Simple Search Engine Report

Assignment 2: Simple Search Engine using Hadoop MapReduce, Cassandra, and other powerful tools
Student: Davydov Danil
Group: DS-01
Innnopolis email address: d.davydov@innopolis.university
Telegram: @tanellD

---

1. Introduction

- In this assignment I was asked to implement simple search engine using Spark, Cassandra, HDFS. I gained knowledge in doing it, also I implemented Map-Reduce pipelines

---

2. Methodology

2.1 Design Choices

- **Data Pipeline**: The script prepare_data.sh and prepare_data.py are used to read parquet file, select 1000 texts and save them into app/data folder, in .sh script those files are copied to hdfs and in .py the csv file is saved into /index/data as index of files. If the user wants to index addditional files he should paste .txt files into a new folder and pass the folder path into index.sh as an argument to index additional documents. The documents are concatenated to the files that were indexed in prepare_data scripts. Next I have 2 Map-Reduce pipelines to store files in cassandra, during this pipelines all metrics for BM25 calculation will be determined and stored in cassandra. The last stage is searching with search.sh and query.py. With this scripts I calculate scores for each document by the query and print top-10 the most relevant documents. The output has filename, score, and part of a text.

The pipeline looks as follows:

```
parquet file ---> prepare_data scripts ---> index.sh ---> 2 Map-Reduce pipelines --->
search for relevant
                                    ^
                                    |
                folder with additional files to index (optional)
```

- **Indexing Strategy**: My cassandra schema is defined by script:

```
CREATE KEYSPACE IF NOT EXISTS document_index
WITH replication = {'class': 'SimpleStrategy', 'replication_factor': '1'};

USE document_index;

CREATE TABLE IF NOT EXISTS vocabulary (
    term text PRIMARY KEY
);

CREATE TABLE IF NOT EXISTS document_lengths (
    doc_id text PRIMARY KEY,
    doc_length int
);

CREATE TABLE IF NOT EXISTS term_frequencies (
    term text,
    doc_id text,
    frequency int,
    PRIMARY KEY (term, doc_id)
);

CREATE TABLE IF NOT EXISTS document_frequencies (
    term text PRIMARY KEY,
    doc_frequency int
);

CREATE TABLE IF NOT EXISTS document_texts (
    doc_id text PRIMARY KEY,
    content text
);
```

The table `document_texts` is created to show slice of a text in output, other tables are needed to calculate BM25 score for documents by query. The initialization of schema is in app.sh script.

- **Ranking Model**: The bm25 has 2 parameters which I defined as k1 = 1.2 and b = 0.75

2.2 Implementation Details

Indexer (MapReduce Pipeline)

- I splitted the code into 2 MapReduce pipelines the first calculates all metrics while the second loads them into cassandra, such that my code is simpler to understand.

Ranker (Spark RDD)

- **Query Processing**: I used the same strategy to preprocess the texts and query, I remove stopwords and stem the tokens.
- **BM25 Calculation**: I use a modification of BM25 to rank documents (Variant Lucene/Elasticsearch), the calculation looks like

BM25 Scoring Formula (Lucene/Elasticsearch Variant)

IDF (Inverse Document Frequency):

$$IDF(t) = \log\left(\frac{N - df(t) + 0.5}{df(t) + 0.5} + 1\right)$$

TF (Term Frequency):

$$TF(t,d) = \frac{tf(t,d) \cdot (k_1 + 1)}{tf(t,d) + k_1 \cdot \left(1 - b + b \cdot \frac{|d|}{avgdl}\right)}$$

Full BM25 Score:

$$\text{Score}(q,d) = \sum_{t \in q} IDF(t) \cdot TF(t,d)$$

Here is the comparison of classic BM25 and Variant Lucene/Elasticsearch:

BM25 vs. Lucene/Elasticsearch BM25 Variants: IDF and Score Ranges

Feature	Classic BM25	Lucene/Elasticsearch BM25
IDF for Common Terms	Can go negative for very common terms	Always non-negative due to the "1+" term
IDF for Terms in All Docs	Negative values possible	Floors at 0 to avoid negative contribution
Score Range - Lower Bound	Can produce negative scores for common terms	Always non-negative scores
Impact on Ranking	Higher contrast between common and rare terms	More balanced influence between common and rare terms

- **Top-K Retrieval**: Spark operations to rank and filter results.
  I use spark RDD for calculating BM25 score in parallel in query.py:

```
# other code
doc_ids_rdd = sc.parallelize(list(doc_lengths.keys()))
# other code
# other code
doc_scores = doc_ids_rdd.map(
    lambda doc_id: (
        doc_id,
        calculate_bm25(
            query_terms,
            doc_id,
            doc_lengths.get(doc_id, 0),
            avg_doc_len,
            term_freqs,
            doc_freqs,
            total_docs
        )
    )
)

top_docs = doc_scores.sortBy(lambda x: -x[1]).take(10)
# other code
```

