

COSC2759-2110 Assignment 3 specifications

Deadline	Sunday 13.06.2021 (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).
- Watch the online recordings on a regular basis if you cannot attend the live sessions.
- Do NOT start the assignment at the last minute.
- **THERE WOULD BE NO CLASS EXTENSION FOR THIS ASSIGNMENT.** Extensions may only be granted for severe personal and medical cases and only based on provided evidence. Issues like working full-time or other assignment would not be accepted as a valid excuse.

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.

If there are any issues with your AWS Educate account, please email course coordinator at shekhar.kalra@rmit.edu.au 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 with the work you did on their application and has been hard at work expanding on the work that you started. They have containerised the application and expanded the CI build to create a container and publish it. Next they are hoping to start using Kubernetes to host their application as they feel that will be a big advantage for them.

They have asked you to expand their CI build to automatically deploy the application to a Kubernetes cluster, implementing a CD pipeline.

The Approach

To make this easier for Stonks Inc., which doesn't have extensive experience with DevOps, we will be using well known SaaS tools 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 your configuration and scaffolding scripts.

Tools to use:

- GitHub (We will be using GitHub Classroom)
- Terraform
- AWS
- CircleCI
- Docker
- Kubernetes
- Helm

Stonks Inc expects **you update your Github classroom repository all the code and documentation required** to run the Continuous Integration build you are creating for them. **This includes all the files.**



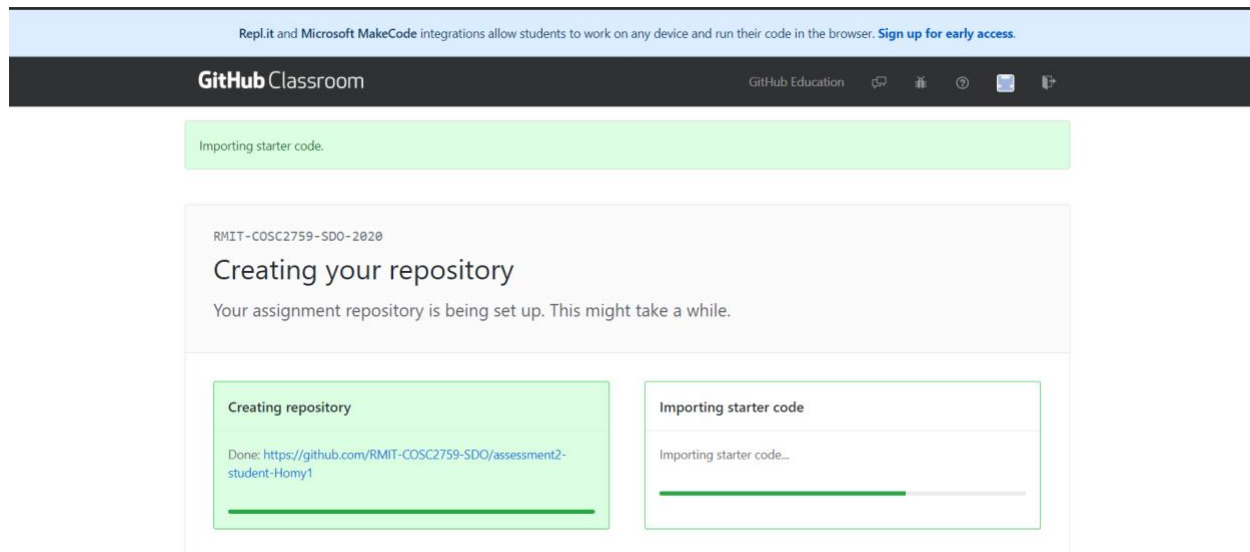


Deliverable

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 #3. You will be expected to work from this repo and make all of your commits to this repo.

