In this task, I used 4 modules on AWS:

* DynamoDB (a NoSQL Database, used for data storage)
* IAM (User management, used for creating role in this task)
* Lambda (Used for computing)
* API Gateway (Used for creating RESTFUL API between front-end and back-end)

1. DynamoDB
   1. Create Table

Go https://ap-southeast-1.console.aws.amazon.com/dynamodb/home?region=ap-southeast-1, choose “Dashboard”, then click “Create table”.

In “Create DynamoDB table”,

Table name: **vote**

Primary key: **timestamp** (Number)

Then click “Create”.

1. IAM
   1. Create Role

Go <https://console.aws.amazon.com/iam/home?region=ap-southeast-1#/roles>,

choose “Create role”

In “Create role”, choose “AWS service” as Select type of trusted entity, then choose “Lambda”, then click “Next: Permissions”.

In “Filter” field, search “AmazonDynamoDBFullAccess” and check “AmazonDynamoDBFullAccess”, then click “Next: Review”.

In Review page,

Role name: **lambda-dynamodb-execution-role**

Role description: **Allows Lambda functions to execute dynamoDB**

Then click “Create role”.

1. Lambda
   1. Create Function for get count

Go <https://ap-southeast-1.console.aws.amazon.com/lambda/home?region=ap-southeast-1#/functions>,

Choose “Functions”, then click “Create function”.

In Create function, choose “Author from scratch”.

Name: **get\_count**

Runtime: **Node.js 4.3**

Role: Choose an existing role

Existing role: **lambda-dynamodb-execution-role**

In Function code,

Copy the content of getCount.js into index.js

* 1. Create Function for voting

Go <https://ap-southeast-1.console.aws.amazon.com/lambda/home?region=ap-southeast-1#/functions>,

Choose “Functions”, then click “Create function”.

In Create function, choose “Author from scratch”.

Name: **vote**

Runtime: **Node.js 4.3**

Role: Choose an existing role

Existing role: **lambda-dynamodb-execution-role**

In Function code,

Copy the content of vote.js into index.js

1. API Gateway
   1. Create API

Go https://ap-southeast-1.console.aws.amazon.com/apigateway/home?region=ap-southeast-1#/apis,

Choose “APIs”, then click “Create API”.

In Create new API page, choose “New API”.

In Settings,

API name: **VoteCount**

Click “Create API”

* 1. Configure API for get\_count

Choose **“VoteCount”**, then choose “Resources”, then choose “Actions”.

In “Actions”, choose “Create Resource”.

Resource Name: **getCount**

Resource Path: /**getcount**

Click “Create Resource”.

In “**/getcount”**, choose “Create Method”.

Under “**/getcount”**, choose “GET”.

In Setup page,

Integration type: Lambda Function

Use Lambda Proxy integration: checked

Lambda Region: **ap-southeast-1**

Lambda Function: **get\_count**

* 1. Configure API for vote

Choose **“VoteCount”**, then choose “Resources”, then choose “Actions”.

In “Actions”, choose “Create Resource”.

Resource Name: **vote**

Resource Path: /**vote**

Click “Create Resource”.

In “**/vote”**, choose “Create Method”.

Under “**/vote”**, choose “GET”.

In Setup page,

Integration type: Lambda Function

Use Lambda Proxy integration: checked

Lambda Region: **ap-southeast-1**

Lambda Function: **vote**

* 1. Deploy API

Choose **“VoteCount”**, then choose “Resources”, then choose “Actions”.

In “Actions”, choose “Deploy API”.

Deployment stage: [New Stage]

Stage name: **demo**

Click “Deploy”.

Now in **VoteCount** -> Stages, you can see the Invoke URL.

1. Usage
   1. Get count

There is a parameter named id.

When **id = 0**, it is getting the sum of vote count of individual candidate in last 10 minutes.

https://*{restapi\_id}*.execute-api.*{region}*.amazonaws.com/*{stage\_name}*/getcount?id=0

e.g. {"message":"**Last 10min total: 13**","input":{"resource":"/getcount","path":"/getcount","httpMethod":"GET","headers":{"accept":"text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8","accept-encoding":"gzip, deflate, br","accept-language":"en-US,en;q=0.9,zh-TW;q=0.8,zh;q=0.7,zh-CN;q=0.6"…

When **id = 1**, it is getting the current cumulative vote count for candidate 1(薯片), and id = 2 for candidate 2(林林), id = 3 for candidate 3(正氣).

https://*{restapi\_id}*.execute-api.*{region}*.amazonaws.com/*{stage\_name}*/getcount?id=1

e.g. {"message":"**薯片(1): 23 votes**","input":{"resource":"/getcount","path":"/getcount","httpMethod":"GET","headers":{"accept":"text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8","accept-encoding":"gzip, deflate, br","accept-language":"en-US,en;q=0.9,zh-TW;q=0.8,zh;q=0.7,zh-CN;q=0.6"…

* 1. Vote

There is a parameter named id.

When **id = 1**, it is for voting to candidate 1(薯片), and id = 2 for candidate 2(林林), id = 3 for candidate 3(正氣).

https://*{restapi\_id}*.execute-api.*{region}*.amazonaws.com/*{stage\_name}*/vote?id=1

e.g. {"message":"**Success voted**","input":{"resource":"/vote","path":"/vote","httpMethod":"GET","headers":{"accept":"text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8","accept-encoding":"gzip, deflate, br","accept-language":"en-US,en;q=0.9,zh-TW;q=0.8,zh;q=0.7,zh-CN;q=0.6"