

CSCI 5408 - DATA Management Warehousing Analytics

Assignment 3

YUGANTHI KRISHNAMURTHY

B00839935

yg617681@dal.ca

A. Cluster Setup:

- Create a cloud account with any cloud service provider

Created a cloud account with the amazon web services (AWS)

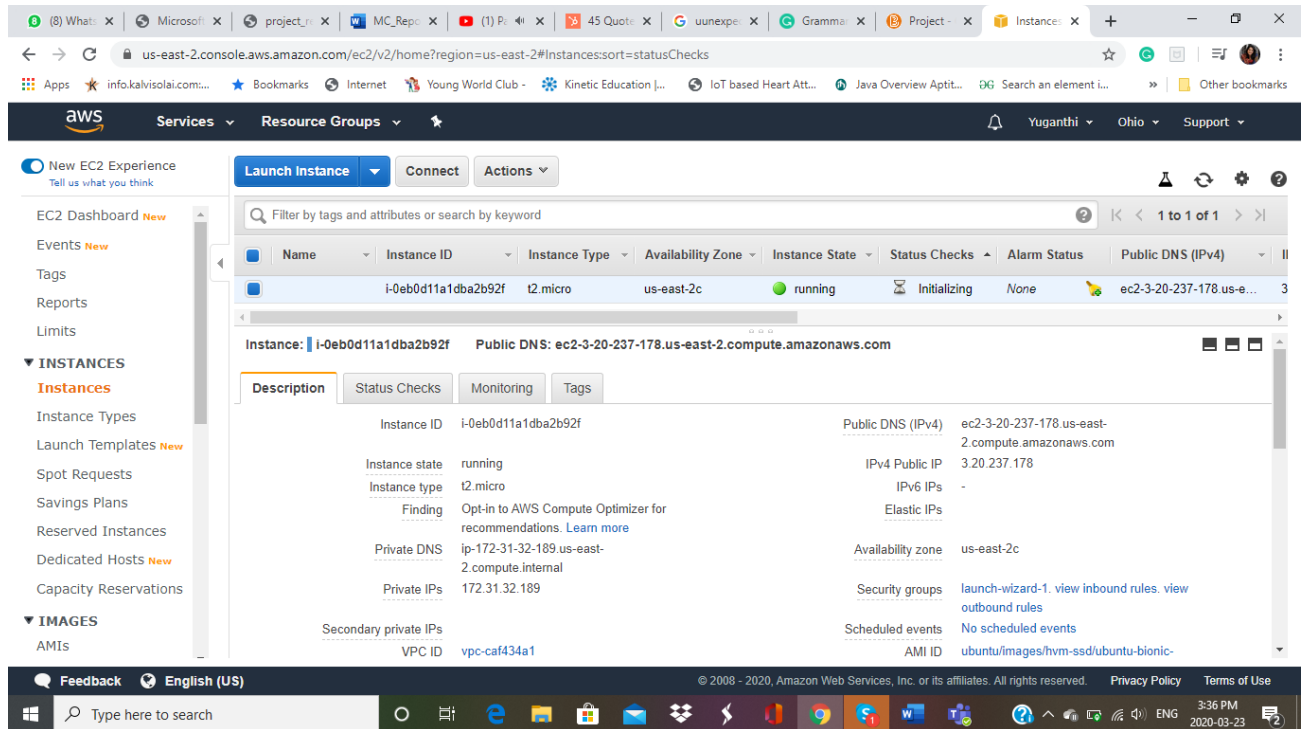


Fig 1: Cloud Account with Amazon Web Services

- Initialize Apache Spark on cloud account

Initialized Apache Spark on cloud account following the tutorials provided in the lab.

