THE UNIVERSITY OF MELBOURNE

QUANCHI CHEN <quanchic@student.unimelb.edu.au>

Ass2 Feedback

1 message

Richard Sinnott <rsinnott@unimelb.edu.au> Mon, Jun 12, 2023 at 6:36 PM
To: "menzhang1@student.unimelb.edu.au" <menzhang1@student.unimelb.edu.au>, "zexue@student.unimelb.edu.au" <zexue@student.unimelb.edu.au>, "zmu1@student.unimelb.edu.au" <zmu1@student.unimelb.edu.au>, "quanchic@student.unimelb.edu.au" <quanchic@student.unimelb.edu.au>, "zixiq@student.unimelb.edu.au" <zixiq@student.unimelb.edu.au>

Hello Folks,

The following is the feedback on your Assignment 2. You will get your individual scores in due course through Canvas. The feedback is based on:

Report+Architecture+WebApplication+MRC Feedback || Software Feedback || Ansible and Dynamic Scaling Feedback

The architecture was well described. The pros and cons of the MRC were very well done. There was no real need to do the pros and cons of couchDB. It was good that used Docker in SWARM mode for making your system more flexible and scalable. Nice that you encrypted the playbook passwords in Ansible vault. Not all teams did this. Your mental health scenarios are very dependent on the set of keywords that you selected. It might have been useful to use a dedicated data dictionary of health terms or apply topic modelling for tweets that mention depression, suicide etc and then any other terms that are associated with them and use these other terms as the basis for your twitter analytics. The front end was fine but it could have been a tad more interactive, e.g. showing real time Mastodon posts, or even colour coding your map to show areas with a higher/lower estimated percentage of people with mental health problems in different LGAs. You might have had scenarios that combined social media and SUDO data on mental health. Right now you support different charts, but not a combined chart, e.g. show the #tweeters in area X and whether they are happier/sadder if they have more mental health support services etc. That said, the report was well structured and easy to follow.	General: Good and tidy effort. Repository: the code is generally well-organized and the READMEs give enough information, but dependencies (requirements.txt) should include package versions to avoid inconsistencies between builds. Harvester: some information (key_words and key_tags belong to external configuration files (possibly JSON), to make the applicaiton more flexible. Processing: good use of argparse and of env. var., but some dictionaries should have been kept in external configuration files. Database: there is no MapReduce code to assess in the repository. Back-end API: Nice use of Golang. Correct layout of a ReSTful API URL space. Front-end: OK	Ansible: Instead of utilizing the Ansible copy module to transfer your source code to the target host and construct the image there, it would be more efficient to employ the git module to extract the source code from Github. This approach enhances the flexibility of your playbook, enabling you to checkout from a specific branch, for instance. You've employed the shell module in your playbook where other modules like uri could have been used. Utilizing the correct module for a particular task offers better error management. In your playbook, the template module has been used for writing static files; a more suitable choice would be the copy module. You used the code provided in the workshop for your assignment. It is essential to appropriately reference such sources in your work to acknowledge the original contributions. Docker: You've used the j2 template to incorporate DB_HOST, DB_USER, and DB_PASS

		into the Dockerfile. As these are environment variables, they should ideally be passed at runtime rather than directly input into the Dockerfile. The same was observed in your harvester deployment. Your Dockerfiles could benefit from simplification, particularly the UI one. Consider referring to Docker best practices for guidance in future tasks. Other: There are portions of redundant code within your playbook. It's best practice to eradicate unused or unwanted code prior to submission. Do not include sensitive information like passwords and private keys in plaintext within your submission. If these credentials must be included, ensure their safety by encrypting them using tools like Ansible Vault.

Kind regards,

Rich

--

Prof. Richard O. Sinnott

Professor of Applied Computing Systems

Faculty of Engineering and Information Technology, The University of Melbourne

Level 5, Melbourne Connect, 700 Swanston Street, Melbourne, Vic 3010, Australia

Tel: +61-(0)435-964-844 Email: rsinnott@unimelb.edu.au

Web: www.eresearch.unimelb.edu.au