



## COSC2759-2110 Assignment 2 specifications

Deadline	Sunday 23.05.2021 (11:59 pm AEST)
% allocated to this assignment	30
To be submitted via	Canvas
To be attempted	Individually

---

### Please read this first

All of us have been affected by the unfortunate COVID-19 scenario and its aftermath. It is often hard to concentrate and study online; but as a student enrolled in this course, it is your responsibility to regularly attend online lecture, your respective pracs and consultation session(s).

- Bring your questions to online discussion board, consultation sessions or email *Homy* ([amirhomayoon.ashrafzadeh@rmit.edu.au](mailto:amirhomayoon.ashrafzadeh@rmit.edu.au)).
- Watch the online recordings on a regular basis if you cannot attend the live sessions.
- Do NOT start the assignment at the last minute.
- Do NOT ask for last minute extensions, these are often rejected. Extensions can only be granted for personal and medical reasons, provided you can supply some evidence.

Besides, DevOps is a very important skill to have. There is a clear lack of training in this area for the graduates and as such doing well in this course will give you a competitive edge over graduates from other universities.

Finally, please do NOT give up on this semester thinking of online study as a waste of time. Global industry projects involving teams at geographically different locations often run fully online. Think of this semester as a training for future. In the fast-evolving CS&IT industry, the online medium is very popular and vastly used.

You have more than 4 weeks to complete this assignment. It is related to AWS. You will need your AWS Educate accounts.

**If there are any issues with your AWS Educate account, please email the head tutor at your earliest. Sending emails late or closer to deadline re account issues will only lead to adverse outcomes.**

Get started!





## Scenario

Stonks Inc. has been very excited about the progress you've made so far and have started looking at the next steps for them.

So far, their application deployments have all been manual using ClickOps, which is prone to human error. They would like you to have a look at how to automate the infrastructure and application deployment. They don't think they are ready for Kubernetes and containers yet, so they would like you to use EC2 instances to host the application.

GitHub Classroom Invite: <https://classroom.github.com/a/VE-QuD7->

## The Approach

To make this easier for Stonks Inc., which is rather immature with DevOps methodologies and practices, you have opted to use SaaS tools where possible to help reduce the learning curve for their development team. You will follow best practice principles and make as much of your solution using code, this includes infrastructure configuration and any scaffolding scripts.

### Tools to use:

- GitHub (We will be using GitHub Classroom)
- Terraform
- Ansible
- AWS
- CircleCI (for HD)

Stonks Inc. expects you to update your Github classroom repository all the code and documentation required to run what you are building for them. This includes the files that define any shell scripts, Docker files, or anything else you use.



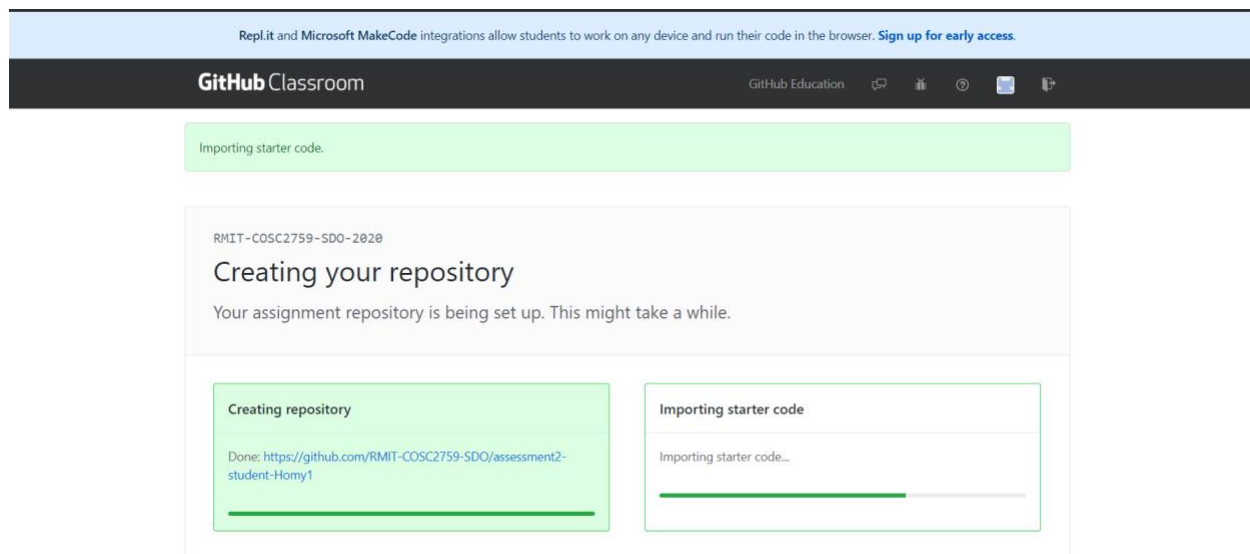


## Deliverable

We will be using Github classroom to edit and manage the code base. All of the core files for the application will be provided to you here. Please accept the invite below to join the classroom and get access to Assignment #1. You will be expected to work from this repo and make all of your commits to this repo.

