**SetUp AchieveHub V2 (จากอันเก่า) สำหรับนำลง LearnerLab โดยเฉพาะ**

**Part 1 : อธิบายทรัพยากรทั้งหมดที่ต้องเตรียมเข้า Lab และต้องเอามาจากไหน**

สามารถโหลดได้จาก github : <https://github.com/TEEiEiYAAA/CS361-Project>

1. **S3 buckets** : ไว้เก็บ static web ที่กลายเป็น dynamic ไปแล้ว

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ตัวอักษร, จำนวน

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

* **achievehub-v2-picture** : ไว้เก็บรูปภาพที่เป็นปก activity (ไม่จำเป็นต้องแยกก็ได้) -> folder ภาพกิจกรรม รูปภาพประกอบด้วย ข้อความ, ตัวอักษร, ภาพหน้าจอ, เครื่องหมาย

  เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

รูปภาพประกอบด้วย ข้อความ, จำนวน, ตัวอักษร, ไลน์

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

Setting : เปิด ACL everyone can read, ติ้ก block public access ออก (ให้เข้าถึงผ่านinternetได้)

* **achieve-hub-ver2** : เป็นหน้าเว็บ static ของเรา -> folder ใน github ตามรูปด้านล่างนี้รูปภาพประกอบด้วย ภาพหน้าจอ, ข้อความ, ตัวอักษร, จำนวน

  เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้องรูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, ตัวอักษร

  เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

ที่เก็บใน S3

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, จำนวน, ซอฟต์แวร์

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

Setting : เปิด ACL everyone can read, ติ้ก block public access ออก (ให้เข้าถึงผ่านinternetได้) \*\*เปิด Static Web Hosting ด้วย\*\* รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ตัวอักษร, ไลน์

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

Link : [http://achieve-hub-ver2.s3-website-us-east-1.amazonaws.com](http://achieve-hub-ver2.s3-website-us-east-1.amazonaws.com )

* **\*ไม่ต้องทำตามจะดีกว่า\* firstone-csv** : เก็บ csv จากตาราง dynamo ของ project ก่อน ไว้ใช้กับ dynamo ปัจจุบันในขั้นตอนตอนแรก เราจะไปทำที่ dynamo แล้วกดตัวเลือก import from s3 (ไม่แนะนำวิธีนี้ เพราะทุกคอลัมน์มันจะมองว่าเป็น string ทั้งหมดซึ่งไม่ถูกต้อง ทำให้ต้องเข้าไปแก้ schema ทีละตารางอยู่ดี) -> Folder csvInDynamo 

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, จำนวน, ซอฟต์แวร์

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

1. Dynamo : ตาราง database ของระบบ มีทั้งหมด 9 ตารางดังนี้ การตั้ง Partition Key และ Sort Key ต้องเป็นไปตามนี้เท่านั้น -> Folder csvInDynamo 

รูปภาพประกอบด้วย ข้อความ, จำนวน, ภาพหน้าจอ, ตัวอักษร

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

โดยมาจากการนำเข้าผ่าน s3 ตามรูป:

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, จำนวน, ตัวอักษร

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

* ตารางที่ 1 Activities : รายละเอียดกิจกรรมต่างๆรูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, จำนวน, ตัวอักษร

  เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง
* ตารางที่ 2 ActivityParticipations : เมื่อนักศึกษากดเข้าร่วมกิจกรรม ข้อมูลจะมาเก็บในตารางนี้

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ตัวอักษร, ไลน์

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

* ตารางที่ 3 Assessments : บันทึกข้อมูลการทำแบบประเมินกิจกรรม

รูปภาพประกอบด้วย ข้อความ, ตัวอักษร, ภาพหน้าจอ, จำนวน

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

* ตารางที่ 4 CompletedSkills : บันทึกสกิลที่นักศึกษาผ่านการทดสอบแล้ว

รูปภาพประกอบด้วย ข้อความ, ตัวอักษร, ภาพหน้าจอ, ไลน์

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

* ตารางที่ 5 QuizAttempts : บันทึกข้อมูลการทำแบบทดสอบของนักศึกษาในสกิลนั้นๆ

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, จำนวน, ตัวอักษร

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

* ตารางที่ 6 QuizQuestions : เก็บคำถามและคำตอบของแบบทดสอบนั้นๆ

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, จำนวน, ตัวอักษร

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

* ตารางที่ 7 Skills : เก็บรายละเอียดสกิลทั้งหมดที่มีในระบบ

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, จำนวน, ตัวอักษร

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

* ตารางที่ 8 Students : ข้อมูลนักศึกษา

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ตัวอักษร, จำนวน

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

ตารางนี้จะมี index เพิ่มมาด้วย \*อย่าลืมตั้งค่า\*

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ตัวอักษร, ไลน์

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

* ตารางที่ 9 Users : ข้อมูลที่ใช้ login (username&password)

