Nekrasov Dmitrii, DS-02 BigData Assignment 2

Methodology:

explanation of design choices and approaches in implementing the search engine

Total execution time of 'docker compose up' is 40min on ryzen 5 5600.
Search for 1 query takes 30+-15 seconds, booting spark takes a large portion of this time.
In my map reduce scheme, i use only 1 mapper and 1 reducer.

In mapper, i map each word to document_id it was in, then sort words giving:

```
3214 word1 100
3214 word1 100
2345 word1 200
2342 word2 100
9079 word2 300
3685 word3 400
```

In reducer, i have O(1) space complexity, writing to .output1 lines for each table entry:

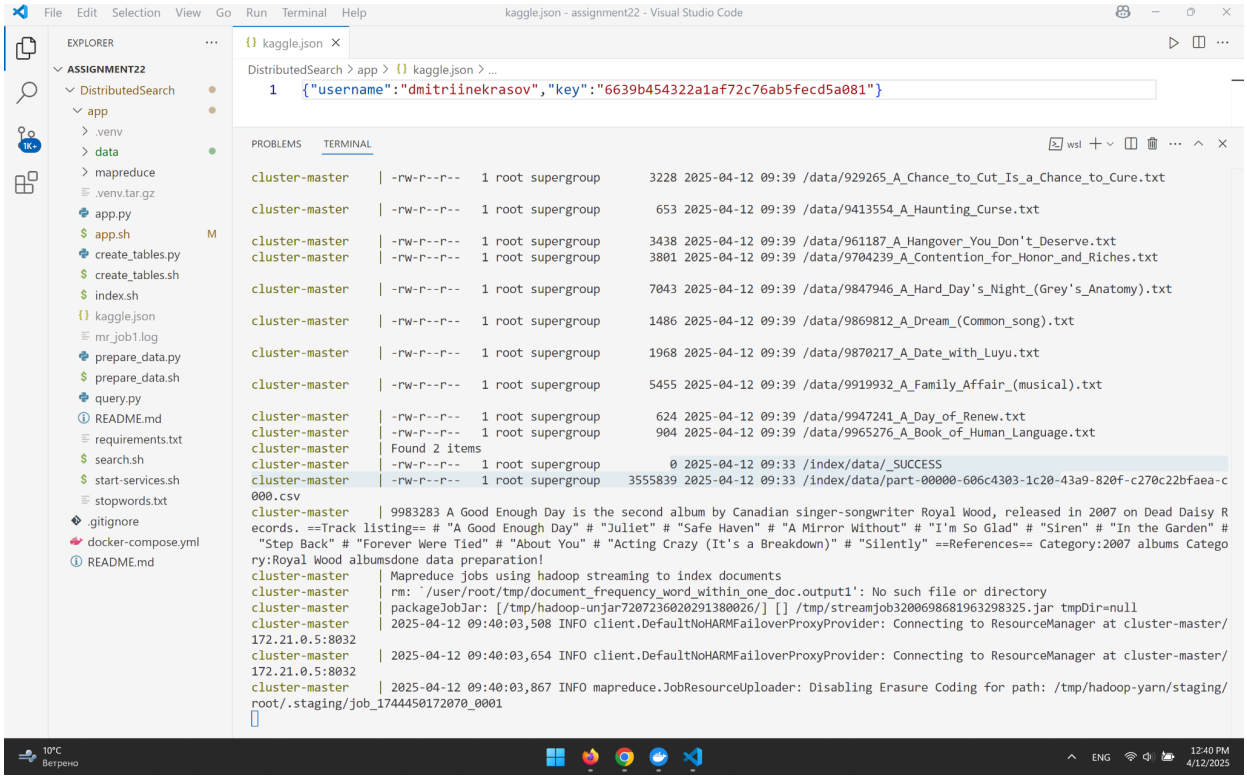
```
print (f'word_within_one_document_count {search_word}
{search_doc} {search_word_count_within_doc}')
print (f'document_frequency {search_word}
{search_word_count_across_docs}' )
print (f'doc_id_len {doc_id} {doc_len}')
```

This is achieved with dynamic updates of search variables (search_word, search_doc).
Output files weights 17110539bytes=16.3 Megabytes for 1000 documents.

In query step i use spark cassandra driver with only tables reading and bm25 fraction calculations.

