Assignment 3

CSCI 5408 - Data Warehousing, Management And Analytics

Submitted by: Nishant Amoli
Banner ID: B00835717

Submitted to: Dr. Saurabh Dey
Faculty of Computer Science
Dalhousie University

Abstract

Cloud computing has brought an innovative change in data storage. It has also been proved very effective in data warehousing. [1] Using cloud databases, a virtualized computer environment can be designed. This can eliminate the need of buying hardware and software resources at an exorbitant price. The resources that are provided by cloud service providers such as Amazon, can simply be used in order to achieve the objective. Many e-commerce websites are actually using cloud services in order to improve their performance.

Introduction

This assignment comprises the fundamentals of Big Data and NoSQL. It is subdivided into various tasks that cover the concepts of extracting data using APIs, cleaning and transformation of data, storing data in NoSQL database (MongoDB) remotely on Amazon Cloud, and finally process the data using MapReduce. Apart from the data processing and working on NoSQL database, this assignment requires the setup of an Amazon cloud account and environment.

Section A - Cluster Setup

1. **Creating Amazon Cloud account and setting up an instance**

   Following are the steps to create an Amazon cloud account as well as creating an EC2 instance [2]:

   1. Visit https://aws.amazon.com. After creating a free tier account, log into AWS account.

   2. Choose “Launch a Virtual Machine with EC2”.

   3. After selecting the “Free Tier” option, click on “Ubuntu Server 18.04 LTS (HVM), SSD Volume Type”.

   4. Click on “Next: Configure Instance Details”, then select “Next: Add Storage”.

   5. Change the size of the volume to 16 gb and then add tags providing suitable key-value pairs.

   6. Add SSH, MySQL and HTTP type of connections and for all the connections, configure the source IPs as 0.0.0.0/0 and ::/0.

7. Now create a new key pair. After this, a private key (*.pem) will be generated which can be used to connect to the VM.