3. Demonstration

3.1 How to Run the Project

Run

```
docker-compose build
docker-compose up
```

If you want to chage query: change it directly in the end of app.sh:

```
bash search.sh "paste_your_query"
```

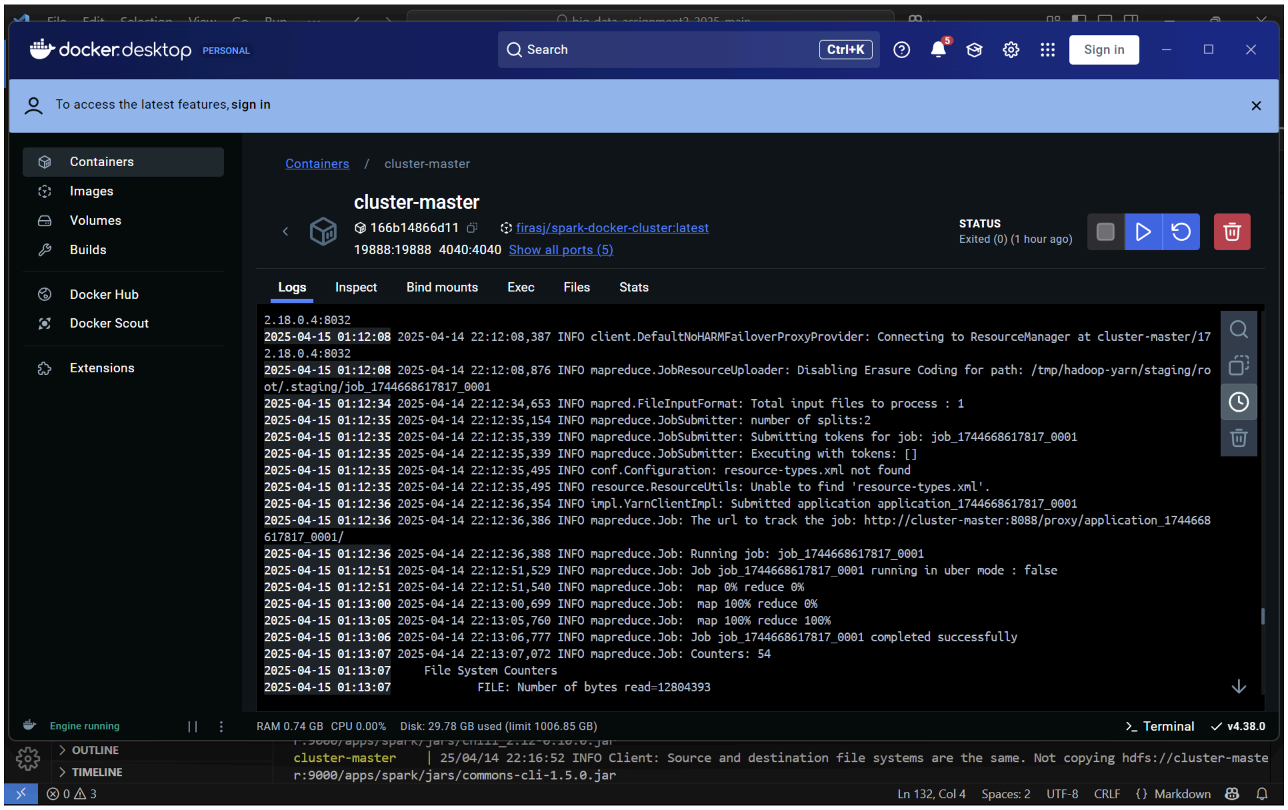
If you want to index additional documents add path to the folder with additional txt files:

```
bash index.sh /app/your_folder
```

3.2 Screenshots & Results

Indexing Success

- **Screenshot 1**: Fullscreen terminal output showing successful indexing of documents.



cluster-master

166b14866d11 firasj/spark-docker-cluster:latest

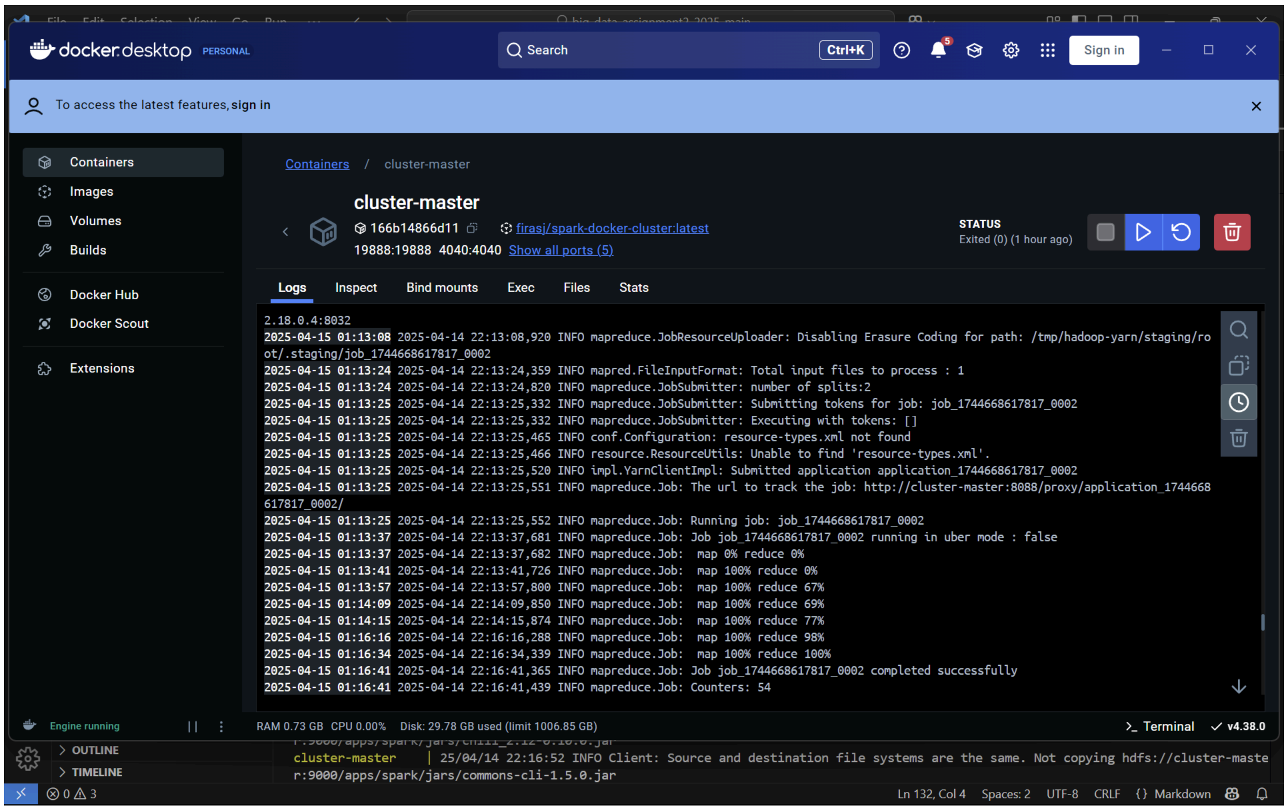
19888:19888 4040:4040 Show all ports (5)

STATUS
Exited (0) (1 hour ago)

```
2.18.0.4:8032
2025-04-15 01:12:08 2025-04-14 22:12:08,387 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at cluster-master/17
2.18.0.4:8032
2025-04-15 01:12:08 2025-04-14 22:12:08,876 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/ro
ot/.staging/job_1744668617817_0001
2025-04-15 01:12:34 2025-04-14 22:12:34,653 INFO mapred.FileInputFormat: Total input files to process : 1
2025-04-15 01:12:35 2025-04-14 22:12:35,154 INFO mapreduce.JobSubmitter: number of splits:2
2025-04-15 01:12:35 2025-04-14 22:12:35,339 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1744668617817_0001
2025-04-15 01:12:35 2025-04-14 22:12:35,339 INFO mapreduce.JobSubmitter: Executing with tokens: []
2025-04-15 01:12:35 2025-04-14 22:12:35,495 INFO conf.Configuration: resource-types.xml not found
2025-04-15 01:12:35 2025-04-14 22:12:35,495 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2025-04-15 01:12:36 2025-04-14 22:12:36,354 INFO impl.YarnClientImpl: Submitted application application_1744668617817_0001
2025-04-15 01:12:36 2025-04-14 22:12:36,386 INFO mapreduce.Job: The url to track the job: http://cluster-master:8088/proxy/application_1744668
617817_0001/
2025-04-15 01:12:36 2025-04-14 22:12:36,388 INFO mapreduce.Job: Running job: job_1744668617817_0001
2025-04-15 01:12:51 2025-04-14 22:12:51,529 INFO mapreduce.Job: Job job_1744668617817_0001 running in uber mode : false
2025-04-15 01:12:51 2025-04-14 22:12:51,540 INFO mapreduce.Job:  map 0% reduce 0%
2025-04-15 01:13:00 2025-04-14 22:13:00,699 INFO mapreduce.Job:  map 100% reduce 0%
2025-04-15 01:13:05 2025-04-14 22:13:05,760 INFO mapreduce.Job:  map 100% reduce 100%
2025-04-15 01:13:06 2025-04-14 22:13:06,777 INFO mapreduce.Job: Job job_1744668617817_0001 completed successfully
2025-04-15 01:13:07 2025-04-14 22:13:07,072 INFO mapreduce.Job: Counters: 54
2025-04-15 01:13:07     File System Counters
2025-04-15 01:13:07             FILE: Number of bytes read=12804393
```

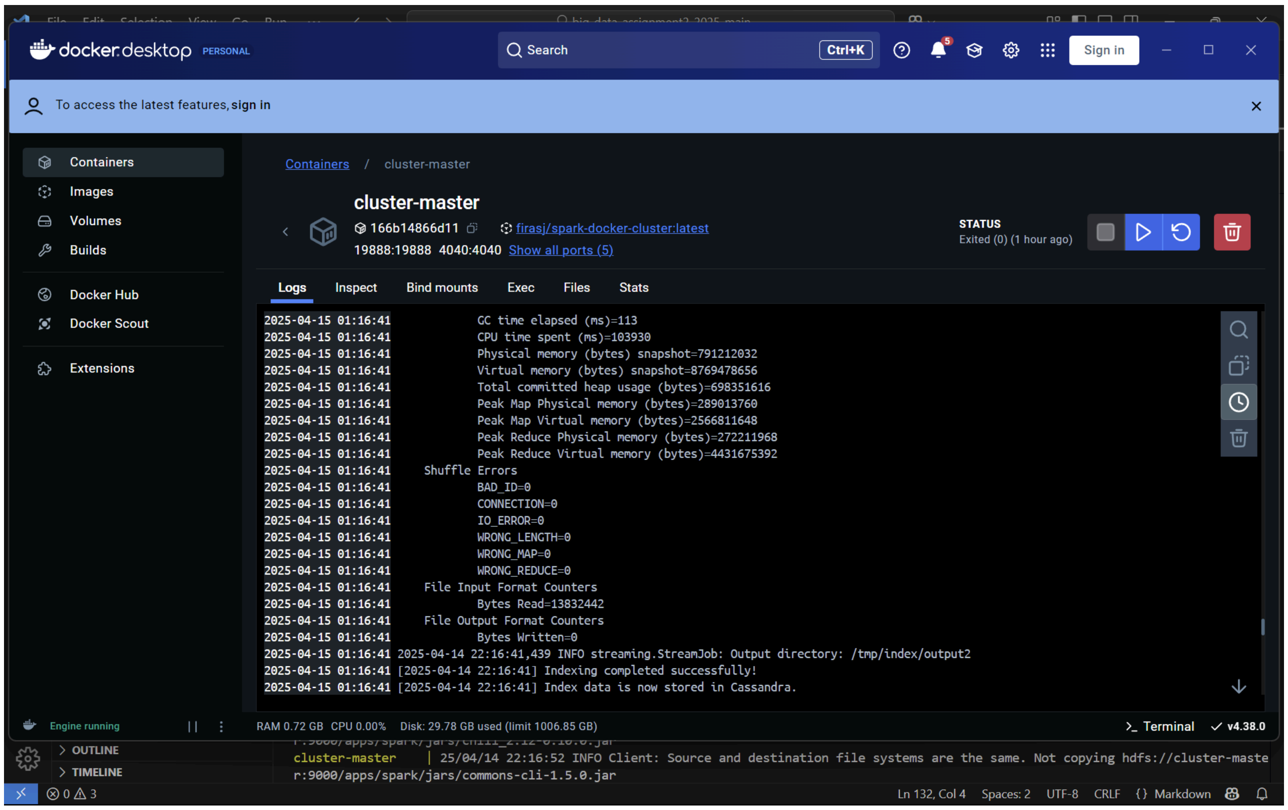
RAM 0.74 GB CPU 0.00% Disk: 29.78 GB used (limit 1006.85 GB)

```
cluster-master     | 25/04/14 22:16:52 INFO Client: Source and destination file systems are the same. Not copying hdfs://cluster-maste
r:9000/apps/spark/jars/commons-cli-1.5.0.jar
```



```
2.18.0.4:8032
2025-04-15 01:13:08 2025-04-14 22:13:08,920 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/ro
ot/.staging/job_1744668617817_0002
2025-04-15 01:13:24 2025-04-14 22:13:24,359 INFO mapred.FileInputFormat: Total input files to process : 1
2025-04-15 01:13:24 2025-04-14 22:13:24,820 INFO mapreduce.JobSubmitter: number of splits:2
2025-04-15 01:13:25 2025-04-14 22:13:25,332 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1744668617817_0002
2025-04-15 01:13:25 2025-04-14 22:13:25,332 INFO mapreduce.JobSubmitter: Executing with tokens: []
2025-04-15 01:13:25 2025-04-14 22:13:25,465 INFO conf.Configuration: resource-types.xml not found
2025-04-15 01:13:25 2025-04-14 22:13:25,466 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2025-04-15 01:13:25 2025-04-14 22:13:25,520 INFO impl.YarnClientImpl: Submitted application application_1744668617817_0002
2025-04-15 01:13:25 2025-04-14 22:13:25,551 INFO mapreduce.Job: The url to track the job: http://cluster-master:8088/proxy/application_1744668
617817_0002/
2025-04-15 01:13:25 2025-04-14 22:13:25,552 INFO mapreduce.Job: Running job: job_1744668617817_0002
2025-04-15 01:13:37 2025-04-14 22:13:37,681 INFO mapreduce.Job: Job job_1744668617817_0002 running in uber mode : false
2025-04-15 01:13:37 2025-04-14 22:13:37,682 INFO mapreduce.Job:  map 0% reduce 0%
2025-04-15 01:13:41 2025-04-14 22:13:41,726 INFO mapreduce.Job:  map 100% reduce 0%
2025-04-15 01:13:57 2025-04-14 22:13:57,800 INFO mapreduce.Job:  map 100% reduce 67%
2025-04-15 01:14:09 2025-04-14 22:14:09,850 INFO mapreduce.Job:  map 100% reduce 69%
2025-04-15 01:14:15 2025-04-14 22:14:15,874 INFO mapreduce.Job:  map 100% reduce 77%
2025-04-15 01:16:16 2025-04-14 22:16:16,288 INFO mapreduce.Job:  map 100% reduce 98%
2025-04-15 01:16:34 2025-04-14 22:16:34,339 INFO mapreduce.Job:  map 100% reduce 100%
2025-04-15 01:16:41 2025-04-14 22:16:41,365 INFO mapreduce.Job: Job job_1744668617817_0002 completed successfully
2025-04-15 01:16:41 2025-04-14 22:16:41,439 INFO mapreduce.Job: Counters: 54
```

```
cluster-master     | 25/04/14 22:16:52 INFO Client: Source and destination file systems are the same. Not copying hdfs://cluster-maste
r:9000/apps/spark/jars/commons-cli-1.5.0.jar
```

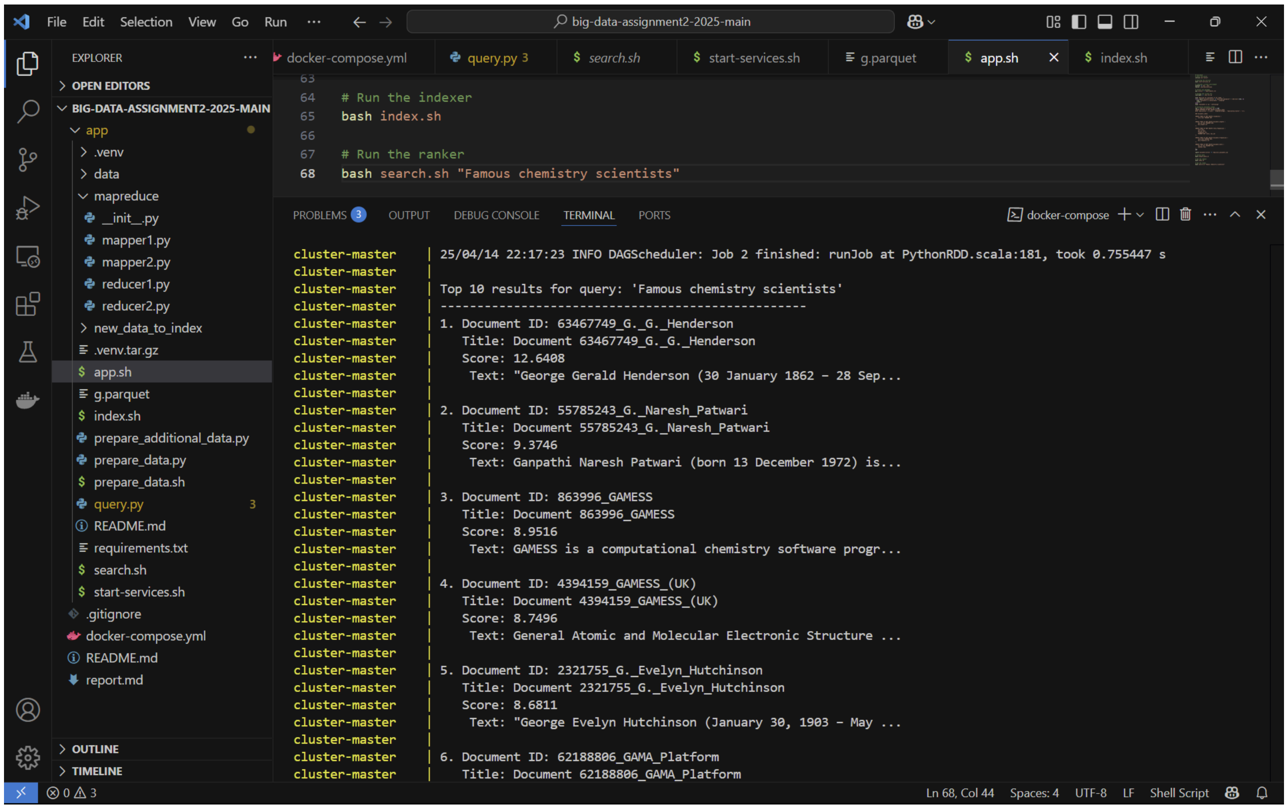


```
2025-04-15 01:16:41             GC time elapsed (ms)=113
2025-04-15 01:16:41             CPU time spent (ms)=103930
2025-04-15 01:16:41             Physical memory (bytes) snapshot=791212032
2025-04-15 01:16:41             Virtual memory (bytes) snapshot=8769478656
2025-04-15 01:16:41             Total committed heap usage (bytes)=698351616
2025-04-15 01:16:41             Peak Map Physical memory (bytes)=289013760
2025-04-15 01:16:41             Peak Map Virtual memory (bytes)=2566811648
2025-04-15 01:16:41             Peak Reduce Physical memory (bytes)=272211968
2025-04-15 01:16:41             Peak Reduce Virtual memory (bytes)=4431675392
2025-04-15 01:16:41     Shuffle Errors
2025-04-15 01:16:41             BAD_ID=0
2025-04-15 01:16:41             CONNECTION=0
2025-04-15 01:16:41             IO_ERROR=0
2025-04-15 01:16:41             WRONG_LENGTH=0
2025-04-15 01:16:41             WRONG_MAP=0
2025-04-15 01:16:41             WRONG_REDUCE=0
2025-04-15 01:16:41     File Input Format Counters
2025-04-15 01:16:41             Bytes Read=13832442
2025-04-15 01:16:41     File Output Format Counters
2025-04-15 01:16:41             Bytes Written=0
2025-04-15 01:16:41 2025-04-14 22:16:41,439 INFO streaming.StreamJob: Output directory: /tmp/index/output2
2025-04-15 01:16:41 [2025-04-14 22:16:41] Indexing completed successfully!
2025-04-15 01:16:41 [2025-04-14 22:16:41] Index data is now stored in Cassandra.
```

```
cluster-master     | 25/04/14 22:16:52 INFO Client: Source and destination file systems are the same. Not copying hdfs://cluster-maste
r:9000/apps/spark/jars/commons-cli-1.5.0.jar
```