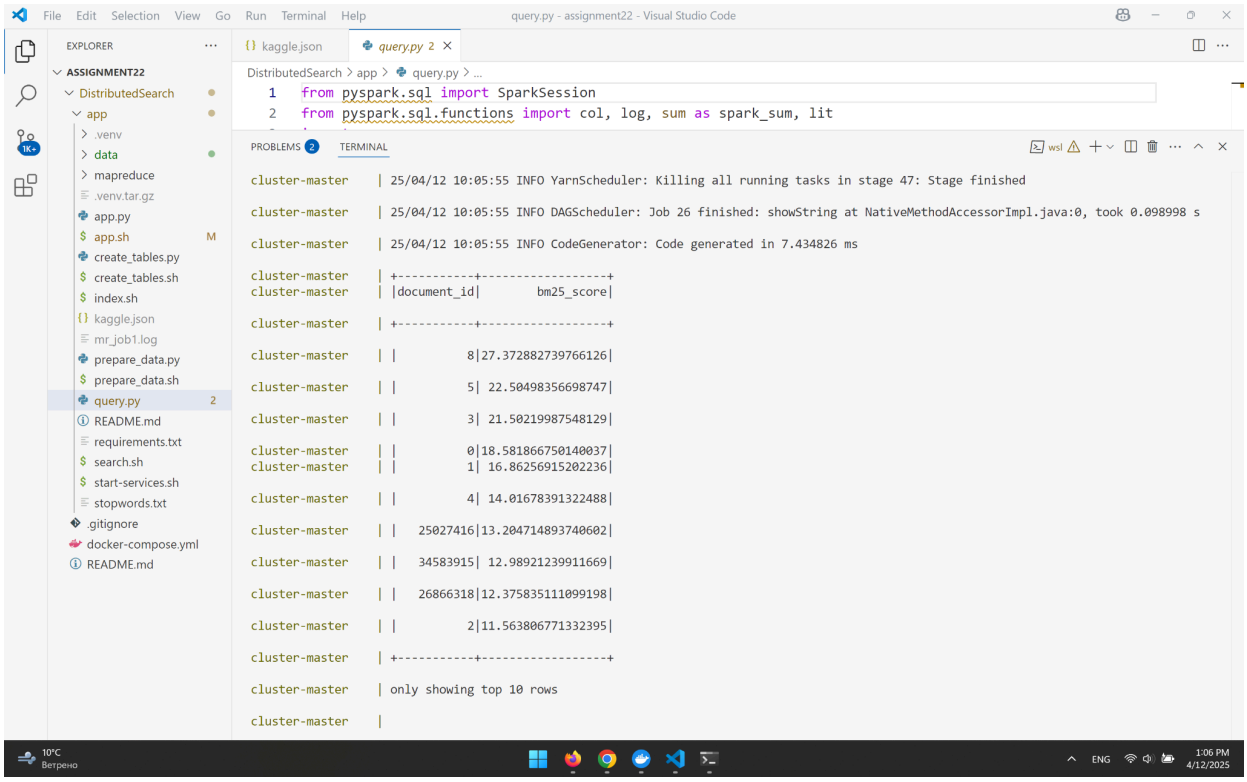
Demonstration

successful indexing of 1000 documents:

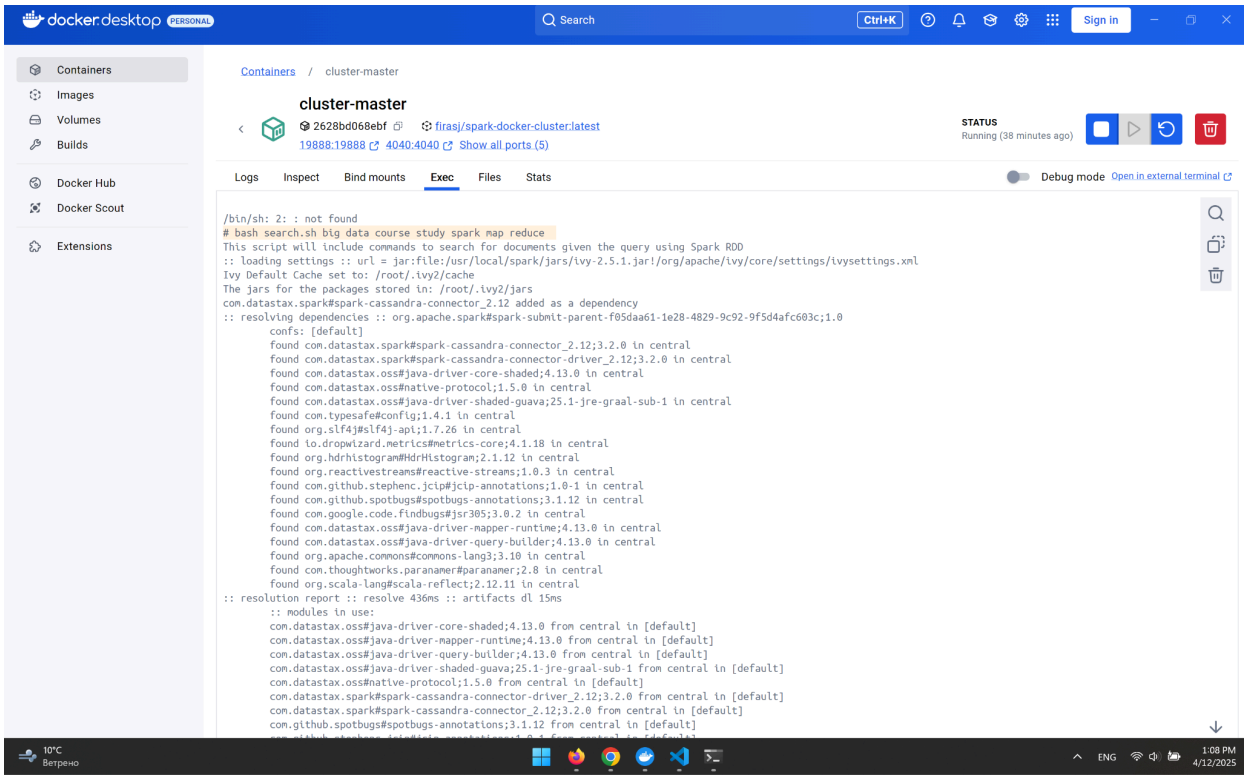


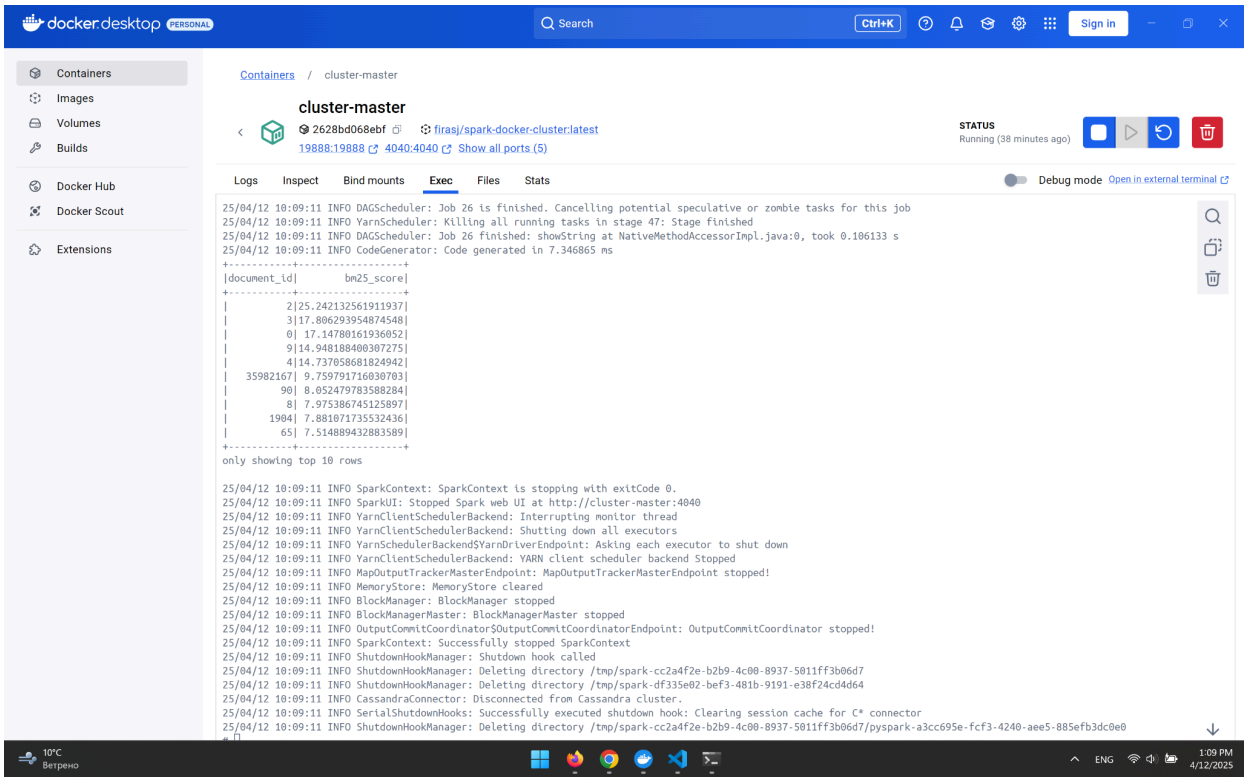
successful execution of the search engine on some queries:

here query is `The Dave Clark Five, Catch Us If You Can` and bm25@10 didnt found actual document



here query is big data course study spark map reduce, documents indeed have those words in content





Bm25 shows good logic in terms of scores have good distribution