中原大學 資訊工程學系

雲端系統實務與開發 － 期末書面報告

Serverless Voted System

資訊四甲 10827107 羅昭艾
資訊四乙 10827257 方思涵
資訊碩一 11177035 林彥輝

授課老師：鍾武君 教授

中華民國一一二年

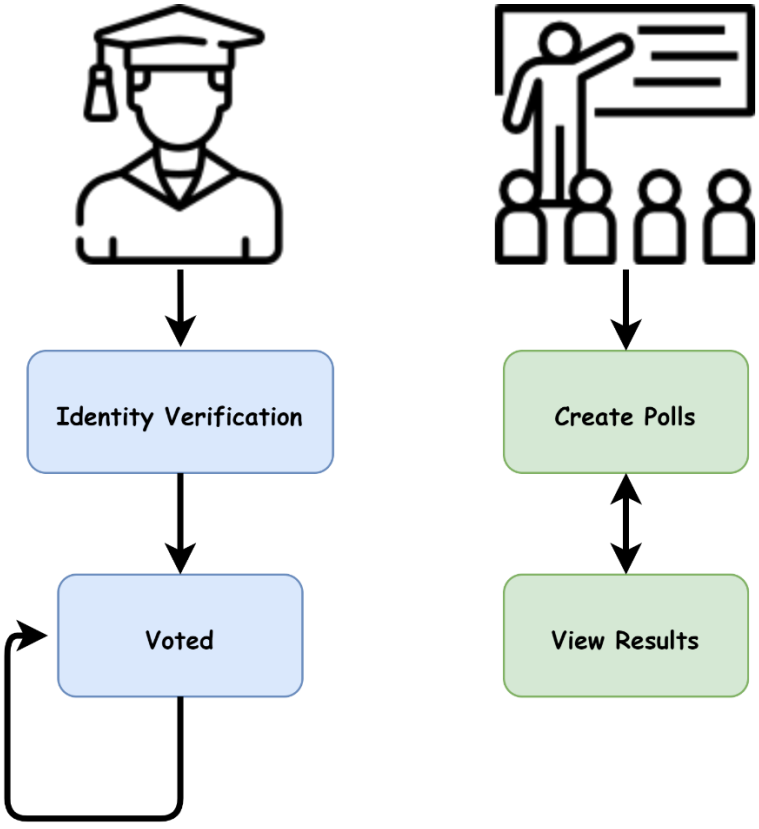
1. User Story

鑒於網路的普及、晶片的快速發展，人手一機的情形已經是常態，任何的學習資源都能在知名的影音串流平台（如 YouTube、Coursera）上透過手邊的便攜裝置進行瀏覽。

在現實的教學環境中，師生可以透過多種方式進行互動，以提高學習成效。其中一種方式是利用紙本隨堂測驗快速進行學習評量，但前置手續成本過高。為此，可以將投票系統引入學校的教學環境中，以匿名方式進行投票。學生可以放心地表達課堂上的意見和反饋，以便師長能夠快速掌握學生的學習狀況。

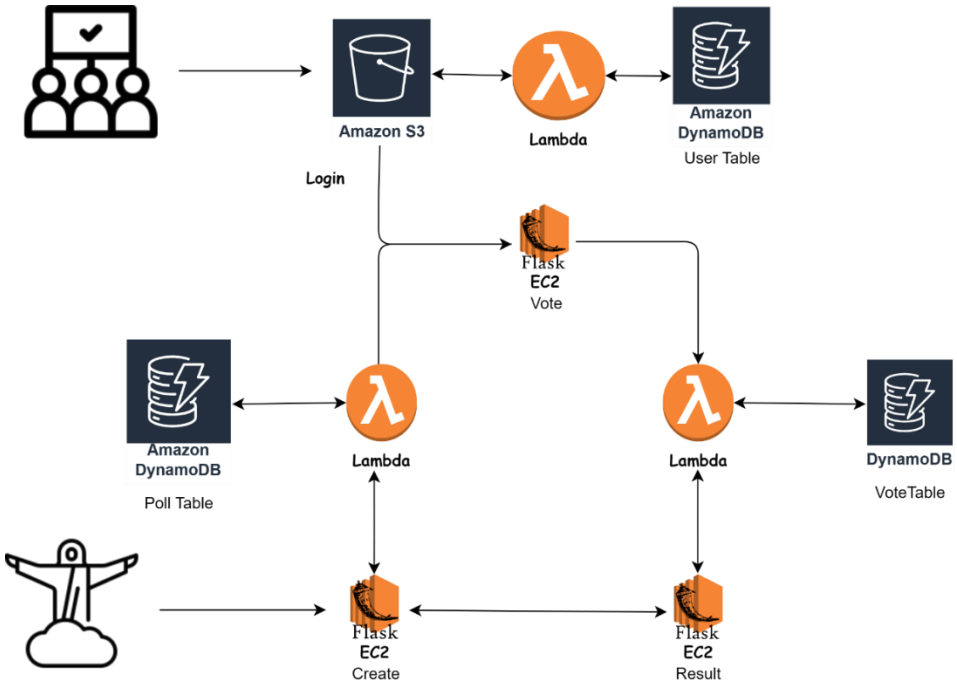
學習科技是近期學習的趨勢，旨在將科技融入於教學當中，透過科技的方式拉近教學的距離。在目前有許多教學輔助軟體 [1]，但這些軟體都存在一些缺陷 [2]，例如人數和投票樣式固定等問題。在這個專題中，我們希望透過一個投票機制，搭配無伺服器的方式來建立此系統。