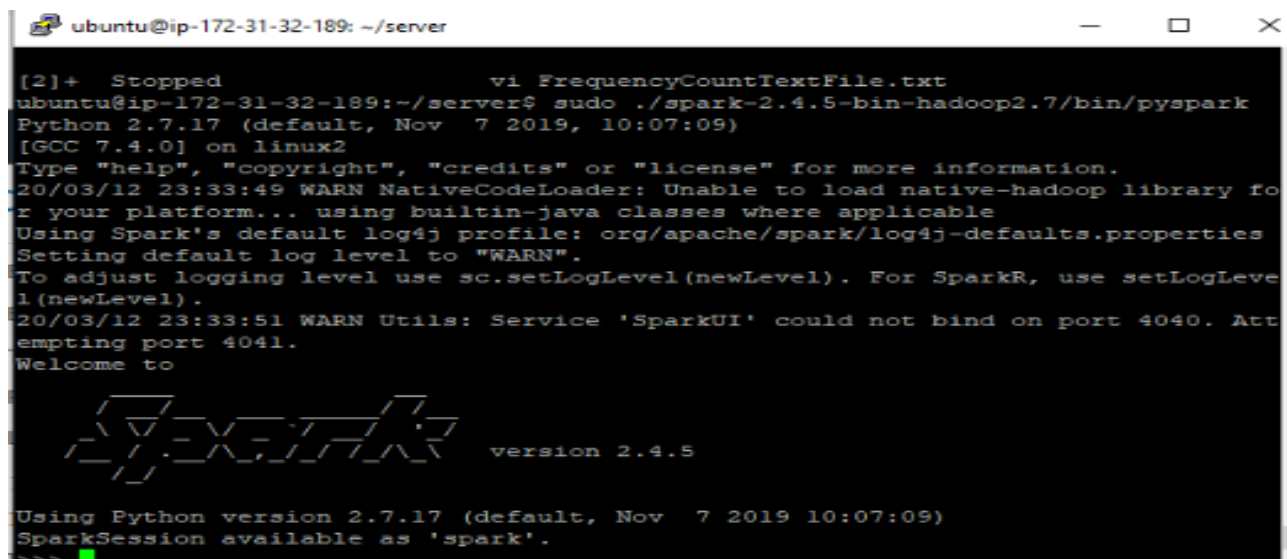
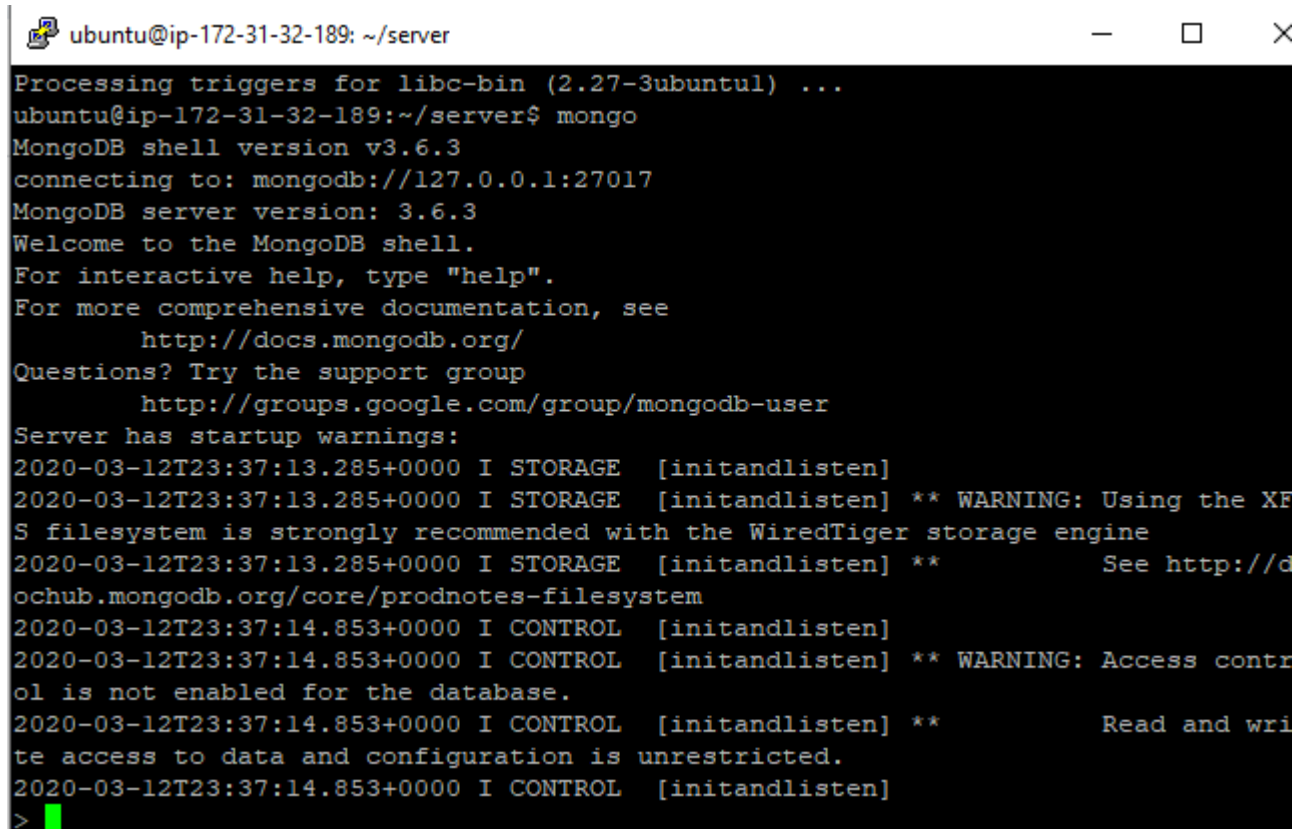


Fig 2: Apache Spark Setup

- Install MongoDB to store the data

A terminal window with a black background and white text. The window title bar shows 'ubuntu@ip-172-31-32-189: ~/server' and standard window controls. The terminal output shows the command 'mongo' being executed, which starts the MongoDB shell. It displays the version (v3.6.3) and connection details (localhost:27017). Several startup warnings are shown, including a recommendation for XFS filesystem and a warning about unrestricted access. The prompt '>' is visible at the bottom.

```
ubuntu@ip-172-31-32-189: ~/server
Processing triggers for libc-bin (2.27-3ubuntu1) ...
ubuntu@ip-172-31-32-189:~/server$ mongo
MongoDB shell version v3.6.3
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.6.3
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
    http://docs.mongodb.org/
Questions? Try the support group
    http://groups.google.com/group/mongodb-user
Server has startup warnings:
2020-03-12T23:37:13.285+0000 I STORAGE  [initandlisten]
2020-03-12T23:37:13.285+0000 I STORAGE  [initandlisten] ** WARNING: Using the XFS
S filesystem is strongly recommended with the WiredTiger storage engine
2020-03-12T23:37:13.285+0000 I STORAGE  [initandlisten] **          See http://d
ochub.mongodb.org/core/prodnotes-filesystem
2020-03-12T23:37:14.853+0000 I CONTROL  [initandlisten]
2020-03-12T23:37:14.853+0000 I CONTROL  [initandlisten] ** WARNING: Access contr
ol is not enabled for the database.
2020-03-12T23:37:14.853+0000 I CONTROL  [initandlisten] **          Read and wri
te access to data and configuration is unrestricted.
2020-03-12T23:37:14.853+0000 I CONTROL  [initandlisten]
>
```

