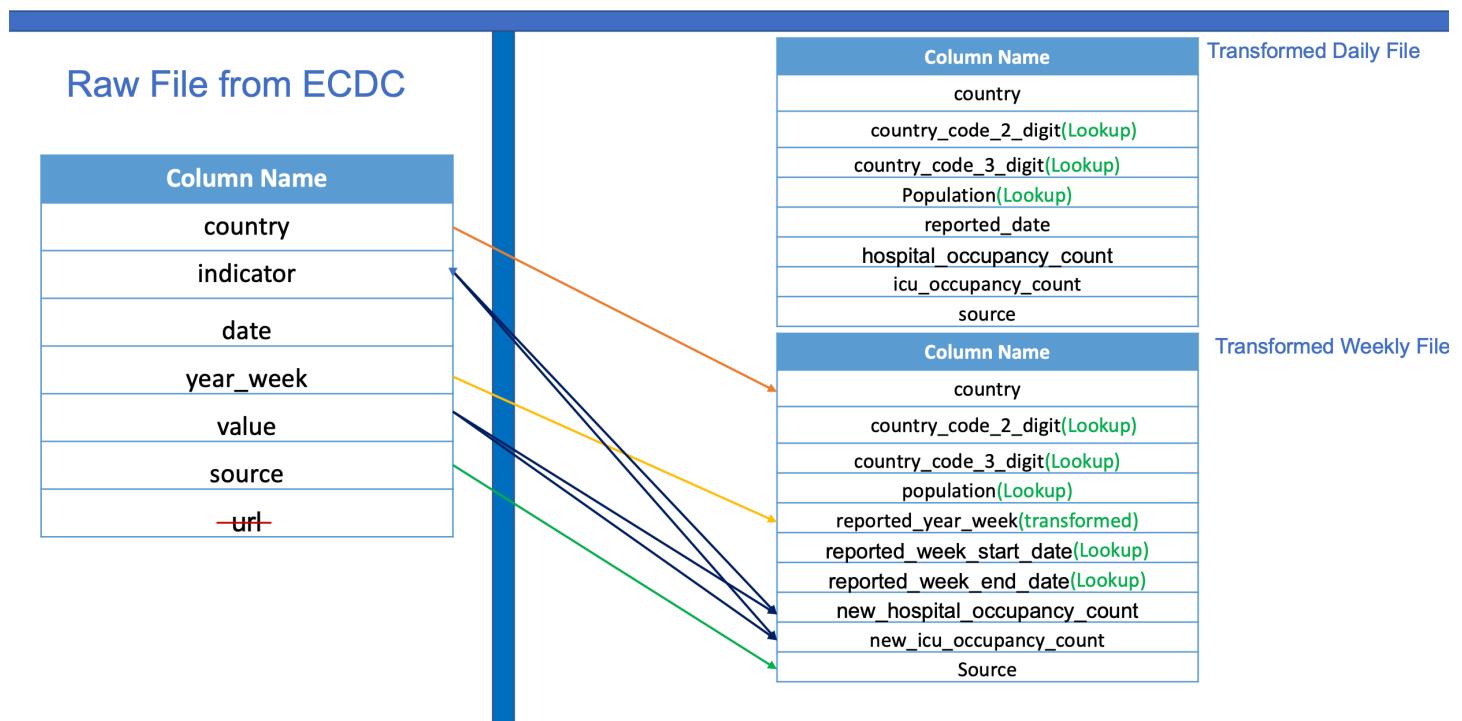
### Pipeline Overview:



## Data Flows (2) Hospital Admissions Data:

Giai pháp:

### Hospital Admissions Data



Các bước thực hiện:

Hospital Admissions Source (Azure Data Lake Storage Gen2 )

Select only the required columns

Lookup Country to get country\_code\_2\_digit,country\_code\_3\_digit columns

Select only the required columns

### **Condition Split Weekly, Daily Split condition**

indicator=='Weekly new hospital admissions per 100k' || indicator=='Weekly new ICU admissions per 100k'

indicator== "Daily hospital occupancy" || indicator=="Daily ICU occupancy"

### **For Weekly Path**

Join with Date to get ecdc\_Year\_week, week\_start\_date, week\_End\_date

Pivot Counts using indicator Columns(confirmed cases, deaths) and get the sum of daily cases count

Sort data using reported\_year\_week ASC and Country DESC

Select only the required columns for the sink

Create a sink dataset (Azure Data Lake Storage Gen2)

