

เว็บแอปพลิเคชันสำหรับการจัดการทรัพยากรและค่าใช้จ่ายใน AWS
(WEB APPLICATION FOR RESOURCE AND COST MONITORING IN
AWS)

โดย

นายชนายุส หนูเอียด

CHANAYUS NUIAD

นายภูชิต รุ่งระวี

POOHCID RUGRAWI

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์ ดร.ลภัส ประดิษฐ์ทัศนีย์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 1 ปีการศึกษา 2564

**WEB APPLICATION FOR RESOURCE AND COST MONITORING IN
AWS**

**CHANAYUS NUIAD
POOHCID RUGRAWI**

**A PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
1/2021**

COPYRIGHT 2021

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ใบรับรองปริญญาโท ประจำปีการศึกษา 2564

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เว็บแอปพลิเคชันสำหรับการจัดการทรัพยากรและค่าใช้จ่ายใน AWS

WEB APPLICATION FOR RESOURCE AND COST
MONITORING IN AWS

ผู้จัดทำ

1. นายชญาต หนูเอียด

รหัสนักศึกษา 61070034

2. นายภูชิต รุ่งระวี

รหัสนักศึกษา 61070166

.....อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร.ลภัส ประดิษฐ์ทัศนีย์)

ใบรับรองโครงการ (PROJECT)

เรื่อง

เว็บแอปพลิเคชันสำหรับการจัดการทรัพยากรและค่าใช้จ่ายใน AWS

WEB APPLICATION FOR RESOURCE AND COST MONITORING IN
AWS

นายชนายุส หนูเอียด

รหัสนักศึกษา 61070034

นายภูชิต รุ่งระวี

รหัสนักศึกษา 61070166

ขอรับรองว่ารายงานฉบับนี้ ข้าพเจ้าไม่ได้คัดลอกมาจากที่ใด

รายงานฉบับนี้ได้รับการตรวจสอบและอนุมัติให้เป็นส่วนหนึ่งของ

การศึกษาวិชาโครงการ หลักสูตรวิทยาศาสตร์บัณฑิต (เทคโนโลยีสารสนเทศ)

ภาคเรียนที่ 1 ปีการศึกษา 2564

.....

(นายชนายุส หนูเอียด)

.....

(นายภูชิต รุ่งระวี)

หัวข้อโครงการ	เว็บแอปพลิเคชันสำหรับการจัดการทรัพยากรและค่าใช้จ่ายใน AWS	
นักศึกษา	นายชนายุส หนูเอียค	รหัสนักศึกษา 61070034
	นายภูชิต รุ่งระวี	รหัสนักศึกษา 61070166
ปริญญา	วิทยาศาสตร์บัณฑิต	
สาขาวิชา	เทคโนโลยีสารสนเทศ	
ปีการศึกษา	2562	
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.ลภัส ประดิษฐ์ทัศนีย์	

บทคัดย่อ

การประมวลผลระบบคลาวด์เป็นเทคโนโลยีที่มีความสำคัญต่อการติดต่อสื่อสารผ่านอินเทอร์เน็ตในการกระจายข้อมูลไปยังส่วนต่าง ๆ ของโลกโดยที่มีการจัดการที่ง่ายและขยายตัวตามความต้องการได้รวดเร็ว ซึ่งจะมีผู้ให้บริการติดตั้งศูนย์กลางข้อมูลกระจายไปทั่วโลก การเรียนรู้โครงสร้างพื้นฐานของระบบสารสนเทศในรูปแบบคลาวด์เป็นการเรียนรู้ที่ง่าย ประหยัด และสามารถไปต่อยอดไปใช้งานในธุรกิจจริงได้ง่าย

Amazon Web Services เป็นหนึ่งในผู้ให้บริการคลาวด์และเป็นแพลตฟอร์มคลาวด์ที่นำมาใช้ในวิชาการพัฒนาคloudแอปพลิเคชันระดับองค์กรและวิชาเทคโนโลยีกลุ่มเมฆเบื้องต้นของคณะเทคโนโลยีสารสนเทศ สจล. โดยมี ผศ. ดร.ลภัส ประดิษฐ์ทัศนีย์ เป็นอาจารย์ประจำวิชาที่ได้พบเจอปัญหาในการสอนวิชาคลาวด์ในส่วนที่เป็นข้อจำกัดของ AWS Management Console ที่ให้การใช้งาน IAM User ให้กับกลุ่มโปรเจกต์นักศึกษาไปใช้งาน โดยปัญหาหลัก ๆ ที่พบเจอคือการเข้าถึงและจัดการทรัพยากรหรือการบริการต่างๆที่นักศึกษาสร้างไว้ทั้งหมดค่อนข้างใช้เวลานานและทรัพยากรที่สร้างขึ้นไม่มีการระบุว่าเป็นของ IAM User คนไหน จึงเกิดเป็นโปรเจกต์เว็บแอปพลิเคชันสำหรับการจัดการทรัพยากรที่จะมีการวางระบบเบื้องหลังในการติดตามการกระทำของ IAM User มีการระบุเจ้าของทรัพยากรที่สร้างขึ้น การลบทรัพยากรที่ง่าย และระบบเสริมที่มาชัดเจนในส่วนที่ AWS Management Console ไม่มีที่จะช่วยอาจารย์ประจำวิชาและนักศึกษาในการลดปัญหาในการทำงานได้ดียิ่งขึ้น

