

# Serverless Voted System

Team2 羅昭艾  
方思涵  
林彥輝



# Agenda



Introduction



Discussion

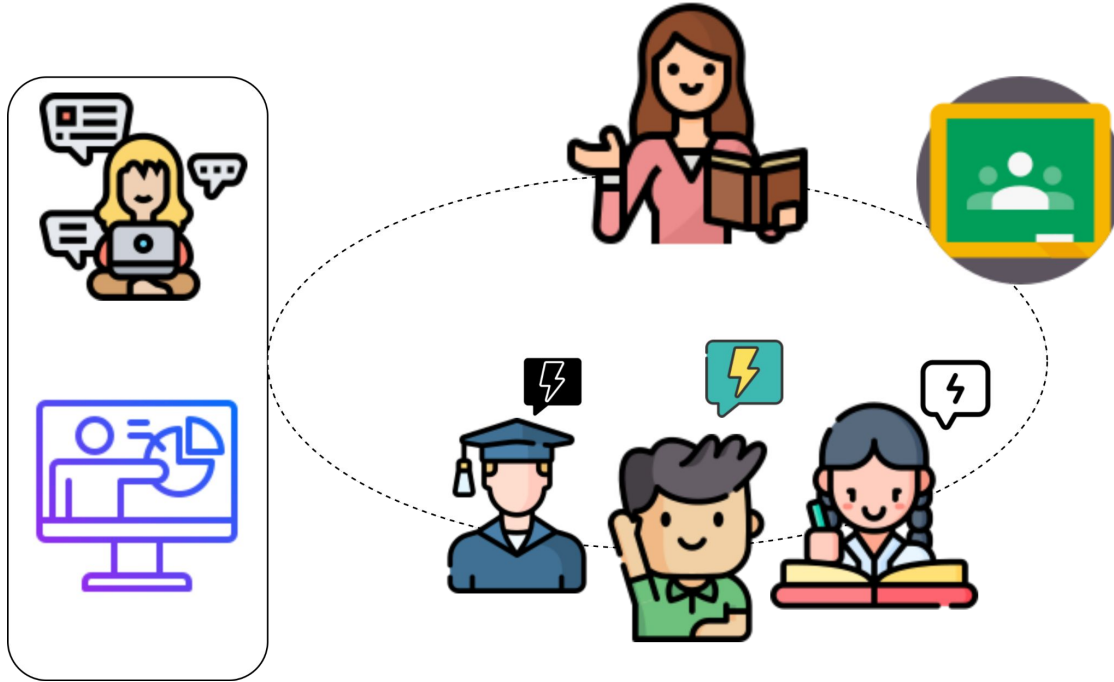
System Architecture



Demonstration

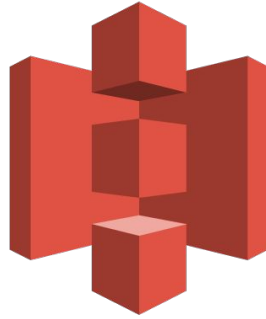


# Introduction



# Resources

- I. AWS S3
- II. AWS EC2
  - A. JQuery
  - B. AjaX
  - C. Python Flask
- III. AWS DynamoDB
- IV. AWS Lambda



S3



EC2



DynamoDB



Lambda



# User Story

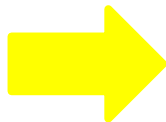




## Polls

Create New Poll

老師端的起始頁面，  
按下create new poll按鈕後  
可以跳轉至建立新投票畫面



建立一個新問題進行投票，  
不同問題之間可以動態新增選項數目



## Create New Poll

Create Date : 20230604

Question : 最喜歡的城市

台北

Add item

桃園

Remove item

台中

Remove item

新竹

Remove item

宜蘭

Remove item

高雄

Remove item

台南

Remove item

Show poll


## 學生進入登入畫面，輸入帳號與密碼進行身分驗證

aws-team2-final.s3.amazonaws.com/AWS-Final.html

應用程式 開放式課程 中原大學 英文學習 程式教學 上課資料 休閒 程式練習競賽 專題 比賽 Facebook - 登入或... Google 其他書籤

aws

Home Architecture Result



# Login Page

David

.....

Submit

[Register/Reset](#)



Show poll :

## 學生以登入身分進入當前正在進行的投票畫面

Question : 最喜歡的城市

Options :

- ☐ 台北
- ☐ 桃園
- ☒ 台中
- ☐ 新竹
- ☐ 宜蘭
- ☐ 高雄
- ☐ 台南

Vote

以單選的方式做勾選，並按下投票按鈕





## Result

Response from AWS Lambda: ['a6b54c20a7b96eeac1a911e6da3124a560fe6dc042ebf270e3676e7095b95652', '台中']

