School of Computing Technologies

COSC2626/2640 Cloud Computing

Assessment 2

✺	Assessment Type: Individual assignment; no group work. Submit online via Canvas→Assignments→Assessment 2. Marks awarded for meeting requirements as closely as possible. Clarifications/updates may be made via announcements/relevant discussion forums.
📅	Due date: 11:59pm, Friday 30/Apr/2021 Please check Canvas→Assignments→Assessment 2 for the most up to date information. As this is a major assignment in which you demonstrate your understanding, a university standard late penalty of 10% per each working day applies for up to 5 working days late, unless special consideration has been granted.
⏬	Weighting: 20 marks

1. Overview

This assignment consists of 2 tasks:

- Task 1: Develop an EC2, S3 and DynamoDB based application based on a programming language of your choice
- Task 2: Further develop your draft proposal for Assessment 3

2. Assessment Criteria

This assessment will develop your ability to:
- Develop cloud-based applications using AWS EC2, S3 and DynamoDB services

3. Learning Outcomes

This assessment is relevant to the following Learning Outcomes:

- Develop and deploy cloud applications using popular cloud platforms
- Design and develop highly scalable cloud-based applications by creating and configuring virtual machines on the cloud and building private cloud.

4. Assessment details

Task-1   [19 marks]

Create a simple online music subscription application with a programming language of your choice, S3 and DynamoDB and host it in a free-tier EC2 instance (with Ubuntu Server 20.04/18.04 LTS AMI) using your learned knowledge in Tute 4 and Tute 5 which will have following components and functions:

1. DynamoDB: 3 marks

    1.1. Create a "login" table in DynamoDB containing 10 entities with the following attributes and values. 1 mark

email (Type: String)	user_name (Type: String)	password (Type: String)
s3######0@student.rmit.edu.au i.e. your RMIT student id+'0@student.rmit.edu.au'	Firstname Lastname0 i.e. your name+'0'	012345
s3######1@student.rmit.edu.au i.e. your RMIT student id+'1@student.rmit.edu.au'	Firstname Lastname1 i.e. your name+'1'	123456
…	…	…
s3######9@student.rmit.edu.au i.e. your RMIT student id+'9@student.rmit.edu.au'	Firstname Lastname9 i.e. your name+'9'	901234

1.2. Write a program to automatically create a "music" table in DynamoDB with the following attributes. 1 mark

title, artist, year, web_url, image_url

1.3. Write a program to automatically load the data from a2.json to your music table. 1 mark

2. Write a program to automatically download all the artist images according to the image_url values in a2.json and upload the images into S3. 2 marks

3. login page 2 marks

The login page contains an "Email" text field, a "Password" field, and a "Login" button as well as a register link. When user clicks the "Login" button, it will validate if the user entered credentials match with the information stored in the login table.

3.1. If the user credential is invalid, the login page will display "email or password is invalid". 1 mark

3.2. If the user credential is valid, it will be redirected to the main page. 1 mark

4. register page 3 marks

The register page contains an "Email" text field, a "Username" a "Password" field, and a "Register" button. When a user clicks the "Register" button, it will validate if the user entered email matches with the email stored in the login table.

4.1. If the entered email matches with the email stored in the login table, the register page will show "The email already exists". 1 mark

4.2. If the entered email is unique,

4.2.1. the new user information will be stored in the login table, and 1 mark

4.2.2. the user will be redirected to the login page, where user can login with the new email and password. 1 mark

5. main page 9 marks

The main page contains three main areas: a user area, a subscription area, a query area and a "Logout" link.