Fig 3: MongoDB installation

B. Data Extraction and Transformation:

- Twitter Data Extraction and Transformation

Twitter data was extracted by creating a twitter developer account. The twitter data was extracted by writing a java program and the data cleaning was also performed. The unwanted tags, URL's were removed. The twitter API was required to access the twitter data using the java program. The twitter API was accessed using the twitter.4j jar. The tweets were collected according the given search keywords. The data extracted is more than 3000 records. Both tweets and retweets were collected along with the provided metadata such as location, time. The retweets were specified with "RT" tag. The extracted data was stored in the csv file. The extracted data is stored in the MongoDB. The folder Assignment3 contains the extracted cleaned twitter data. The java program for the twitter data extraction is attached in the folder.

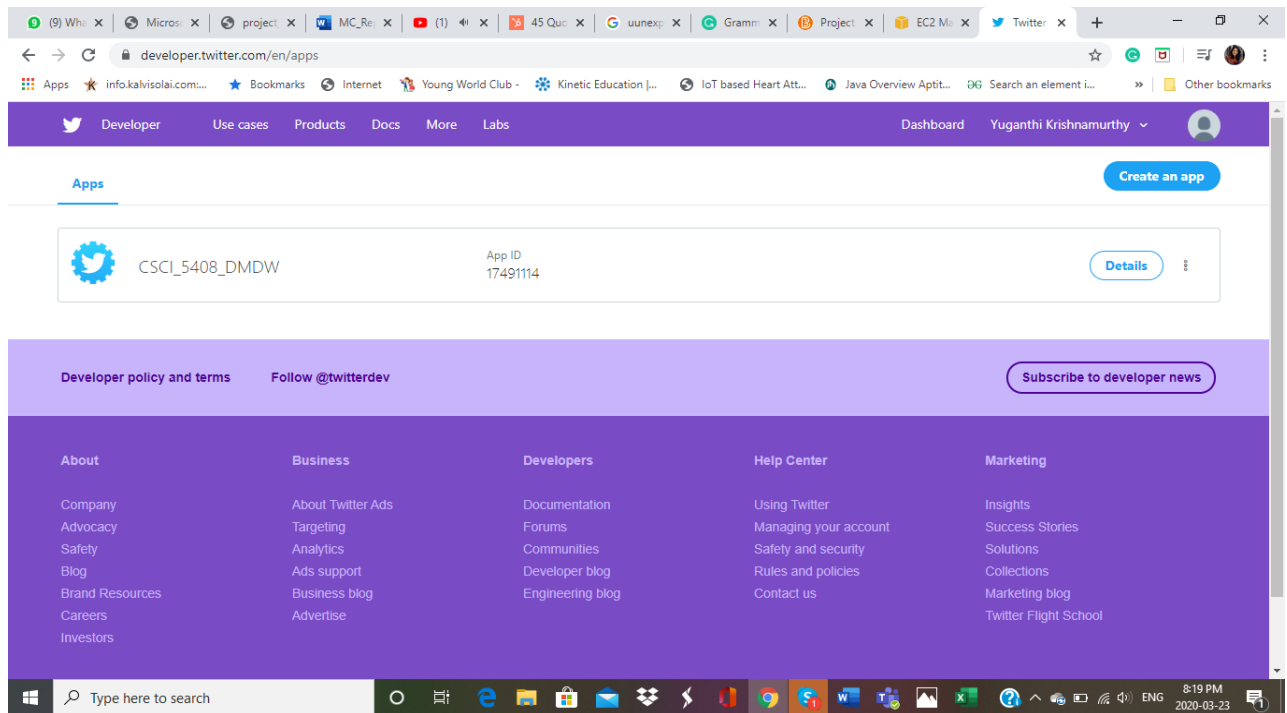


Fig 4: Twitter developer account

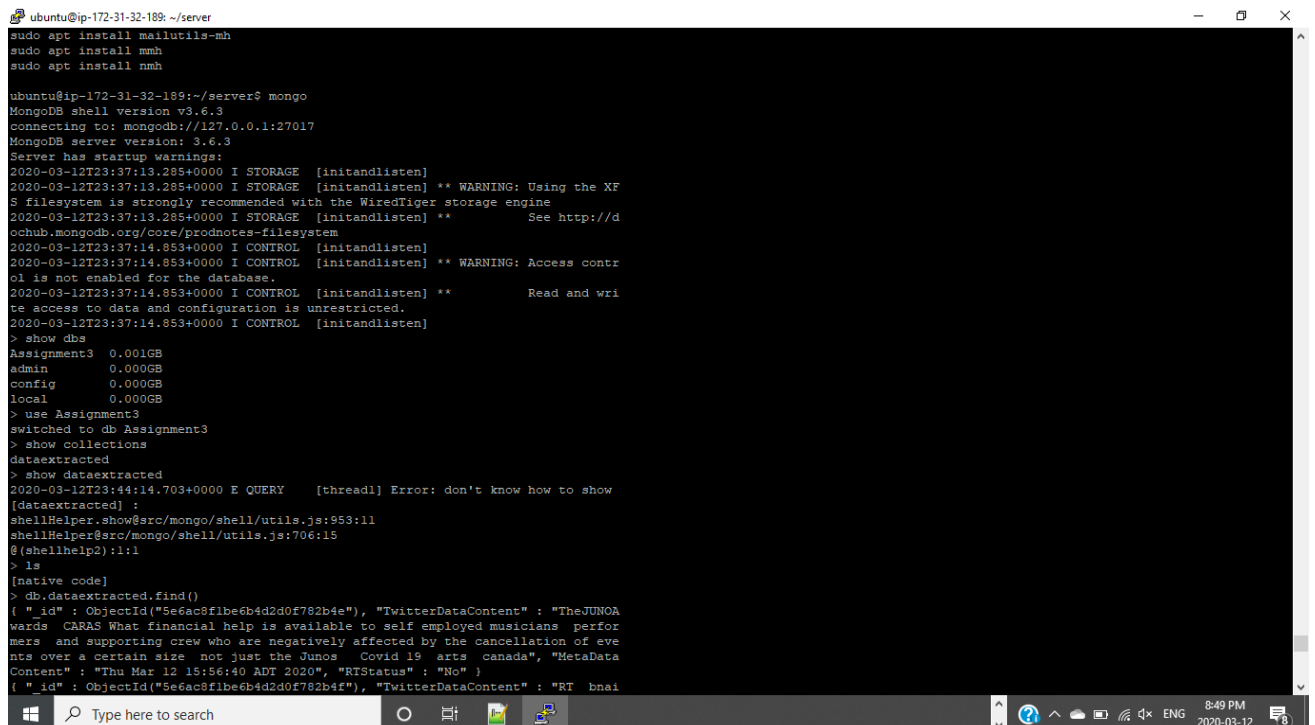


Fig 5: Extracted Data Stored in MongoDB

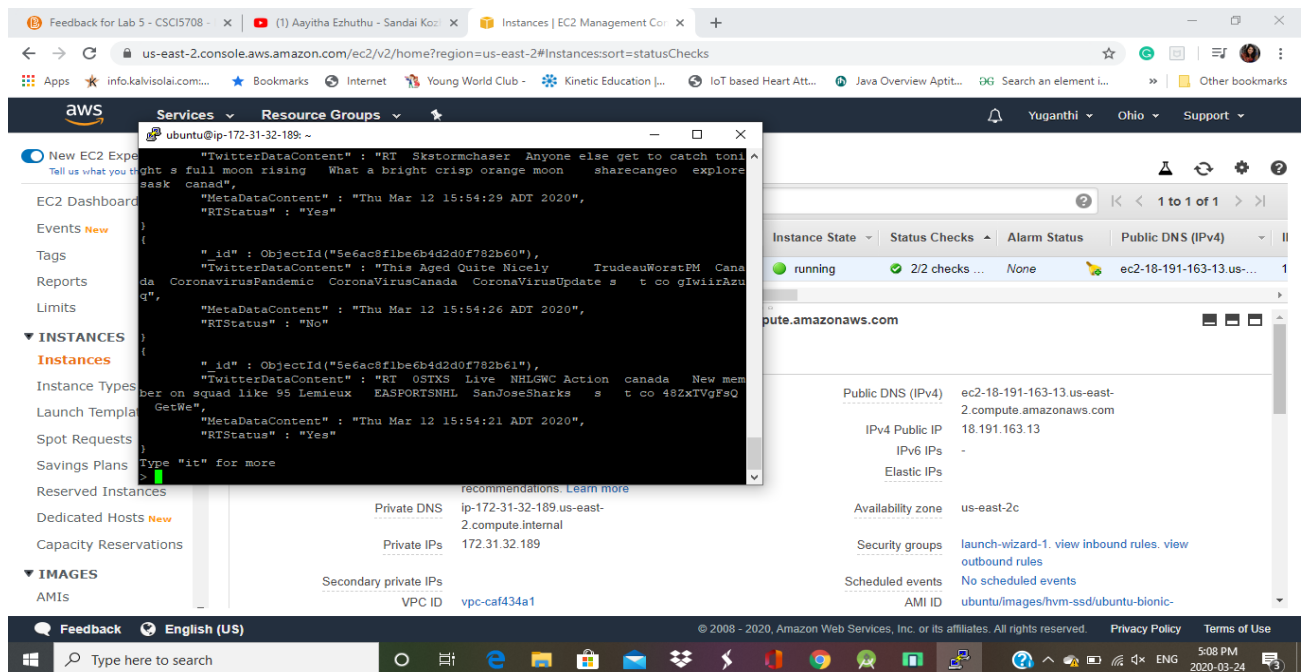


Fig 6: Twitter Data Extracted and Stored in MongoDB

- **News Article Data Extraction and Transformation:**

News article data was extracted and transformed using the java program. The news API <https://newsapi.org/> was used to access the data by creating a developer account. The data was extracted according to the given keywords. The data cleaning is also performed using the java program. The cleaned data is stored in the cleanedNewsArticle csv file. The java program used to extract and clean the data is attached in the folder. The response returned was in the JSON format. While cleaning the data, the fields that were considered are “author, title, content, description”. The extracted and the cleaned data is stored in the MongoDB. The folder newdata in Assignment3 folder contains cleaned news data.