Schedule Trigger

### **For Daily Path**

Pivot Counts using indicator Columns(confirmed cases, deaths) and get the sum of daily cases count

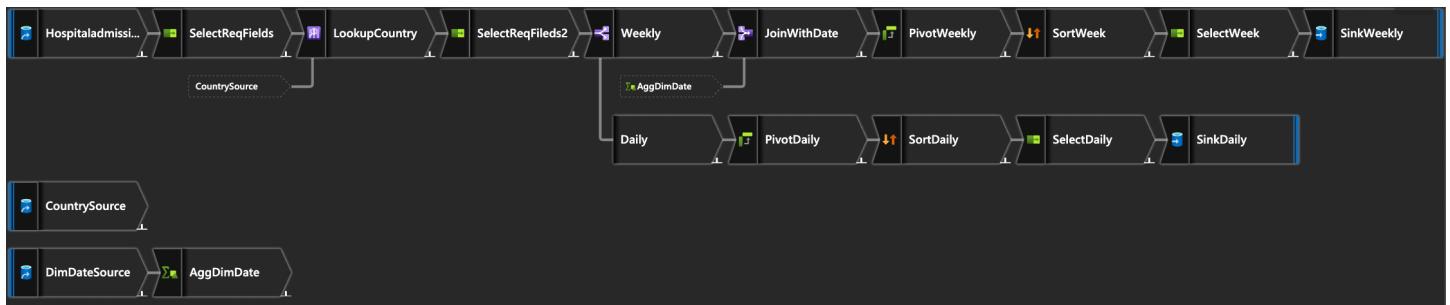
Sort Data using reported\_year\_week ASC and Country DESC

Select only the required columns for the sink

Create a sink dataset (Azure Data Lake Storage Gen2)

Used Schedule Trigger

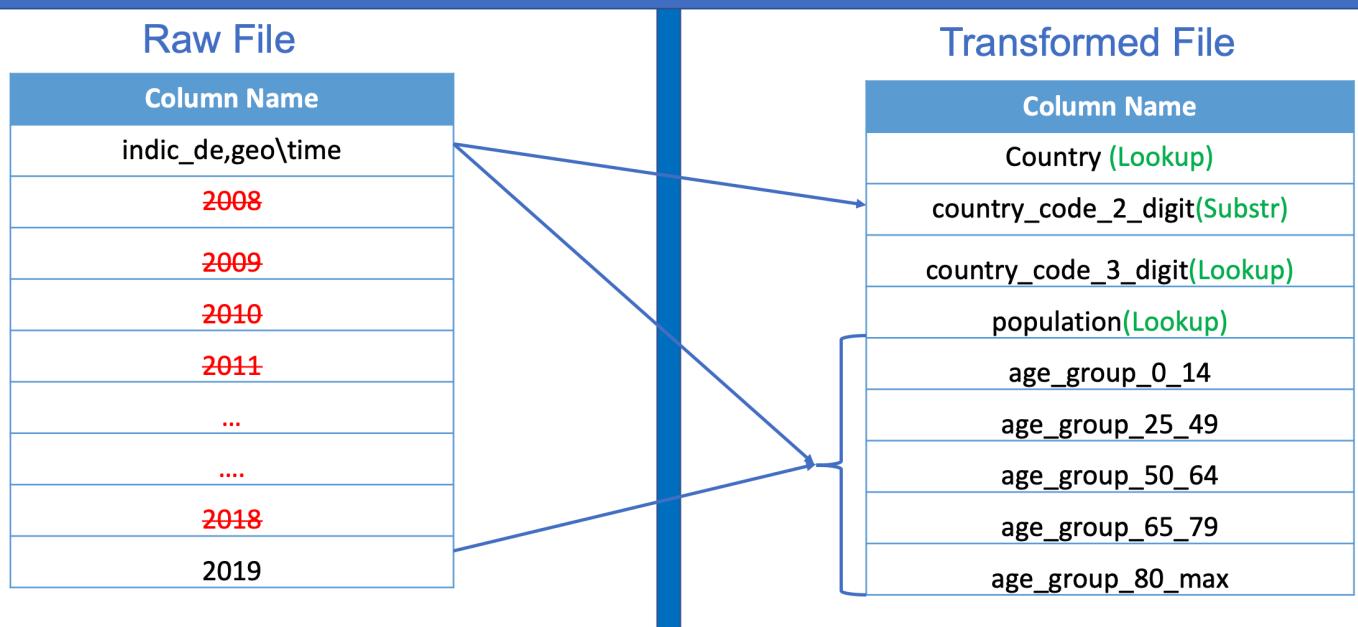
### **Pipeline Overview:**



## Databricks Activity (3) -- Population File:

Giải pháp:

### Transform Population By Age Data



File xử lý:

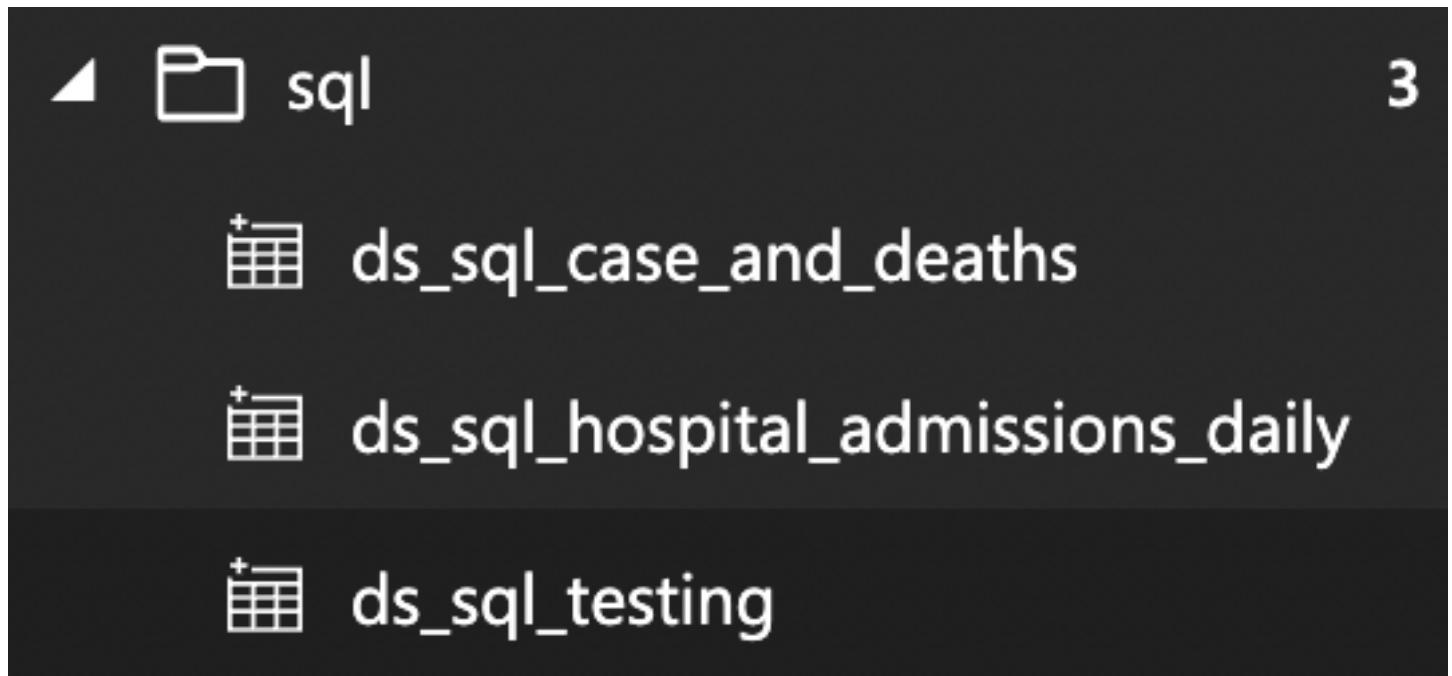
[@Project\\_on\\_Covid19/Databricks/pyspark\\_notebooks/transform\\_population\\_dat...](#)

## Copy Data to Azure SQL ( Data WareHouse )

1- Copy Cases and Deaths

2- Copy hospital admissions data

3- Copy testing data



## Login SQL DataWarehouse

Sever: nthanhdat.database.windows.net

User name: dat\_admin

Pass: 01202596666Nguyen\$

Database: covid-tb

The screenshot shows the Azure Data Studio interface. On the left, the object explorer displays the 'covid-tb (dat\_admin)' database with tables like 'cases\_and\_deaths', 'hospital\_admissions\_daily', and 'testing'. In the center, 'Query 2' is selected and contains the following T-SQL code:

```
1 SELECT TOP (1000) * FROM [covid_reporting].[cases_and_deaths]
2 SELECT TOP (1000) * FROM [covid_reporting].[hospital_admissions_daily]
3 SELECT TOP (1000) * FROM [covid_reporting].[testing]
```