8. Download the key-value pair.

9. The EC2 instance will be successfully created.

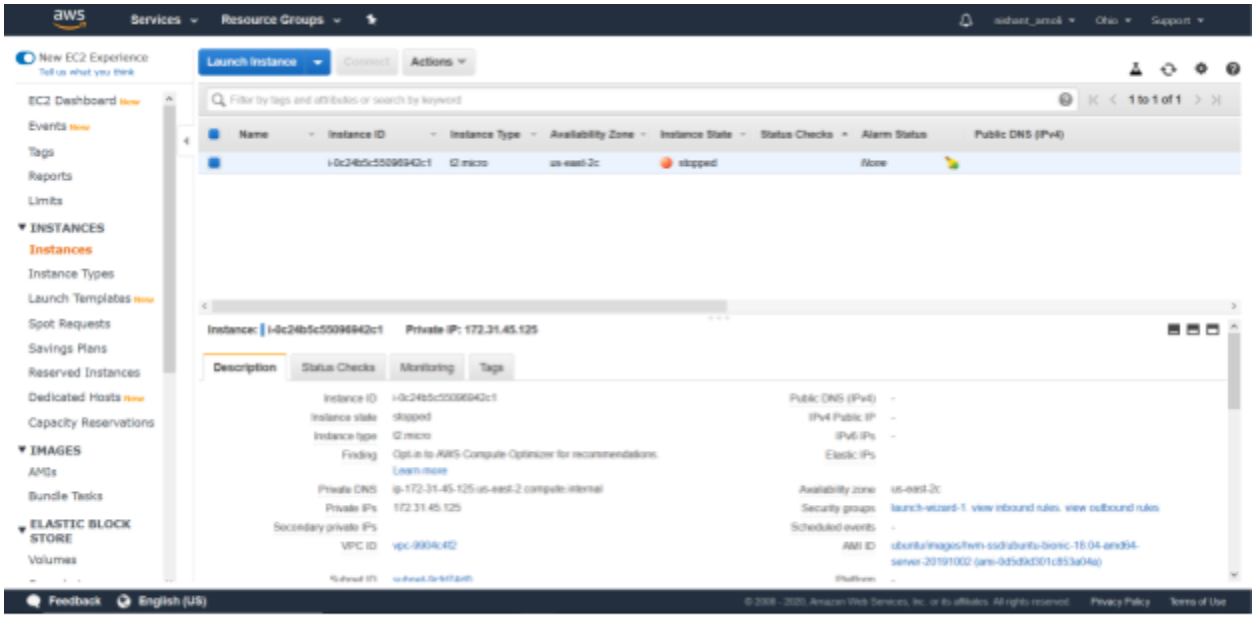

Figure 1: Figure showing an instance on the AWS EC2 Dashboard.

2. Initializing Apache Spark on Amazon Cloud account

Following are the steps to initialize Apache Spark on Amazon cloud account [3]:

1. Run the following commands to add Java PPA to apt:
   a. sudo add-apt-repository –y ppa:webupd8team/java
   b. sudo apt-get update

2. Run the following command to install Oracle or Open JDK:
   a. sudo apt-get –y install openjdk-8-jdk-headless

3. Run the following command to install Python:
   a. sudo apt-get install python3

4. Using the following command, create a directory to install spark:
   a. mkdir server
   b. cd server

5. Run the following commands to download and unpack Apache spark:
    a. wget http://apache.forsale.plus/spark/spark-2.4.4/spark-2.4.4-bin- hadoop2.7.tgz
    b. sudo tar zxvf spark-2.4.4-bin-hadoop2.7.tgz

6. Run the following commands to export path variables in ~/.profile:
    a. sudo nano ~/.profile
    b. export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/
    c. export SPARK_HOME=~/server/spark-2.4.4-bin-hadoop2.7
    d. export PYSPARK_PYTHON=python3

7. Start master node using the following command:
    a. sudo ./spark-2.4.4-bin-hadoop2.7/sbin/start-master.sh

8. Go to <IPv4 Public IP>:8080 in the browser and copy the URL

9. Run the following command to start slave node
    a. sudo ./spark-2.4.4-bin-hadoop2.7/sbin/start-slave.sh spark://ip-< paste the IP copied in the previous step here >

3. Initialling MongoDB on Amazon Cloud instance

Following are the steps to install MongoDB on Amazon cloud instance [4]:

1. Run the following command to import the MongoDB public GPG key:
    a. wget -qO - https://www.mongodb.org/static/pgp/server-4.2.asc | sudo apt-key add -

2. Run the following command to create a list file for MongoDB on Ubuntu:
    a. echo "deb [arch=amd64,arm64] https://repo.mongodb.org/apt/ubuntu bionic/mongodb-org/4.2 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.2.list

3. Run the following command to reload the local package database:
    a. sudo apt-get update

4. Using the following command, install the latest version of MongoDB:
    a. sudo apt-get install -y mongodb-org

5. Now that MongoDB is installed, run the following command to start the mongod process:
    a. sudo systemctl start mongod

6. Run the following command to verify the status of the mongod process:
    a. sudo systemctl status mongod

7. Run the following command to stop the process:
    a. sudo systemctl stop mongod

Section B - (i) Twitter Data Extraction and Transformation

In order to use tweepy API to extract the data from Twitter, a developer account needs to be created. After that a new application can be created followed by generating api keys which will be used to authenticate any request to get the data from Twitter.

The python script used to extract the tweets using the Cursor object can be found in the folder ***csci5409-a3 Scripts*** with the name ***TwitterApi.py***.

The tweets are extracted based on the keywords “Canada”, “University”, “Dalhousie University”, “Halifax” and “Canada Education”. A total of 3165 tweets are being extracted and the raw and unprocessed data is being stored in the json format in the file named ***tweets.json*** within the folder ***json files***. This file is then used in order to read the stored data and retrieve only the filtered fields “name”, “screen_name”, “location”, “created_at” and “full_text”. After this the tweets are cleaned, all the unicode characters, smileys, special symbols and websites are removed and the cleaned tweets are stored in the file named ***cleaned_tweets.json*** in the same folder.