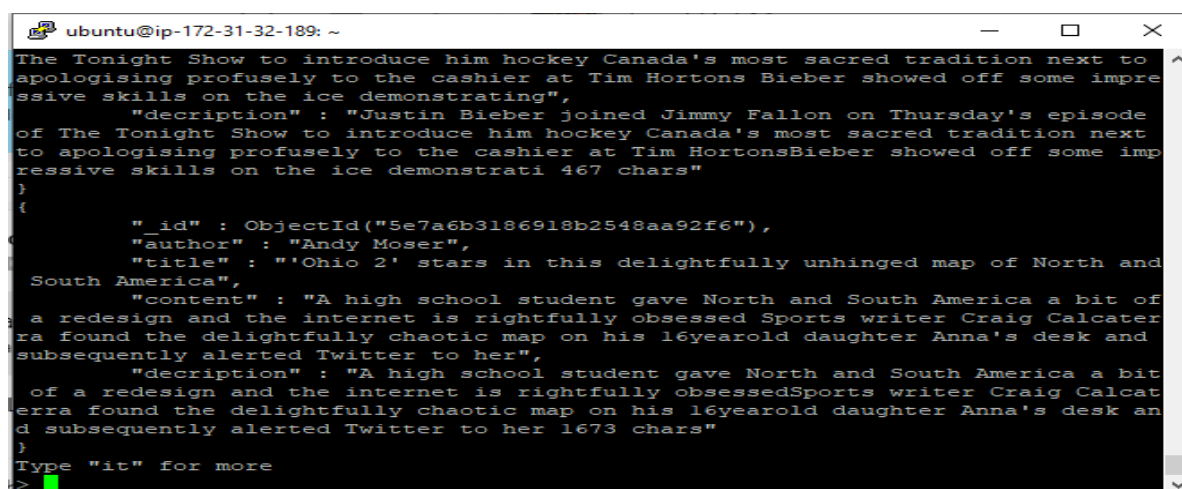
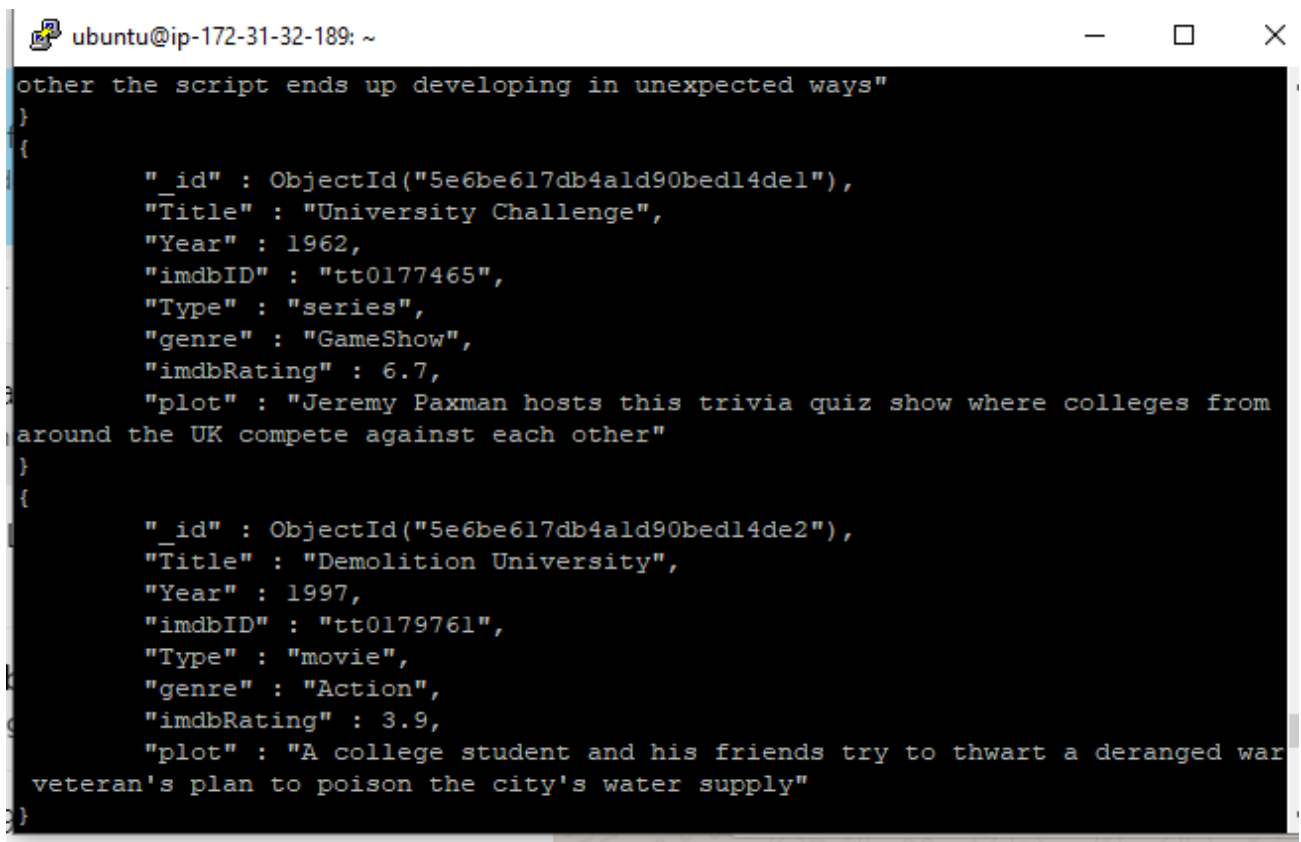


Fig 7: News Data Extracted and Stored in MongoDB

- **Movie Data Extraction and Transformation:**

Movie data was extracted and transformed using the java program. The movie API <http://www.omdbapi.com/> was used to access the data by creating a developer account. The data was extracted according to the given keywords. The data cleaning is also performed using the java program. The cleaned data is stored in the cleanedMovieArticles csv file. The java program used to extract and clean the data is attached in the folder. The response returned was in the JSON format. While cleaning the data, the fields that were considered are “title, year, imdbID, Type, genre, imdbRating, plot”. The extracted and the cleaned data is stored in the MongoDB. The folder moviesArticles contain the extracted and cleaned movie data.