Project Title	Web Application for Resource and Cost Monitoring in AWS	
Students	Mr. Chanayus Nuiad	Student ID 61070034
	Mr. Poohcid Rugravi	Student ID 61070166
Degree	Bachelor of Science	
Program	Information Technology	
Academic Year	2021	
Advisor	Assoc. Prof. Dr. Lapat Praditthasanee	

ABSTRACT

Cloud computing is a critical technology for internet communication and information distribution around the world. Cloud computing allows easy management and rapid expansion on demand, which has service providers located around the world. The infrastructure of cloud-based information systems is simple to learn, economical, and easily extended to use in business.

Amazon Web Services (AWS) is a cloud provider and cloud platform used in enterprise cloud application development, which is an introductory cloud technology course offered by the Faculty of Information Technology at KMITL. by Asst. Prof. Dr. Lapat Praditthasanee is an instructor who has struggled to teach cloud subjects due to the limitations of the AWS Management Console. The main problem encountered is that accessing and managing all student-created resources or services takes a long time and resources to create. It does not specify which IAM user it belongs to. Therefore, we developed a web application for resource management in AWS that includes a background event tracking system for IAM users. The owner of the created resource is specified. Easy resource removal and a complementary system that compensates for parts that the AWS Management Console does not have. to help course instructors and students reduce problems with work.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้จะประสบผลสำเร็จได้เป็นอย่างดี ด้วยความกรุณาของอาจารย์ที่ปรึกษาผู้ช่วยศาสตราจารย์ ดร.ลภัส ประดิษฐ์ทัศนีย์ ที่ให้ข้อมูลอันสำคัญในการวิจัย ทรัพยากร และคำชี้แนะต่าง ๆ อีกทั้ง คอยติดตามความก้าวหน้าของโครงการอยู่เป็นระยะ ๆ เพื่อให้ปริญญานิพนธ์ฉบับนี้เกิดประสบผลสำเร็จที่จะมาซึ่งในการแก้ไขปัญหาที่มีอยู่และสร้างเป็นผลิตภัณฑ์ที่จะถูกนำไปใช้งานจริงในการเรียนการสอนในโอกาสต่อไป

ขอขอบคุณคณาจารย์และเจ้าหน้าที่ในคณะเทคโนโลยีสารสนเทศ ที่เป็น กรรมการผู้ช่วยตรวจสอบปริญญานิพนธ์ฉบับนี้ให้ประสบผลสำเร็จ ตลอดจนให้ทรัพยากรในการวิจัยและสร้างผลิตภัณฑ์ขึ้นมา ก่อให้เกิดประโยชน์แก่คณะเทคโนโลยีสารสนเทศ อีกทั้งนำไปประยุกต์ใช้งานในการทำงานในอนาคตได้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	V
สารบัญรูป.....	VI

บทที่

1. บทนำ.....	1
1.1 ความเป็นมาของปัญหา.....	1
1.2 วัตถุประสงค์.....	1
1.3 ขอบเขตของระบบ.....	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.5 ศัพท์นิยามเฉพาะที่ใช้ในโครงการ.....	2
2. การทบทวนวรรณกรรม.....	3
2.1 การบริการและทรัพยากรใน AWS เพื่อใช้ในการสร้างแท็กแบบอัตโนมัติ.....	3
2.2 เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์.....	5
3. วิธีการดำเนินการวิจัย.....	7
3.1 การวิเคราะห์และออกแบบระบบ.....	7
3.2 แผนดำเนินงาน.....	9
3.3 หลักการทำงานของระบบ.....	11
บรรณานุกรม.....	19

