Cloud setup steps:

Below is the sequence of steps followed in setting up the cloud for this project.

- An Amazon **AWS** account was created initially [1], and a new virtual machine is created.
- A new instance is created and SSH, MySQL, HTTP type of connections were added to it and the source IPs were configured as 0.0.0.0/0 and ::/0 for all three types. The ports 8080, 8081 on AWS Linux machine instance were enabled.
- Connection to the **EC2** instance was created using the terminal in local machine.
- Installed java, **python3, Apache spark and Mongo DB** in the EC2 instance using the tutorial 4,5 from the labs [2]. Master and slave nodes were started in the EC2 instance.
- **Pymongo** is installed to store the extracted tweets and also to retrieve them from MongoDB during MapReduce.

Data extraction process:

1. Extraction of articles from sgm files
   - As there are 1578 articles in the two given sgm files, each of the article is stored in a separate file.
   - A python script was written to parse the sgm files and find the <text> and </text> tags in the provided sgm files. The data between <text>, </text> tags was identified and written to a new text file using a python logic.
   - Total of 1578 files generated from the two sgm files and are stored in a directory separating from rest of the file structure.
   - The python script used for parsing the sgm files **(SgmFiles_parsing.py**) is provided in the zip folder.
2. Extraction of tweets from twitter
   - A twitter developer account was created initially.
   - A Twitter application is created, and keys & tokens are generated for getting access to extract the tweets from twitter.
   - A total of 2000 tweets have been extracted from twitter using the tweepy streaming and searching APIs in python script [4]. The keywords “Canada”, “Canada import”, “Canada export”, “Canada vehicle sales”, “Canada Education” are used for finding the tweets.
   - Tweets and retweets have been extracted along with their metadata like location, time, tweet id, username, etc.,
   - The tweets along with their metadata have been inserted into mongo DB database after cleaning the tweet’s text.

- The logic used for extracting the tweets and storing them in Mongo DB is provided in **Tweets_extraction.py** python script which is included in the submitted zip folder.

Cleaning process:

All the tweets extracted using the streaming API and searching API are cleaned before storing in the Mongo DB database.

- When the tweet is extracted using the tweepy streaming/searching API, first all the url's present in the tweet's text are removed using the python script.
- Tweet's text is analysed to check if there are any emoticons are present and all the emoticons are removed from the text of a tweet before inserting into the database.
- The python code used to clean the tweets is provided in **Tweets_extraction.py** file in the provided zip folder.

Data processing:

- The tweets that are stored in MongoDB are retrieved using python script **(tweets_from_MongoDB.py)** and stored in a separate file along with the 1578 articles in the same directory.
- I have written Pyspark script (**MapReducer.py)** to calculate the word count of all the provided key words from 2000 tweets extracted from twitter and 1578 articles extracted from the provided 2 SGM files. The key words for which word count is calculated: "oil", "vehicle", "university", "dalhousie", expensive" , "good school" or "good schools" , "bad school" or "bad schools" or "poor school" or "poor schools" , "population" , "bus" or "buses", "agriculture" , "economy".
- In this Pyspark script, all the 1579 files (1578 articles and tweets file) are looped one by one to extract the words that are matched with above list of key words. Then MapReduce is performed on these words to find out the word count [5].
- The output file **(WordCount_output.txt)** with all the frequencies of given words is included in the submitted zip folder.

Sample JSON/XML/or any other formats of data file:

- Sample JSON file (**Extracted_tweets.json**) with tweets extracted using search and streaming API's is included in the zip folder.
- Sample text file **(Sample_ article _extracted from SGM file.txt**) with article extracted from provided SGM file is included in the zip folder.

References:

[1] Dal.brightspace.com, 2019. [Online]. Available: https://dal.brightspace.com/d2l/le/content/98340/viewContent/1333088/View. [Accessed: 01- Jul- 2019].

[2] Dal.brightspace.com, 2019. [Online]. Available: https://dal.brightspace.com/d2l/le/content/98340/viewContent/1335327/View. [Accessed: 01- Jul- 2019].

[3] Dal.brightspace.com, 2019. [Online]. Available: https://dal.brightspace.com/d2l/le/content/98340/viewContent/1337724/View. [Accessed: 01- Jul- 2019].

[4]"Standard search API", Developer.twitter.com, 2019. [Online]. Available: https://developer.twitter.com/en/docs/tweets/search/api-reference/get-search-tweets.html. [Accessed: 01- Jul- 2019].

[5]"PySpark Word count Program – Geoinsyssoft", Geoinsyssoft.com, 2019. [Online]. Available: http://geoinsyssoft.com/pyspark-wordcount-program/. [Accessed: 01- Jul- 2019].