

Assignment 2

CSCI 5410 (Serverless Data Processing)

Date Given: Oct 7, 2022

Due Date: Oct 19, 2022 at 11:59 pm

Late Submissions are not accepted.

A deduction of 10% per day will be applied for late submission.

To avoid any additional charges for resource consumption - Delete the GCP resources after fulfilling the assignment submission requirements.

Objective:

This assignment covers concepts of containerization and Serverless components of cloud computing. The primary objective of this assignment is to introduce you to the cloud computing containerization application using Docker.

Plagiarism Policy:

- This assignment is an individual task. Collaboration of any type amounts to a violation of the academic integrity policy and will be reported to the AIO.
- Content cannot be copied verbatim from any source(s). Please understand the concept and write in your own words. In addition, cite the actual source. Failing to do so will be considered as plagiarism and/or cheating.
- 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

Assignment Rubric - based on the discussion board rubric (McKinney, 2018)

	Excellent (25%)	Proficient (15%)	Marginal (5%)	Unacceptable (0%)	Problem # where applied
Completeness including Citation	All required tasks are completed	Submission highlights tasks completion. However, missed some tasks in between, which created a disconnection	Some tasks are completed, which are disjoint in nature.	Incorrect and irrelevant	Part A Part B
Correctness	All parts of the given tasks are correct	Most of the given tasks are correct. However, some portions need minor modifications.	Most of the given tasks are incorrect. The submission requires major modifications.	Incorrect and unacceptable	Part A Part B
Novelty	The submission contains novel contribution in key segments, which is a clear indication	The submission lacks novel contributions. There are some evidence of novelty, however, it is not significant	The submission does not contain novel contributions. However, there is	There is no novelty	Part A Part B

	of application knowledge.		an evidence of some effort.		
Clarity	The written or graphical materials, and developed applications provide a clear picture of the concept and highlights the clarity.	The written or graphical materials, and developed applications do not show clear picture of the concept. There is room for improvement	The written or graphical materials, and developed applications fail to prove the clarity. Background knowledge is needed.	Failed to prove the clarity. Need proper background knowledge to perform the tasks.	Part A Part B

Citation:

McKinney, B. (2018). The impact of program-wide discussion board grading rubrics on students' and faculty satisfaction. Online Learning, 22(2), 289-299.

Tasks:

This assignment has two parts. Part A has a reading task, and part B has a small programming tasks:

Part A. Read the following paper and write a summary (visit IEEE from libraries.dal.ca)

S. Garg and S. Garg, "Automated Cloud Infrastructure, Continuous Integration and Continuous Delivery using Docker with Robust Container Security," 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), 2019, pp. 467-470, doi: 10.1109/MIPR.2019.00094.

URL: <https://ieeexplore-ieee-org.ezproxy.library.dal.ca/document/8695332>

- It can be approximately 1 page summary and must be written in your own words. **Your in-depth review of the paper is expected.**
- There should be **inline citations** in an acceptable format.
- The summary should include - (a) what the authors have presented in the paper, (b) if any specific issue is addressed, (c) if any experiments or studies performed, (d) analysis or findings made by the authors.

Part A - Submission requirement: A pdf file with the summary is needed.

Please write in the form of summary only (i.e., interconnected sentences with continuous flow forming meaningful paragraph). Do not write it in question-answer format.