**跳轉至結果畫面，代表投票成功**



## Result

老師端建立完新的問題後,  
跳轉至投票結果顯示頁面

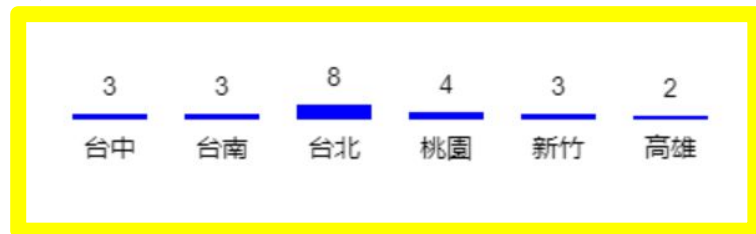
使用DynamoDB trigger來即時更新  
網頁上的票數

1  
台中



## Result

使用長條圖來顯示各選項投數結果

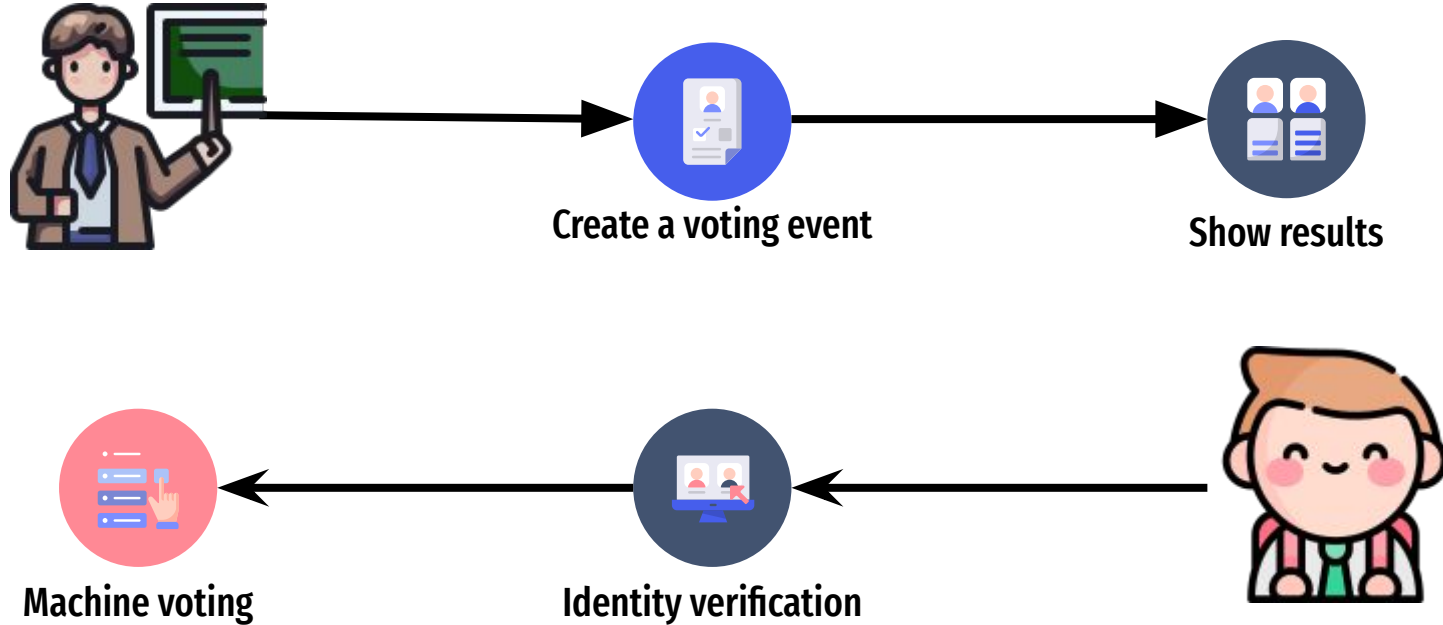


# System Architecture

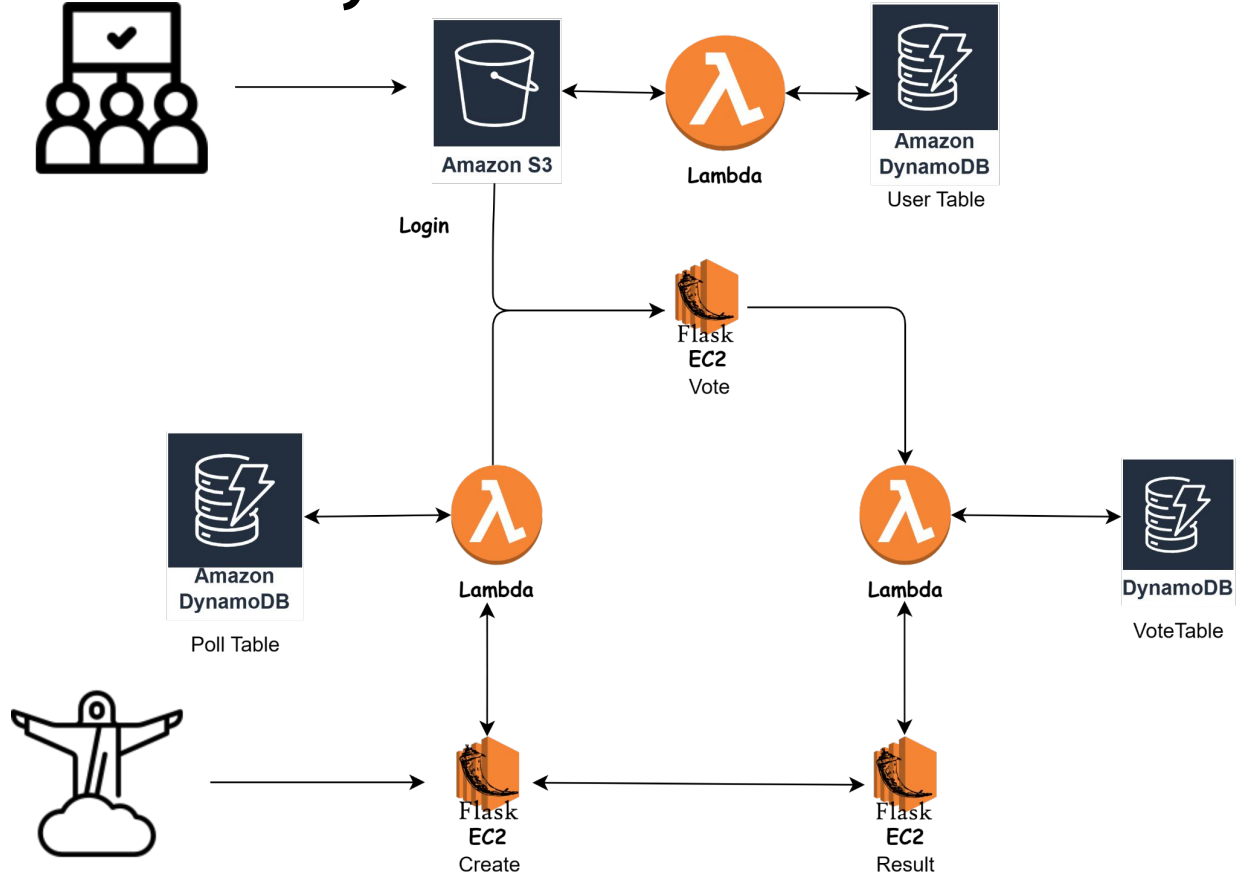


# Voting process infographics

## Voting methods



# System Architecture



```

import urllib.parse
import urllib.request
import json
from flask import Flask, render_template, request, redirect, url_for, jsonify

app = Flask(__name__)

userName = ""

@app.route('/polls/vote', methods=['GET', 'POST'])
def vote_poll():

    global userName

    if request.method == 'POST':

        lambda_endpoint = 'https://d7etplxo89.execute-api.us-east-1.amazonaws.com/default/SaveUserVote'

        vote = request.form.get('option')

        payload = {
            'name': userName,
            'option': vote
        }

        data = json.dumps(payload).encode('utf-8')

        # 構建請求對象
        req = urllib.request.Request(lambda_endpoint, data=data, method='POST')
        req.add_header('Content-Type', 'application/json')

        # 發送請求
        with urllib.request.urlopen(req) as response:
            result = json.loads(response.read())

        # 將結果傳遞給HTML
        return render_template('result.html', result=result)

```

## Student\_app.py

http://18.209.63.6:8080/polls/vote?name=f464d7d71c06e47a535ce441aa202aa717cddeab902a45b0c283aac7a9a090d7

```

else:

    userName = request.args['name']
    lambda_endpoint1 = 'https://c7jganrqgj.execute-api.us-east-1.amazonaws.com/default/TakePoll'

    # 構建請求對象
    req = urllib.request.Request(lambda_endpoint1, method='GET')
    req.add_header('Content-Type', 'application/json')

    # 發送請求
    with urllib.request.urlopen(req) as response:
        result = json.loads(response.read())

    question = result['question']
    options = result['options']

    return render_template('vote_poll.html', question=question, options=options)

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=8080, debug=True)

```

## 在EC2中開啟一個instance, 用來執行學生端與老師端的投票系統與網頁

EC2 > 執行個體 > i-03935ac587b3571d7

### i-03935ac587b3571d7 (pollVM) 的執行個體摘要 [資訊](#)

less than a minute 前更新



連線

執行個體狀態 ▼

動作 ▼

執行個體 ID

i-03935ac587b3571d7 (pollVM)

IPv6 地址

—

主機名稱類型

IP 名稱 : ip-172-31-21-227.ec2.internal

回答私有資源 DNS 名稱

IPv4 (A)

自動指派的 IP 地址

18.209.63.6 [公有 IP]

公有 IPv4 地址

18.209.63.6 [開啟地址](#)

執行個體狀態

執行中

私有 IP DNS 名稱 (僅限 IPv4)

ip-172-31-21-227.ec2.internal

執行個體類型

t2.micro

VPC ID

vpc-02ebcd20171d3b3a7

私有 IPv4 地址

172.31.21.227

公有 IPv4 DNS

ec2-18-209-63-6.compute-1.amazonaws.com | [開啟地址](#)

彈性 IP 地址

—

AWS Compute Optimizer 發現項目

選擇使用 [AWS Compute Optimizer](#) 獲得建議。

| [進一步了解](#)

## DynamoDB

儀表板

資料表

更新設定

探索項目

PartiQL 編輯器

備份

匯出至 S3

從 S3 匯入

預留容量

設定

### ▼ DAX

叢集

子網路群組

參數群組

事件

DynamoDB中的VoteTable -  
負責儲存user與其投票結果的資料表，  
其中user name是紀錄經過雜湊的數值

傳回的項目 (19)



動作 ▼



1

name option

4adfbede2d7d7a502a... 台中

6b9d9fd27f02fb9e92... 台中

c8a83340a9af... 桃園

688787d8ff144c502c... 台北

66e1f9646066d8c32... 台北

4fe1a1a5ebd98b1a4a... 台北

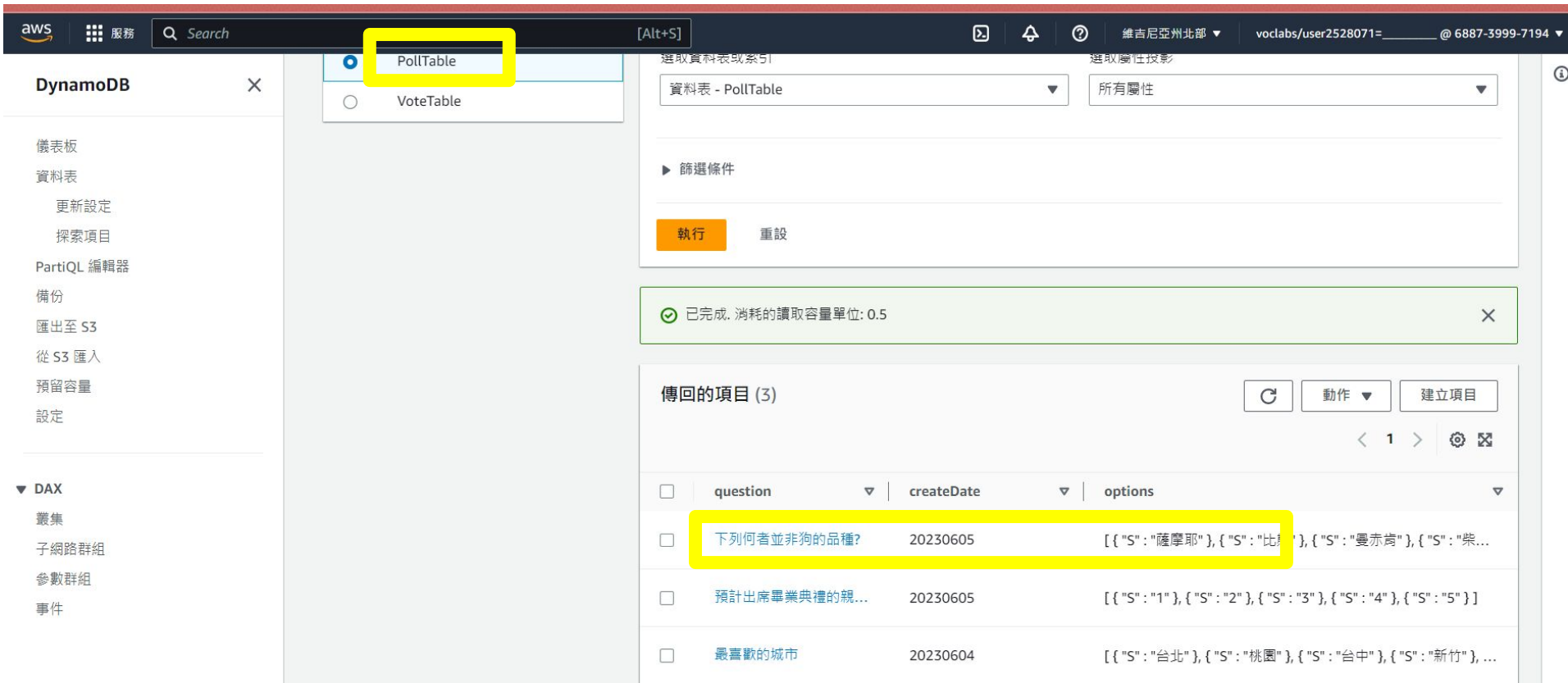
9d9075a75711d3c27... 宜蘭

b662575be5192061a... 台南

一筆資料



# DynamoDB中的PollTable - 負責儲存歷屆投票的題目與選項，並記錄建立時間




The screenshot shows the AWS IAM console interface. The left sidebar contains navigation links for various AWS services. The main content area displays the IAM role configuration for the PollTable. The table is highlighted with a yellow box. The table's schema is also visible, showing columns for question, createDate, and options.

**Table: PollTable**

**Schema:**




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"}, ...]

# 使用 Lambda 從 DynamoDB 中獲取與儲存資料，每一個Lambda 函式會負責一種網頁與資料庫的互動模式(抓取、插入、監控...)

 服務

Search

[Alt+S]

維吉尼亞州北部 ▼

voclabs/user2528071= @ 6887-3999-7194 ▼

AWS Lambda

儀表板

應用程式

函數

▼ 其他資源

程式碼簽署組態

層

複本

▼ 相關的 AWS 資源

Step Functions 狀態機器

Lambda > 函數

函數 (9)

上次已擷取 1 小時前

操作 ▼

建立函式

以標籤和屬性篩選或以關鍵字搜尋

<input type="checkbox"/>	函數名稱 ▼	描述 ▼	套件類型 ▼	執行時間 ▼	上次修改時間 ▼
<input type="checkbox"/>	SaveUserVote	A simple backend (read/write to DynamoDB) with a RESTful API endpoint using Amazon API Gateway.	Zip	Python 3.7	3 小時前
<input type="checkbox"/>	PushPoll	A simple backend (read/write to DynamoDB) with a RESTful API endpoint using Amazon API Gateway.	Zip	Python 3.7	5 小時前
<input type="checkbox"/>	GetAllResult	A simple backend (read/write to DynamoDB) with a RESTful API endpoint using Amazon API Gateway.	Zip	Python 3.7	4 小時前
<input type="checkbox"/>	RedshiftOverwatch	Deletes Redshift Cluster if the count is more than 2.	Zip	Python 3.8	上個月
<input type="checkbox"/>	RoleCreationFunction	Create SLR if absent	Zip	Python 3.8	上個月
<input type="checkbox"/>	trigger	An Amazon DynamoDB trigger that logs the updates made to a table.	Zip	Python 3.7	3 小時前
<input type="checkbox"/>	TakePoll	A simple backend (read/write to DynamoDB) with a RESTful API endpoint using Amazon API Gateway.	Zip	Python 3.7	4 小時前