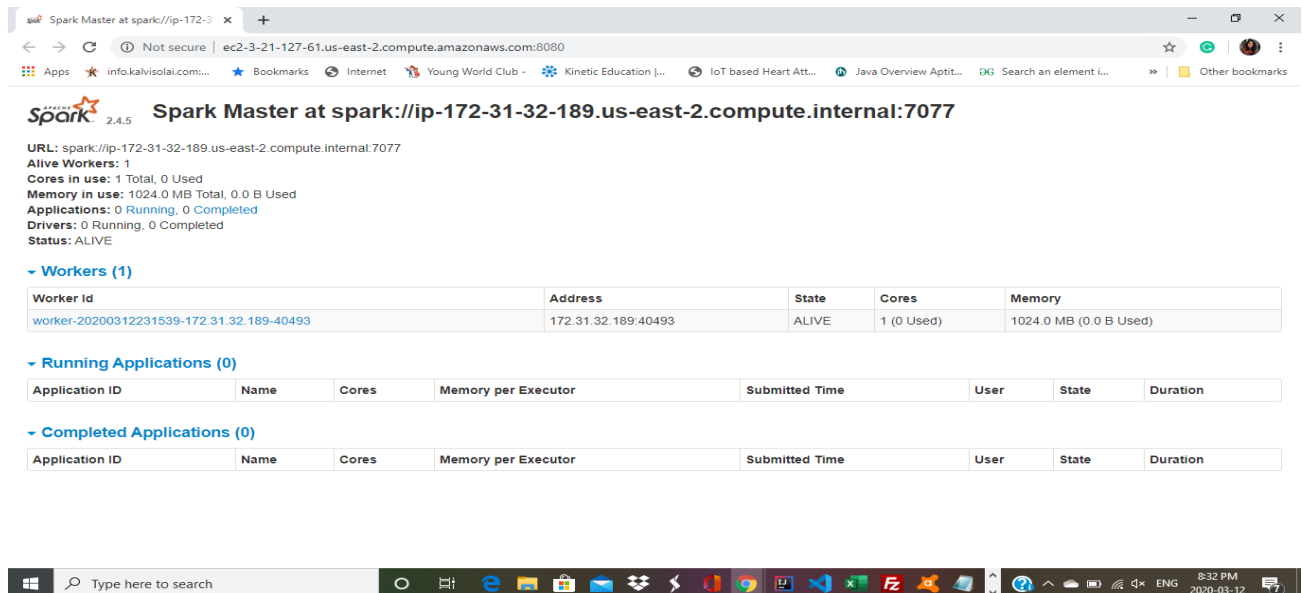


```
ubuntu@ip-172-31-32-189: ~  
other the script ends up developing in unexpected ways"  
}  
{  
  "_id" : ObjectId("5e6be617db4ald90bed14de1"),  
  "Title" : "University Challenge",  
  "Year" : 1962,  
  "imdbID" : "tt0177465",  
  "Type" : "series",  
  "genre" : "GameShow",  
  "imdbRating" : 6.7,  
  "plot" : "Jeremy Paxman hosts this trivia quiz show where colleges from  
around the UK compete against each other"  
}  
{  
  "_id" : ObjectId("5e6be617db4ald90bed14de2"),  
  "Title" : "Demolition University",  
  "Year" : 1997,  
  "imdbID" : "tt0179761",  
  "Type" : "movie",  
  "genre" : "Action",  
  "imdbRating" : 3.9,  
  "plot" : "A college student and his friends try to thwart a deranged war  
veteran's plan to poison the city's water supply"  
}
```

Fig 8: Movie Data Extracted and Stored in MongoDB

C. Data Processing (Spark):

The spark framework performs a frequency count using MapReduce for the given words. The stored tweets and the news article were considered to perform the frequency count. The Python program was used to perform the map reduce and the frequency count.



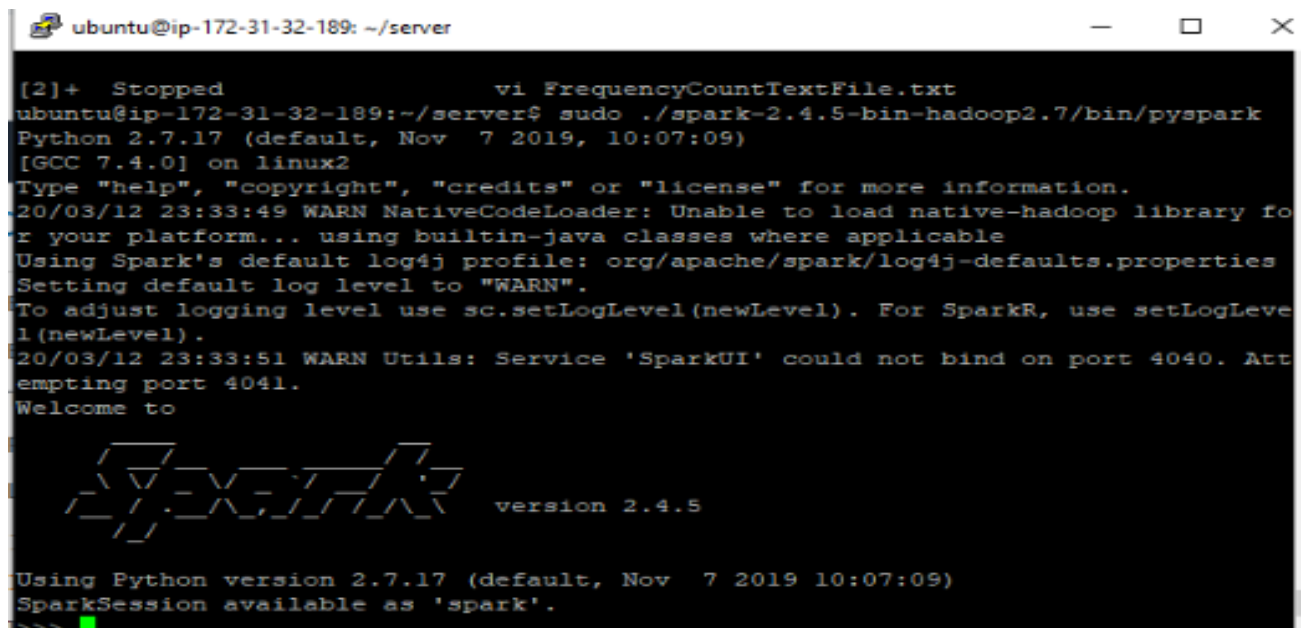
The screenshot shows the Spark Master web interface at `spark://ip-172-31-32-189.us-east-2.compute.internal:7077`. The interface displays the following information:

- URL:** `spark://ip-172-31-32-189.us-east-2.compute.internal:7077`
- Alive Workers:** 1
- Cores in use:** 1 Total, 0 Used
- Memory in use:** 1024.0 MB Total, 0.0 B Used
- Applications:** 0 Running, 0 Completed
- Drivers:** 0 Running, 0 Completed
- Status:** ALIVE

Below this summary, there are three expandable sections:

- Workers (1):** A table showing one worker with ID `worker-20200312231539-172.31.32.189-40493`, address `172.31.32.189:40493`, state `ALIVE`, 1 core used, and 1024.0 MB memory used.
- Running Applications (0):** An empty table with columns: Application ID, Name, Cores, Memory per Executor, Submitted Time, User, State, and Duration.
- Completed Applications (0):** An empty table with the same columns as the Running Applications section.

Fig 9: Slave Master Setup



```
ubuntu@ip-172-31-32-189: ~/server
[2]+  Stopped                  vi FrequencyCountTextFile.txt
ubuntu@ip-172-31-32-189:~/server$ sudo ./spark-2.4.5-bin-hadoop2.7/bin/pyspark
Python 2.7.17 (default, Nov  7 2019, 10:07:09)
[GCC 7.4.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
20/03/12 23:33:49 WARN NativeCodeLoader: Unable to load native-hadoop library for
your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLeve
l(newLevel).
20/03/12 23:33:51 WARN Utils: Service 'SparkUI' could not bind on port 4040. Att
empting port 4041.
Welcome to

      _ _ _ _ _
     / _ _ _ _ \   version 2.4.5
    / _ _ _ _ \
   / _ _ _ _ \

Using Python version 2.7.17 (default, Nov  7 2019 10:07:09)
SparkSession available as 'spark'.
>>>
```

Fig 10: pyspark setup

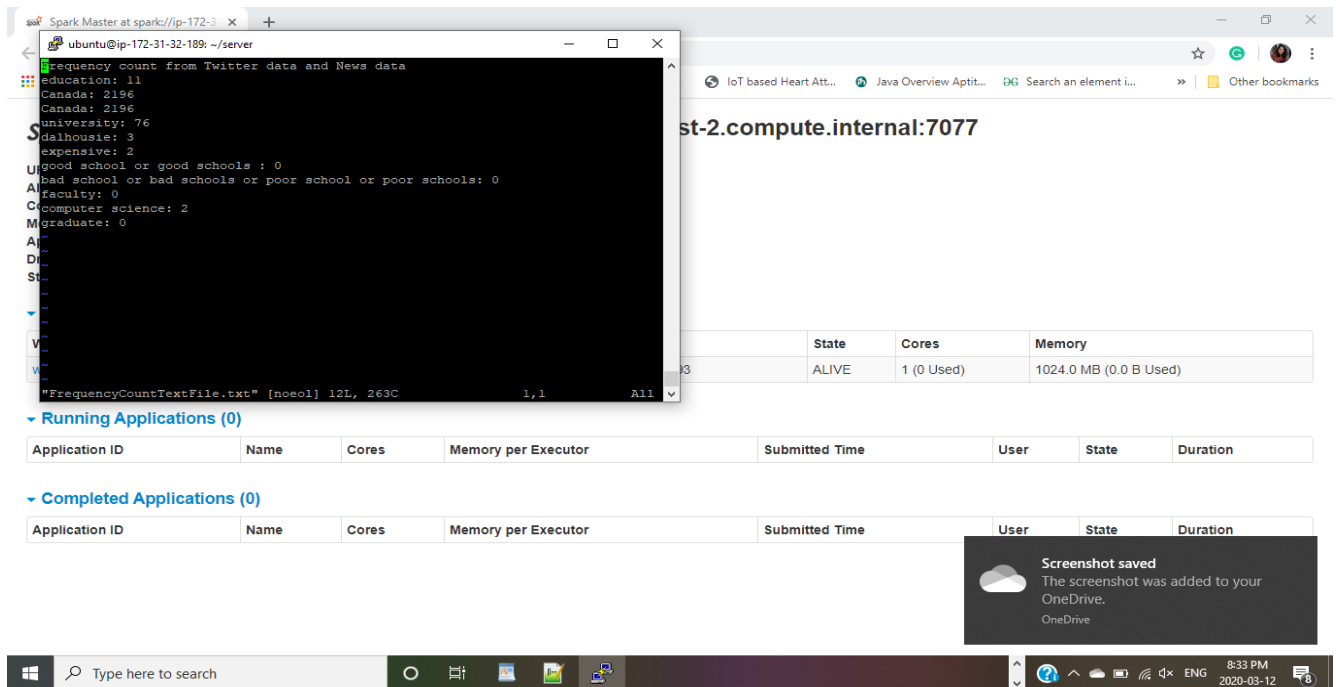


Fig 11: Frequency Count using Map Reduce

- Program/script to extract movie rating, genre, plot, from the stored data:

Java program is used to extract the movie rating, genre, plot from the stored movie data. The java program is written on the MongoDB and required fields is extracted from the stored movie data.