Query Execution

- **Query 1**: "Famous chemistry scientists"

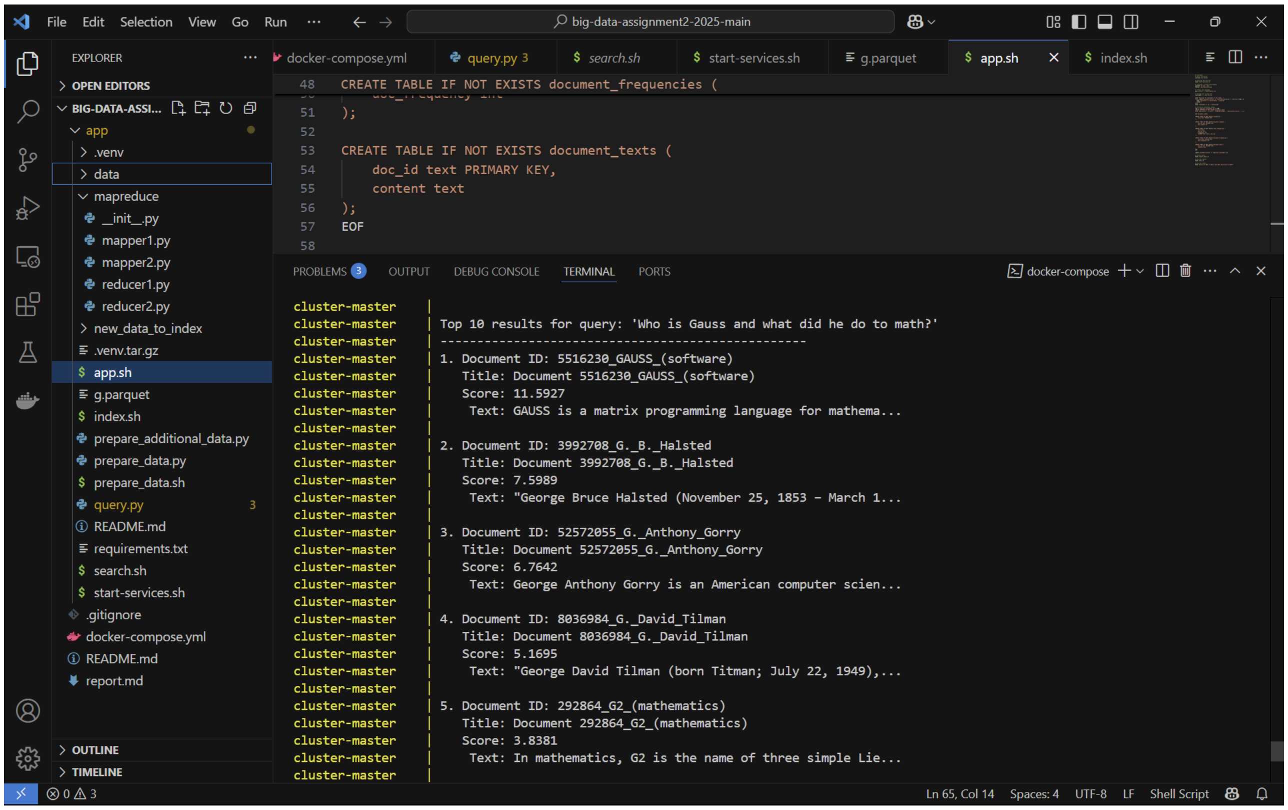


```
# Run the indexer
bash index.sh

# Run the ranker
bash search.sh "Famous chemistry scientists"
```

```
cluster-master  | 25/04/14 22:17:23 INFO DAGScheduler: Job 2 finished: runJob at PythonRDD.scala:181, took 0.755447 s
cluster-master  |
cluster-master  | Top 10 results for query: 'Famous chemistry scientists'
cluster-master  | --------------------------------------------------
cluster-master  | 1. Document ID: 63467749_G._G._Henderson
cluster-master  |    Title: Document 63467749_G._G._Henderson
cluster-master  |    Score: 12.6408
cluster-master  |     Text: "George Gerald Henderson (30 January 1862 – 28 Sep...
cluster-master  |
cluster-master  | 2. Document ID: 55785243_G._Naresh_Patwari
cluster-master  |    Title: Document 55785243_G._Naresh_Patwari
cluster-master  |    Score: 9.3746
cluster-master  |     Text: Ganpathi Naresh Patwari (born 13 December 1972) is...
cluster-master  |
cluster-master  | 3. Document ID: 863996_GAMESS
cluster-master  |    Title: Document 863996_GAMESS
cluster-master  |    Score: 8.9516
cluster-master  |     Text: GAMESS is a computational chemistry software progr...
cluster-master  |
cluster-master  | 4. Document ID: 4394159_GAMESS_(UK)
cluster-master  |    Title: Document 4394159_GAMESS_(UK)
cluster-master  |    Score: 8.7496
cluster-master  |     Text: General Atomic and Molecular Electronic Structure ...
cluster-master  |
cluster-master  | 5. Document ID: 2321755_G._Evelyn_Hutchinson
cluster-master  |    Title: Document 2321755_G._Evelyn_Hutchinson
cluster-master  |    Score: 8.6811
cluster-master  |     Text: "George Evelyn Hutchinson (January 30, 1903 – May ...
cluster-master  |
cluster-master  | 6. Document ID: 62188806_GAMA_Platform
cluster-master  |    Title: Document 62188806_GAMA_Platform
```

- **Query 2**: "Who is Gauss and what did he do to math?"

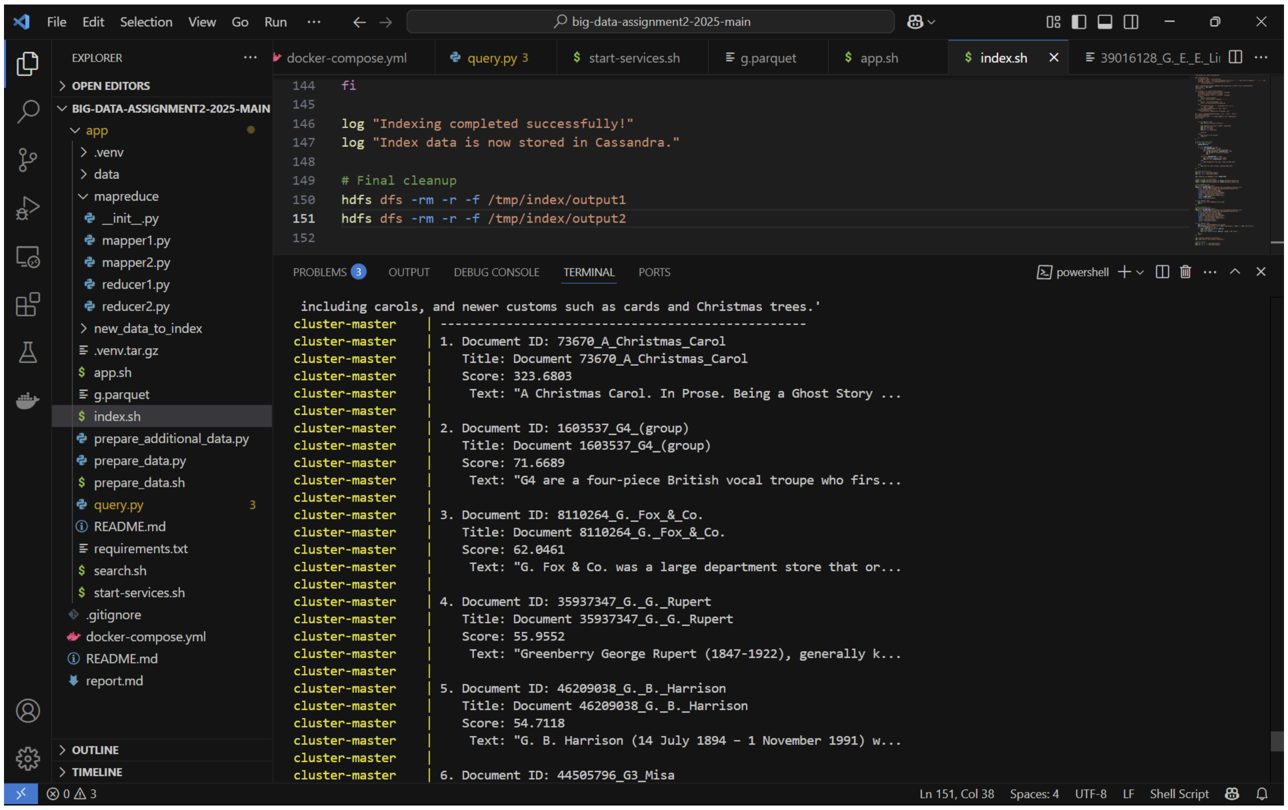


```
CREATE TABLE IF NOT EXISTS document_frequencies (
);

CREATE TABLE IF NOT EXISTS document_texts (
    doc_id text PRIMARY KEY,
    content text
);
EOF
```

```
cluster-master  |
cluster-master  | Top 10 results for query: 'Who is Gauss and what did he do to math?'
cluster-master  | --------------------------------------------------
cluster-master  | 1. Document ID: 5516230_GAUSS_(software)
cluster-master  |    Title: Document 5516230_GAUSS_(software)
cluster-master  |    Score: 11.5927
cluster-master  |     Text: GAUSS is a matrix programming language for mathema...
cluster-master  |
cluster-master  | 2. Document ID: 3992708_G._B._Halsted
cluster-master  |    Title: Document 3992708_G._B._Halsted
cluster-master  |    Score: 7.5989
cluster-master  |     Text: "George Bruce Halsted (November 25, 1853 – March 1...
cluster-master  |
cluster-master  | 3. Document ID: 52572055_G._Anthony_Gorry
cluster-master  |    Title: Document 52572055_G._Anthony_Gorry
cluster-master  |    Score: 6.7642
cluster-master  |     Text: George Anthony Gorry is an American computer scien...
cluster-master  |
cluster-master  | 4. Document ID: 8036984_G._David_Tilman
cluster-master  |    Title: Document 8036984_G._David_Tilman
cluster-master  |    Score: 5.1695
cluster-master  |     Text: "George David Tilman (born Titman; July 22, 1949),...
cluster-master  |
cluster-master  | 5. Document ID: 292864_G2_(mathematics)
cluster-master  |    Title: Document 292864_G2_(mathematics)
cluster-master  |    Score: 3.8381
cluster-master  |     Text: In mathematics, G2 is the name of three simple Lie...
cluster-master  |
```