Section B - (ii) News Article Data Extraction and Transformation

Just like twitter, the data from https://newsapi.org/ can be extracted after creating a developer account and generating an api key. However, the free tier account will not only allow to extract the news articles beyond the period of one month.

The python script used to extract the news articles using the GET request can be found in the folder ***csci5409-a3 Scripts*** with the name ***NewsApi.py***.

The news articles are extracted based on the keywords “Canada”, “University”, “Dalhousie University”, “Halifax”, “Canada Education”, “Moncton” and “Toronto”. As there is a limit on extracting the articles using this api, only the 100 articles are extracted and after filtering the data of the following attributes: “author”, “title” and “description” are stored in the json format in the file ***news.json*** within the folder ***json files***. After this the news articles are cleaned, all the

unicode characters and special symbols are removed and the cleaned news are stored in the file named ***cleaned_news.json*** within the same folder.

Section B - (iii) Movie Data Extraction and Transformation

Data related to movies and tv series such as plot, poster, e.t.c. can be retrieved from http://www.omdbapi.com/ using the API. The website allows the users to create a paid or a free account. With the free account there is a restriction of 1000 articles per day. For this assignment, the free version is being used.

The python script used to extract the news articles using the REQUEST object can be found in the folder ***csci5409-a3 Scripts*** with the name ***OmdbApi.py***.

The movie data has been extracted based on the keywords “Canada”, “University”, “Moncton”, “Halifax”, “Toronto”, “Vancouver”, “Alberta”, “Niagara”. While retrieving the data, the fields are filtered and only the attributes “title”, “year”, “genre”, “type” and “plot” are stored in the json format in the file ***movies.json*** which can be found in the folder named ***json files***. After this the movies data is cleaned, all the unicode characters and special symbols are removed and the cleaned movies data is stored in the file named ***cleaned_movies.json*** within the same folder.

Data Storage in MongoDB on AWS Instance

In the first phase of data storage, after retrieving the data from the three websites mentioned above, the data has been cleaned and transformed according to the need and then stored in the json files. In the final phase, the cleaned and transformed data are further stored in MongoDB, but not on the local machine. After MongoDB is successfully installed on an AWS instance using the steps mentioned in Section A, the data can be stored in the MongoDB server of that instance. The python script used to achieve this task can be found in the folder ***csci5409-a3 Scripts*** with the name ***mongo.py***. The name of the database where the data is stored is named ***csci5408a3*** [Figure 2]. Further the transformed data are stored in three different collections in this database [Figure 3]. The collection ***tweets*** contains documents from the cleaned tweets, ***news*** contains documents from the transformed news articles, and ***movies*** contains the transformed data from omdb. In the python code, these three tasks are performed using three different methods within the class named ***mongo***.