5.1. After a user logs in, the user area will show the corresponding user_name. 1 mark

5.2. The subscription area

    5.2.1. The subscription area will show all the user subscribed music information (title, artist, and year) stored in DynamoDB. 1 mark

    5.2.2. Each music information is followed by the corresponding artist image retrieved from S3 and a "Remove" button. 1 mark

    5.2.3. If the user clicks a "Remove" button, the corresponding subscribed music information and artist information will be removed from the subscription area and the corresponding table in DynamoDB. 1 mark

5.3. The query area should contain three text areas, "Title", "Year", "Artist" and a "Query" button. Once the user enters some information in any (or all) these text areas and clicks the "Query" button,

    5.3.1. If the queried information is not contained in the entities' corresponding attribute value(s) in the music table, it will show "No result is retrieved. Please query again". 1 mark

    5.3.2. If the queried information is contained in (one or more) entities' corresponding attribute value(s) in the music table, the area will show

        5.3.2.1. All the retrieved music information (title, artist, and year). 1 mark

        5.3.2.2. Each music information is followed by the corresponding artist images retrieved from S3 and a "Subscribe" button. 1 mark

        5.3.2.3. If the user clicks a "Subscribe" button, the subscribed music information and the corresponding artist image will be added into the subscription area and the subscribed music information will be stored in DynamoDB. 1 mark

5.4. If the user clicks the "Logout" link, the user will be redirected to the login page. 1 mark

Important Notes:

1. The whole application must be **COMPLETELY** programmed by yourself. You **CANNOT** use any external source code.

2. The whole application must be **FULLY** deployed in the designated EC2 instance, otherwise **NO MARK** will be given for Task 1-3 to Task 1-5.

Task-2 [1 mark]

Write at least 400 words to describe a more detailed plan of your Assessment 3, including the motivation, the objective, the possible system components, the proposed functions and the possibly adopted AWS services and/or external APIs and the functions of each service/API.

Note:

1) References should be included when needed (excluded in the word count);

2) This task is only designed to encourage you to further develop and improve your previously submitted draft plan for Assessment 3. We won't assess the quality of Task-2 content. You may discuss the details of your plan with your tutor in weekly workshops.

5. Referencing guidelines

What: This is an individual assignment, and all submitted contents must be your **OWN**. If you have used sources of information other than the contents directly under Canvas→Modules, you must give acknowledge the sources and give references using IEEE referencing style.
Where: Add a code comment near the work to be referenced and include the reference in the IEEE style.
How: To generate a valid IEEE style reference, please use the citethisforme tool if unfamiliar with this style. Add the detailed reference before any relevant code (within code comments).

6. Submission format

Create a .txt file and name it [your_student_number].txt (e.g. s3369312.txt). This .txt file will contain items in following order

1. Your Name and student number
2. Code of Task 1 (all python/html/php code in your project)
3. Texts of Task 2

Submit the txt file into Canvas→Assignments→Assessment 2 before the deadline. You will be marked during your demo time. This submission is only for keeping the records. However, your assessment will NOT be allowed to be demonstrated in Canvas until you submit your file.

7. Demonstration

You must demonstrate your project online to an available tutor (**unnecessarily your tutor**) by making an appointment with the tutor in Week 9. The demo booking will be made available to students in Week 5. All demonstrations must be completed by Week 9. There will be **penalty** if you fail to complete and demonstrate your work by Week 9. The demo is around **15 minutes** for each student. Keep everything ready and make your application live during your demo.

Note: Each student can only attend one demonstration for Assessment 2. **DON'T** overbook otherwise you will receive penalty. If you are not satisfied with your mark, you may attend the online remarking session organized by the course coordinator after Week 9 (remarking time and booking will be made available to students through announcements).

8. Academic integrity and plagiarism (standard warning)

Academic integrity is about honest presentation of your academic work. It means acknowledging the work of others while developing your own insights, knowledge and ideas. You should take extreme care that you have:

- Acknowledged words, data, diagrams, models, frameworks and/or ideas of others you have quoted (i.e. directly copied), summarised, paraphrased, discussed or mentioned in your assessment through the appropriate referencing methods,

- Provided a reference list of the publication details so your reader can locate the source if necessary. This includes material taken from Internet sites.

If you do not acknowledge the sources of your material, you may be accused of plagiarism because you have passed off the work and ideas of another person without appropriate referencing, as if they were your own. RMIT University treats plagiarism as a very serious offence constituting misconduct. Plagiarism covers a variety of inappropriate behaviours, including:

- Failure to properly document a source
- Copyright material from the internet or databases
- Collusion between students

For further information on our policies and procedures, please refer to the University website.

9. Assessment declaration

When you submit work electronically, you agree to the assessment declaration.