# **Building an ETL Pipeline using Azure Data Services**

**DESCRIPTION**

Use the data analytics stack to build a data pipeline using Data Factory, Databricks and Synapse.

**Problem Statement:**

As a Data Engineer, you’ve been asked to access the services that can help with ETL of data in the cloud data storage to enable analytics through Synapse. In this POC, we will be collecting the data from SQL Database using ADF and the transformed data will be the source for databricks to run complex transformations and once data is analysed using Databricks, it is synced into synapse analytics data warehouse as historical dataset for enabling various analytics.

# **Overview Of Task 1**

Graphical user interface, application

Description automatically generated

**Prerequisites for Task 1:**

To develop a ETL (Extract Transform Loading) Assignment, we need some prerequisites for performing the following operation.

Create a Resource Group so that I can perform my task inside the resource. Create a resource Group named ETLAssignment as shown in below screenshot.

Graphical user interface, text, application, email

Description automatically generated

Create a data factory workspace, using the workspace I can perform my data flow and pipeline activity

Configure Data Factory as below.

Graphical user interface, text, application

Description automatically generated

Once configured and Workspace is created, we can see the below screen.

Graphical user interface, text, application

Description automatically generated

Create Azure Data Lake Storage Account and place the input as below

**In Azure 🡪Go To Storage Account🡪Create🡪Select Resource Group (ETLAssignment)🡪give Storage Account name🡪select Redundancy as LRS🡪In Advanced tab Enable hierarchical namespace🡪Review and Create .**Refer Below Screenshot.

Graphical user interface, text, application

Description automatically generated

Once ADLS Gen 2 Account Created .we can see the below screen.

Graphical user interface, application, email

Description automatically generated

Create a Storage Container to store the data inside Storage account and place the input movies.csv file there

Graphical user interface, text, application, email

Description automatically generated

To store the output of CountMoviesBasedOnGenre data Stream ,we need Azure Data Lake Storage Gen 1

Graphical user interface, application

Description automatically generated

The output of JoinMovieCountWithCleanData should be stored in SQL Database, so need to create new SQL db and server

Graphical user interface, text, application, email

Description automatically generated

Creating SQL DB named **ETLDB** and creating new server for this resource group.

Server name – etladmin

Pass – Visi@2022

Graphical user interface, application

Description automatically generated

Configuring as shown below

Graphical user interface, application

Description automatically generated

After deployment gets completed, adding client IP in Firewall settings for connecting with Studio

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Connecting server with studio for opening Database in studio and to create table

Below Query is used for creating table Movie\_Clean\_Data

Graphical user interface, text, application

Description automatically generated

For Using in Linked Service, need to give YES in Allow Azure service and resource to access server

Graphical user interface, text, application

Description automatically generated

**Task 1: Create a dataflow with the following requirement:**

1.Create a data stream named CleaningGenreRomance and perform data cleansing on the Genre column using Derived Column and case expression. (While collecting data it was observed that some genres have spelling mistakes like romance, Romence for Romance, comedy, Comdy for Comedy.)

Steps:

Create a data flow and add source by creating new dataset ADLS gen2 with delimited text format and using linked service to connect with etladlsgen2 storage account for input data

Graphical user interface, application

Description automatically generated

Connecting with new linked service with below information

A screenshot of a computer

Description automatically generated

The input data location is selected in wildcard path under source option

Graphical user interface, text, application

Description automatically generated

The datatype and format of each column should be mentioned accurately in projection

A screenshot of a computer

Description automatically generated

The input data can be viewed from Data Preview if needed.

Graphical user interface, text, application, email

Description automatically generated

Adding derived Column from sourceMovie named **derivedColumn1**

The below condition is used in settings for getting cleansed data

Case(Genre ==’romance’, ‘Romance’, Genre == ‘Romence’, ‘Romance’, Genre == ‘Comdy’ , ‘Comedy’)

Graphical user interface, application

Description automatically generated

The output of derivedColumn1 will be cleansed data with 8 column. It can be viewed in datapreview

A screenshot of a computer

Description automatically generated

Point 1 Completed..