# 設定Lambda函式的程式1

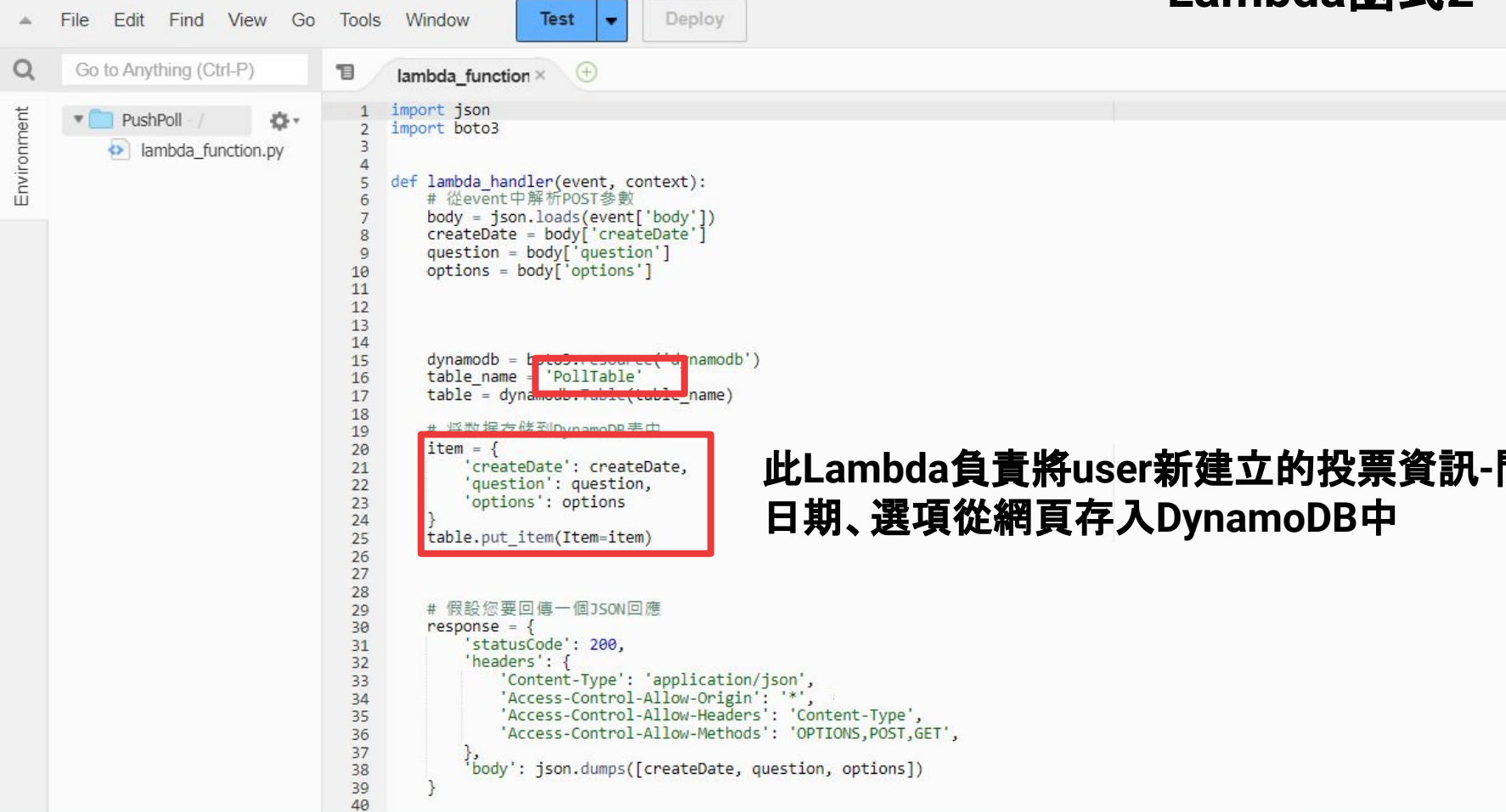
The screenshot shows a code editor with the following Python code:

```
1 import json
2 import boto3
3
4 def lambda_handler(event, context):
5     # 從event中解析POST參數
6     body = json.loads(event['body'])
7     userName = body['name']
8     option = body['option']
9
10
11
12
13
14     dynamodb = boto3.resource('dynamodb')
15     table_name = 'VoteTable'
16     table = dynamodb.Table(table_name)
17
18
19     item = {
20         'name': userName,
21         'option': option
22     }
23     table.put_item(Item=item)
24
25
26
27     # 假設您要回傳一個JSON回應
28     response = {
29         'statusCode': 200,
30         'headers': {
31             'Content-Type': 'application/json',
32             'Access-Control-Allow-Origin': '*',
33             'Access-Control-Allow-Headers': 'Content-Type',
34             'Access-Control-Allow-Methods': 'OPTIONS,POST,GET',
35         },
36         'body': json.dumps([userName, option])
37     }
38
39     return response
```

**設定目標DynamoDB**

**數據的資料結構**

**此Lambda負責從網頁將使用者輸入的投票資料儲存在VoteTable中**



The screenshot shows a code editor with a file named `lambda_function.py`. The code is a Python Lambda function that handles events and stores data in DynamoDB. The function `lambda_handler` takes an event and context as input. It parses the event body into a dictionary, extracting `createDate`, `question`, and `options`. It then initializes a DynamoDB client, creates a table named `PollTable`, and stores the event data as an item in the table. Finally, it returns a JSON response with a 200 status code and headers.

```
1 import json
2 import boto3
3
4 def lambda_handler(event, context):
5     # 從event中解析POST參數
6     body = json.loads(event['body'])
7     createDate = body['createDate']
8     question = body['question']
9     options = body['options']
10
11
12
13
14
15     dynamodb = boto3.resource('dynamodb')
16     table_name = 'PollTable'
17     table = dynamodb.Table(table_name)
18
19     # 將數據存儲到DynamoDB表中
20     item = {
21         'createDate': createDate,
22         'question': question,
23         'options': options
24     }
25     table.put_item(Item=item)
26
27
28
29     # 假設您要回傳一個JSON回應
30     response = {
31         'statusCode': 200,
32         'headers': {
33             'Content-Type': 'application/json',
34             'Access-Control-Allow-Origin': '*',
35             'Access-Control-Allow-Headers': 'Content-Type',
36             'Access-Control-Allow-Methods': 'OPTIONS,POST,GET',
37         },
38         'body': json.dumps([createDate, question, options])
39     }
40
```

此Lambda負責將user新建立的投票資訊-問題、日期、選項從網頁存入DynamoDB中

此Lambda3負責從VoteTable資料庫中提取所有user之投票結果，計算後利用長條圖將票數顯示於網頁中

```

2 import boto3
3
4 def lambda_handler(event, context):
5     dynamodb = boto3.resource('dynamodb')
6     table = dynamodb.Table('VoteTable')
7
8     response = table.scan()
9     items = response['Items']
10
11     response = {
12         'statusCode': 200,
13         'body': json.dumps(items),
14         'headers': {
15             'Content-Type': 'application/json',
16             'Access-Control-Allow-Origin': '*',
17             'Access-Control-Allow-Headers': 'Content-Type',
18             'Access-Control-Allow-Methods': 'OPTIONS,POST,GET',
19         },
20     }
21     return response

```

```

2 import requests
3
4
5
6 print('Loading function')
7
8
9 def lambda_handler(event, context):
10     #print("Received event: " + json.dumps(event, indent=2))
11     for record in event['Records']:
12         print(record['eventID'])
13         print(record['eventName'])
14         print("DynamoDB Record: " + json.dumps(record['dynamodb'], indent=2))
15         url = 'http://18.209.63.6:8888/trigger'
16
17         response = requests.get(url, data=record)
18

```

Lambda4:觸發並更新網頁的票數

aws

服務

Search

[Alt+S]

維吉尼亞州北部

voclabs/user2528071

FileEditFindViewGoToolsWindow

Test

Deploy

Go to Anything (Ctrl-P)

Environment

TakePoll - /

lambda\_function.py

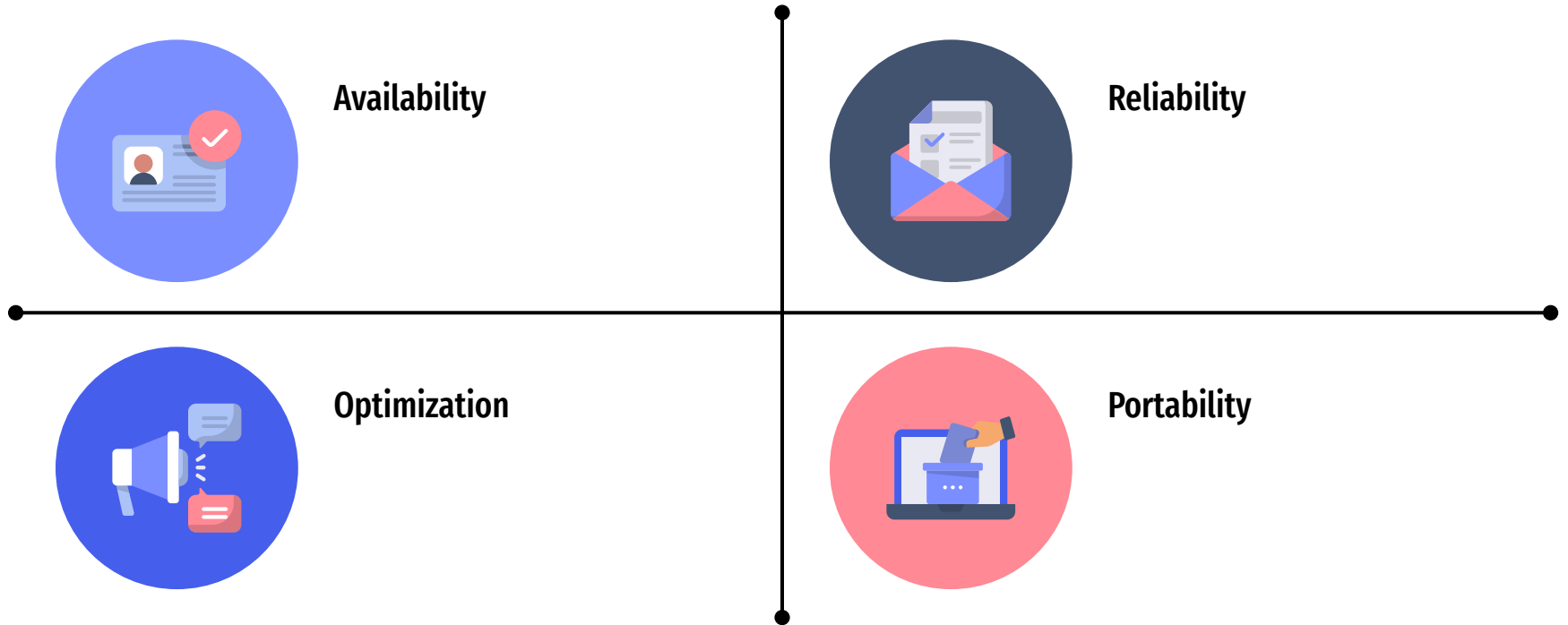
lambda\_function ×

Execution results ×

```
1 import json
2 import boto3
3
4 def lambda_handler(event, context):
5     dynamodb = boto3.resource('dynamodb')
6     table = dynamodb.Table('PollTable') # 替換為你的 DynamoDB 表格名稱
7
8     response = table.scan()
9     items = response['Items']
10
11     # 使用 sorted 函式進行排序，指定欄位名稱為 'YourSortField'
12     sorted_items = sorted(items, key=lambda x: x['createDate'], reverse=True)
13
14     response = {
15         'statusCode': 200,
16         'body': json.dumps(sorted_items[0]),
17         'headers': {
18             'Content-Type': 'application/json',
19             'Access-Control-Allow-Origin': '*',
20             'Access-Control-Allow-Headers': 'Content-Type',
21             'Access-Control-Allow-Methods': 'OPTIONS,POST,GET',
22         },
23     }
24     return response
```