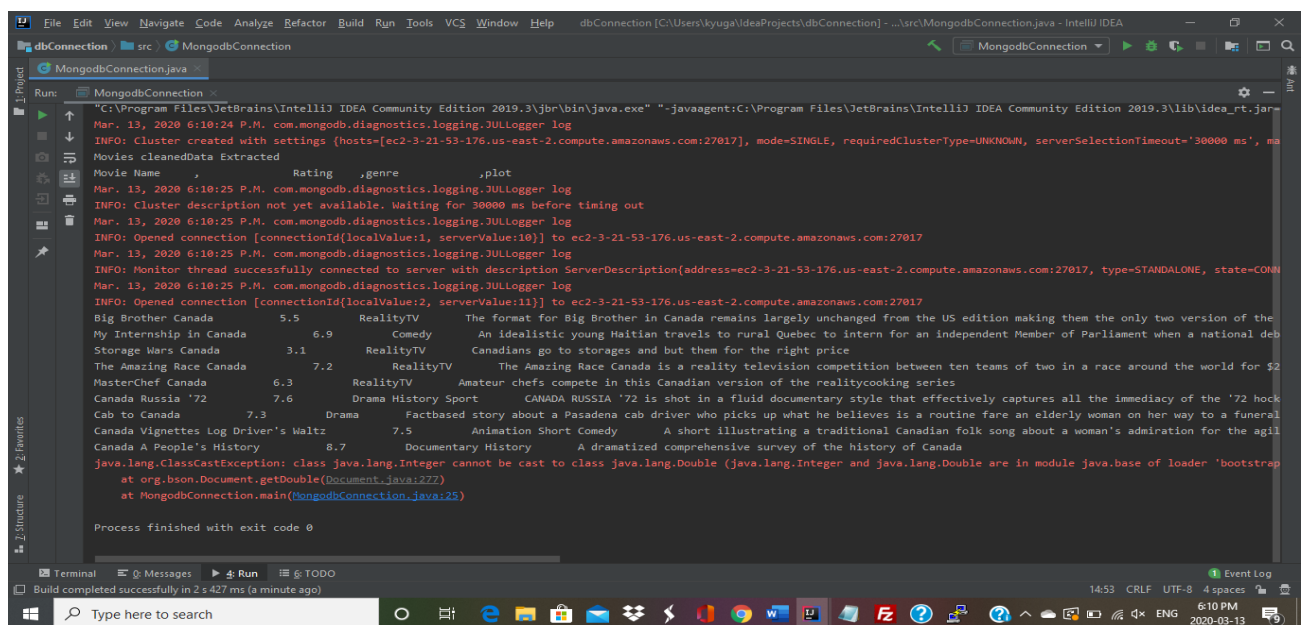


Fig 12: Java Program to Extract Movie Data

D. References:

- [1]"How to Parse Nested JSON using JAVA", *YouTube*, 2020. [Online]. Available: <https://www.youtube.com/watch?v=h5VLKYOQOjM&t=461s>. [Accessed: 24- Mar- 2020]
- [2]*Developer.twitter.com*, 2020. [Online]. Available: <https://developer.twitter.com/en/apps>. [Accessed: 24- Mar- 2020]
- [3]"Twitter4J - A Java library for the Twitter API", *Twitter4j.org*, 2020. [Online]. Available: <http://twitter4j.org/en/>. [Accessed: 24- Mar- 2020]
- [4]"News API - A JSON API for live news and blog articles", *Newsapi.org*, 2020. [Online]. Available: <https://newsapi.org/>. [Accessed: 24- Mar- 2020]
- [5]"OMDb API - The Open Movie Database", *Omdbapi.com*, 2020. [Online]. Available: <http://www.omdbapi.com/>. [Accessed: 24- Mar- 2020]
- [6]"Apache Spark™ - Unified Analytics Engine for Big Data", *Spark.apache.org*, 2020. [Online]. Available: <https://spark.apache.org/>. [Accessed: 24- Mar- 2020]
- [7]"AWS Educate", *Amazon Web Services, Inc.*, 2020. [Online]. Available: <https://aws.amazon.com/education/awseducate/>. [Accessed: 24- Mar- 2020]
- [8]"Welcome to Spark Python API Docs! — PySpark 2.4.5 documentation", *Spark.apache.org*, 2020. [Online]. Available: <https://spark.apache.org/docs/latest/api/python/index.html>. [Accessed: 24- Mar- 2020]
- [9]2020. [Online]. Available: <https://www.digitalocean.com/community/tutorials/how-to-install-mongodb-on-ubuntu18-04>. [Accessed: 24- Mar- 2020]
- [10]"BionicBeaver/ReleaseNotes - Ubuntu Wiki", *Wiki.ubuntu.com*, 2020. [Online]. Available: <https://wiki.ubuntu.com/BionicBeaver/ReleaseNotes>. [Accessed: 24- Mar- 2020]