**Azure Data Factory Capstone - Covid Use Case:**

**Document Status: v1.1**

**Date Created: 01/29/2023**

Contents

[1 Overview 3](#_Toc115187810)

[2 Requirements 3](#_Toc115187811)

[2.1 Azure resources 3](#_Toc115187812)

[Create the following Azure resource to create the end-to-end data pipeline 3](#_Toc115187813)

[2.2 Storage 3](#_Toc115187814)

[2.2.1 ADLS 3](#_Toc115187815)

[2.3 Azure Data Factory Configuration 3](#_Toc115187816)

[2.4 Pipeline Structure – 3](#_Toc115187817)

[3 Data Warehouse Table Definitions 4](#_Toc115187818)

[3.1 Data Dictionary – 4](#_Toc115187819)

[4 Transformation 5](#_Toc115187820)

[5 Data Verification 5](#_Toc115187821)

[6 Success Criteria 5](#_Toc115187822)

# Overview

The purpose of the Covid use case exercise is to learn how to build a real-world data pipeline in Azure Data Factory (ADF) to analyze the covid trend across the regions using Azure cloud data services. By performing this case study, you will learn

* How to ingest data from flat files into Azure Data Lake Gen2 and Azure Synapse using Azure Data Factory (ADF)
* How to transform data using Data Flows in Azure Data Factory (ADF) and load into Azure Synapse

Through this exercise, you will be having a hands-on experience on Storage, ADF Pipeline, Mapping Dataflow, Azure Synapse

# Requirements

* Basic understanding about cloud computing will be useful, but not necessary

## Azure resources

## Create the following Azure resource to create the end-to-end data pipeline

|  |  |
| --- | --- |
| **Azure Service** | **Azure Service Name** |
| Resource Group | covid-rg |
| Azure Data Lake Storage Gen2 Account | Covid-adls |
| Azure Data Lake Storage Gen2 Container | Covid |
| Azure Synapse Workspace | Covid-Synapse-Workplace |
| Azure Data Factory | Covid-ADF |

## Storage

### ADLS

Unzip the attached file and upload all the files to the “Covid/Ingest” container



## Azure Data Factory Configuration

* Data Factory- **ADFCovidUseCaseDataFactory**
* Linked Service – LS\_adls, LS\_dw
* Data Set – DS\_adls, DS\_dw

## Pipeline Structure –

You may follow different proper naming conventions. One example given below:

DataFactory – **ADFCovidUseCaseDataFactory**

1. ***CopyCovidData-ADLS-SQL-Pipeline:***

The pipeline will be configuration managed. The pipeline will able to copy the datasets in ADLSGen2 dynamically in a loop. The pipeline will have the following activities.

* Lookup Activity: Look Up the names of the files to be copied.

This activity will invoke a Store Procedure (*Suggested Name – sp\_GetConfigSrcFileNames*). The Stored Procedure will get the file names from a configuration table (*Suggested Name – config\_srcfileNames*).

**Source Dataset**

Name: configDataSet

LinkedService – dwLinkedService

* ForEach Activity: Loop through the file names fetched by the Lookup activity and invoke the copy activity for each file name.
* Copy Activity: Copies data from datalake to datawarehouse:

**Source Dataset:**

Name:  adlsV2Dataset

LinkedService – adlsLinkedService

**Target Dataset:**

Name:  dwDataset

LinkedService – dwLinkedService

# Data Warehouse Table Definitions

## Data Dictionary –

**Dbo.DeathsInUKAndIndia**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Technical Name | Column Datatype | Nullable? | PK? | FK? | Column Notes |
| Country | Varchar | Yes |  |  |  |
| CountryCode | Varchar | Yes |  |  |  |
| Continent | Varchar | Yes |  |  |  |
| Population | BigInt | Yes |  |  |  |
| Indicator | Varchar | Yes |  |  |  |
| DailyCount | BIgInt | Yes |  |  |  |
| Date | Date | Yes |  |  |  |
| Rate\_14\_Day | Decimal | Yes |  |  |  |
| Source | Varchar | Yes |  |  |  |