Lambda5:負責從PollTable資料庫中提取問題與投票選項，並顯示於網頁上

# Cloud Properties

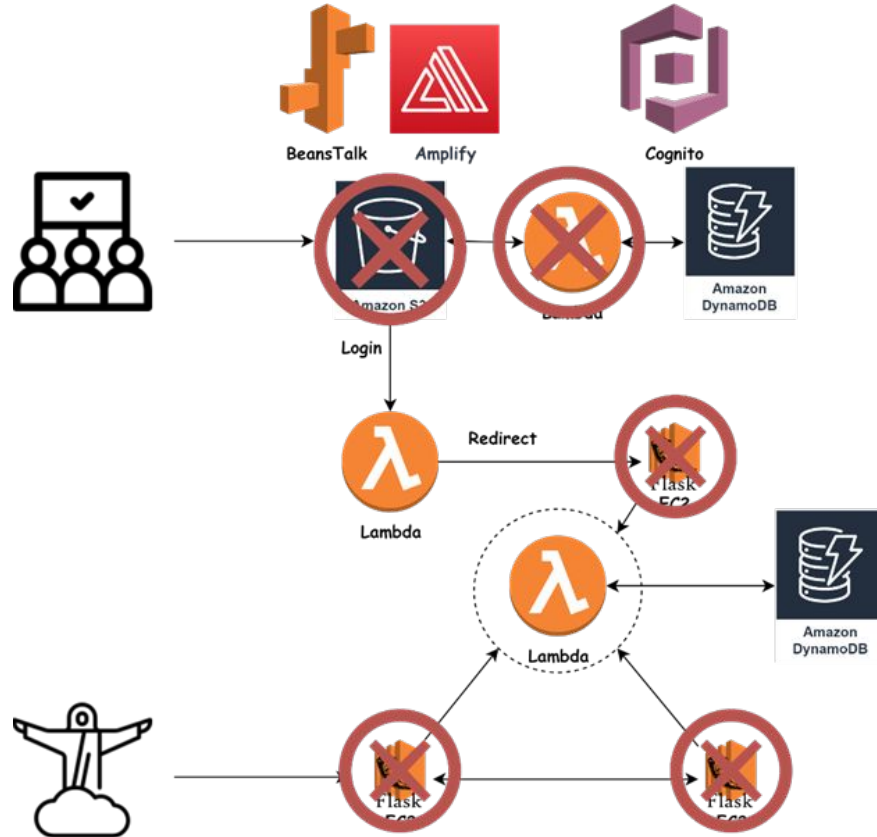


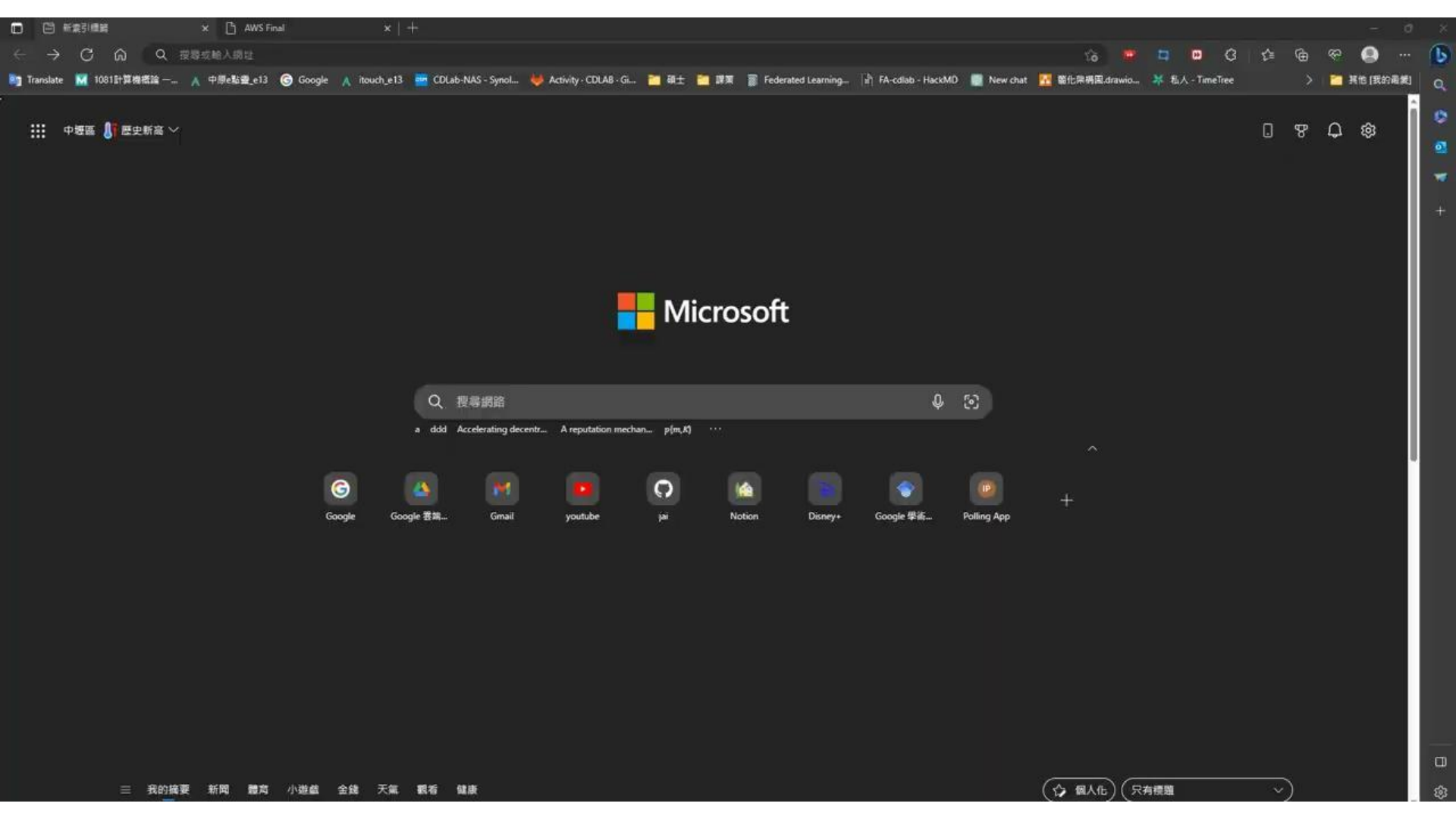
# Lesson Learns from Project & Problems

1. Coding Skill ( Flask、Javascript、 RESTful API)
2. Serverless
  - a. Azure vs. AWS
3. 過程中換題目
  - a. Limited GPU & Inference Type VM
  - b. IAM



# Future Work?





**Thanks for  
listening!**

Learner Lab

0604-test - Lambda

項目 | Amazon DynamoDB Mar

us-east-1.console.aws.amazon.com/dynamodbv2/home?region=us-east-1#item-explorer?table=Account-DynamoDB

更新

aws

服務

Search

[Alt+S]

維吉尼亞州北部

voclabs/user1545578=Lin\_Yan-Hui @ 2799-8503-8536

DynamoDB

儀表板

資料表

更新設定

探索項目

PartiQL 編輯器

備份

匯出至 S3

從 S3 匯入

預留容量

設定

test-table-hui

資料表 - Account-DynamoDB

所有屬性

▶ 篩選條件

執行重設

✔ 已完成. 消耗的讀取容量單位: 0.5

傳回的項目 (9)

↺

動作 ▼

建立項目

< 1 >

<input type="checkbox"/>	accountID ▼	createDate ▼	hash-password ▼
<input type="checkbox"/>	qwer <div><div></div><div></div></div>	2023/5/30	f6f2ea8f45d8a057c9566a33f99474da2e5c6a6604d736121650e2730c6fb0a3
<input type="checkbox"/>	10727211	20000526	ce29c5d5f5578adadab9523dc10db7be0e758eea3fee129afedf9b30ce8bdf14
<input type="checkbox"/>	apple	2023/6/4	3a7bd3e2360a3d29eea436fcfb7e44c735d117c42d1c1835420b6b9942dd4f1b
<input type="checkbox"/>	qaz	2023/6/4	8d6c5597d25eca212ea6c6cacc0a02e247b8c631343a70147cb81374ff72f414
<input type="checkbox"/>	asdf	2023/6/3	f0e4c2f76c58916ec258f246851bea091d14d4247a2fc3e18694461b1816e13b
<input type="checkbox"/>	lemon	2023/6/4	f464d7d71c06e47a535ce441aa202aa717cddeab902a45b0c283aac7a9a090d7
<input type="checkbox"/>	11111111	20230527	ee79976c9380d5e337fc1c095ece8c8f22f91f306ceeb161fa51fecede2c4ba1
<input type="checkbox"/>	123	2023/6/4	a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3
<input type="checkbox"/>	11177035	20230526	526f2a21851696e53b79e754f46462adcfdae94f30541bdc484376f78e28e58a

CloudShell

意見回饋

語言

© 2023, Amazon Web Services, Inc. 或其關係企業。

隱私權

條款

Cookie 喜好設定