- **Query 3**: "A Christmas Carol. In Prose. Being a Ghost Story of Christmas, commonly known as A Christmas Carol, is a novella by Charles Dickens, first published in London by Chapman & Hall in 1843 and illustrated by John Leech. A Christmas Carol recounts the story of Ebenezer Scrooge, an elderly miser who is visited by the ghost of his former business partner Jacob Marley and the spirits of Christmas Past, Present and Yet to Come. After their visits, Scrooge is transformed into a kinder, gentler man. Dickens wrote A Christmas Carol during a period when the British were exploring and re-evaluating past Christmas traditions, including carols, and newer customs such as cards and Christmas trees."



```
fi

log "Indexing completed successfully!"
log "Index data is now stored in Cassandra."

# Final cleanup
hdfs dfs -rm -r -f /tmp/index/output1
hdfs dfs -rm -r -f /tmp/index/output2

```

```
 including carols, and newer customs such as cards and Christmas trees.'
cluster-master    | -----------------------------------------------
cluster-master    | 1. Document ID: 73670_A_Christmas_Carol
cluster-master    |    Title: Document 73670_A_Christmas_Carol
cluster-master    |    Score: 323.6803
cluster-master    |     Text: "A Christmas Carol. In Prose. Being a Ghost Story ...
cluster-master    |
cluster-master    | 2. Document ID: 1603537_G4_(group)
cluster-master    |    Title: Document 1603537_G4_(group)
cluster-master    |    Score: 71.6689
cluster-master    |     Text: "G4 are a four-piece British vocal troupe who firs...
cluster-master    |
cluster-master    | 3. Document ID: 8110264_G._Fox_&_Co.
cluster-master    |    Title: Document 8110264_G._Fox_&_Co.
cluster-master    |    Score: 62.0461
cluster-master    |     Text: "G. Fox & Co. was a large department store that or...
cluster-master    |
cluster-master    | 4. Document ID: 35937347_G._G._Rupert
cluster-master    |    Title: Document 35937347_G._G._Rupert
cluster-master    |    Score: 55.9552
cluster-master    |     Text: "Greenberry George Rupert (1847-1922), generally k...
cluster-master    |
cluster-master    | 5. Document ID: 46209038_G._B._Harrison
cluster-master    |    Title: Document 46209038_G._B._Harrison
cluster-master    |    Score: 54.7118
cluster-master    |     Text: "G. B. Harrison (14 July 1894 – 1 November 1991) w...
cluster-master    |
cluster-master    | 6. Document ID: 44505796_G3_Misa
```

Caption: this query is used with additional documents that were indexed, the folder is app/new_data_to_index
this path was used as the argument to index.sh script. This query is a demonstration that my code can index additional documents