這個專題搭建了一個基於無伺服器的系統，如圖一所示，系統將分為兩種角色進行使用。第一種是教師的身分，教師能夠創建一個投票活動並查看投票結果；另一種是學生，學生能根據教師所開啟的活動進行投票。學生的身份會經過 SHA256 [3] 加密，因此無法從加密後的密文反推學生的真實身份，學生能更加放心的作答。



圖一、使用流程圖

2. System Architecture and Technologies



圖二、系統架構圖

學生（圖二上半部）可透過 Amazon S3 的連結開啟登入頁面，此登入頁面透過 Lambda 與 DynamoDB 做加密登入的帳密驗證，登入成功後再導向至 EC2 Flask 所架出的 Vote 頁面進行投票，投票後的結果將傳送到 DynamoDB 當中。

老師（圖二下半部）可直接進入 EC2 Flask 所架出的 Create 頁面創建投票，並將投票相關的基本資訊透過 Lambda 傳送到 DynamoDB，創建完投票後將重導向到 Result 頁面查看結果，本專題一共使用了以下技術：

1. AWS S3 [4]

   是一種 AWS Object Storage 的服務，可用於儲存各式檔案，並能託管網站，本專題將帳號登入頁面交由 S3 託管，並額外在 JavaScript 部分使用 JQuery.ajax() [5] 進行 RESTful API 的呼叫；CSS 部分採用 NicePage [6] 進行美編。

2. AWS EC2 [7]

   是 AWS 的 VM 服務，本專題使用 EC2 來建置 Python 環境，並使用 Python Flask [8] 來搭建專題剩餘的網站並額外在 JavaScript 部分使用 JQuery.ajax() [5] 進行 RESTful API 的呼叫；有關數據視覺化部分採用 Canvas [9] 套件來進行長條圖的繪畫。

3. AWS DynamoDB [10]

   是 AWS 的資料庫服務，提供高可穩定性用於儲存 Semi-Structure 格式的資料，本專題在資料庫當中創建三個 Table，分別用於儲存身分、投票欄位、投票歷史紀錄。

4. AWS Lambda [11] & API Gateway [12]

   是 AWS 的 Serverless 服務，提供 Function as a Service 能使使用者將部分程式佈署至雲端供應商做託管，為程式提供更好的解耦穩定性。本專題採用兩種 Trigger 方法（HTTP Trigger、DynamoDB Trigger）作為執行程式的條件，並搭配 API Gateway 可以產生一個端點供 RESTful API 呼叫。

在 DynamoDB 當中，資訊存放方式如下圖：

accountID	createDate	hash-password
Sadie	2023/6/5	0e9f7139fe5ac2fb2124541f125d0299345dec149375c17032044151bcd3daaf
Hui	2023/6/5	9e6d51ee8e05e878f46fc7fc64dcc4f137e8dc4350bf304db3da872252eee011
Jai	2023/6/5	b769d75224ca678bf1de9e6f20bb322725db0586ce9bb7d0d67f9a9ef7b8e17d

圖三、帳號密碼欄位：{[String]帳號、[String]創立時間、[String]密碼}

question	createDate	options
下列何者並非狗的品種?	20230605	[{"S":"薩摩耶"},{"S":"比熊"},{"S":"曼赤肯"},{"S":"柴...
預計出席畢業典禮的親...	20230605	[{"S":"1"},{"S":"2"},{"S":"3"},{"S":"4"},{"S":"5"}]
最喜歡的城市	20230604	[{"S":"台北"},{"S":"桃園"},{"S":"台中"},{"S":"新竹"},...