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, จำนวน, ตัวอักษร

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

1. Lambda function : โค้ดที่เรียกการทำงานให้เป็นไปตามที่ user ต้องการ (มี 15 ฟังก์ชัน)

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, จำนวน, ไลน์

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

โหลดโค้ดได้จาก github folder นี้ รูปภาพประกอบด้วย ข้อความ, ตัวอักษร, ภาพหน้าจอ, กราฟิก

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

1. API Gateway : เป็นตัวเชื่อมระหว่าง lambda กับ dynamo ซึ่งถูกเรียกใช้ใน ไฟล์ js ที่ถูกเก็บอยู่ใน s3 เป็นฝั่ง frontend (มี 15 เส้น)

* ทุก 1 api จะเชื่อมต่อกับ 1 lambda function ไม่ซ้ำกัน
* สร้างแบบ REST API

รูปภาพประกอบด้วย ข้อความ, ซอฟต์แวร์, ไอคอนคอมพิวเตอร์, จำนวน

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

รูปภาพประกอบด้วย ข้อความ, อิเล็กทรอนิกส์, ภาพหน้าจอ, โทรศัพท์มือถือ

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้องรูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, โทรศัพท์มือถือ, ตัวอักษร

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้องรูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, โทรศัพท์มือถือ, ตัวอักษร

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้องรูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ตัวอักษร, โทรศัพท์มือถือ

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้องรูปภาพประกอบด้วย ข้อความ, อิเล็กทรอนิกส์, ภาพหน้าจอ, ตัวอักษร

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

**รายละเอียดว่าแต่ละเส้นเชื่อมกับฟังก์ชันไหนเพื่อทำอะไร และถูกเรียกใช้ใน JS ใด ให้ดูตามนี้ :**

1. /activities - GET – getActivities ดึงเอาข้อมูลกิจกรรมมาแสดงผลในหน้ากิจกรรม
2. /activities/{activityId} - GET - getActivityDetail รายละเอียดเวลากดเข้าไปแต่ละกิจกรรม
3. /activities/{activityId}/assessment - POST – submitAssesment บันทึกรายละเอียดการทำแบบประเมินของกิจกรรมนั้นๆ
4. /activities/{activityId}/participants - GET - getActivityParticipants --> JS advisor-overall ดูจำนวนผู้เข้าร่วม ใครบ้าง มีactivities ยังไง
5. /activities/register - POST - registerActivity --> JS detailed-activities นักศึกษากดสมัครเข้าร่วมกิจกรรม
6. /activities/verify-qr - POST - verifyActivityCode --> JS my-activities.js สำหรับยืนยันเข้าร่วมกิจกรรม ให้กรอกโค้ด มีกำหนดแพทเทิร์นความถูกต้องด้วย
7. /advisors/{advisorId}/students - GET - getAdvisorStudents --> JS advisor-dashboard.js ดึงข้อมูลของนักศึกษาในที่ปรึกษามาแสดงผล
8. /login - POST - login --> JS Login --> JS login.js ล็อกอินไปหน้าตามบทบาท
9. /quiz/questions/{skillId} - GET - getQuizQuestions --> JS quiz.js เป็นแบบทดสอบ จะดึงคำถามและช้อยมาแสดงผล
10. /quiz/submit - POST - submitQuizAnswers --> JS quiz.js
11. /requiredSkills/{yearLevel} - GET -getRequiredSkills --> JS current-skill.js ดึงทักษะบังคับของชั้นปีมา
12. /skills/all - GET - getAllSkills --> JS current-skill.js ดึงสกิลทั้งหมดของระบบมาวิเคราะห์เพื่อแสดงผลให้ถูกต้องตามชั้นปี (สกิลบังคับ)
13. /students/{studentId}/activities - GET - getStudentActivities --> JS my-activities.js ดึงกิจกรรมที่นศคนนั้นเข้าร่วมมา
14. /students/{studentId}/info - GET - getStudentInfo --> JS student-dashboard.js แสดงรายละเอียดข้อมูลส่วนตัวของนศ.
15. /students/{studentId}/skills - GET – getStudentSkills ดึงสกิลทั้งหมดของนศที่ได้รับมา
16. Cloudwatch : ใช้ในการ monitor ดู error ที่เกิดขึ้นจาการเรียกใช้ api

