Cloud Computing Project 2

Data Processing step:

- This code performs a comprehensive analysis of Twitter hashtags using Apache Spark. It begins by creating a Spark session to facilitate distributed data processing and then reads JSON tweet data from an S3 bucket.
- The core functionality revolves around a series of functions that extract hashtags from various text fields within the DataFrame, including user descriptions and tweet content.
- By cleaning the text and using regular expressions, the code identifies hashtags and combines them into a single DataFrame for further analysis. It then counts the occurrences of each unique hashtag, ultimately writing the top 20 hashtags and their corresponding counts to a specified CSV file in the S3 bucket.
  A screenshot of the output is attached here:

	A	B	C
1	hashtag	count	
2	#melbourn	1739	
3	#stkilda	962	
4	#australia	502	
5	#beach	431	
6	#ausgp	358	
7	#love	313	
8	#f1	303	
9	#sunset	273	
10	#portmelb	226	
11	#votearian	212	
12	#stkildabe	210	
13	#albertpar	206	
14	#buybreak	204	
15	#summer	196	
16	#italianalit	110	
17	#food	96	
18	#victoria	94	
19	#travel	91	
20	#coffee	89	
21	#repost	86	

- Additionally, the top hashtags are displayed for immediate review, making the code a powerful tool for understanding trending topics within the tweet dataset.

Overview of the steps performed on AWS:

- To create the EMR cluster on AWS, I first initiated an Amazon EMR cluster configuration using two **m5.xlarge** instances for the primary and core nodes.
- The **m5.xlarge** instances were chosen for their balanced compute, memory, and networking resources, making them suitable for data processing tasks. This decision was made after encountering performance issues with **c5.xlarge** instances, which resulted in excessively long execution times for the processing steps.
- After configuring the cluster specifications, I added a step named **p2-step-Aditya Sambhaji Jadhav** to the cluster that included my Python script. This script is designed to perform hashtag analysis on Twitter data, processing the input from an S3 bucket.
- Upon execution, the results are automatically saved in the S3 bucket **p2dataadityajadhav**, specifically within the **/output** folder. This setup allows for efficient handling of large datasets using Apache Spark on the EMR platform.