圖四、投票紀錄欄位：{[String]題目、[String]創立時間、[List]結果}

在 Lambda 當中，程式碼展示：

HTTP Trigger 的 Lambda 能夠接收 POST 的訊息，並且將內容（投票資訊）透過 boto 函數連線並存放至 DynamoDB 特定 Table 當中。並且 Response 資訊，資訊中需要特別設定 CORS。

lambda_function

```python
import json
import boto3


def lambda_handler(event, context):
    # 從event中解析POST參數
    body = json.loads(event['body'])
    createDate = body['createDate']
    question = body['question']
    options = body['options']



    dynamodb = boto3.resource('dynamodb')
    table_name = 'PollTable'
    table = dynamodb.Table(table_name)


    item = {
        'createDate': createDate,
        'question': question,
        'options': options
    }
    table.put_item(Item=item)



    response = {
        'statusCode': 200,
        'headers': {
            'Content-Type': 'application/json',
            'Access-Control-Allow-Origin': '*',
            'Access-Control-Allow-Headers': 'Content-Type',
            'Access-Control-Allow-Methods': 'OPTIONS,POST,GET',
        },
        'body': json.dumps([createDate, question, options])
    }
```

圖五、Lambda 程式展示之一

DynamoDB Trigger 的 Lambda 能在資料庫每次更新時將資訊透過 GET 傳到特定 Flask 的網頁中，網頁就能及時更新投票結果。

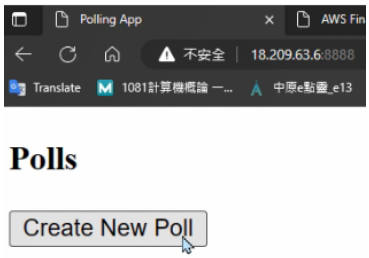
lambda_function

```python
import requests



print('Loading function')


def lambda_handler(event, context):
    #print("Received event: " + json.dumps(event, indent=2))
    for record in event['Records']:
        print(record['eventID'])
        print(record['eventName'])
        print("DynamoDB Record: " + json.dumps(record['dynamodb'], indent=2))
        url = 'http://18.209.63.6:8888/trigger'

        response = requests.get(url, data=record)

    return 'Successfully processed {} records.'.format(len(event['Records']))
```

圖六、Lambda 程式展示之二

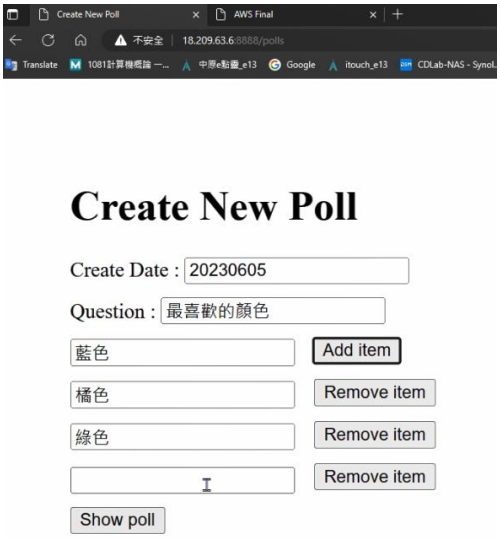
3. Demonstration

I. 老師可透過 Browser 連線至創建投票表單的網頁。



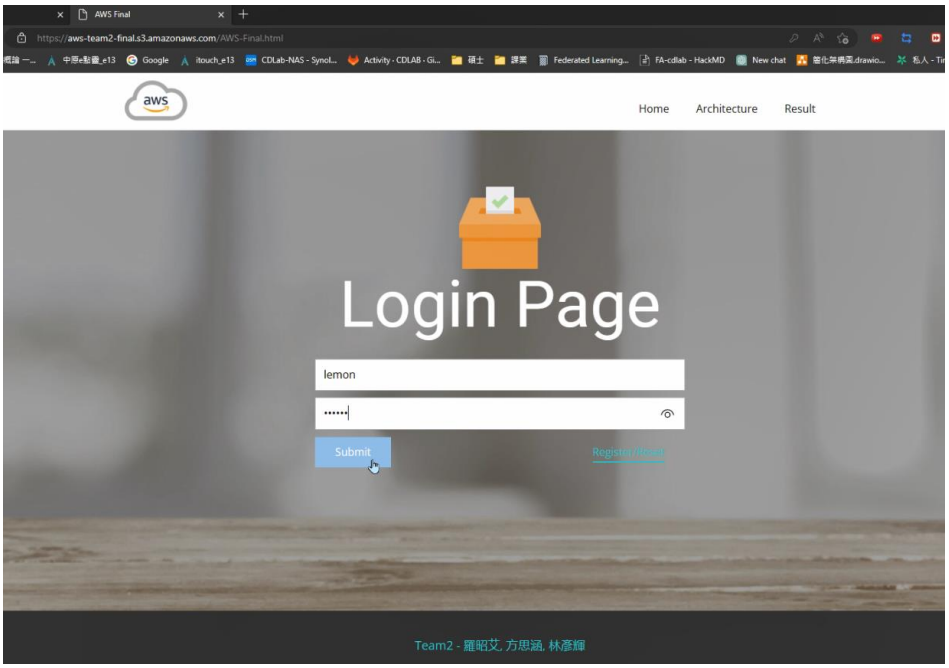
圖七、老師建立投票介面

II. 按下 Create New Poll 後，老師可填入創建的日期、表單問題，以及根據所提出的問題動態地新增或刪除選項，當按下 Show Poll 後，便會自動轉到顯示投票統計結果的 Result 網頁。



圖八、老師建立投票細節

III. 學生同樣可透過 Browser 連線至登入驗證畫面。

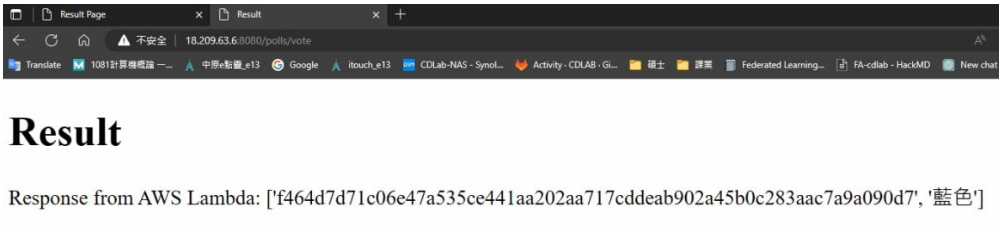


圖九、學生登入頁面

IV. 成功登入後便會自動跳轉到先前由老師所創建的投票表單，並且由於帳號及密碼是經過 Hash 加密，因此所有學生都是匿名投票。

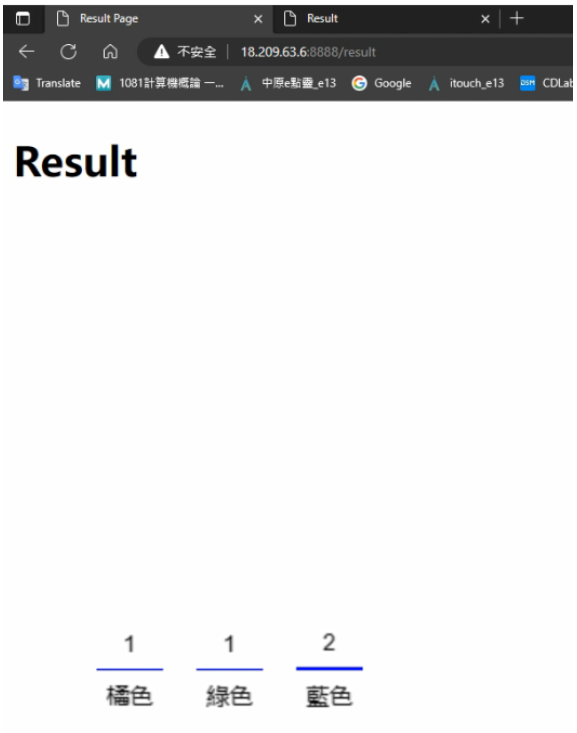
圖十、學生作答頁面

V. 當學生按下 Vote 後，會顯示出成功投票的檢視畫面。



圖十一、學生投票結束確認視窗

VI. 此時顯示統計結果的頁面便會自動記錄目前所投票的票數，方便老師進行觀察學生們的投票進度及結果。



圖十二、老師確認結果頁面

4. Discuss the cloud properties

本專題是基於 AWS Learner Lab 所建立，所擁有的雲端屬性包含以下：

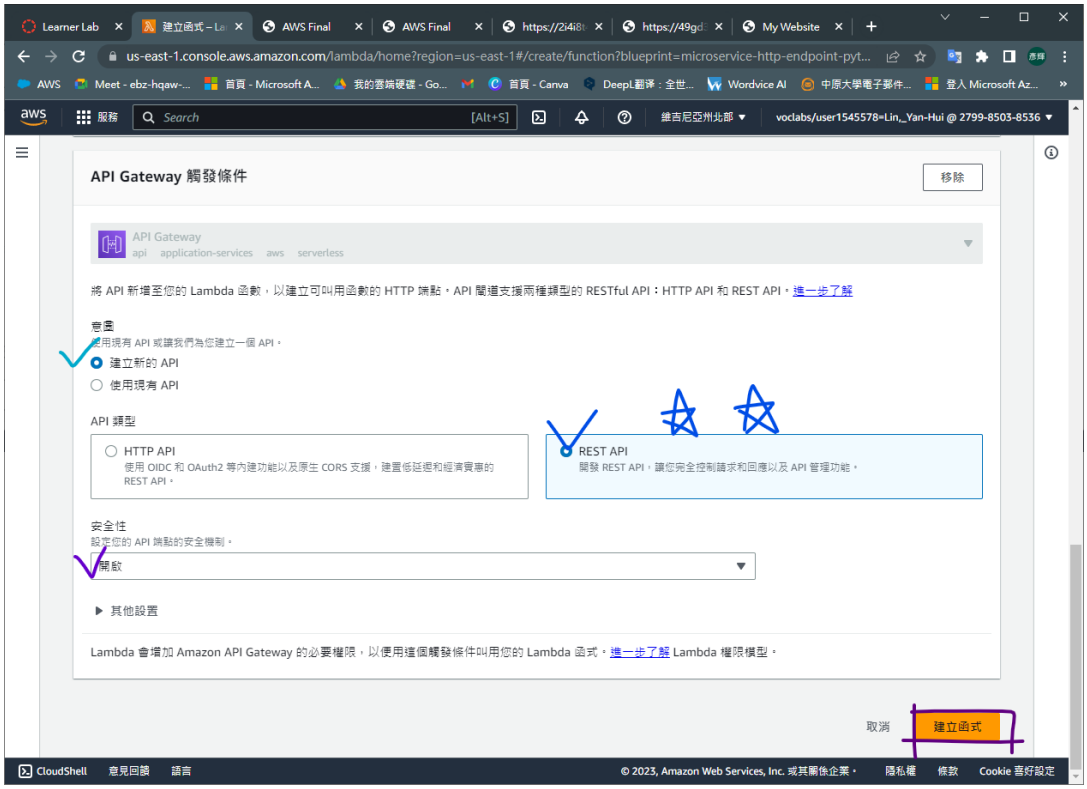
- Availability：透過雲端供應商的高服務水準協議（SLA），使得系統在理想上擁有高可用性。然而，由於部分的 Web 應用仍然基於 EC2 開發，因此運算能力將受限於 EC2 的運算能力。不過此問題仍能透過 Auto Scaling 調整運算資源、或是將 Web 應用交由 AWS 做託管。
- Reliability：在資料儲存方面，使用 AWS DynamoDB 進行帳號密碼的儲存，以及投票資訊的紀錄。DynamoDB 能在故障問題發生時保持高度的可靠性，讓使用者享有 always-on 的體驗[13]。
- Optimization：這個系統使用了非常多的 Serverless 呼叫微服務。Serverless 能夠快速地調整運算資源，以彈性地符合使用者發出的請求。
- Portability：所有的系統使用方式都是透過 Web 呈現，因此使用者只需要具備一個能夠連網的裝置，並且能夠透過瀏覽器開啟網頁即可使用此系統。

