**DAMG 7275 Project Group 4**

**P3 Implementation**

**Netflix Data Analysis – Cosmos DB Multi-Model**

**Implementation Process (Document Database):**

1. The data source in our project will be a CSV file containing data of all Netflix Shows and Movies.
2. The columns contained in the CSV file are (show\_id,type,title,director,cast,country,date\_added,release\_year,rating,duration,listed\_in,description)
3. This file is first dropped in AWS S3 bucket (input Bucket).
4. On the occurrence of above event, an AWS Lambda Function will be triggered, which processes the CSV file, makes certain necessary transformations in the file, converts the data in Document format and saves in a JSON file in another S3 bucket (Output Bucket).
5. On the other hand, we have an Azure Data Factory, Data Flow in place which executes regularly after a fixed time period.
6. This Data Flow reads the JSON file from S3 Output Bucket and loads that data into Cosmos DB Document Database.
7. The entire process is set up such that it inserts every new record and updates any existing record with any changes in their data values.
8. This document data is further utilized by the visualization tool, PowerBI to fetch necessary KPIs and insights from the data.

**Screenshots (Document Database):**

**S3 Buckets :**

**A screenshot of a web page

Description automatically generated with medium confidence**

**AWS Lambda Function :**

A screenshot of a computer

Description automatically generated with medium confidence

**AWS Lambda Code:**

import boto3

import csv

import json

import datetime

def lambda\_handler(event, context):

# S3 bucket names

source\_bucket = 'kd-project-input-bucket'

destination\_bucket = 'kd-project-output-bucket-final'

# Input and output file names

source\_file\_name = 'sample.csv'

destination\_file\_name = 'output.json'

# Create S3 client

s3 = boto3.client('s3')

try:

# Read CSV file from source bucket

response = s3.get\_object(Bucket=source\_bucket, Key=source\_file\_name)

csv\_data = response['Body'].read().decode('utf-8-sig').splitlines()

# Convert CSV to JSON with field transformation

csv\_reader = csv.DictReader(csv\_data)

json\_data = []

for row in csv\_reader:

# Perform field transformation for 'director', 'cast', 'country', and 'listed\_in' fields

directors = row['director'].split(',')

row['director'] = [director.strip() for director in directors]

casts = row['cast'].split(',')

row['cast'] = [cast.strip() for cast in casts]

countries = row['country'].split(',')

row['country'] = [country.strip() for country in countries]

listed\_ins = row['listed\_in'].split(',')

row['listed\_in'] = [listed\_in.strip() for listed\_in in listed\_ins]

# Convert date\_added format if necessary

date\_added = row['date\_added'].strip()

if date\_added and not date\_added.isdigit():

date\_added = datetime.datetime.strptime(date\_added, '%B %d, %Y').strftime('%d-%b-%y')

row['date\_added'] = date\_added

json\_data.append(row)

# Save JSON file to destination bucket

s3.put\_object(Body=json.dumps(json\_data), Bucket=destination\_bucket, Key=destination\_file\_name)

# Delete the source CSV file

s3.delete\_object(Bucket=source\_bucket, Key=source\_file\_name)

return {

'statusCode': 200,

'body': 'File converted and saved successfully'

}

except Exception as e:

return {

'statusCode': 500,

'body': f'Error: {str(e)}'

}

**Below is the Azure Data Factory Pipeline:**

A screenshot of a computer

Description automatically generated

**Loaded first two documents in Cosmos DB:**

A screenshot of a computer

Description automatically generated with medium confidence

**New incoming csv file contains record “s1” with a change and a new record “s3”. Below two screenshots showing proper UPSERT happening in Cosmos DB:**

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

**Below two screenshots showing periodic trigger enabled on Azure Data Factory Pipeline:**  
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence