




# COSC2626/2640 Cloud Computing

## Assessment 1

	<b>Assessment Type:</b> This is an <b>individual</b> assignment; <b>NO</b> group work is allowed. Submit your solutions online via Canvas→Assignments→Assessment 1. Marks are awarded for meeting the requirements as closely as possible. Clarifications/updates may be made via announcements/relevant discussion forums.
	<b>Due date:</b> <b>11:59PM on Friday, 07 April 2023.</b> Please check Canvas→Assignments→Assessment 1 for the most up-to-date information. A university standard late penalty of <b>10% (= 4 marks)</b> per each working day applies for the days that you are late to submit your work for, unless a special consideration has been granted.
	<b>Weighting:</b> 40 marks

### 1. Overview

Develop an AWS EC2, S3, API Gateway, Lambda and DynamoDB based application based on a programming language of your choice.

### 2. Assessment Criteria

This assessment will assess your ability to develop cloud-based applications using AWS EC2, S3, API Gateway, Lambda, 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

Create a simple online music subscription application with a programming language of your choice using AWS 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 which will have following components and functions.

#### Important Note:

1. The application should be **FULLY hosted** in a web server environment (e.g., Apache2 or suitable alternative) and be able to be rendered in a web browser by directly entering the root Public IPv4 DNS (in either https or http) of the EC2 instance hosting the application.
2. Using Elastic Beanstalk is **NOT allowed** as a valid means of deploying your application. Using non-standard HTTP ports (i.e. ones other than 80/443) are also **NOT allowed**.

## Task Description

**Note:** AWS Academy account does not allow to create any IAM roles. However, a role named LabRole has been pre-created for you. Use **LabRole** instead.

### 1. (2 + 2 + 2 = 6 marks) AWS DynamoDB

- 1.1. Create a “login” table in DynamoDB containing 10 entities with the following attributes and values.

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 table titled “music” in DynamoDB with the following Attributes (some attributes might not be used to implement any functionalities in the subsequent sections): title, artist, year, web\_url, image\_url

- 1.3. Write a program to automatically load the data from a1.json to your music table.

### 2. (2 marks) Write a program to automatically download all the artist images according to the image\_url values in a1.json and upload the images into S3.

### 3. (2 + 2 = 4 marks) Login page

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.

If the user credential is invalid, the login page will display “email or password is invalid”; it will be redirected to the main page otherwise.

### 4. (1 + 2 + 2 = 5 marks) Register page

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. (1 mark) If the entered email matches with the email stored in the login table, the register page will show “The email already exists”.

- 4.2. (2 + 2 = 4 marks) If the entered email is unique, the new user information will be stored in the login table, and the user will be redirected to the login page, where user can login with the new email and password.

**5. (15 marks) Main page**

The main page contains three areas (a user area, a subscription area, and a query area) and a “Logout” link.

5.1. (1 mark) After a user logs in, the **user area** will show the corresponding user\_name.

5.2. (1 + 2 + 3 = 6 marks) The **subscription area**

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

5.2.2. (2 marks) Each music information is followed by the corresponding artist image retrieved from S3 and a “Remove” button.

5.2.3. (3 marks) 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.

5.3. (1 + 2 + 2 + 2 = 7 marks) The **query area** should contain three text areas, “Title”, “Year”, “Artist” and a “Query” button. The user may enter some information in any (or all) of these text areas and click the “Query” button,

5.3.1. (1 mark) 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”.

5.3.2. (2 + 2 + 2 = 6 marks) 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).

Note: If a user enters more than one query conditions, the query conditions are connected by “AND” operator(s) by default.

5.3.2.2. Each music information is followed by the corresponding artist images retrieved from S3 and a “Subscribe” button.

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.

5.4. (1 mark) If the user clicks the “Logout” link, the user will be redirected to the login page.

**6. (8 marks) Advanced task:** the use of API Gateway and Lambda functions.

The application needs to read and update the records (user, music, and subscription) in DynamoDB through API Gateway and Lambda function.

**Important Notes:**

1. The whole application must be **COMPLETELY** programmed by yourself. You **CANNOT** use any external source code (e.g., website templates, etc.).
2. The whole application must be **FULLY** deployed in the designated EC2 instance, otherwise **NO MARK** will be given for Task 3, 4 and 5.

## 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 .zip file and name it **[your\_student\_number].zip** (e.g., s3369312.zip). This .zip file will contain the code you developed.

Submit the .zip file on Canvas 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 the tutor assigned to your class by making an appointment in Week 6. The demo booking will be made available to students in early **Week 6**. All demonstrations must be completed by **Week 7**. There will be **penalties** if you fail to complete and demonstrate your work by **Week 7** unless you have a valid extension or special consideration granted. The demo will take around **15 minutes** for each student. Further details on the demonstrations and their structure will be announced in **Week 6**.

**Note:** Each student can only attend one demonstration for Assessment 2. **DO NOT** overbook slots. If you do, you will receive a penalty of **10% (= 4 marks)**.

## 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 behaviors, 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](#).