CSCI 5408 - DATA MANAGEMENT AND WAREHOUSING ASSIGNMENT – 3 REPORT

GOWRI PRASHANTH KANAGARAJ- B00942544

Gitlab: https://git.cs.dal.ca/kanagaraj/csci5408_s23_b00942544_gowriprashanth_kanagaraj

lo#1: Perform research on NoSQL and data processing – To achieve this task, you need to read and understand the usage of spark framework, MongoDB and then implement a programming framework for big data processing, and store.

Problem 1A:

Reuter News Data Reading & Transformation and storing in MogoDb

MongoDb Atlas was used to create a shared cloud cluster [1]

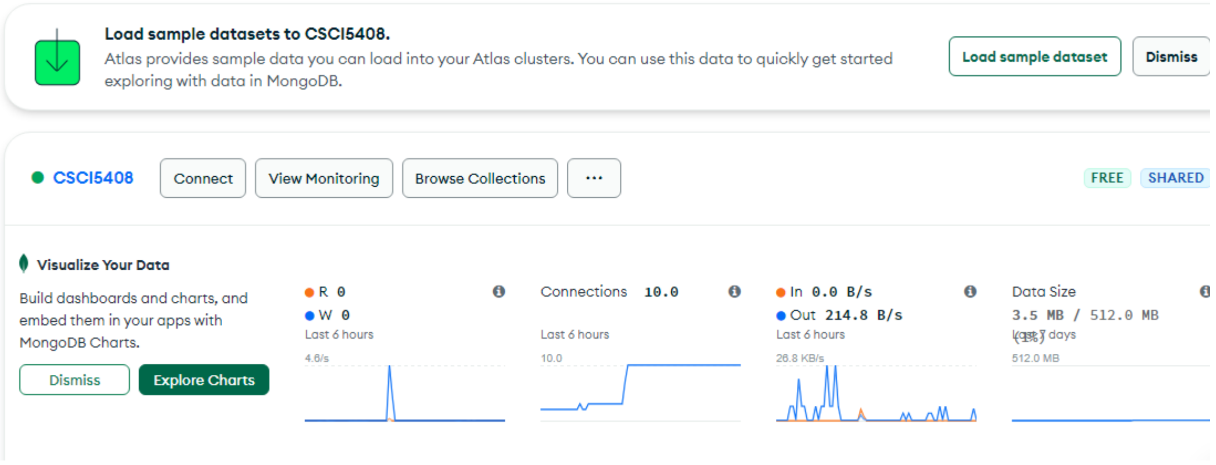


Fig 1: MongoDb Atlas Cluster

A connection was made to the cluster from MongoDb Compass

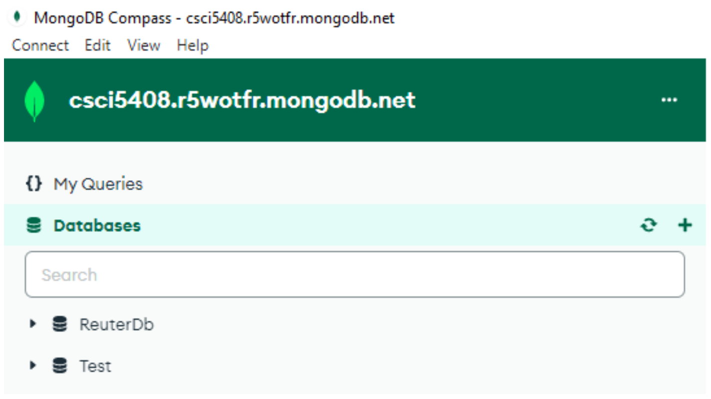

MongoDB Compass - csci5408.r5wotfr.mongodb.net
Connect Edit View Help
csci5408.r5wotfr.mongodb.net
My Queries
Databases
Search
ReuterDb
Test

Fig 2: MongoDb Atlas Cluster

The source files for news articles: reut2-009.sgm, and reut2-014.sgm were downloaded and used as the source for database ReuterDb

Algorithm - Reuters program:

Create a ReutRead class with the main method:

- Instantiate a Parser object.
- Set up the MongoDB connection settings with the URI.
- Create a ServerApi object with the specified version.
- Create the MongoClientSettings with the connection string and server API version.
- Define the MongoDB database name (ReuterDb) and collection names (newsArticles).
- Define the paths for newsFile1 and newsFile2.
- Inside a try-with-resources block, create a MongoClient with the given settings.
- Within the try block, get the MongoDB database instance and call the parseAndStoreNews method twice with different SGML files to parse and store the news articles in the MongoDB collection.

Create a Parser class with the following methods:

- parseAndStoreNews: This method takes the MongoDatabase, collection name, and file name as inputs.
- Read the content of the SGML file specified by fileName into a String.
- Split the content by the <REUTERS> tags to get an array of news articles.
- For each news article in the array:
- Parse the title and text content using the parseTagContent method.
- Create a MongoDB Document containing the title and text.
- Insert the document into the specified collection of the database.
- parseTagContent: This method takes the content of an SGML news article and a tag name as inputs.
- Find the starting and ending tag positions for the given tagName.
- Extract the content between the starting and ending tags.
- Return the parsed content.

Flowchart- Reuters data transformation program:

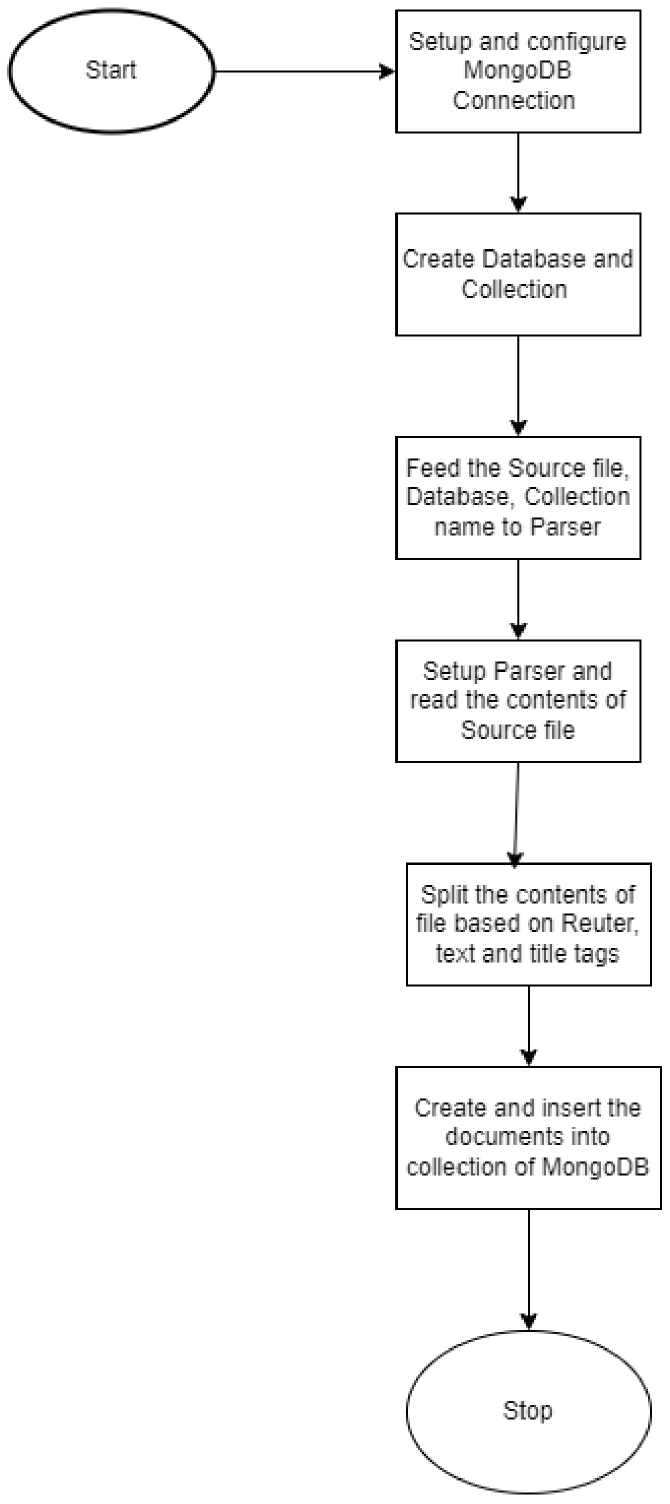
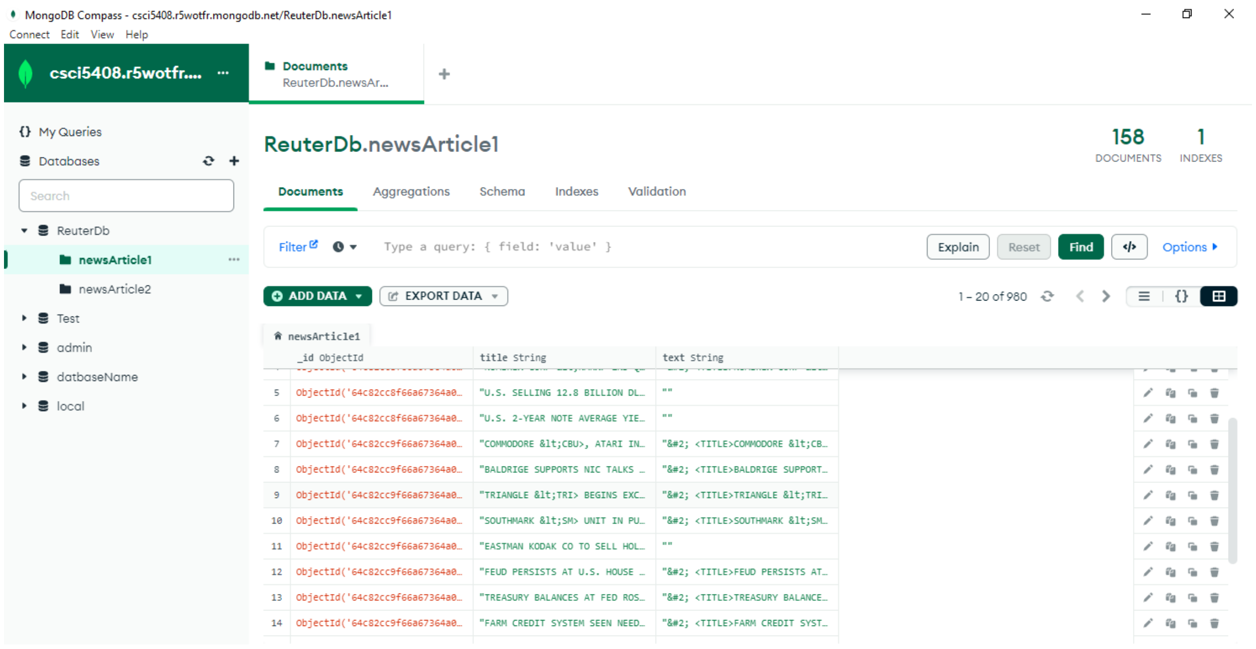

Start
Setup and configure MongoDB Connection
Create Database and Collection
Feed the Source file, Database, Collection name to Parser
Setup Parser and read the contents of Source file
Split the contents of file based on Reuter, text and title tags
Create and insert the documents into collection of MongoDB
Stop

Fig 3: Flowchart for Reuters data transformation program

The Database ReuterDb is successfully created and the news articles are stored in Collections newsArticle1 and newsArticle2



ReuterDb.newsArticle1

158 DOCUMENTS 1 INDEXES

Type a query: { field: 'value' }

newsArticle1

	_id ObjectId	title String	text String
5	ObjectId('64c82cc8f66a67364a0…	"U.S. SELLING 12.8 BILLION DL…	""
6	ObjectId('64c82cc8f66a67364a0…	"U.S. 2-YEAR NOTE AVERAGE YIE…	""
7	ObjectId('64c82cc9f66a67364a0…	"COMMODORE <CBU>, ATARI IN…	"&#2; <TITLE>COMMODORE &lt;CB…
8	ObjectId('64c82cc9f66a67364a0…	"BALDRIGE SUPPORTS NIC TALKS …	"&#2; <TITLE>BALDRIGE SUPPORT…
9	ObjectId('64c82cc9f66a67364a0…	"TRIANGLE <TRI> BEGINS EXC…	"&#2; <TITLE>TRIANGLE &lt;TRI…
10	ObjectId('64c82cc9f66a67364a0…	"SOUTHMARK <SM> UNIT IN PU…	"&#2; <TITLE>SOUTHMARK &lt;SM…
11	ObjectId('64c82cc9f66a67364a0…	"EASTMAN KODAK CO TO SELL HOL…	""
12	ObjectId('64c82cc9f66a67364a0…	"FEUD PERSISTS AT U.S. HOUSE …	"&#2; <TITLE>FEUD PERSISTS AT…
13	ObjectId('64c82cc9f66a67364a0…	"TREASURY BALANCES AT FED ROS…	"&#2; <TITLE>TREASURY BALANCE…
14	ObjectId('64c82cc9f66a67364a0…	"FARM CREDIT SYSTEM SEEN NEED…	"&#2; <TITLE>FARM CREDIT SYST…

Fig 4: MongoDb - ReuterDb

Problem 1B:

Configure and initialize Apache Spark cluster in GCP cloud

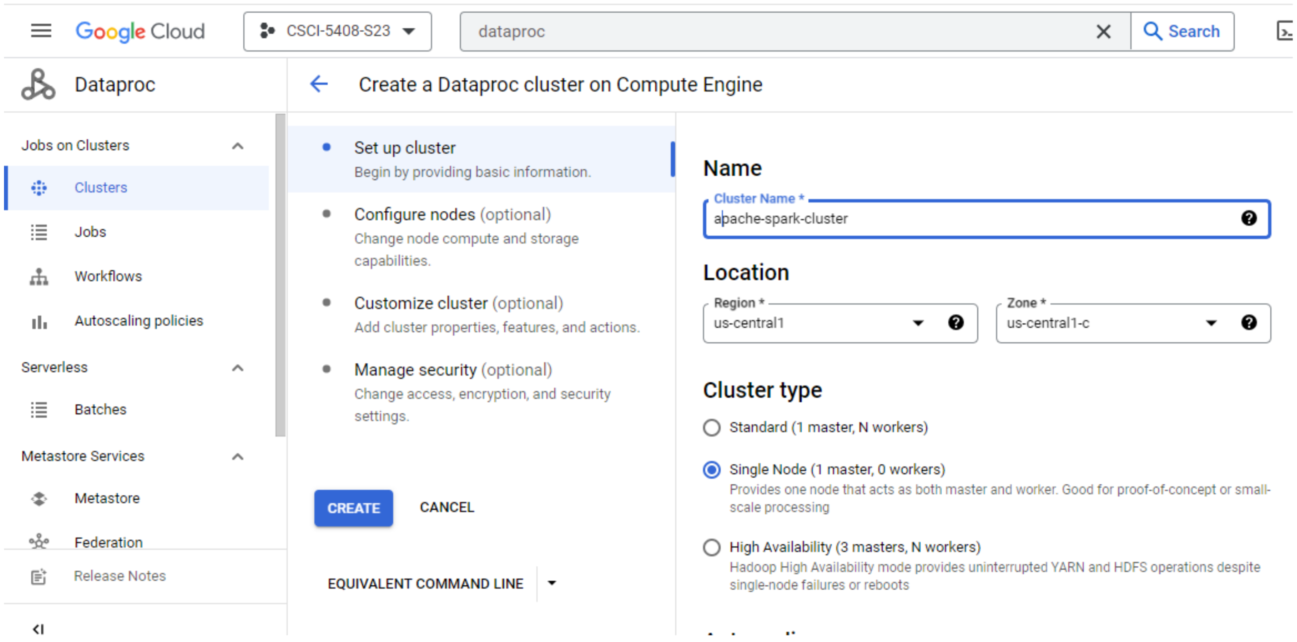
Step 1: Create a Dataproc instance in GCP .

Dataproc is a fully managed cloud service that allows you to run Apache Spark

Navigate to the Google Cloud Console and Enable the Dataproc API

Create the Dataproc Cluster and perform the Cluster Configuration.

Select the Cluster Name, Region, Cluster type and configure nodes for the cluster aswell



Create a Dataproc cluster on Compute Engine

- Set up cluster
  Begin by providing basic information.
- Configure nodes (optional)
  Change node compute and storage capabilities.
- Customize cluster (optional)
  Add cluster properties, features, and actions.
- Manage security (optional)
  Change access, encryption, and security settings.

Name

Cluster Name *: apache-spark-cluster

Location

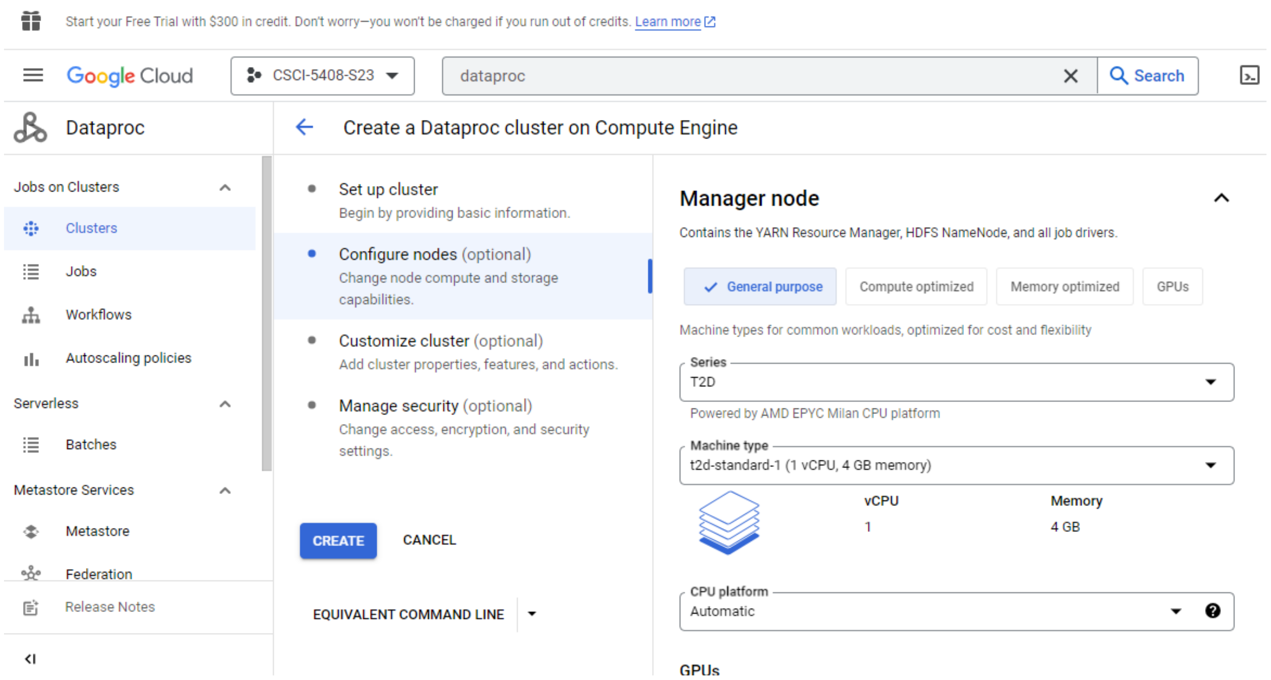
Region *: us-central1

Zone *: us-central1-c

Cluster type

- Standard (1 master, N workers)
- [✓ Single Node (1 master, 0 workers)]
  Provides one node that acts as both master and worker. Good for proof-of-concept or small-scale processing
- High Availability (3 masters, N workers)
  Hadoop High Availability mode provides uninterrupted YARN and HDFS operations despite single-node failures or reboots

Fig 5: Setting up Dataproc cluster in GCP


Start your Free Trial with $300 in credit. Don't worry—you won't be charged if you run out of credits. Learn more

Create a Dataproc cluster on Compute Engine

- Set up cluster — Begin by providing basic information.
- Configure nodes (optional) — Change node compute and storage capabilities.
- Customize cluster (optional) — Add cluster properties, features, and actions.
- Manage security (optional) — Change access, encryption, and security settings.

Manager node

Contains the YARN Resource Manager, HDFS NameNode, and all job drivers.

✓ General purpose | Compute optimized | Memory optimized | GPUs

Machine types for common workloads, optimized for cost and flexibility

Series: T2D

Powered by AMD EPYC Milan CPU platform

Machine type: t2d-standard-1 (1 vCPU, 4 GB memory)

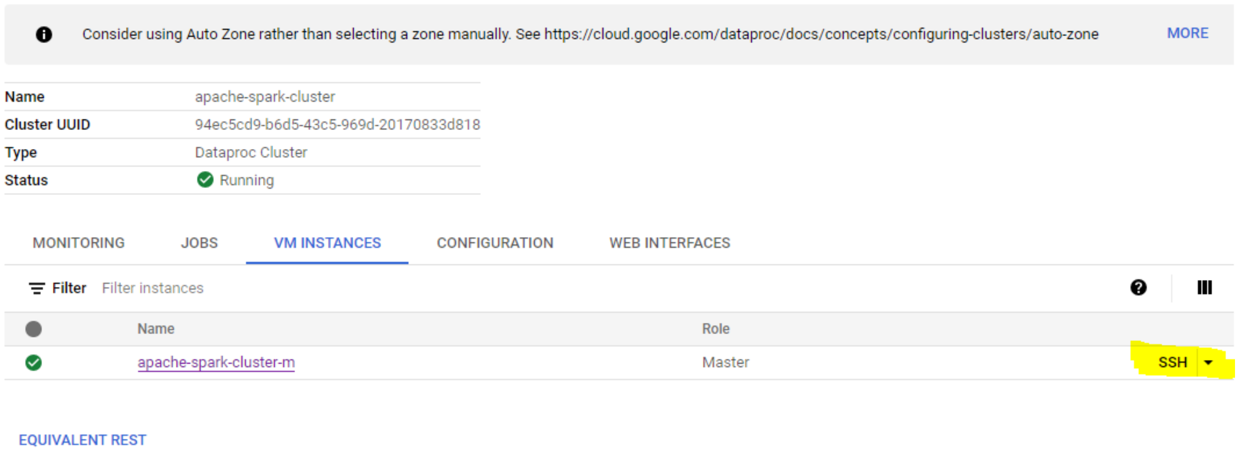
vCPU	Memory
1	4 GB

CPU platform: Automatic

Fig 6: Configuring nodes - Dataproc cluster in GCP

Click on Create and wait for the cluster to be created

Step 2: Once the cluster is created, Connect to Apache Spark SSH instance and click on authorize to proceed to the instance


Consider using Auto Zone rather than selecting a zone manually. See https://cloud.google.com/dataproc/docs/concepts/configuring-clusters/auto-zone MORE

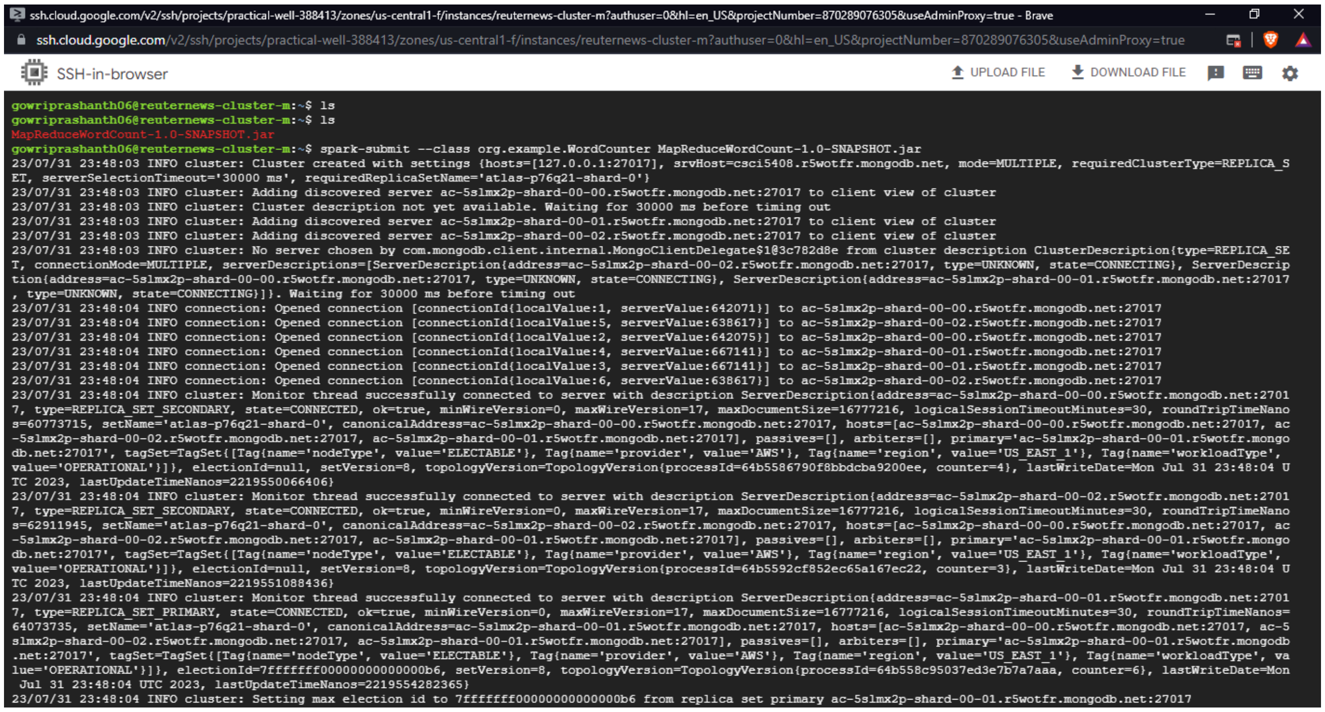
Name	apache-spark-cluster
Cluster UUID	94ec5cd9-b6d5-43c5-969d-20170833d818
Type	Dataproc Cluster
Status	Running

	Name	Role	
✓	apache-spark-cluster-m	Master	SSH

Fig 7: Initializing Apache Spark SSH instance

Step 3:

Once the instance is started the necessary processing can be made


ssh.cloud.google.com/v2/ssh/projects/practical-well-388413/zones/us-central1-f/instances/reuternews-cluster-m?authuser=0&hl=en_US&projectNumber=870289076305&useAdminProxy=true - Brave
ssh.cloud.google.com/v2/ssh/projects/practical-well-388413/zones/us-central1-f/instances/reuternews-cluster-m?authuser=0&hl=en_US&projectNumber=870289076305&useAdminProxy=true
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE

```
gowriprashanth06@reuternews-cluster-m:~$ ls
gowriprashanth06@reuternews-cluster-m:~$ ls
MapReduceWordCount-1.0-SNAPSHOT.jar
gowriprashanth06@reuternews-cluster-m:~$ spark-submit --class org.example.WordCounter MapReduceWordCount-1.0-SNAPSHOT.jar
23/07/31 23:48:03 INFO cluster: Cluster created with settings {hosts=[127.0.0.1:27017], srvHost=csci5408.r5wotfr.mongodb.net, mode=MULTIPLE, requiredClusterType=REPLICA_S
ET, serverSelectionTimeout='30000 ms', requiredReplicaSetName='atlas-p76q21-shard-0'}
23/07/31 23:48:03 INFO cluster: Adding discovered server ac-5slmx2p-shard-00-00.r5wotfr.mongodb.net:27017 to client view of cluster
23/07/31 23:48:03 INFO cluster: Cluster description not yet available. Waiting for 30000 ms before timing out
23/07/31 23:48:03 INFO cluster: Adding discovered server ac-5slmx2p-shard-00-01.r5wotfr.mongodb.net:27017 to client view of cluster
23/07/31 23:48:03 INFO cluster: Adding discovered server ac-5slmx2p-shard-00-02.r5wotfr.mongodb.net:27017 to client view of cluster
23/07/31 23:48:03 INFO cluster: No server chosen by com.mongodb.client.internal.MongoClientDelegate$1@3c782d8e from cluster description ClusterDescription{type=REPLICA_SE
T, connectionMode=MULTIPLE, serverDescriptions=[ServerDescription{address=ac-5slmx2p-shard-00-02.r5wotfr.mongodb.net:27017, type=UNKNOWN, state=CONNECTING}, ServerDescrip
tion{address=ac-5slmx2p-shard-00-00.r5wotfr.mongodb.net:27017, type=UNKNOWN, state=CONNECTING}, ServerDescription{address=ac-5slmx2p-shard-00-01.r5wotfr.mongodb.net:27017
, type=UNKNOWN, state=CONNECTING}]}. Waiting for 30000 ms before timing out
23/07/31 23:48:04 INFO connection: Opened connection [connectionId{localValue:1, serverValue:642071}] to ac-5slmx2p-shard-00-00.r5wotfr.mongodb.net:27017
23/07/31 23:48:04 INFO connection: Opened connection [connectionId{localValue:5, serverValue:638617}] to ac-5slmx2p-shard-00-02.r5wotfr.mongodb.net:27017
23/07/31 23:48:04 INFO connection: Opened connection [connectionId{localValue:2, serverValue:642075}] to ac-5slmx2p-shard-00-00.r5wotfr.mongodb.net:27017
23/07/31 23:48:04 INFO connection: Opened connection [connectionId{localValue:4, serverValue:667141}] to ac-5slmx2p-shard-00-01.r5wotfr.mongodb.net:27017
23/07/31 23:48:04 INFO connection: Opened connection [connectionId{localValue:3, serverValue:667141}] to ac-5slmx2p-shard-00-01.r5wotfr.mongodb.net:27017
23/07/31 23:48:04 INFO connection: Opened connection [connectionId{localValue:6, serverValue:638617}] to ac-5slmx2p-shard-00-02.r5wotfr.mongodb.net:27017
23/07/31 23:48:04 INFO cluster: Monitor thread successfully connected to server with description ServerDescription{address=ac-5slmx2p-shard-00-00.r5wotfr.mongodb.net:2701
7, type=REPLICA_SET_SECONDARY, state=CONNECTED, ok=true, minWireVersion=0, maxWireVersion=17, maxDocumentSize=16777216, logicalSessionTimeoutMinutes=30, roundTripTimeNano
s=60773715, setName='atlas-p76q21-shard-0', canonicalAddress=ac-5slmx2p-shard-00-00.r5wotfr.mongodb.net:27017, hosts=[ac-5slmx2p-shard-00-00.r5wotfr.mongodb.net:27017, ac
-5slmx2p-shard-00-02.r5wotfr.mongodb.net:27017, ac-5slmx2p-shard-00-01.r5wotfr.mongodb.net:27017], passives=[], arbiters=[], primary='ac-5slmx2p-shard-00-01.r5wotfr.mongo
db.net:27017', tagSet=TagSet{[Tag{name='nodeType', value='ELECTABLE'}, Tag{name='provider', value='AWS'}, Tag{name='region', value='US_EAST_1'}, Tag{name='workloadType',
value='OPERATIONAL'}]}, electionId=null, setVersion=8, topologyVersion=TopologyVersion{processId=64b5586790f8bbdcba9200ee, counter=4}, lastWriteDate=Mon Jul 31 23:48:04 U
TC 2023, lastUpdateTimeNanos=2219550066406}
23/07/31 23:48:04 INFO cluster: Monitor thread successfully connected to server with description ServerDescription{address=ac-5slmx2p-shard-00-02.r5wotfr.mongodb.net:2701
7, type=REPLICA_SET_SECONDARY, state=CONNECTED, ok=true, minWireVersion=0, maxWireVersion=17, maxDocumentSize=16777216, logicalSessionTimeoutMinutes=30, roundTripTimeNano
s=62911945, setName='atlas-p76q21-shard-0', canonicalAddress=ac-5slmx2p-shard-00-02.r5wotfr.mongodb.net:27017, hosts=[ac-5slmx2p-shard-00-00.r5wotfr.mongodb.net:27017, ac
-5slmx2p-shard-00-02.r5wotfr.mongodb.net:27017, ac-5slmx2p-shard-00-01.r5wotfr.mongodb.net:27017], passives=[], arbiters=[], primary='ac-5slmx2p-shard-00-01.r5wotfr.mongo
db.net:27017', tagSet=TagSet{[Tag{name='nodeType', value='ELECTABLE'}, Tag{name='provider', value='AWS'}, Tag{name='region', value='US_EAST_1'}, Tag{name='workloadType',
value='OPERATIONAL'}]}, electionId=null, setVersion=8, topologyVersion=TopologyVersion{processId=64b5592cf852ec65a167ec22, counter=3}, lastWriteDate=Mon Jul 31 23:48:04 U
TC 2023, lastUpdateTimeNanos=2219551088436}
23/07/31 23:48:04 INFO cluster: Monitor thread successfully connected to server with description ServerDescription{address=ac-5slmx2p-shard-00-01.r5wotfr.mongodb.net:2701
7, type=REPLICA_SET_PRIMARY, state=CONNECTED, ok=true, minWireVersion=0, maxWireVersion=17, maxDocumentSize=16777216, logicalSessionTimeoutMinutes=30, roundTripTimeNanos=
64073735, setName='atlas-p76q21-shard-0', canonicalAddress=ac-5slmx2p-shard-00-01.r5wotfr.mongodb.net:27017, hosts=[ac-5slmx2p-shard-00-00.r5wotfr.mongodb.net:27017, ac-5
slmx2p-shard-00-02.r5wotfr.mongodb.net:27017, ac-5slmx2p-shard-00-01.r5wotfr.mongodb.net:27017], passives=[], arbiters=[], primary='ac-5slmx2p-shard-00-01.r5wotfr.mongodb
.net:27017', tagSet=TagSet{[Tag{name='nodeType', value='ELECTABLE'}, Tag{name='provider', value='AWS'}, Tag{name='region', value='US_EAST_1'}, Tag{name='workloadType', va
lue='OPERATIONAL'}]}, electionId=7fffffff0000000000000b6, setVersion=8, topologyVersion=TopologyVersion{processId=64b558c95037ed3e7b7a7aaa, counter=6}, lastWriteDate=Mon
 Jul 31 23:48:04 UTC 2023, lastUpdateTimeNanos=2219554282365}
23/07/31 23:48:04 INFO cluster: Setting max election id to 7fffffff0000000000000b6 from replica set primary ac-5slmx2p-shard-00-01.r5wotfr.mongodb.net:27017
```

Fig 8: Apache Spark SSH instance

MapReduce program using Java (WordCounter.java Engine) to count (frequency count) the unique words found in reut2-009.sgm (newsArticle1 Collection) was created.

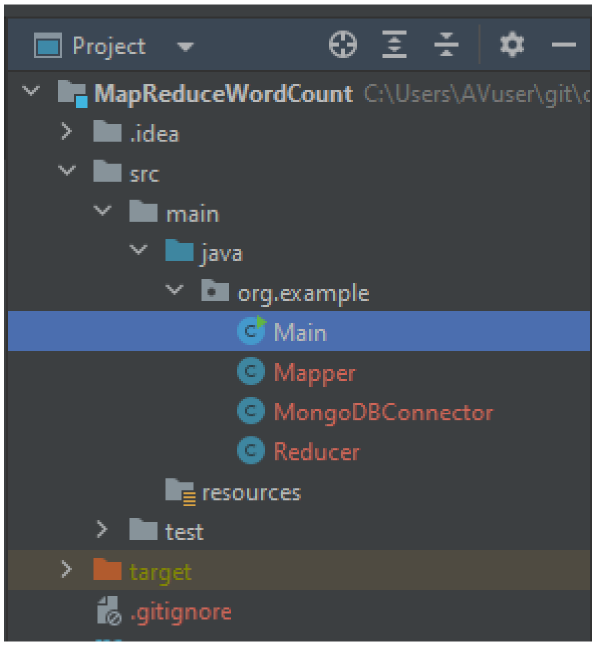


Fig 8: Project Structure

MapReduce Algorithm:

- Initialize the MongoDB connection by creating a MongoDBConnector object with the connection string, database name - "ReuterDb”, and collection name "newsArticle1".

- Get the MongoDB collection using the getCollection() method of the MongoDBConnector.

- Create a Mapper object and a Reducer object to perform the map and reduce operations, respectively.

- **Mapper:**

    1. Start the map function by taking two parameters: line and output
    2. Trim the line to remove leading and trailing spaces.
    3. Split the line into individual words using a regular expression (\\s+ matches one or more whitespace characters).
    4. Initialize a loop to iterate through each word in the words array.
    5. Within the loop, check if the word is already present in the output map using output.containsKey(word).
    6. If the word is not in the output map, add it with a count of 1 using output.put(word, 1).
    7. If the word is already in the output map, retrieve its current count using output.get(word), and then update the count by incrementing it by 1.
    8. Continue the loop until all words in the line have been processed.
    9. The output map now contains intermediate key-value pairs, where each key is a unique word, and the value is the count of occurrences of that word in the line.

- **Reducer:**
    1. Start the reduce function by taking one parameter: intermediate
    2. Create a new result map to store the final output after reducing.
    3. Initialize a loop to iterate through each entry in the intermediate.entrySet().
    4. Within the loop, retrieve the word and count from the entry.
    5. Add the word and its count directly to the result map using result.put(word, count).
    6. Continue the loop until all entries in the intermediate map have been processed.
    7. The result map now contains the final output, where each key is a unique word, and the value is the total count of occurrences of that word across all lines processed.

- Initialize an empty HashMap called intermediate to store the word frequencies after mapping.

- Iterate over each document in the MongoDB collection:

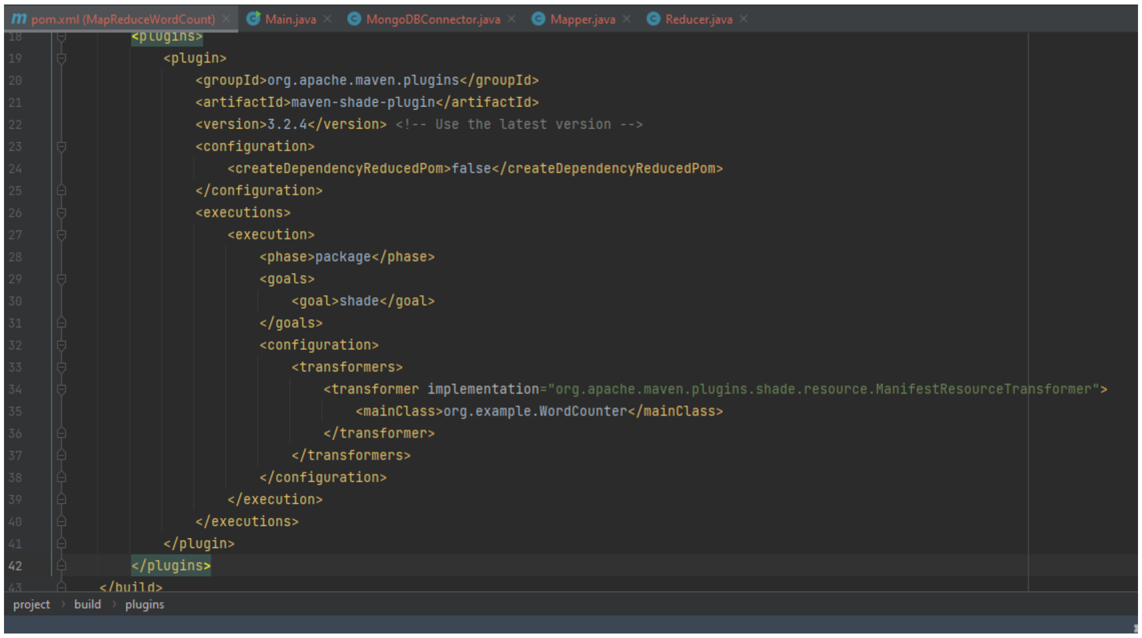
- Retrieve the text from the current document.

- Call the map method of the Mapper object, passing the text and the intermediate map to update the word frequencies.

- After processing all the documents, create an empty HashMap called output to store the final word frequencies.
- Call the reduce method of the Reducer object, passing the intermediate map to get the final word frequencies. Store the result in the output map.
- Iterate over the output map and print each word along with its frequency to the console.
- Close the MongoDB connection using the closeConnection() method of the MongoDBConnector.

Packaged the maven project to obtain the JAR file

Used the below plugin to obtain .jar file along with the dependencies (MongoDb) to enable it in apache spark.

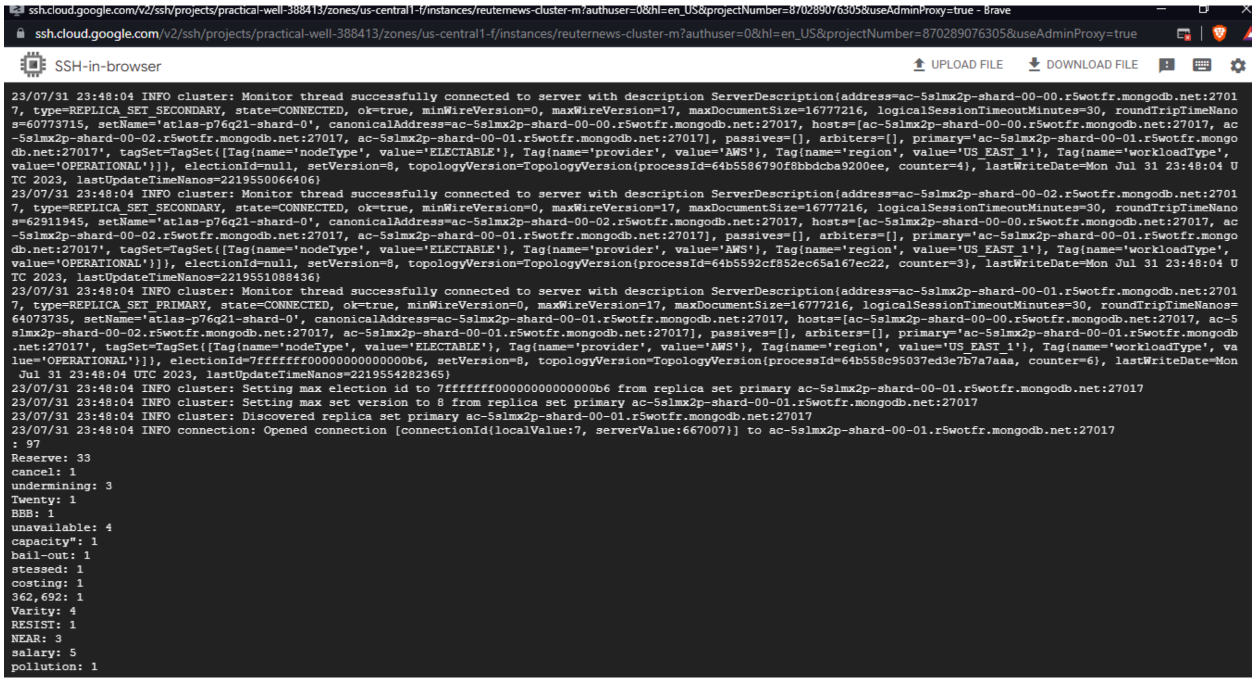


```
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-shade-plugin</artifactId>
                <version>3.2.4</version> <!-- Use the latest version -->
                <configuration>
                    <createDependencyReducedPom>false</createDependencyReducedPom>
                </configuration>
                <executions>
                    <execution>
                        <phase>package</phase>
                        <goals>
                            <goal>shade</goal>
                        </goals>
                        <configuration>
                            <transformers>
                                <transformer implementation="org.apache.maven.plugins.shade.resource.ManifestResourceTransformer">
                                    <mainClass>org.example.WordCounter</mainClass>
                                </transformer>
                            </transformers>
                        </configuration>
                    </execution>
                </executions>
            </plugin>
        </plugins>
    </build>
```

Fig 9 : Plugin to create runnable JAR with dependencies

Submitted the jar file in apache spark and obtained the output for the program

Used the command : spark-submit --class org.example.WordCounter MapReduceWordCount-1.0-SNAPSHOT.jar



```
23/07/31 23:48:04 INFO cluster: Monitor thread successfully connected to server with description ServerDescription{address=ac-5slmx2p-shard-00-00.r5wotfr.mongodb.net:2701
7, type=REPLICA_SET_SECONDARY, state=CONNECTED, ok=true, minWireVersion=0, maxWireVersion=17, maxDocumentSize=16777216, logicalSessionTimeoutMinutes=30, roundTripTimeNano
s=60773715, setName='atlas-p76q21-shard-0', canonicalAddress=ac-5slmx2p-shard-00-00.r5wotfr.mongodb.net:27017, hosts=[ac-5slmx2p-shard-00-00.r5wotfr.mongodb.net:27017, ac
-5slmx2p-shard-00-02.r5wotfr.mongodb.net:27017, ac-5slmx2p-shard-00-01.r5wotfr.mongodb.net:27017], passives=[], arbiters=[], primary='ac-5slmx2p-shard-00-01.r5wotfr.mongo
db.net:27017', tagSet=TagSet{[Tag{name='nodeType', value='ELECTABLE'}, Tag{name='provider', value='AWS'}, Tag{name='region', value='US_EAST_1'}, Tag{name='workloadType',
value='OPERATIONAL'}]}, electionId=null, setVersion=8, topologyVersion=TopologyVersion{processId=64b5586790f8bbdcba9200ee, counter=4}, lastWriteDate=Mon Jul 31 23:48:04 U
TC 2023, lastUpdateTimeNanos=2219550066406}
23/07/31 23:48:04 INFO cluster: Monitor thread successfully connected to server with description ServerDescription{address=ac-5slmx2p-shard-00-02.r5wotfr.mongodb.net:2701
7, type=REPLICA_SET_SECONDARY, state=CONNECTED, ok=true, minWireVersion=0, maxWireVersion=17, maxDocumentSize=16777216, logicalSessionTimeoutMinutes=30, roundTripTimeNano
s=62911945, setName='atlas-p76q21-shard-0', canonicalAddress=ac-5slmx2p-shard-00-02.r5wotfr.mongodb.net:27017, hosts=[ac-5slmx2p-shard-00-00.r5wotfr.mongodb.net:27017, ac
-5slmx2p-shard-00-02.r5wotfr.mongodb.net:27017, ac-5slmx2p-shard-00-01.r5wotfr.mongodb.net:27017], passives=[], arbiters=[], primary='ac-5slmx2p-shard-00-01.r5wotfr.mongo
db.net:27017', tagSet=TagSet{[Tag{name='nodeType', value='ELECTABLE'}, Tag{name='provider', value='AWS'}, Tag{name='region', value='US_EAST_1'}, Tag{name='workloadType',
value='OPERATIONAL'}]}, electionId=null, setVersion=8, topologyVersion=TopologyVersion{processId=64b5592cf852ec65a167ec22, counter=3}, lastWriteDate=Mon Jul 31 23:48:04 U
TC 2023, lastUpdateTimeNanos=2219551088436}
23/07/31 23:48:04 INFO cluster: Monitor thread successfully connected to server with description ServerDescription{address=ac-5slmx2p-shard-00-01.r5wotfr.mongodb.net:2701
7, type=REPLICA_SET_PRIMARY, state=CONNECTED, ok=true, minWireVersion=0, maxWireVersion=17, maxDocumentSize=16777216, logicalSessionTimeoutMinutes=30, roundTripTimeNanos=
64073735, setName='atlas-p76q21-shard-0', canonicalAddress=ac-5slmx2p-shard-00-01.r5wotfr.mongodb.net:27017, hosts=[ac-5slmx2p-shard-00-00.r5wotfr.mongodb.net:27017, ac-5
slmx2p-shard-00-02.r5wotfr.mongodb.net:27017, ac-5slmx2p-shard-00-01.r5wotfr.mongodb.net:27017], passives=[], arbiters=[], primary='ac-5slmx2p-shard-00-01.r5wotfr.mongodb
.net:27017', tagSet=TagSet{[Tag{name='nodeType', value='ELECTABLE'}, Tag{name='provider', value='AWS'}, Tag{name='region', value='US_EAST_1'}, Tag{name='workloadType', va
lue='OPERATIONAL'}]}, electionId=7fffffff0000000000000b6, setVersion=8, topologyVersion=TopologyVersion{processId=64b558c95037ed3e7b7a7aaa, counter=6}, lastWriteDate=Mon
 Jul 31 23:48:04 UTC 2023, lastUpdateTimeNanos=2219554282365}
23/07/31 23:48:04 INFO cluster: Setting max election id to 7fffffff0000000000000b6 from replica set primary ac-5slmx2p-shard-00-01.r5wotfr.mongodb.net:27017
23/07/31 23:48:04 INFO cluster: Setting max set version to 8 from replica set primary ac-5slmx2p-shard-00-01.r5wotfr.mongodb.net:27017
23/07/31 23:48:04 INFO cluster: Discovered replica set primary ac-5slmx2p-shard-00-01.r5wotfr.mongodb.net:27017
23/07/31 23:48:04 INFO connection: Opened connection [connectionId{localValue:7, serverValue:667007}] to ac-5slmx2p-shard-00-01.r5wotfr.mongodb.net:27017
: 97
Reserve: 33
cancel: 1
undermining: 3
Twenty: 1
BBB: 1
unavailable: 4
capacity": 1
bail-out: 1
stessed: 1
costing: 1
362,692: 1
Varity: 4
RESIST: 1
NEAR: 3
salary: 5
pollution: 1
```

Fig 10 : Output from Apache Spark instance

lo#2: Build a light-weight analytics engine, which will perform custom ETL operation, and one specific analysis (sentiment and semantic)

Problem 2: Sentiment Analysis using BOW model on title of Reuters News Articles

To build the light weight Sentiment Analysis engine, the Project was created.

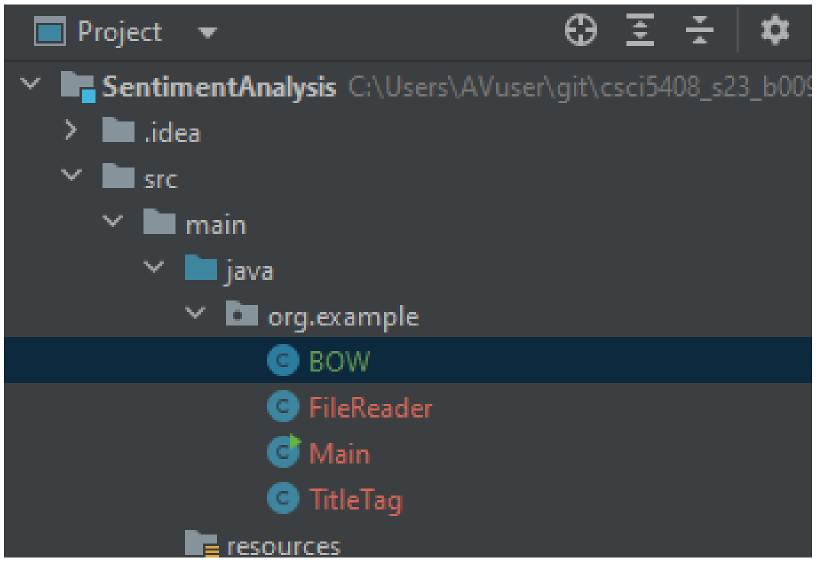


Fig 11 : Project structure

1. Bag of words(BOW):

The method BagOfWords processes the title to create a "bag of words" representation, where it counts the occurrences of each word and stores the results in a map.

The regex library is used with following regex pattern: [3]

Pattern wordPattern = Pattern.compile("\\b\\w+\\b"); [4]

In the pattern ("\\b\\w+\\b")

- \\b: The \\b is a word boundary anchor. It matches the position between a word character (\\w) and a non-word character
- \\w+: The \\w is a shorthand character class that matches any word character (letters, digits, or underscores). The + quantifier means "one or more times," so \\w+ matches one or more word characters
- \\b: Another word boundary anchor, which matches the end of a word.

Combining these elements, the regular expression \\b\\w+\\b will match any word in the input text.

The bag of words is case-insensitive, so the counts are not affected by the word's capitalization.

The method returns the resulting bag of words as a map, where each word from the input title is a key, and the corresponding value is its frequency (number of occurrences).

2.TitleTag

The following files are used to list of positive and negative words.

String filePath1 = "C:/Users/AVuser/Documents/Data/Assignment_3/positive-words.txt"; [5]

String filePath2 = "C:/Users/AVuser/Documents/Data/Assignment_3/negative-words.txt"; [6]

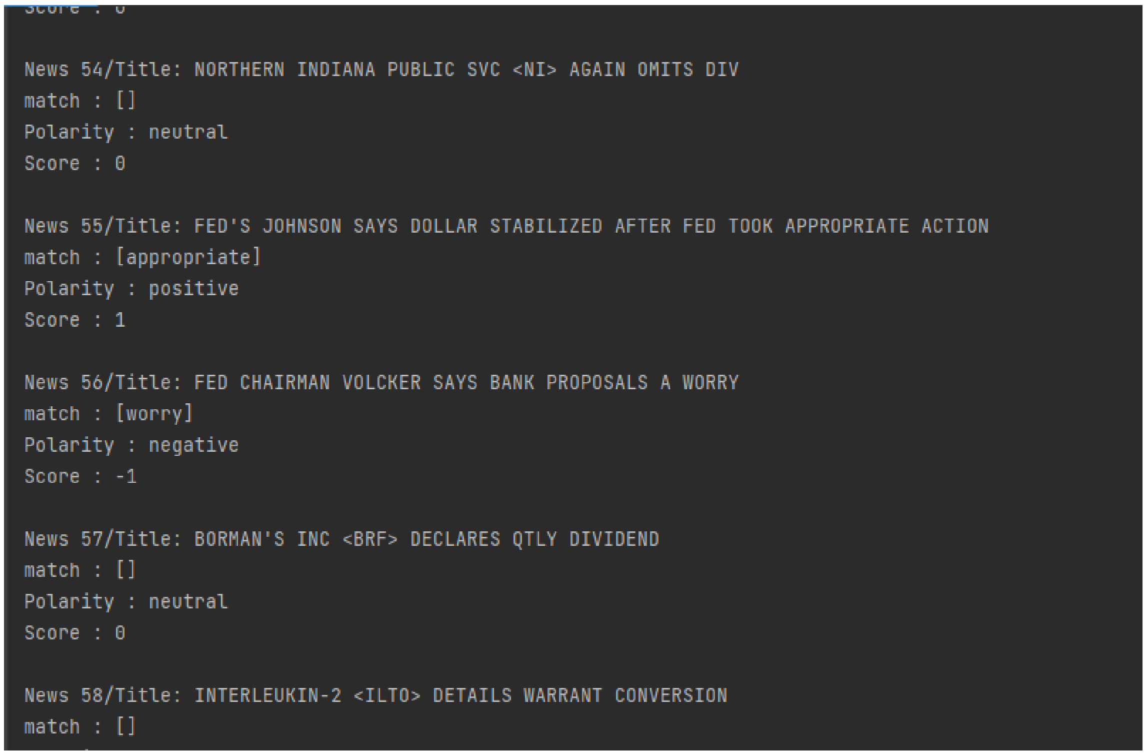
They were downloaded from GitHub and from the user:

mkulakowski2 [5], [6], [7], [8]

The words are extracted from the files ad stored in String List. A word by word comparison with a list of positive and negative words and the words in BOW is made. Scores are tagged with the words as :
positive words : 1
negative words : -1
neutral words : 0

3.Sentiment Analysis

- A MongoDB connection is made to ReuterDb database and the news titles are fetched.
- The data in title is cleaned (“&lt” is replaced with “<”)
- Each titles is tagged with positive, negative or neutral based on the score of their words.
- The match word, polarity and score is displayed.



```
News 54/Title: NORTHERN INDIANA PUBLIC SVC <NI> AGAIN OMITS DIV
match : []
Polarity : neutral
Score : 0

News 55/Title: FED'S JOHNSON SAYS DOLLAR STABILIZED AFTER FED TOOK APPROPRIATE ACTION
match : [appropriate]
Polarity : positive
Score : 1

News 56/Title: FED CHAIRMAN VOLCKER SAYS BANK PROPOSALS A WORRY
match : [worry]
Polarity : negative
Score : -1

News 57/Title: BORMAN'S INC <BRF> DECLARES QTLY DIVIDEND
match : []
Polarity : neutral
Score : 0

News 58/Title: INTERLEUKIN-2 <ILTO> DETAILS WARRANT CONVERSION
match : []
```

Fig 12 : Output from Sentiment Analysis

References:

[1] "Connect To MongoDB," MongoDb, [Online]. Available: https://www.mongodb.com/docs/drivers/java/sync/v4.3/fundamentals/connection/connect/#std-label-connect-to-mongodb [Accessed: July 29, 2023].

[2] "Java Regular Expressions," w3schools, [Online]. Available: https://www.w3schools.com/java/java_regex.asp [Accessed: July 29, 2023].

[3] " Difference between \w and \b regular expression meta characters," stackoverflow, [Online]. Available: https://www.w3schools.com/java/java_regex.asp [Accessed: July 29, 2023].

[4] " Class Pattern," oracle, [Online]. Available: https://docs.oracle.com/javase/8/docs/api/java/util/regex/Pattern.html [Accessed: July 29, 2023].

[5] " mkulakowski2/negative-words.txtstackoverflow", Github Gist [Online]. Available: https://gist.github.com/mkulakowski2/4289441 [Accessed: July 29, 2023].

[6] "mkulakowski2/positive-words.txt", Github Gist [Online]. Available: https://gist.github.com/mkulakowski2/4289437 [Accessed: July 29, 2023].

[7] "Mining and Summarizing Customer Reviews", University of Illinois Chicago [Online]. Available: https://www.cs.uic.edu/~liub/publications/kdd04-revSummary.pdf [Accessed: July 29, 2023].

[8] "Opinion Observer: Analyzing and Comparing Opinions on the Web", University of Illinois Chicago [Online]. Available: https://www.cs.uic.edu/~liub/publications/www05-p536.pdf [Accessed: July 29, 2023].