The 'Results' tab is active, showing the output of the third query:

country	country_code	country_code_2_digit	country_code_3_digit	case
Austria	AUT	AT	AUT	754
Andorra	AND	AD	AND	0
Czechia	CZE	CZ	CZE	9720
Azerbaijan	AZE	AZ	AZE	230
Greece	CYP	CY	CYP	20

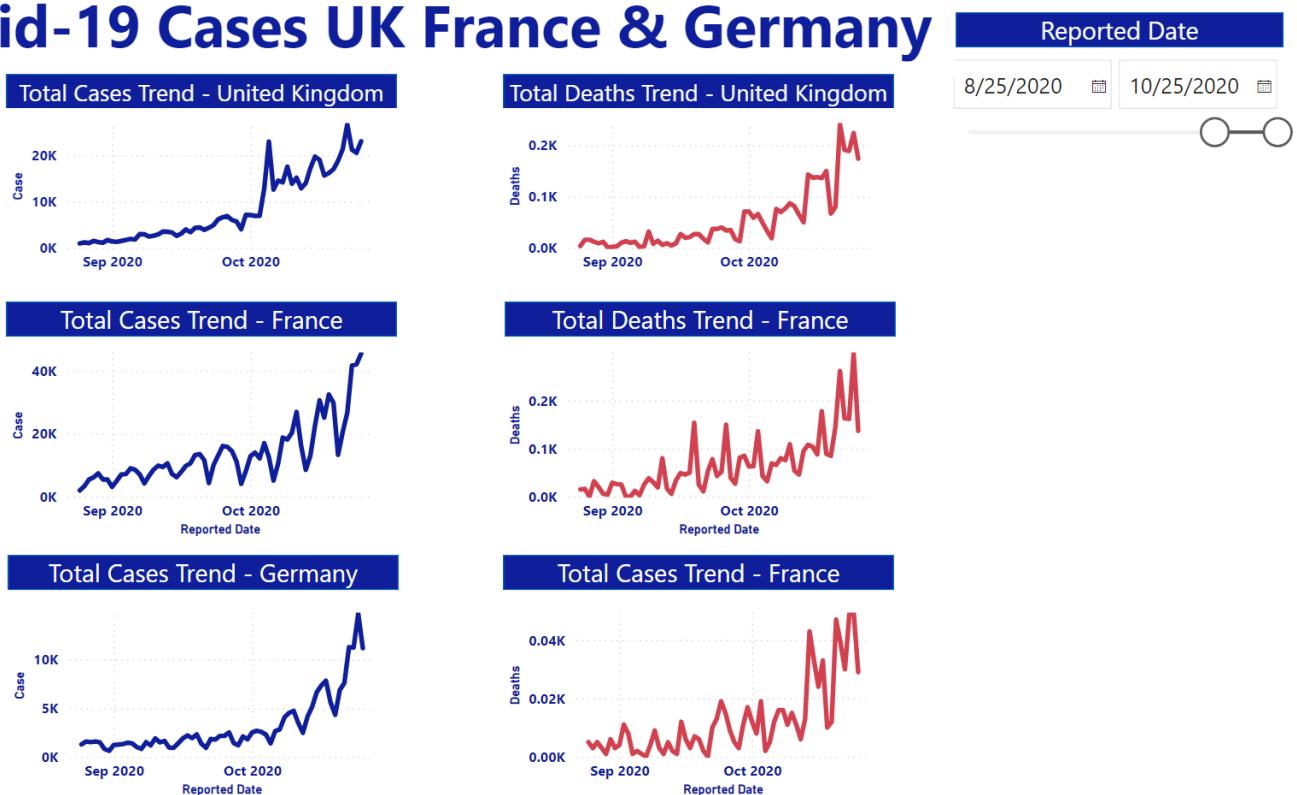
At the bottom, a green bar indicates the query succeeded.