5. Lesson learns from your final project

1. Serverless

在本次的專題當中，組員接觸了微服務的概念，使一個大型系統能拆分成不同的物件，在這次的例子當中，是將與資料庫相關的功能皆移植到雲端以 Function as a Service 的方式執行，除了保持高 HA 以外，也能使得專題分工更加明確，每個人能更專注於自己的部分。

過程中碰到了 Lambda API Gateway 的設定問題，使用傳統的 HTTP API 將無法使用 JavaScript 以 RESTful API 的方式進行溝通，會產生 HTTP Error 405 (Method Not Allowed)的問題，需要特別留意在建立 Function 時的 API Gateway 類型，範例如圖十三。



圖十三、創立 Gateway 流程

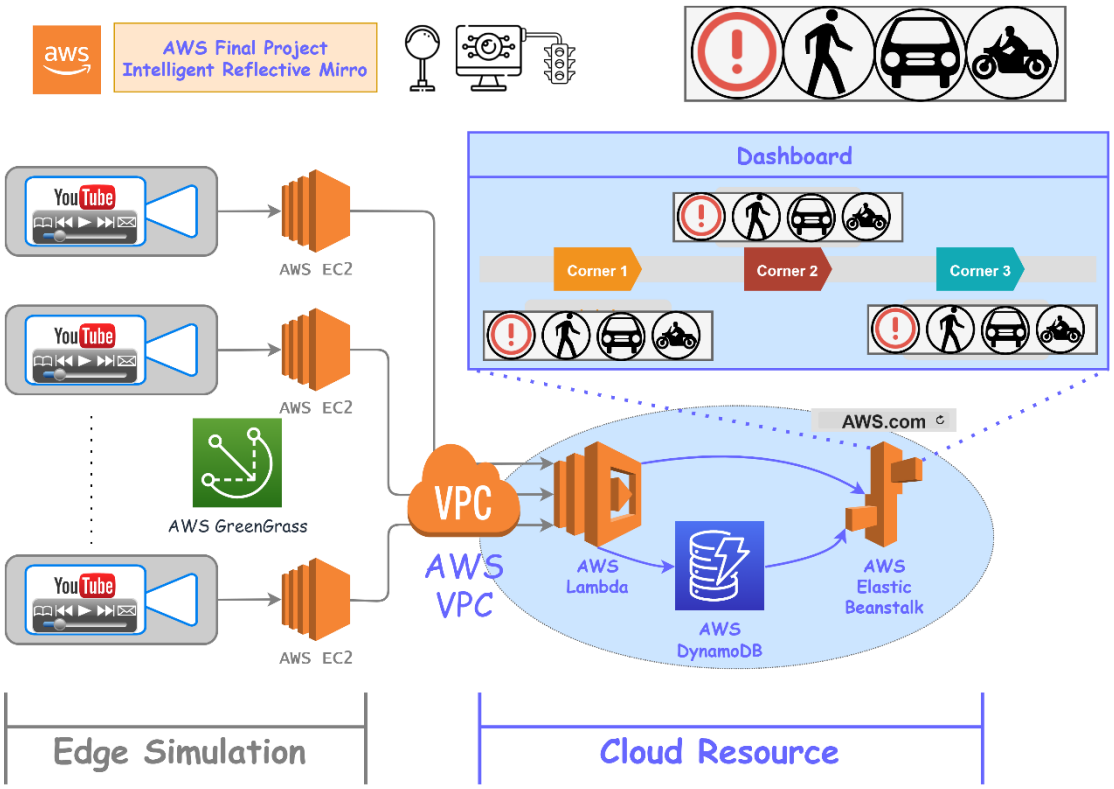
2. Coding Technique

在本次的專題當中，能對於一個產品的前後端有更深入的了解，並習得各式程式語言及電腦系統的架構。這些技能包含：

A. Python Flask 的架站、HTTP GET/POST 概念。
B. HTML、JavaScript 的 RESTful API。
C. 使用 NicePage 軟體協助網頁 CSS 美化。
D. 規劃一個良好的系統架構、設計界面。