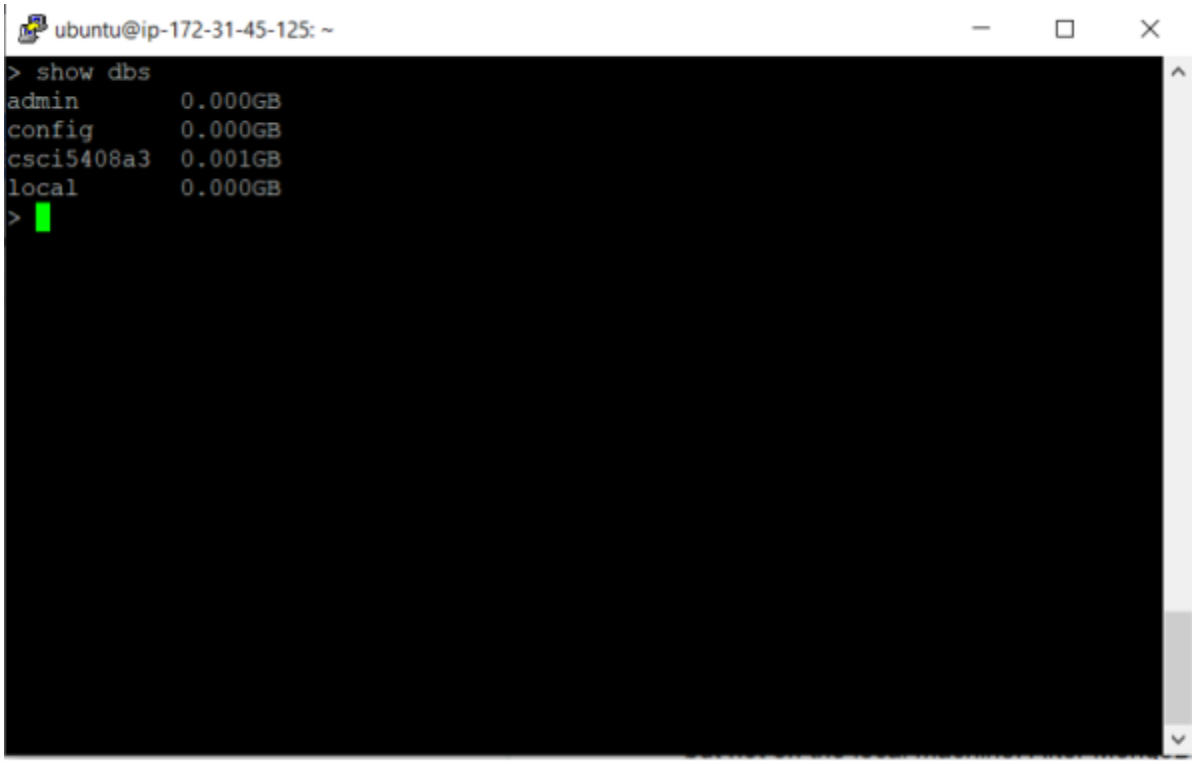

Figure 2: The MongoDB databases on the AWS instance.