# Power BI Reporting

- 1- Tạo kết nối từ Azure SQL đến Power BI và tải dữ liệu
- 2- Phân tích dữ liệu để có được tổng số ca được xác nhận và số ca tử vong
- 3- Xác định xu hướng dữ liệu dựa trên ngày báo cáo
- 4- Xuất bản báo cáo lên Power BI Server
- 5- Đăng lên web

## Covid-19 Trend in the EU/EEA & UK 2020 by Cases, Deaths, Hospital Occupancy, and ICU Occupancy

### Covid-19 Cases UK France & Germany



## Covid-19 Cases and Death breakdown by population in the UK, France, and Germany

# Covid-19 Cases EU/EEA & UK

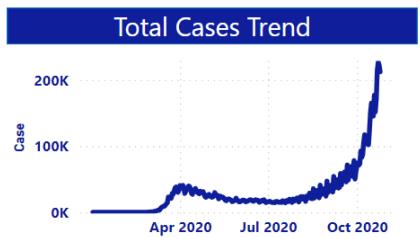
Reported Date

1/2/2020

10/25/2020

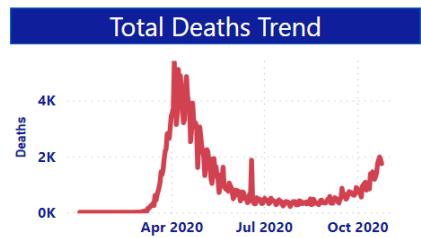
Total Confirmed Cases

**8730K**



Total Deaths

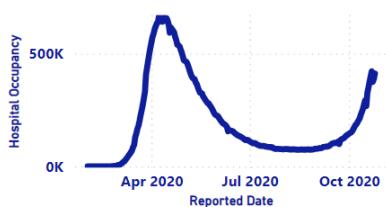
**263K**



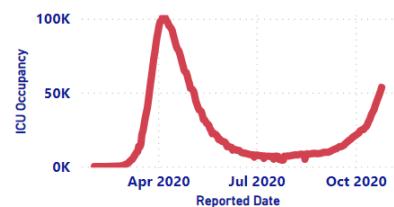
Country

- Austria
- Belgium
- Bulgaria
- Croatia
- Cyprus
- Czechia
- Denmark
- Estonia
- Finland
- France
- Germany
- Greece
- Hungary
- Iceland

Hospital Occupancy



ICU Occupancy



## Total Number of Covid tests carried out vs. Confirmed Cases

### Covid-19 Testing EU/EEA & UK

Reported Date

1/2/2020

10/25/2020

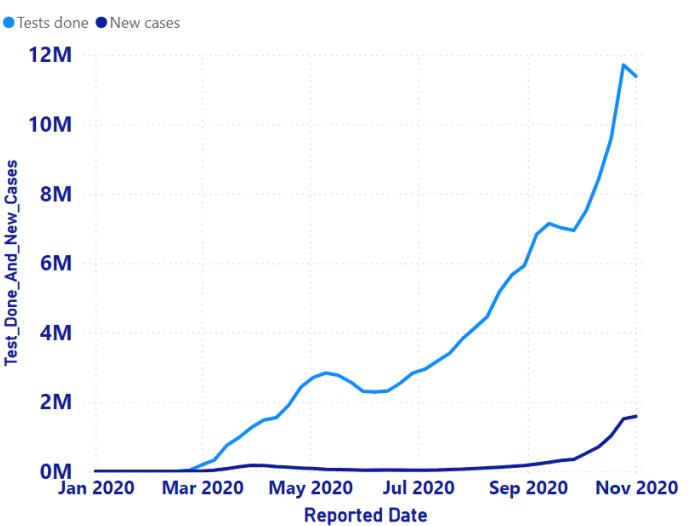
Country

All

Test Done By Country

- country
- United Kingdom (34M (22.79%))
  - Germany (25M (16.73%))
  - France (19M (12.73%))
  - Spain (14M (9.45%))
  - Italy (6M (3.89%))
  - Denmark (5M (3.55%))
  - Belgium (5M (...))
  - Poland (3M (2.23%))
  - Netherlands (2M (1.56%))
  - Portugal (2M (1.1%))
  - Romania (1M (0.75%))
  - Sweden (1M (0.75%))
  - Czechia (1M (0.75%))
  - Austria (1M (0.75%))
  - Norway (0M (0.25%))
  - Greece (0M (0.25%))