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, จำนวน, ตัวอักษร

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

ถ้ามีการเรียกใช้มันก็จะมาแสดงผลในนี้ **-> อันนี้ไม่จำเป็นต้องเข้าไปทำไร** ไว้ดู log เฉยๆ

**Part 2 : วิธีการสร้างและติดตั้งใน AWS Learner Lab**

1. **S3** : เราจะมี bucket ทั้งหมด 3 bucket

-bucket ที่ไว้ hosting web

-bucket สำหรับเก็บรูปหน้าปกของกิจกรรม

-bucket สำหรับเก็บตาราง database

**สำหรับ Bucket Hosting Web และ bucket สำหรับเก็บรูปหน้าปกของกิจกรรม :** ตั้งชื่ออะไรก็ได้ และตั้งค่าตามนี้

รูปภาพประกอบด้วย ข้อความ, ไลน์, จำนวน, ตัวอักษร

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ตัวอักษร, ไลน์

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง รูปภาพประกอบด้วย ข้อความ, ตัวอักษร, จำนวน, ภาพหน้าจอ

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

อย่างอื่นปล่อย default และ กด create bucket

หลังจากมันสร้างเสร็จ สำหรับ **Bucket Hosting Web** อย่าลืมมาเปิด Static website hosting (Properties > Static website hosting)

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ไลน์, ตัวอักษร

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง รูปภาพประกอบด้วย ข้อความ, ซอฟต์แวร์, จำนวน, ไลน์

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

จากนั้นกด Save Changes

ปล.ถ้าไม่เปิด ACL สามารถตั้ง Bucket Policy เลยก็ได้ อาจจะสะดวกกว่า คือให้เข้าถึงผ่าน internet ได้ทั้งbucket เมื่อมีการแก้ไขไฟล์ อัพโหลดไฟล์ใหม่มันก็จะเข้าถึงได้ทันทีโดยที่ไม่ต้อง ให้สิทธิ read ผ่าน ACL ทุกครั้งที่อัพโหลดไฟล์ใหม่

ส่วน **bucket สำหรับเก็บตาราง database จะตั้งชื่ออะไรก็ได้ ตอนสร้างไม่จำเป็นต้องเปิด ACL หรือ Block all public access** คือตั้งแค่ชื่อ ที่เหลือปล่อยค่า default ไม่ต้องกดอะไรทั้งนั้น > Create Bucket ตามรูปด้านล่าง**รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, จำนวน, ตัวอักษร

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง** **รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, จำนวน

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง**

1. DynamoDB

ต่อจากด้านบน เมื่อนำตารางเข้า s3 แล้ว รูปแบบ folder และ ไฟล์ต่างๆ จะต้องมี path ที่ถูกต้องครบถ้วนตามในรูป (อย่าลืมเช็ค) ในรูปโฟลเดอร์อาจจะเป็นเลขแปลกๆ แต่ที่เตรียมไว้ให้จะเป็นชื่อตารางเลย 9 โฟลเดอร์

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ไลน์, ตัวอักษร

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้องรูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, จำนวน, ตัวอักษร

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้องรูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, จำนวน, ตัวอักษร

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้องรูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, จำนวน, ตัวอักษร

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

วิธีนำเข้า dynamo :

ไปที่ dynamo > import from s3 ทำตามรูป

รูปภาพประกอบด้วย ข้อความ, ซอฟต์แวร์, จำนวน, ไลน์

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้องรูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, จำนวน

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

อย่าลืมตั้งค่า Partition Key, Sort Key ให้ถูกต้องในแต่ละตาราง ตามข้อมูลที่ให้ไว้ใน Part 1