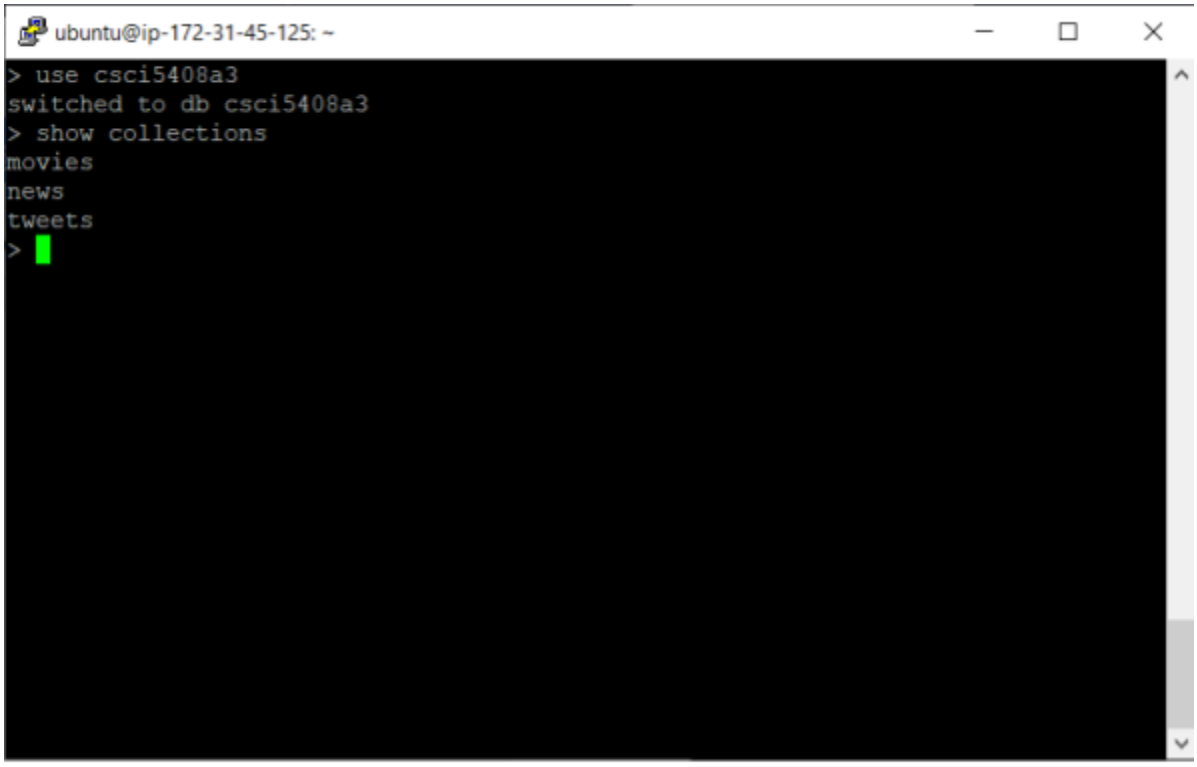

Figure 3: The collections present in the MongoDB csci5408a3

Section C - (i) Data Processing (Spark)

The instructions in Section 1 can be used to install and set up Spark in an AWS instance. It also contains the steps for starting the master and slave nodes. After the master and slave nodes have been started, PySpark can be opened. Please note that although the cleaned data have been previously stored in mongoDB, in order to get the frequency count using PySpark, a text file was created using a python script which contains all the tokenized data from the cleaned tweets and cleaned news articles without any new line or carriage return character. This was done to make the data processing using mapReduce easier. In order to transfer this text file to the aws instance, the tool **WinSCP** was used. This text file is named ***tweets_and_articles.txt*** and this file along with the python script named ***a3qc.py*** can be found within the folder ***PySpark***. Below is the screenshot [Figure 4] of the shell commands as well as output of the frequency count of the given terms: 'education', 'canada', 'university', 'dalhousie', 'expensive', 'good school', 'good schools', 'bad school', 'bad schools' ,'poor school' , 'poor schools', 'faculty', 'computer science', and 'graduate'. For readability, the shell commands are also stored in a text file called ***PySpark Shell Commands.txt*** within the same folder.