## Create Github Classroom for the assignment

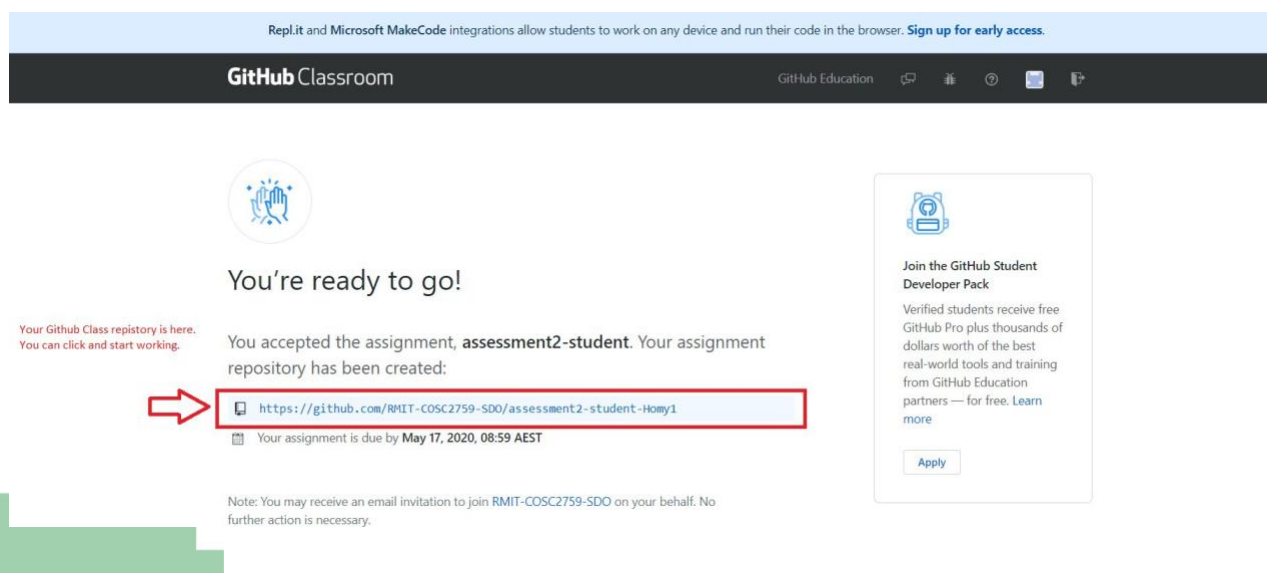
Please accept [this invitation](#), (you must login with your Github account).



You will see above picture after you accept the invitation, wait until the process is done.

Then you must be able to see the below page. You can see your repository for you on SDO Github classroom. Please click on that and then (The URL in here is as an example and your Github URL must be as below:

“<https://github.com/RMIT-SDO/assignment-1-student-<your Github Username>>”



you must be able to see your Github repository as below (pictures are as examples your URL can be different):

The screenshot shows a GitHub repository page for 'assessment2-student-Homy1' created by GitHub Classroom. The repository is private and has 1 commit, 1 branch, 0 packages, and 0 releases. The repository address is 'RMIT-COSC2759-SDO / assessment2-student-Homy1'. The repository contains files: 'ansible', 'infra', '.gitignore', and 'README.md', all with initial commits 9 hours ago. The README.md file is visible, showing the title 'Servian TechTestApp' and the word 'dependencies'.

Now, you can clone and start your work on the assignment.

Please do not forget to add, commit and push all the steps as you do.

Please update the README.md file as you are adding steps.

**Please be advised that you are only allowed to use this repository for the assignment. Your scaffold with a folder structure and some support files is available in your Github repository.**

Please use your AWS Educate account to test the solution as you build it.



## Tasks

For all queries pertaining to this assignment, be it specification-related or extension, ONLY send email to Homy and no one else.

## Pass

### A. Document the solution in the Readme.md file

Please document the solution including all the steps of your work so it can be deployed by someone with no prior knowledge of the approach. This means a step by step guide to deploying the application including screenshots.

As a piece of guidance, we would recommend you document as you go and as you build out specific features.

Please consider:

- Analysis of the problem
- Explain and justify the solution
- Writing quality, layout, and accuracy

### B. Run the environment bootstrap templates & also generate an artifact that you can use in the next few steps (refer to the Readme &/or Makefile for both).

### C. (10 marks) Create a VPC using Terraform ready to host the application.

Create a VPC in terraform with 3 layers across 3 availability zones (9 subnets). Public, Private, and Data.


