Assignment 3 (10%)
Date Given: Mar 4, 2020
Submission Due: Mar 17, 2020 at 11:59 pm (midnight)
**** Late submissions are not accepted and will result in a 0 on the assignment**

Objective:

This assignment focuses on building a light-weight application using Docker containers.

Grading Scheme:

- Job1: Report: 10%
- Job2: Extraction Engine: 10%
- Job3: Web Page (Search Page/ Note Submission Page): 10%
- Job4: Docker Containers: 45%
- Job5: Deployment: 5%
- Job6: Testing: 15%
- Adding citation in IEEE/ACM Format only. Use reliable information source: 5%

Academic Integrity:

- This assignment does not require group work. Therefore, each student is expected to complete their work by themselves. Collaboration of any type amounts to a violation of the academic integrity policy and will be reported to the AIO.
- Do not copy texts verbatim from online or printed materials
- Do not copy texts from other's work
- Do not submit other's work
- If you obtain help from Tutor(s), please acknowledge
- Provide citation for texts, images, tables, data etc.
- The Dalhousie Academic Integrity policy applies to all material submitted as part of this course. Please understand the policy, which is available at: https://www.dal.ca/dept/university_secretariat/academic-integrity.html

Job Description:

Using the given architecture develop a cloud application. Follow the given steps:

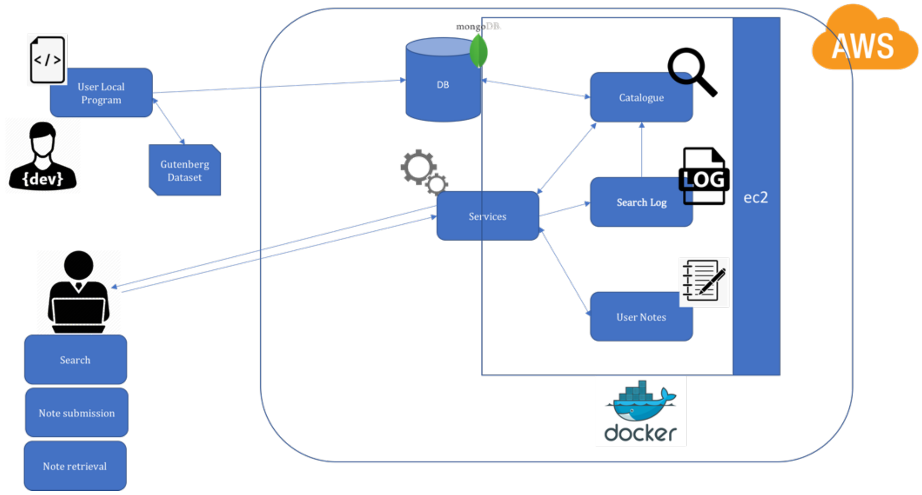

Fig: Book Title, Author Search Application (CSCI 5409 Model Project)

Job1:

Report: Write a 1 page report on your understanding of the given architecture. You should mention your configuration, design, and implementation details in the report. **Expectations**: Documenting containers, API calls, and design require background theoretical foundation. Therefore, read official documentation provided by Docker, AWS – provide citation.

Job2:

Extraction Engine: You need to create a simple program using Python/ Java/ C++/ Perl to extract title, and authors of English books given in the link
http://www.gutenberg.org/wiki/Gutenberg:Offline_Catalogs
You need to extract information from all files - 1996(earlier) to 2020. Your program should process each file and store in the AWS MongoDb database. After processing each file, the program should introduce a delay of 5 minutes. In addition, you should capture and store in MongoDb the file process start time, and end time.

Job3:

Web Page: You need to create a simple web page using Php/JSP/ Angular/ React with one keyword search option, one note submission option (which appears after keyword search), and note retrieval option. It is a single user application, so no login or session management required.

Job4:

Docker Containers: You need to create containerized services as given in the architecture. Follow the arrows and directions to understand the interactions, and API calls.

- If a user type a keyword and hit "search" – The time of search request and the keyword are stored in Search Log. The search log also count frequency of keywords searched.
- If the search is successful the data (author, title) is placed in catalogue and the result is returned to the browser client
- For a successful search, the user can enter a note and submit. This notes are handled as JSON structure in User Notes. Multiple notes for the same keyword can be entered
- Notes related to a keyword can also be retrieved.

Job5:

Deployment: Successful deployment of the system is very important. You must follow the given architecture. This application does not require any load balancing, auto scaling, or huge resources.

Job6:

Testing: Write small test cases for different components of your application, and show evidence of testing. You can check this URL for some examples: https://www.softwaretestinghelp.com/how-to-write-effective-test-cases-test-cases-procedures-and-definitions/

Submission Instruction:

- Create a Folder with your name and B00 number, and store all your files –
  - Program Source code with readme and/or libraries
  - PDF file containing all answers
  - Screenshots and image files (if any)
- Compress the folder and create a .ZIP file (do not use other compression formats)
- Upload the .ZIP file on Brightspace.
- Submission Due: **Mar 17, 2020 at 11:59 pm (midnight)**