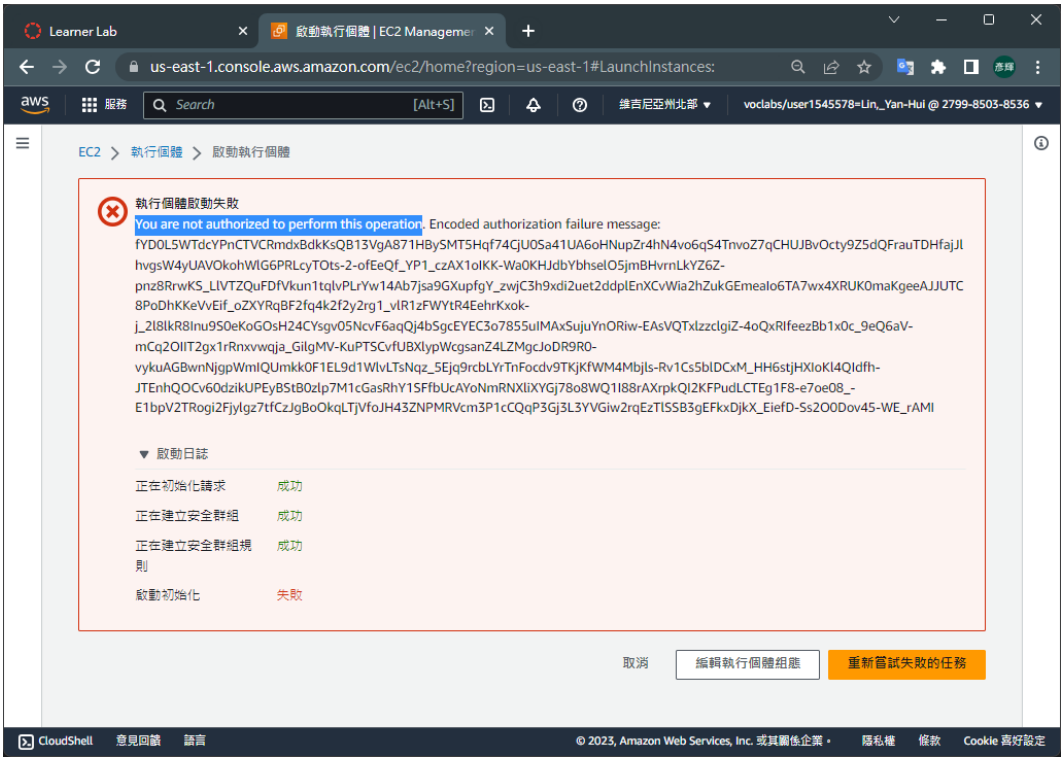
3. 更換題目

在學期中段規劃了一個題目叫做智慧反射鏡，希望透過軟體的方式輔助辨識傳統反射鏡的物件。期待透過 EC2 模擬 Edge 的 Device，並使用 YouTube 的直播畫面作為輸入，並將辨識結果透過 Lambda 傳到前端網頁展示，完整架構圖如圖十四。



圖十四、系統架構圖

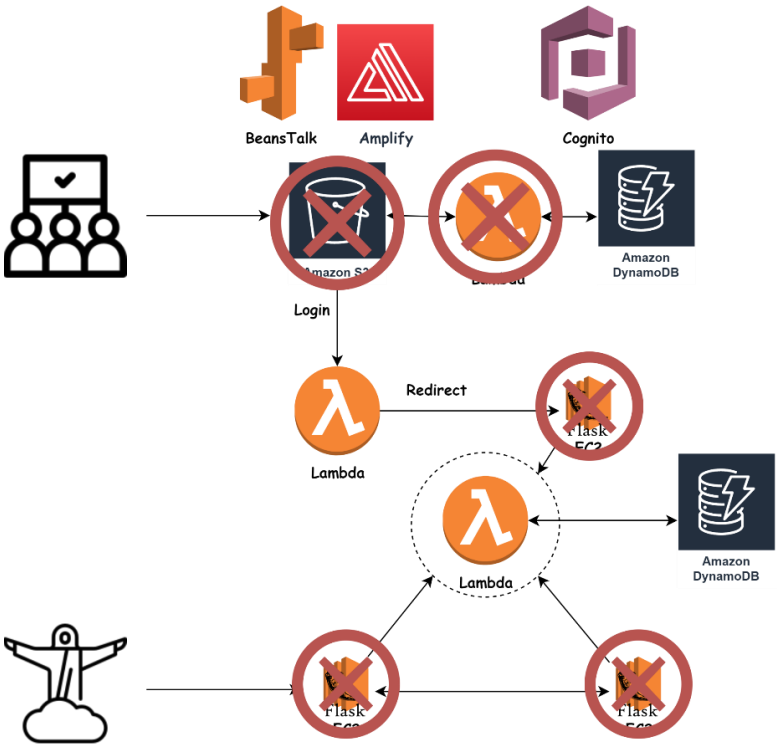
但在規劃統整 Learner Lab 可用資源的過程發現，預設的 IAM 僅能挑選既有的 LabRole，不能新增新的 IAM 權限，也無法擴增 LabRole 的權限，因此在開啟 Inference Type 的 EC2 時，發生了權限不足的問題，截圖如圖十五所示，此外，在 Amplify 或是 AWS App Runner 也有無法開啟資源的問題。因此才毅然決然更改題目為 Serverless Voted System。