2. Create a data stream named CountMoviesBasedOnGenre that can calculate number of films for each genre and store it as a separate dataset in ADLS under folder name “solution/genreCount”

Cloning derivedColumn1 from previous one and considering that cleansed data as a input for counting number of films

Adding pipeline named Aggregate1 which will calculate the number of films for each genre.

Under group by => genre is selected

Under Aggregate => column name is filmCount and expression is count(film).

Graphical user interface, application

Description automatically generated

And the Output is shown as Below with Film Count.

Graphical user interface, text, application, email

Description automatically generated

And Now to store the output in adls gen 1

Create a sink with new linked service with storage account name etladlsgen1

Graphical user interface, text, application, email

Description automatically generated

To mention the destination path for gen1 account Dataset > ADLSGen1 > under connection mention file path as shown below

Graphical user interface, text

Description automatically generated

Creating a sink with outputStream name **CountMovieBasedonGenre**

Graphical user interface, application

Description automatically generated

Selecting a destination dataset ADLSgen1

Under settings, filename options is Output to single file and the name given as MovieCount

Graphical user interface, text, application, email

Description automatically generated

Now the Output with two column is stored in Adls Gen1 Folder

Point 2 completed….

3. Create a new stream named JoinMovieCountWithCleanData. Perform join operation on CountMoviesBasedOnGenre with CleaningGenreRomance stream and store the same in the Azure SQL Database.

Performing join operation where left stream is CleaningGenreRomance and right stream is aggregate1

Join type is InnerJoin and condition is CleaningGenreRomance@Genre == genre

Graphical user interface, application

Description automatically generated

Now the Output of join operation Contains 10 Column(2 genre column) ,So using **RemoveColumn** remove the duplicate Column and now there is 8 column

Graphical user interface, application

Description automatically generated

And the output of removecolumn is passed in map drifted where in derived column settings, the output stream name is MapDrifted1, incoming stream is RemoveColumn1

Condition is represented as **toString(byName(‘Genre’)) ,**using map Drifted add the Genre Column ,now including FilmCount there is total 9 column

A screenshot of a computer

Description automatically generated

Create a sink with output stream name JoinMovieCountWithCleanData, incoming is mapdrifted

Using the created server information in linked service and mentioning Database name

A screenshot of a computer

Description automatically generated

Under sink dataset, add a destination sql table name

A screenshot of a computer

Description automatically generated

Using Mapping map the input with the Table column name properly

Graphical user interface, application

Description automatically generated

And Now To execute the data Flow and the data flow in the pipeline with proper settings

Graphical user interface, application

Description automatically generated

Now Validate and Publish ,once published with out error ,then trigger now .

Once the Data Flow is triggered, will see the below

As per requirement output of CountMoviesBasedOnGenre is stored under adls gen1 under folder name “solution/genreCount”

Graphical user interface, text, application, email

Description automatically generated

And the output of JoinMovieCountWithCleanData as per required we can see the data stored in e Azure SQL Database under table name Movie\_clean\_data.

A screenshot of a computer

Description automatically generated with medium confidence

-------------------------------------TASK 1 COMPLETED-------------------------------------

# **Overview Of Task 2**

**A picture containing diagram

Description automatically generated**

**Prerequisites for Task 2.**

Create a Azure Data Bricks Workspace.

Graphical user interface, text, application, email

Description automatically generated

Launch the data brick Workspace and Create a new Cluster like below.

Graphical user interface, text, application

Description automatically generated

And the cluster created and shown as below

Graphical user interface, text, application

Description automatically generated

Create a notebook with the created cluster

Graphical user interface, application

Description automatically generated

Create a Synapse Workspace to store the output of task 2 in Sql Pool

Graphical user interface, text, application, email

Description automatically generated

Create a new Sql Pool with the following configuration

Graphical user interface, text, application

Description automatically generated

Once Created ,Connect the Sql Pool via SSM using dedicated sql endpoint.Create a Table in the Sql pool as follows

Graphical user interface, text

Description automatically generated

**Task 2: Create the following activity pipeline**

Create a pipeline CopyDataFromSqlToADLSGen2 and configure as belows,To copy data from input of task 1 from SQL database to the ALDS GEN2 Account folder.