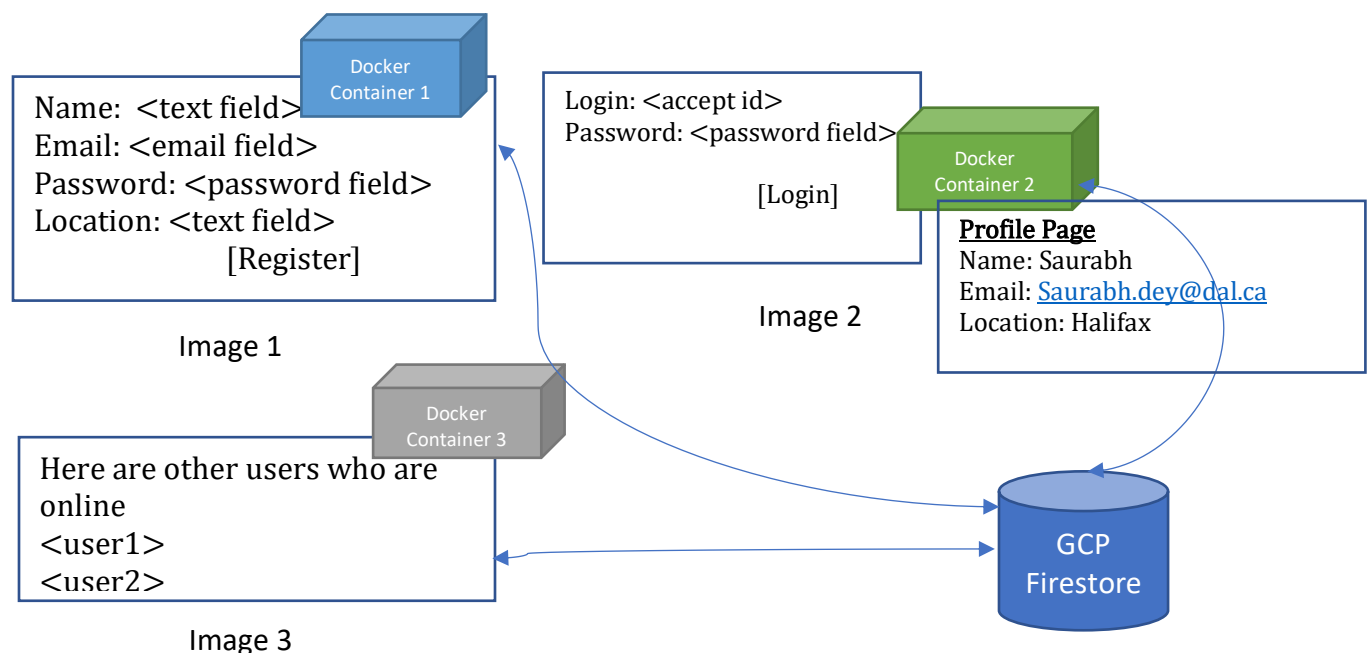
Part B. Build, deploy, and run a Containerized Application using GCP.

Using GCP create and validate an online meeting account.

take screenshots at every step and submit as part of the PDF:

- Create three microservices and package these in three different containers, which will be responsible for the front-end and backend logic in this application. The database you will be using here is, **Firestore**
- Code and required dependencies in **Container #1** are responsible for accepting registration details from frontend and store it in backend database. (image 1)
- Code and required dependencies in **Container #2** are responsible for validating the Login information during a login process, and launching a profile page with

- the registration information of the user, which will come from Firestore (image 2)
- Once a user is logged in – the state of the user changes to online in the FireStore database
 - Your database should contain only 2 documents. One document to contain **registration data** (Name, Password, Email, Location), another document to contain user **state** (online, offline, timestamp etc.) information with the name and email of the users.
 - Once the user signs out from the profile page, it should load code and dependencies of Container #3.
 - Code and required dependencies in **Container #3** are responsible for extracting state information from database. E.g. who was online at what time. You need to maintain the session from login to logout. The session must expire after clicking the logout, which should update the **state** item in the Firestore database.
 - Once the docker images are built, you need to push those container images to artifact registry repository (GCR is deprecated now). Once it is done, you need to deploy those in **Cloud Run**.
 - In order complete the tasks, and perform interaction, you need build 3 simple web pages (or 1), using any technology of your choice.
 - There is no direct communication between these containers, all data update, and select is done through Firestore only. You can build 3 web-pages for the 3 services mentioned.
 - Write test case to test your application and perform testing. Provide screenshots as evidence for all steps.
 - You need to study Google Cloud Run, GCR/ Artifact registry, Docker Container documentations, and write a summary within 1 page explaining how you have used these technologies in your application. ***If needed, you can add more containers to design your application



Part A - Submission requirement:

For (a to j), submit screenshots of every steps. Please do not exclude any steps.

Submit screenshots of empty and populated Firestore database

For (a to j), submit your program files (Source code on Gitlab) as well.

However, from the source code - add the important methods (pseudocode), or program instructions as part of the pdf. E.g. Login Validation method etc.

For (k), add the test cases and screenshots of tests in the pdf

For (l), add the summary as part of the pdf file