สารบัญตาราง

หน้า

ตารางที่

3.1 ปฏิทินแผนดำเนินงาน.....	9
-----------------------------	---

สารบัญภาพ

รูปที่

หน้า

2.1 สัญลักษณ์ AWS CloudTrail	3
2.2 สัญลักษณ์ AWS CloudWatch.....	3
2.3 สัญลักษณ์ AWS Lambda.....	4
2.4 สัญลักษณ์ IAM	4
2.5 สัญลักษณ์ Boto 3	5
2.6 สัญลักษณ์ NEXT.JS.....	5
2.7 สัญลักษณ์ MongoDB	6
3.1 Use Case Diagram.....	8
3.2 Auto-Tagging Solution Workflow	11
3.3 โค้ด Boto 3 (1)	12
3.4 โค้ด Boto 3 (2)	12
3.5 อธิบายการเปิดใช้งาน AWS CloudTrail	12
3.6 อธิบายการเพิ่ม Event Rule ใน AWS CloudWatch	13
3.7 อธิบายการเพิ่ม Event ที่ต้องการจะดักผ่าน JSON	13
3.8 อธิบายการใส่ Targets Trigger Function ใน CloudWatch.....	14
3.9 ฟังก์ชันโค้ด AWS SDK (boto3).....	14
3.10 การสร้าง API Gateway เพื่อเรียกใช้งานฟังก์ชัน (1).....	15
3.11 การสร้าง API Gateway เพื่อเรียกใช้งานฟังก์ชัน (2).....	15
3.12 หน้า Log in.....	16
3.13 หน้า Dashboard ของอาจารย์ผู้สอน	16

3.14 หน้าแสดงผล Resource ที่กำลังใช้งานในแต่ละ Region	17
3.15 หน้าแสดง IAM User ทั้งหมด	17
3.16 หน้ารายละเอียดของ Resource	18
3.17 หน้าแสดงผล IAM User ที่ใช้งาน Resource	23

บทที่ 1

บทนำ

1.1 ความเป็นมาของปัญหา

การที่มีนักศึกษาในการเรียนวิชาที่เกี่ยวข้องกับระบบ Cloud Computing เป็นจำนวนมากจึงส่งผลให้อาจารย์ประจำวิชาเข้าถึงและจัดการ Resource หรือ Service ต่างๆที่นักศึกษาสร้างไว้ทั้งหมดค่อนข้างใช้เวลานานและยากลำบาก ทางกลุ่มของเราจึงต้องการทำโครงการเกี่ยวกับเว็บแอปพลิเคชันสำหรับจัดการ Resource และ Service ต่างๆ ใน AWS เพื่อให้สามารถเข้าถึงและติดตามได้สะดวกมากยิ่งขึ้น

1.2 วัตถุประสงค์

1. เพื่อเพิ่มความรวดเร็วและความสะดวกสบายให้กับอาจารย์ ในการจัดการ Resource ต่างๆ ใน AWS ที่นักศึกษасร้างขึ้นในวิชาเรียนของคณะเทคโนโลยีสารสนเทศสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
2. ลดข้อผิดพลาดในการบริหารจัดการระบบ ยกตัวอย่างเช่น กรณีที่นักศึกษาลืมใส่ Tag ให้กับ Resource เพื่อมั่นใจได้ว่า Resource ที่ถูกสร้างขึ้นทุกที่เป็นของกลุ่มไหนและสามารถลบได้ตามกรอบ Policy ที่กำหนด
3. เพิ่มฟังก์ชันในส่วนที่ AWS Management Console ไม่มี สำหรับอาจารย์ผู้รับผิดชอบวิชา เช่น ฟังก์ชันการดูค่าใช้จ่าย ฟังก์ชันในการติดตั้งและตั้งค่า Resources ที่นอกเหนือในการเรียนในวิชา Cloud

1.3 ขอบเขตของระบบ

โปรเจกต์จัดทำขึ้นเพื่อใช้สำหรับวิชา Cloud-Based Enterprise Application Development หรือ วิชาที่เรียนทางด้าน Cloud Computing โดยใช้บริการของ AWS เท่านั้น เพื่อแก้ไขปัญหาของอาจารย์ประจำวิชาและนักศึกษาที่เรียนในวิชานี้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. อาจารย์ประจำวิชามีความสะดวกสบายในการจัดการ Resources
2. นักศึกษาที่เรียนในวิชา Cloud วางแผนการคำนวณค่าใช้จ่ายได้ง่ายขึ้น
3. อาจารย์ประจำวิชาที่เป็นเจ้าของ Account ไม่ต้องเสียค่าใช้จ่ายส่วนเกินที่ได้รับมา
4. นักศึกษาสามารถทดลองการใช้ Resource ได้เต็มที่เพื่อให้โปรเจกสำเร็จรวดเร็ว
5. คณะผู้จัดทำได้ความรู้และประสบการณ์ในการทำ Infrastructure as Code ของระบบ AWS

