**Data Engineer - Technical Assessment**

**Approach**:

Let’s we data coming from livestreaming in the given format.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Row key** | **user\_id** | **timestamp** | **click\_data\_url** | **country** | **city** | **browser** | **os** | **device** |
| 1 | user123 | 26-06-2023 10:12 | example.com/page1 | USA | New York | Chrome | Windows | Desktop |
| 2 | user456 | 26-06-2023 11:15 | example.com/page2 | USA | Los Angeles | Safari | iOS | Mobile |
| 3 | user789 | 26-06-2023 12:30 | example.com/page1 | UK | London | Firefox | Windows | Desktop |
| 4 | user123 | 26-06-2023 13:45 | example.com/page3 | Canada | Toronto | Chrome | macOS | Laptop |
| 5 | user456 | 26-06-2023 14:20 | example.com/page1 | USA | New York | Chrome | Windows | Desktop |
| 6 | user789 | 26-06-2023 15:55 | example.com/page2 | UK | London | Safari | iOS | Mobile |

We need to install dependencies :

Pyspark

kafka-python

azure-cosmos

we have created python file with script.py

to execute the kafka live streaming :

spark-submit --master local[2] --packages org.apache.spark:spark-streaming-kafka-0-8\_2.11:2.0.2 script.py

**Analysis of Data:**

To analyze the number of clicks, unique users, and average time spent on each URL by users from each country, you can use the data stored in Azure Cosmos DB and perform the required calculations

# Create a SparkSession

spark= SparkSession.builder.master("local").appName("ClickstreamAnalysis").getOrCreate()

# Load data from Azure Cosmos DB into a Spark DataFrame

data = spark.read.format("cosmos.oltp").option("spark.cosmos.container", container\_name).load()

# Register the DataFrame as a temporary view

data.createOrReplaceTempView("clickstream")

# Perform analysis using Spark SQL: number of clicks, unique users, and average time spent on each URL by users from each country

analysis\_df = spark.sql("""

SELECT country, url, COUNT(url) AS clicks, COUNT( DISTINCT user\_id ) AS unique\_users, AVG(timestamp) AS avg\_time\_spent

FROM clickstream

GROUP BY country, url

""")

# Show the analysis results

analysis\_df.show()

# Stop the SparkSession

spark.stop()