COSC349 Assignment 2: your software in the cloud

Submission information

Due date: Friday, 2nd October 2020, at 11:59 PM.

Weight: This assignment is worth 20% of the mark for the paper.

Participation: You are permitted to work individually or in pairs.

How you will make submissions: A 'test' on the COSC349 Blackboard section will collect the following three pieces of information:

1. the URL of your live application, running in the cloud;
2. the URL to a git repository—or even better, to a specific git commit, git branch or git tag—that contains the version of your project's files that you want to be marked; and
3. a project report in PDF format.

All students must make their own personal submission within Blackboard, however members of pairs will both submit exactly the same material.

Requirements:

- The software you write for your assignment should be deployed to the AWS cloud using your Amazon Educate Classroom credentials—however, other arrangements can be made where use of these AWS facilities is impractical.
- You should expect to leave your cloud application running and internet accessible from the time of the deadline until you receive notification that your assignment has been marked.
- You are expected to work on your assignment within a git repository. The easiest way to acquire URLs that reach your repository is to push your work to a (free) cloud-based git service, such as GitHub, Bitbucket, GitLab, or Altitude (a GitLab server run locally within the Department of Computer Science that you can access). The aforementioned cloud services allow you to create private repositories.
- You must invite teaching staff to collaborate on your repository, so that they can access the URL that you submit. Inviting David is sufficient: he is user `dme26` on GitHub, Bitbucket, and GitLab. Otherwise you can send an email invitation to `dme@cs.otago.ac.nz`.
- Do not apply `git` history-rewriting operations to your repository: it is important that the markers can see how your commits showed you completing your assignment over time.
- You are welcome to either extend the same repository as you used for assignment 1 or create a new repository: do whatever suits you best, but be sure to submit the URLs and report relevant for assignment 2 through Blackboard.

- Your repository must include all of the files required to build your system, other than stable, publicly-available, external resources downloaded during your build process. For example you do *not* need to include Vagrant box files, or data that can be downloaded from cloud-based repository servers (e.g., GitHub, etc.).
- You must also prepare a report in PDF format. This report should indicate who you are. The marking guide below indicates a number of points your report must cover.

Late submissions: The commit date of your submitted commit ID will be taken as your submission timestamp. COSC349 applies a late submissions policy typical of other COSC papers. Late submissions will incur a 10% penalty per working day, rolling over at 11:59 PM. Submissions that are more than five days overdue will not be accepted.

Learning objectives

The aims of the assignment include the following learning objectives:

- to demonstrate that you can build and deploy software onto cloud hosting environments;
- to employ services made available by cloud providers within your software;
- to carry out research to discover open source projects and materials that you can use within your application development work; and
- to deliver software that is well tested and documented.

Problem description

For this assignment you will deploy an application to the cloud. The application will be of your design, and will use multiple interacting virtual machines. You are welcome to build on the application you designed and deployed for assignment 1. As for assignment 1, assignment 2 is not focused on the application itself, the assignment is focused on how you deployed your application and what cloud services you have used.

You do not need to fully automate your deployment (e.g., using Vagrant), but if you take manual steps to deploy your application into the cloud, these should be documented in your report.

An easy way for a cloud application to provide an interface to (non-developer) users is through the use of web technologies. Because this is not a web technology paper, marking will not focus on the details of your web implementation. Any other form of interface is acceptable, provided that it does not require your potential users to have advanced computing qualifications to use it.

You are welcome to use others' code within software components and as a starting point for your assignment, but you still need to build an application of your own design and deploy it into the cloud. Any use of others' resources must be attributed by you in your in-code documentation and your report. Further, the history of commits in your `git` repository will be examined to determine what, and how much code and configuration you added into your project.

You should make use of at least one cloud-provided service in addition to the cloud provider facilitating your use of virtual machines. On AWS, for example, this could include using S3 for storage, or one of the AWS-based database services. Describe your choice in your report.

Example applications

If you cannot think of an application to build and deploy, some suggestions are listed below. You may want to consider developing a project that you can present to future potential employers. Also, given that the assignment does not focus on the application itself, you are welcome to rework and extend material that you have completed for other assignments within Computer Science or Information Science papers.

- A world timezone converter: one VM could run a web interface, another VM could run a database storing users' specific display preferences, and a third VM could provide tools for batch querying of timezone conversions.
- A customised editor of some sort: e.g., shared note-taking tools, or an assistant for recording minutes of meetings. Different VMs could run a web interface for the editor, a database server, and a tool for producing reports, or calendar files of due dates for items.
- A site for visualising data: one VM could provide the interface users use to upload their data and retrieve their results. Another VM could store the users' data in a database or other storage system. A third VM could provide administrative functionality.

Rough marking guide

The assignment is worth 20% of your COSC349 mark but will be marked out of 100. A breakdown of marks is included below. This is not a strict marking scheme, as having one tends to disadvantage students. For example, if you are unable to achieve one of the points in the marking breakdown below, you may be able to gain compensatory marks by explaining in your report what you intended to do, what problem arose, and what compromise you reached. Also, it may be considered that you have gone well beyond the call of duty, and should be awarded additional marks for a section, thus helping compensate for some minor problems elsewhere.

40% Your application is running in the public cloud.

- Your report should describe how you deployed your application.
- Your report should describe how to reach your application in the cloud, and what a user can do easily to interact with it.
- Your report should include brief evidence—such as screen-captures with explanatory text—that show your application being used in the cloud.

30% A cloud service is used in your application (beyond using the cloud to create your VMs). Your report describes your choice of service and how it is used. (Example services include Amazon S3 or a cloud database API.)

20% How you incrementally develop and debug your project is clear, from examining the commit history in your `git` repository, and your report.

10% Your application should be built using at least three VMs that interact, with one exception. If your application for assignment one used a database server VM, and you shift your database(s) to use a cloud database service instead of a VM, your application only requires two interacting VMs, alongside use of the cloud database service.

- Your report explains the design of your application and how the VMs and other cloud APIs are used.
- Your application needs to be of your design, even if you use others' open-source components to help build it.