Create Github Classroom for the assignment

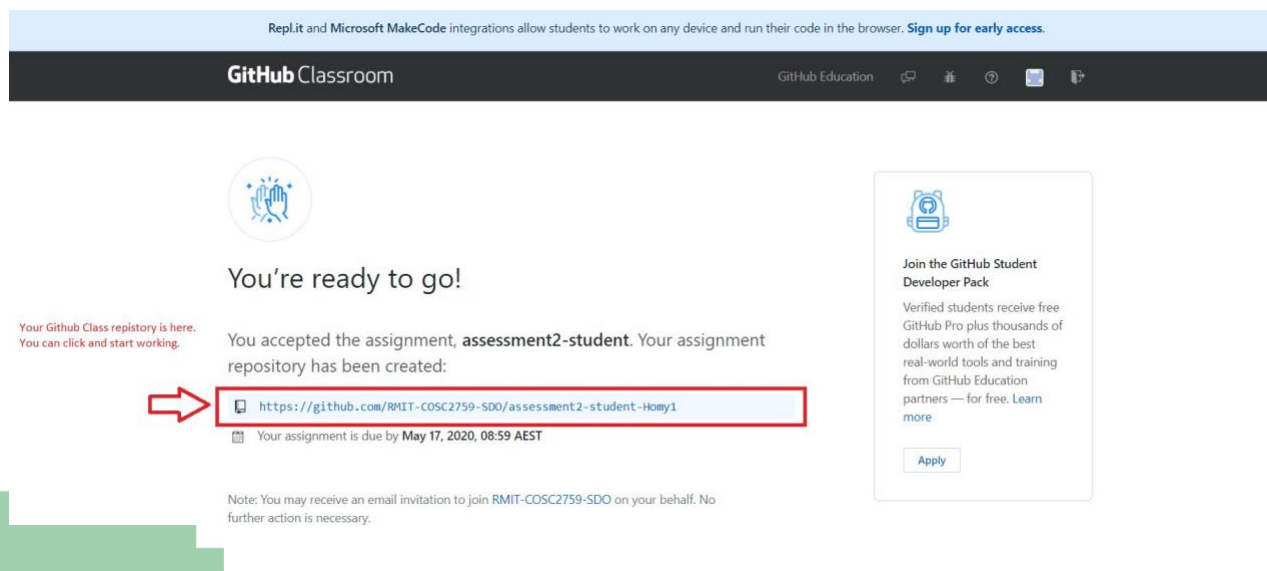
Please accept [this invitation](#), (you must login with your Github account).



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. Set up your environment:

- a. 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).
 - i. This will create a new terraform state bucket, kops state bucket & container repo for you to use for this assignment.
 1. You are welcome to use your old state buckets/resources if you wish however for the sake of this assignment it may be easier to start again.
 2. Deploy a new KOPS K8s Cluster for you to work out of.
 3. Ensure you have gone through the CI pipeline.yaml file and updated the KOPS
 4. Ensure you have updated the bucket details in the Makefile in the Infra folder.
 5. Ensure the environment variables in your package step are updated to the newly created ECR.
- b. Deploy your kubernetes cluster using KOPS.
- c. Ensure your environment variables are set in your CircleCI Project to allow your CICD Tool to communicate with AWS.
- d. Update the Makefile in the root repository to create a new namespace in your Kubernetes cluster, one namespace exist for test.

C. (15 marks) Create a HELM chart to deploy the application to Kubernetes

- The files should be created under the helm folder. The helm chart name should be called 'todo.'
- Include both a Deployment and a Service manifest
 - **Service Manifest:**



- Your service manifest should be of type loadbalancer and ensure that the ports are listening on the correct port and targeting the port listening for the application.
 - **Deployment Manifest:**
 - Use a variable to allow the Image to be passed in
 - Use a variable to allow the database endpoint to be passed in.
- D. Deploy the application into a non-production (test) environment. Make sure the following is implemented in the CircleCI configuration file and add a screenshot of the successful execution.
- a. Deploy a database to back the application
 - i. Infrastructure-as-Code has already been provided for you to do this.
 - ii. Please ensure that you update the terraform.tfvars file with the correct details required i.e The Subnet IDs & VPC IDs that were created by KOPS to host your kubernetes cluster.
 - 1. Echo the mongodb URL connection string out to a file in the root folder i.e. `../db_connection_string.txt`, you will need to use this in the next step and read it into the `DB_URL` environment variable. Hint, you can use Terraform output for this to get the username, password and URL and transform it into the mongodb connection string.
 - b. Deploy HELM Chart to the Kubernetes cluster.
 - i. Ensure that you read in the image name (check the package step in the pipeline to confirm where the image name is stored) and db connection strings. Hint: you can read and parse the contents of a file into a cli command with the cat command and by using command substitution i.e. `$(cat ...)`
 - c. Ensure that the smoke test command is run at the end of the deployment job to verify if the LB is up and reachable.

Credit:


Update the end-to-end test to run against the non-production environment you have just created and add it into the workflow for the CICD pipeline. This should take place after the non-production (test) environment has been deployed.

Distinction:

Deploy the application into a production environment.

- A. Create a namespace for the application in Kubernetes (You will need to update the makefile in the first step to create the namespace for production).
- B. Make sure the following is implemented in the CircleCI configuration file and add a screenshot of the successful execution.
 - i) Deploy a database to back the application
 - ii) Deploy HELM Chart to the Kubernetes cluster
 - iii) Ensure that the smoke test command is run at the end of the deployment job to verify if the LB is up and reachable.



- 
- C. Create a stage gate before deployments are allowed into production Make sure it is implemented in the CircleCI configuration file and add a screenshot of the successful execution.

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

- A. Integrate logging to the solution so any logs from the Kubernetes cluster are automatically stored in AWS CloudWatch for the future.

Provide a screenshot of the TODO app logs being displayed in CloudWatch (Hint: use the querying function in Log Insights), as well as the code used to deploy the logging solution to the cluster. Add the screenshot to your Readme file.

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
- Does the provided Zip file contain everything required for ACME corp to be able to use the deployment you have designed?
- 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 Homy: (amirhomayoon.ashrafzadeh@rmit.edu.au) for late submission or extension related queries.