Graphical user interface, text, application

Description automatically generated

And Also configure the output path to store the data in adls gen2 to store sql dab data.

A screenshot of a computer

Description automatically generated

Link CopyDataFromSqlToADLSGen2 with new pipeline CopyDataFromSqlToGen1,Only After this activity Rank Operation must be performed in data bricks as per requirement

Graphical user interface

Description automatically generated

Configure Sink to store the data in adls gen1

Graphical user interface

Description automatically generated

Before Linking Databricks with CopyDataFromSqlToGen1 .Login with DataBricksand open the created notebook and perform the rank operation and store the output in adls gen1 folder and save the notebook with name EtlProjectNotebook

**NOTEBOOK:** **EtlProjectNotebook**

from pyspark.sql import SparkSession

from pyspark.sql.types import \*

from pyspark.sql.window import Window

from pyspark.sql.functions import rank

from pyspark.sql.functions import row\_number

account\_name = "etladlsgen2"

container\_name = "task2output"

relative\_path = "rankoutput"

adls\_path = 'abfss://%s@%s.dfs.core.windows.net/%s' % (container\_name, account\_name, relative\_path)

spark.conf.set("fs.azure.account.auth.type.%s.dfs.core.windows.net" %account\_name, "SharedKey")

spark.conf.set("fs.azure.account.key.%s.dfs.core.windows.net" %account\_name ,"6IuB2nl5g7+x6syTpT2DLKPB2c5uSPRKfFGqQdKcwTrU/NFL1v9pmarTJWAN1KjwTImZnb4LOzet+AStCXG0+w==")

df1 = spark.read.option('header', 'true') \

.option('delimiter', ',').option('inferSchema', 'true')\

.csv(adls\_path)

Window\_Spec = Window.partitionBy("Genre").orderBy("Rotten\_Tomatoes\_in\_pcent")

Result=df1.withColumn("rank",rank().over(Window\_Spec))

spark.conf.set("fs.azure.account.key.etladlsgen1.blob.core.windows.net","yqcUoV0fdn88gifc5ANMhA/bRUq+m5JRv9EGslFoIQDoj7vz6f3fb0fcxQRt8MmGlZvdMZ+q1Qj2+AStkHUSKA==")

Result.write.csv("wasbs://countmovies@etladlsgen1.blob.core.windows.net/Fold/AzureCostAnalysis.csv", header="true")

Graphical user interface, text, application

Description automatically generated

Link DataBricks with the CopyDataFromSqlToGen1,Once the above activity is performed,run the configured databricks by using linked service add databricks notebook **EtlProjectNotebook with ETLCLuster**

Graphical user interface, text, application, email

Description automatically generated

When the above data bricks executed the output will be stored under adls gen1

Now link the databricks pipeline with copydata pipeline wich transfer data from adlsgen1 to synapse created SQLPOOL :**ETLSYNAPSESQLPOOL**

Under Source tab configure the adlsgen1 path where the output of databricks stored

Graphical user interface, application

Description automatically generated

Under the sink Tabe Configure the synapse database and the sqlpool table name

Graphical user interface, text, application, email

Description automatically generated

Under mapping map the input of adls gen1 with the synapse table name

Graphical user interface, application

Description automatically generated

Under Settings enable staging with linked service name configured as below

Graphical user interface

Description automatically generated

And now validate and publish the task 2 .once published trigger now the pipeline

And now the output of CopyDataFromSqlToADLSGen2 ,which is copied from sql database from task1 to adls gen2

A screenshot of a computer

Description automatically generated

And now the output of CopyDataFromSqlToADLSGen1 ,which is copied from sql database from task1 to adls gen1 ,once this activity done then only the databricks must be performed

Graphical user interface, text, application, email

Description automatically generated

And the output of rank operation of data bricks is stored under adls gen1 as mentioned

Graphical user interface, text, application, email

Description automatically generated

And atlast the data of rank operation stored in adls gen1 which is copied to synapse sqlpool database with table name Movie.

Graphical user interface, application, table

Description automatically generated

--------------------------------------------------------Task 2 Completed--------------------------------------------------

-------------------------------------------------------Project Completed---------------------------------------------------