### D. Create 3 layer application infrastructure using Terraform There should be:

- a Load balancer deployed in the Public layer, with a listener and target group (Note we are not using an auto scaling configuration in this assignment).
- an EC2 instance deployed in the Private layer (use the latest Amazon Linux image & deploy a t2.Micro instance size)
- a Database deployed in the Data Layer (DocumentDB).

There should be security groups defined to limit access to what is required.

As the application we are using requires MongoDB, we will be deploying a Database service called DocumentDB. Like RDS, DocumentDB is Amazon's fully managed DocumentDB Service that provides seamless MongoDB integration and enables us to use the same libraries, drivers and tools to connect with it meaning no changes have to be made to the application itself! (Winning!). To ensure you can talk to the DB with our application we need to ensure the DB is set up with no TLS. To do this you need to provide the ID of the parameter group that you generated in the bootstrap step, if you don't know what this is, hint: `terraform output`





E. Automate deployment of the application to the deployed ec2 instance. Set up an ansible playbook that deploys and configures the application on the ec2 instance you deployed in the previous step.

It should:

- Automatically generate the inventory file (hint: a shell script & `terraform output`)
- Copy the artefact you have generated to the remote machine (Hint, to generate the artifact you can run `make pack`)
- Configure dependencies and install node dependencies.
- Configure the application with the correct database endpoint and credentials.
  - These should be automatically be fed in, rather than manually entered (hint: ``terraform output``, how you choose to set these are up to you)
- Set the application up as a service using SystemD so it will automatically start if the server is rebooted.



## Credit

Set up Terraform to use a remote backend. This includes:

- State files stored in S3
- Leveraging DynamoDB to lock the statefile.

Hint: Resources should already be provisioned as part of the Bootstrap process, check the outputs (`terraform output`) of Bootstrap to confirm IDs & names of resources provisioned.

## Distinction

If you can finish all the above tasks, then you can start working on D tasks.

Update your Terraform Templates to automatically use the latest Amazon Linux 2 AMI as opposed to being statically assigned. Hint: Look at Terraform Data Sources.

You will need to update the `aws_instance` resource to obtain the AMI ID from the datasource you created as opposed to using a hardcoded value.

### Useful resources:

<https://www.terraform.io/docs/language/data-sources/index.html>

<https://registry.terraform.io/providers/hashicorp/aws/latest/docs/data-sources/ami>

## High Distinction

If you can finish all the above tasks, then you can start working on HD tasks.

**No help or consultation will be provided for these tasks. You are welcome to ask general questions regarding the tasks, but this is a strictly self-research section**

Deploy the application automatically using CircleCI





## Assessment criteria

You will be assessed on:

- Your ability to automate the deployment of the AWS infrastructure using Terraform.
- Your ability to deploy the application and configure it automatically using Ansible
- Your ability to document and explain the solution so the scripts provided can be executed by someone with no prior context
- How well you analyse the problem and justify your solution for each task
- Do you use the branching and commit changes often?

## PLAGIARISM:

All assignments will be checked with plagiarism-detection software; any student found to have plagiarised would be subject to disciplinary action. Plagiarism includes:

- CONTRACT CHEATING: paying someone to do your work
- CONTRACT CHEATING: getting someone else to write the test or attend demo
- submitting work that is not your own or submitting text that is not your own
- copying work from/of previous/current semester students
- allowing others to copy your work via email, printouts, social media etc.
- posting assignment questions (in full or partial) on external technical forums
- sending or passing your work to your friends
- posting assignment questions on technical forums to get themselves

A disciplinary action can lead to

- a meeting with the disciplinary committee
- a score of zero for the assignment
- a permanent record of copying in your personal university records and/or
- expulsion from the university, in some severe cases

All plagiarism will be penalised. There are no exceptions and no excuses. You have been warned. For more details please read RMIT's page on Academic Integrity at

<https://www.rmit.edu.au/students/student-essentials/assessment-and-exams/academic-integrity>





## Submission Procedure

Each submission must include a README.txt file containing your full name, Student ID and any other relevant information (if you are working in a group, then please mention the details of your partner).

ACME corp. expects **you push all the code and documentation required** to run the Continuous Integration build you are creating for **them into your Github classroom repository**. This includes the files that define any shell scripts, Docker files, or anything else you use.

**Please submit ONLY the URL of your GitHub Classroom repository into Canvas, your repository must have a format like below:**

`https://github.com/RMIT-SDO/assignment-1-student-<your github username>`

## Late submissions and Extension-related information

A penalty of 10% per day of the total marks for each assignment will apply for each day a submission is late, including both weekdays and the weekend. After 5 days, you will receive zero marks for that assignment.

Email your head tutor, Homy: ([amirhomayoon.ashrafzadeh@rmit.edu.au](mailto:amirhomayoon.ashrafzadeh@rmit.edu.au)) for extension related queries.