圖十五、創立資源警告截圖

6. Future Work

受限於 IAM 的權限，網頁採用 AWS EC2 進行維運，但目前的系統未採用 Scaling 機制來動態調整運算資源以處理使用者的請求，因此在未來展望中可以用更正式的網頁託管服務，例如 Amplify [14]、BeansTalk [15]，此外，Cognito 是 AWS 上的身分認證服務，可以使得系統能支援更多平台的帳號來進行登入，系統架構示意圖如圖十六。



圖十六、未來展望系統架構圖

Reference

1. TA 培訓：高互動線上教學技巧. Available online: https://www.youtube.com/watch?v=ySRekvBNiJo (accessed on 10 June 2023).
2. How many players can play a kahoot? Available online: https://support.kahoot.com/hc/en-us/articles/115003072287-How-many-players-can-join-a-game- (accessed on 10 June 2023).
3. Appel, A.W. Verification of a cryptographic primitive: SHA-256. *ACM Transactions on Programming Languages and Systems (TOPLAS)* **2015**, *37*, 1-31.
4. Cloud Object Storage – Amazon S3 – Amazon Web Services. Available online: https://aws.amazon.com/s3/?nc1=h_ls (accessed on 10 June 2023).
5. jQuery.ajax() | jQuery API Documentation. Available online: https://api.jquery.com/jquery.ajax/ (accessed on 10 June 2023).
6. Nicepage - Your Powerful Website Designer. Available online: https://nicepage.me/ (accessed on 10 June 2023).
7. Secure and resizable cloud compute – Amazon EC2 – Amazon Web Services. Available online: https://aws.amazon.com/ec2/?nc1=h_ls (accessed on 10 June 2023).
8. Welcome to Flask - Flask Documentation (2.3.x). Available online: https://flask.palletsprojects.com/en/2.3.x/ (accessed on 10 June 2023).
9. Canvas 教學文件 - Web APIs | MDN. Available online: https://developer.mozilla.org/zh-TW/docs/Web/API/Canvas_API/Tutorial (accessed on 10 June 2023).
10. Fast NoSQL Key-Value Database – Amazon DynamoDB – Amazon Web Services. Available online: https://aws.amazon.com/dynamodb/ (accessed on 10 June 2023).
11. Serverless Computing - AWS Lambda - Amazon Web Services. Available online: https://aws.amazon.com/lambda/?nc1=h_ls (accessed on 10 June 2023).
12. Amazon API Gateway | API Management | Amazon Web Services. Available online: https://aws.amazon.com/api-gateway/ (accessed on 10 June 2023).
13. DeCandia, G.; Hastorun, D.; Jampani, M.; Kakulapati, G.; Lakshman, A.; Pilchin, A.; Sivasubramanian, S.; Vosshall, P.; Vogels, W. Dynamo: Amazon's highly available key-value store. *ACM SIGOPS operating systems review* **2007**, *41*, 205-220.
14. Full Stack Development - Web and Mobile Apps - AWS Amplify. Available online: https://aws.amazon.com/amplify/?nc1=h_ls (accessed on 10 June 2023).
15. Website & Web App Deployment - AWS Elastic Beanstalk - AWS. Available online: https://aws.amazon.com/elasticbeanstalk/?nc1=h_ls (accessed on 10 June 2023).