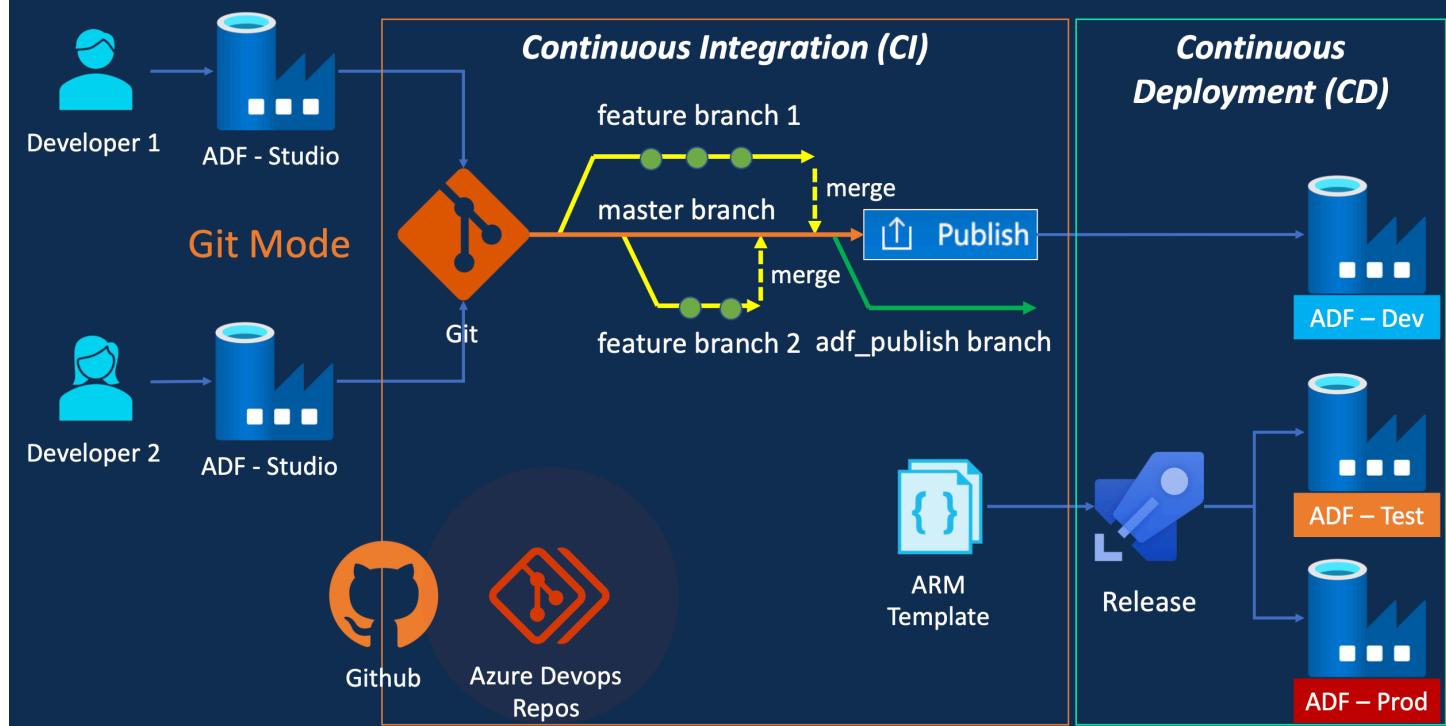
Test done Vs Confirmed Cases



# Continuous Integration / Continuous Delivery ( CI/CD )

## CI/CD Option 1 – Using ADF Publish:

### CI/CD Option 1 – Using ADF Publish



**Phát triển trên Nhánh Tính Năng:** Nhà phát triển làm việc trên nhánh Git riêng và hợp nhất vào nhánh chính khi hoàn tất.