1.5 ศัพท์นิยามเฉพาะที่ใช้ในโครงการ

1. SDK (Software Development Kit) คือ ชุดเครื่องมือที่ใช้สำหรับการพัฒนาซอฟต์แวร์ที่ถูกสร้างโดยผู้พัฒนา เพื่อให้ผู้ใช้ซอฟต์แวร์สามารถนำเครื่องมือนี้ไปสร้างหรือพัฒนางานของตนเองได้
2. AWS Resource คือ ทรัพยากรต่างๆที่มีให้บริการใน Amazon Web Service เช่น AWS Lambda, AWS CloudTrail และ AWS S3 เป็นต้น
3. IAM User คือ ผู้ใช้ในส่วนของนักศึกษาที่ใช้งาน AWS Management Console ของ Account อาจารย์ผู้สอน

บทที่ 2

การทบทวนวรรณกรรม

2.1 การบริการและทรัพยากรใน AWS เพื่อใช้ในการสร้างแท็กแบบอัตโนมัติ



รูปที่ 2.1 สัญลักษณ์ AWS CloudTrail

2.1.1 AWS CloudTrail

เป็น Service ที่จะเก็บ Log ทุกการกระทำของ IAM user ทั้งหมด โดยการเก็บ Log จะขึ้นอยู่กับ Policy ที่ตั้งไว้ เมื่อ User ใช้งานตาม Policy ที่กำหนดจะมีการเก็บ Log โดย CloudTrail ลง S3 Bucket



รูปที่ 2.2 สัญลักษณ์ AWS CloudWatch

2.1.2 AWS CloudWatch

เป็น Service ที่จะคอยตรวจสอบ รวบรวมข้อมูลและการดำเนินการต่างๆ เห็นภาพรวมของทรัพยากร แอปพลิเคชัน และบริการต่างๆ ของ AWS ใน 1 Region



รูปที่ 2.3 สัญลักษณ์ AWS Lambda

2.1.3 AWS Lambda

เป็น Service ที่ทำให้ผู้ใช้สามารถรันโค้ดโดยไม่ต้องมี Server เป็นของตัวเอง ซึ่งส่งผลให้ผู้ใช้ไม่ต้องทำการตั้งค่าต่างๆเกี่ยวกับ Server ผู้ใช้เพียงแค่นำโค้ดขึ้นไป AWS Lambda โค้ดจะสามารถทำงานได้ทันที



รูปที่ 2.4 สัญลักษณ์ AWS IAM

2.1.4 IAM

เป็น Service ที่ช่วยจัดการสิทธิ์การเข้าถึงทรัพยากรและบริการของ AWS อย่างปลอดภัย สามารถสร้างและจัดการผู้ใช้และกลุ่ม AWS ได้โดยใช้ IAM และใช้สิทธิ์เพื่อบุคคลหรือปฏิเสธไม่ให้ผู้ใช้เข้าถึงทรัพยากร AWS ได้



รูปที่ 2.5 สัญลักษณ์ Boto 3

2.1.3 Boto 3

Boto 3 เป็น AWS SDK สำหรับภาษา Python ที่ช่วยให้นักพัฒนาสามารถพัฒนาซอฟต์แวร์ให้ทำงานร่วมกับ Amazon Service ได้สะดวกมากขึ้น

2.2 เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์

2.2.1 เครื่องมือสำหรับการพัฒนาเว็บแอปพลิเคชัน



รูปที่ 2.6 สัญลักษณ์ NEXT.JS

2.2.1.1 Next JS

Next Js เป็น React Framework พัฒนาโดย Vercel ที่ใช้สำหรับการพัฒนาเว็บแอปพลิเคชันทั้งในฝั่ง Front-end โดยจุดเด่นหลักๆของ Next JS คือ ทาง Vercel ได้นำเครื่องมือต่างๆที่จำเป็นมาไว้ใน Next.JS ให้แล้วเช่น การทำ Routing, การทำ API และมีการ Optimize ทำให้มี Performance ที่ดียิ่งขึ้น



