

COSC349 Assignment 2: your software in the cloud

Submission information

Due date: Monday, 2nd October 2023, 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 Blackboard ‘assignment’ and a Blackboard ‘test’ on the COSC349 Blackboard section will together collect the following four pieces of information:

1. a URL such that the command ‘`git clone URL`’ would download your repository (assuming that the repository has been shared with the requesting user);
2. a Git commit ID within your repository that you want markers to assess;
3. a URL to a screen recording showing you using your software when it is running in the cloud;
4. a project report in PDF format.

All students must make their own personal submission within Blackboard. Members of pairs must both submit exactly the same material.

Requirements:

- Typically your software would be deployed to the AWS cloud using your Amazon Academy Learner Lab credentials. Other arrangements can be made where use of these AWS facilities is impractical.
- You should expect to leave your cloud application functional and internet accessible from the time of the assignment deadline until you receive notice 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 hosted locally that you as School of Computing students 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 in the cloud. He is user `dme` on Altitude. 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 completed your assignment, and how you worked on the assignment over time.
- You are welcome to either extend the same repository as you used for assignment 1 or create a new repository, but be sure to submit the URLs and report relevant for assignment 2 through Blackboard.

- Your repository must include all of the data and code required to deploy your application, although you do *not* need to include content that can be publicly downloaded from cloud-based repository servers (e.g., GitHub, etc.), instead just describing where that content comes from (e.g., via URL).
- Your screen recording must be no longer than two minutes. Do not add additional graphics to your video beyond what is seen on screen. You may narrate your screen recording, or leave the audio track blank, and instead briefly describe in your report the important time points in your recording. You may use any tool to create your screen recording, but note that Zoom facilitates this (join your own meeting, screen share into that meeting, and then record the meeting onto your computer). You can share your recording with David using your University of Otago OneDrive storage (or YouTube, etc.).
- Your repository should be set up so as to allow a new developer to join your project, and to rapidly get up to speed regarding how to use and extend your work. For example an informative ‘README’ file of some sort is expected. Assume that such a developer will not receive your submitted report. Ideally your repository will not contain too many indicators that it is aiming to satisfy a university assignment.
- Your project report is where you can communicate with your markers, explaining aspects of your project that might not be reflected clearly within your repository, such as unsuccessful attempts that you might have made to use particular technologies. The marking guide, below, indicates a number of aspects that your report must cover.

Late submissions: The timestamp of the Git commit that you select for marking will be taken as when you submitted your assignment. 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, and **provide instructions** on how to use it. The application will be of your design,

and will use multiple interacting virtual machines—**at least two virtual machines must be used**. These VMs can be EC2 instances, but they can also be containers, if you would prefer.

You must use a non-EC2 cloud-based service to provide storage to your application. For example, this could be a relational database hosted by the Amazon Relational Database Service (RDS), or you could employ Amazon S3 object storage. There are many more storage services that you can research and use. **You must** choose and integrate one further non-EC2 cloud service, different from the one you have used for storage. **Describe and justify** your choices of non-EC2 cloud-provided services within your project report.

Also, **you must** include in your project report a rough estimation of the running costs of your application—see the marking guide below.

You are welcome to build on the application that 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 tools such as Terraform, or suitable plug-ins for Vagrant), but where you take manual steps to deploy your application into the cloud, these steps should be explained within the documentation included **within your repository**, for the sake of a hypothetical new developer joining your team. (Material intended only for the marking team can be included **within your project report**.)

It is typical for cloud applications to provide an interface to (non-developer) users through the use of web technologies. Because COSC349 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 comments and within documentation included within your repository. 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.

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 rental share-house budget management helper: one VM could provide the interface renters use to enter bills, payments, and view the results of queries. Use a cloud service to store the users' data in a database or other storage system. A second VM could provide functionality for the owner/manager of the property.

- An online service for booking use of shared resources of some type. One VM could run a web interface for users, a cloud service provides the database that records allocation of the shared resource, and a second VM could provide administrative functionality.
- 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, and a tool for producing reports, or calendar files of due dates for items.

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 aspects within 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.

30% Your application is running in the public cloud.

- Documentation within your repository and your project application, taken together, 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 screen recording—no longer than two minutes—makes it clear how a user should use your application. You may narrate your screen recording using audio, or instead include a textual description of your screen recording’s timeline within your report.

40% Your application should be built using at least two VMs that interact.

- Documentation within your repository and your project report explains the design of your application and how the VMs interact.
- A non-EC2 cloud service is used for providing storage to your application. Your report describes and justifies your choice of service and how it is used.
- A further non-EC2 cloud service, different from that used to provide storage, is used by your application. Your report describes and justifies your choice of service and how it is used.
- Your application needs to be of your design, even if you use others’ open-source components to help build it.

20% Your project report includes a rough estimation of the ongoing running costs of this application in the cloud: (a) when it is entirely idle, and (b) when it is being lightly used.

10% How you incrementally develop and debug your project is clear, from examining the commit history in your Git repository, and your report.