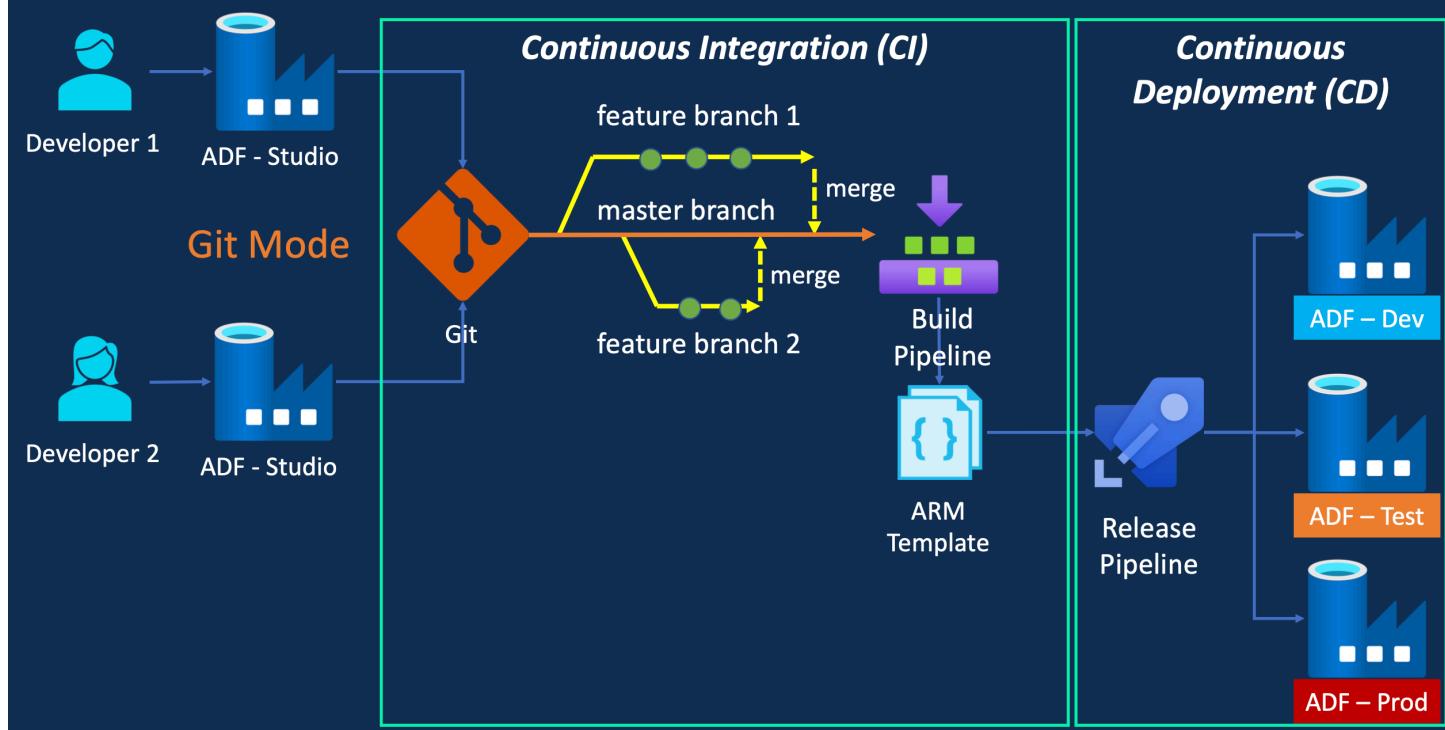
**Tạo Nhánh adf\_publish:** ADF tự động tạo nhánh này với ARM Template, lưu cấu hình ADF.

**Triển khai qua Release Pipeline:** ARM Template từ adf\_publish được triển khai qua các môi trường Test và Prod, có thể tự động hoặc cần phê duyệt.

**Kiểm thử và Giám sát:** Các thay đổi được kiểm thử trên Dev và Test, rồi triển khai lên Prod với giám sát qua Azure Monitor.

## CI/CD Option 2 – Using Build Pipeline:

# CI/CD Option 2 – Using Build Pipeline



Phương thức này sử dụng một build pipeline tự động để kiểm tra và xây dựng ARM template. Các nhánh mã sẽ được hợp nhất và xây dựng tự động để tạo ra các artefact (đối tượng) sẵn sàng triển khai qua các môi trường.

Quy trình tự động triển khai ARM template từ nhánh chính vào môi trường Test và Prod được thực hiện qua release pipeline, giúp quản lý phiên bản và đồng bộ hóa các thay đổi dễ dàng.

## Quy trình:

**Phát triển và Kiểm thử:** Nhà phát triển làm việc trên nhánh Git riêng, kiểm thử pipeline trong môi trường Dev.

**Tạo ARM Templates:** Sau khi hợp nhất các nhánh, build pipeline tạo ra ARM template cho cấu hình ADF.

**Tự động triển khai:** Release pipeline tự động triển khai ARM template qua các môi trường Test và Prod.

## Điểm chính:

- **Tự động hóa** giảm lỗi và tăng nhất quán.
- **ARM Templates** chuẩn hóa cấu hình qua các môi trường.
- **Tích hợp với Azure DevOps** quản lý mã nguồn và giám sát hiệu quả.



# Thank you!