**Dbo.AllOverDeaths**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Technical Name | Column Datatype | Nullable? | PK? | FK? | Column Notes |
| Country | Varchar | Yes |  |  |  |
| CountryCode | Varchar | Yes |  |  |  |
| Continent | Varchar | Yes |  |  |  |
| Population | BigInt | Yes |  |  |  |
| Indicator | Varchar | Yes |  |  |  |
| DailyCount | BIgInt | Yes |  |  |  |
| Date | Date | Yes |  |  |  |
| Rate\_14\_Day | Decimal | Yes |  |  |  |
| Source | Varchar | Yes |  |  |  |

**Dbo.CountryWiseResponseMeasure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Technical Name | Column Datatype | Nullable? | PK? | FK? | Column Notes |
| Country | Varchar | Yes |  |  |  |
| ResponseMeasure | Varchar | Yes |  |  |  |
| DateStart | Varchar | Yes |  |  |  |
| DateEnd | Varchar | Yes |  |  |  |

**Dbo.HospitalAdmissions**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Technical Name | Column Datatype | Nullable? | PK? | FK? | Column Notes |
| Country | Varchar | Yes |  |  |  |
| Indicator | Varchar | Yes |  |  |  |
| Date | Datetime | Yes |  |  |  |
| Year\_Week | Varchar | Yes |  |  |  |
| Value | Decimal | Yes |  |  |  |
| Source | Varchar | Yes |  |  |  |
| URL | Varchar | Yes |  |  |  |

**Dbo.Testing**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Technical Name | Column Datatype | Nullable? | PK? | FK? | Column Notes |
| Country | Varchar | Yes |  |  |  |
| CountryCode | Varchar | Yes |  |  |  |
| Year\_Week | Varchar | Yes |  |  |  |
| New\_Cases | BigInt | Yes |  |  |  |
| Tests\_Done | BigInt | Yes |  |  |  |
| Population | BigInt | Yes |  |  |  |
| Testing\_Rate | Decimal | Yes |  |  |  |
| Positivity\_Rate | Decimal | Yes |  |  |  |
| Testing\_Data\_Source | Varchar | Yes |  |  |  |

# Transformation

Get the data from ingest path and after transformation using mapping dataflow store in a ADLS different storage container named ‘transformPath’. We need to do Aggregate and Rank transformation to find the continents which have max daily death data in a single file in ADLS.

# Assuming you have already read the data from the ingest path and transformed it using mapping dataflow

# Let's assume the transformed data is stored in a DataFrame called "transformed\_data"

# Aggregate by continent and find the max daily death data

aggregated\_data = transformed\_data.groupby("continent")["daily\_deaths"].max()

# Rank the aggregated data

ranked\_data = aggregated\_data.rank(ascending=False)

# Create a new DataFrame with continent, max daily deaths, and rank

result\_df = pd.DataFrame({

"continent": aggregated\_data.index,

"max\_daily\_deaths": aggregated\_data.values,

"rank": ranked\_data.values

})

# Save the result DataFrame to a CSV file in the "transformPath" ADLS container

result\_df.to\_csv("adls://transformPath/result.csv", index=False)

# Print a success message

print("Aggregated and ranked data saved to result.csv in the 'transformPath' ADLS container.")

# Data Verification

|  |  |  |
| --- | --- | --- |
|  | **How?** | **Description / Notes** |
| **Step 1** | Open Synapse Studio and Open a new Query editor | Write a Query to check the confirmed cases in India in March’2020 and verify with source data in excel |
| **Step 2** | Open Synapse Studio and Open a new Query editor | Write a query to show country wise confirmed case in 2020 |

# Success Criteria

Should be able to create the below Azure Cloud Resources:

* Azure data lake storage account
* Container within Azure data lake storage
* Upload the attached files in container
* Create Azure Synapse workspace and dedicated pool
* Create required tables in dedicated pool
* Create Azure Data Factory Pipeline to load all the files into Azure Synapse dedicated pool
* Query the loaded data to generate meaningful reports