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, จำนวน, ตัวอักษร

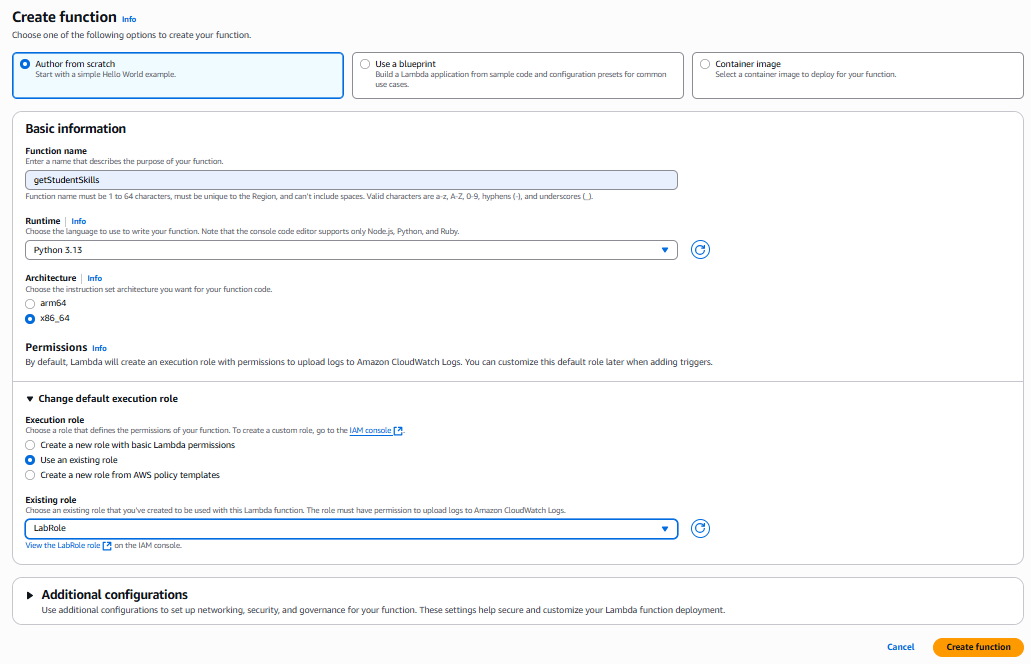
เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้องรูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, จำนวน, ไลน์

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

ปล. สามารถนำเข้าได้ทีละตารางเท่านั้น โดยจะส่งไฟล์ไว้ให้ เป็นรูปแบบ dynamo json (ดีกว่า csvมากๆ type ถูกต้อง)

1. Lambda

ไปที่ Lambda > Functions > Create function โดยตั้งชื่อฟังก์ชันตามข้อมูลที่ไว้ใน Part 1 และตั้งค่าตามรูป



เมื่อกด create function แล้ว ให้เลื่อนลงมาข้างล่าง ที่ code หลังจากนั้นอย่าลืมกด Deploy

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ซอฟต์แวร์, ไอคอนคอมพิวเตอร์

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

1. API Gateway

ไปที่ API Gateway > APIs > Create API > REST API > Build ตั้งชื่ออะไรก็ได้

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ตัวอักษร

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, จำนวน, ซอฟต์แวร์

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

หลังจากกด Create API แล้ว มันจะเข้ามาหน้านี้ รูปภาพประกอบด้วย ข้อความ, ซอฟต์แวร์, จำนวน, ไอคอนคอมพิวเตอร์

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

กด Create Resource ตั้งค่าตามนี้ \*อย่าลืมติ้กCORS\* แล้วกดสร้าง

รูปภาพประกอบด้วย ข้อความ, ไลน์, ภาพหน้าจอ, ตัวอักษร

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ตัวอักษร, จำนวน

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

จากนั้นให้กด Create Method จะเข้ามาหน้านี้ ให้เลือก GET/POST ดูตามข้อมูลใน Part 1

\*อย่าลืมติ้ก Lambda proxy integration\* เลือกฟังก์ชันที่ถูกต้อง แล้ว Create Method เลย

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ขนาน, แสดง

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

ผลลัพธ์ทีได้จะเป็นแบบนี้ :

รูปภาพประกอบด้วย ข้อความ, ตัวอักษร, จำนวน, ไลน์

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

จากนั้นให้กดที่ /activities > Enable CORS

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, ตัวอักษร, จำนวน

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

แล้วให้ติ้กตามรูป > save

รูปภาพประกอบด้วย ข้อความ, ซอฟต์แวร์, หน้าเว็บ, ไอคอนคอมพิวเตอร์

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

จากนั้นให้กดที่ GET แล้วกด Deploy API

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, จำนวน, ตัวอักษร

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

เมื่อเราจะ deploy ครั้งแรก ให้เลือก Newstage ตั้งชื่ออะไรก็ได้ ในที่นี้จะตั้งว่า prod แล้ว deploy

ผลลัพธ์ที่ได้มันจะมาแสดงหน้า Stage :

รูปภาพประกอบด้วย ข้อความ, ภาพหน้าจอ, จำนวน, ซอฟต์แวร์

เนื้อหาที่สร้างโดย AI อาจไม่ถูกต้อง

ปล.Deploy เป็นสิ่งสุดท้ายที่ต้องทำหลังตั้งค่าอะไรเสร็จ ต้องกด Deploy ทีละเส้นเมื่อสร้างเสร็จ และต้อง deploy ไปที่ stage เดียวกัน คือ prod เท่านั้น เมื่อทำจนครบผลลัพธ์ต้องได้ตามรูปใน Part 1

------------------------------------- จบ -------------------------------------