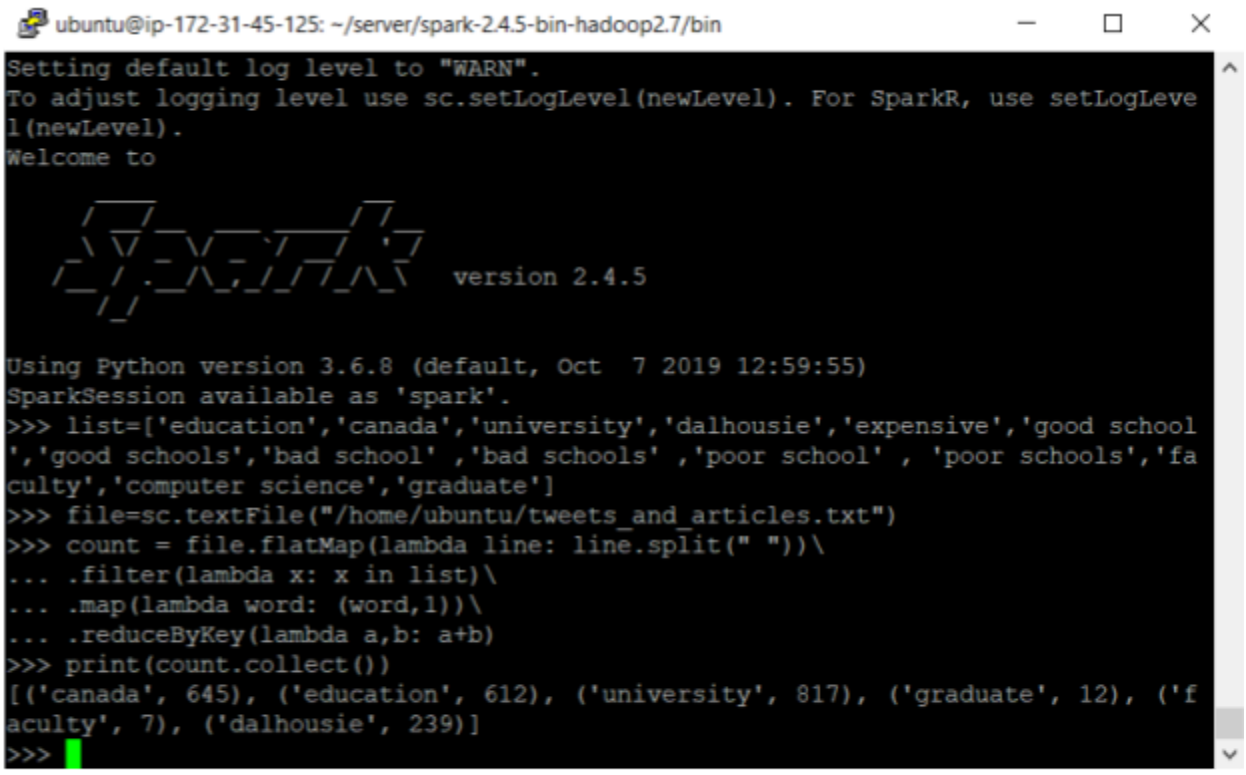

Figure 4: PySpark shell commands to count the frequency of the given terms along with the output.

Section C - (ii) Script to extract movie rating, genre, and plot from the stored movie data

As mentioned in the previous section, the extracted movie data along with the cleaned tweets and news articles are stored in the mongoDB on an EC2 instance. For this task, a separate function called ***extract_movie_ratings()*** has been written that creates connection with the remote mongoDB and extracts and prints movie title, genre, plot and ratings and is present in the class ***a3qc*** in same script ***a3qc.py*** that created ***tweets_and_articles.txt*** file. Please find the script within the folder ***PySpark***.

Conclusion

In this assignment, concepts of data retriever, processing and storage were covered, and along with that tools and technologies including NoSQL database MongoDB, Apache Spark, MapReduce and Hadoop. Data extraction, data transformation and data processing are important aspects of the field of Data Science and this assignment has provided an exposure to the importance of all of these.

References

[1] W. Al Shehri, "Cloud Database Database as a Service", *International Journal of Database Management Systems*, vol. 5, no. 2, pp. 1-12, 2013. Available: 10.5121/ijdms.2013.5201.

[2] M. Bhanderi, K. Amilmani, and G. S. Dhillon, *Brightspace - Dalhousie University*. [Online]. Available: https://dal.brightspace.com/d2l/le/content/110354/viewContent/1596093/View. [Accessed: 19-Mar-2020].

[3] M. S. Bhanderi, K. S. Amilmani, and G. S. Dhillon, *Brightspace - Dalhousie University*. [Online]. Available: https://dal.brightspace.com/d2l/le/content/110354/viewContent/1608781/View. [Accessed: 19-Mar-2020].

[4] “Install MongoDB Community Edition on Ubuntu¶,” *Install MongoDB Community Edition on Ubuntu - MongoDB Manual*. [Online]. Available: https://docs.mongodb.com/manual/tutorial/install-mongodb-on-ubuntu/. [Accessed: 20-Mar-2020].

“Developer,” *Twitter*. [Online]. Available: https://developer.twitter.com/. [Accessed: 21-Mar-2020].

“News API - A JSON API for live news and blog articles,” *News API - A JSON API for live news and blog articles*. [Online]. Available: https://newsapi.org/. [Accessed: 21-Mar-2020].

“OMDb API,” *OMDb API - The Open Movie Database*. [Online]. Available: http://www.omdbapi.com/. [Accessed: 21-Mar-2020].