รูปที่ 2.7 สัญลักษณ์ MongoDB

2.2.1.2 MongoDB

MongoDB เป็น Database โดยมีพื้นฐานเป็น NoSQL จึงไม่มีความสัมพันธ์ของข้อมูล

เหมือนกับ Database แบบ SQL ซึ่ง MongoDB จะบันทึกข้อมูลลงในไฟล์รูปแบบ

JSON

ซึ่งต่างจาก Database SQL ที่เก็บข้อมูลในรูปแบบตาราง

บทที่ 3

วิธีการดำเนินการวิจัย

3.1 การวิเคราะห์และออกแบบระบบ

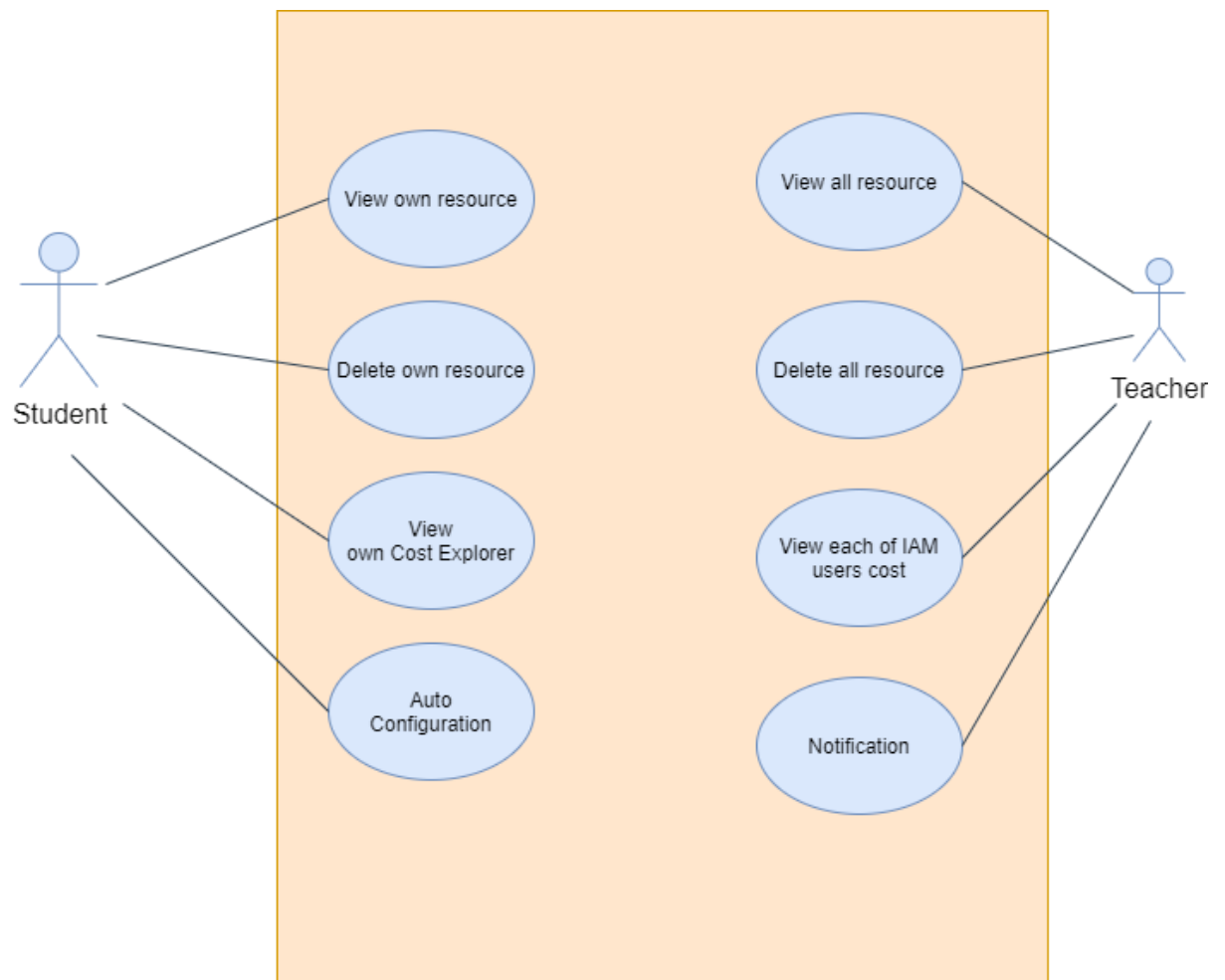
3.1.1 วิเคราะห์ปัญหา

- ทรัพยากรที่ถูกสร้างขึ้นจะไม่มีการระบุว่าเป็นของผู้ใช้งานคนไหน
- การดูแลทรัพยากรทุกอย่างใน AWS เป็นเรื่องที่ยุ่งยาก
- การจะลบทรัพยากรต้องใช้ระยะเวลาจนถึงจะลบหมด
- IAM User ไม่มีหน้าการคำนวณค่าใช้จ่ายบน AWS Management Console

3.1.2 ออกแบบความต้องการของระบบ

- วางระบบสร้างแท็กอัตโนมัติเมื่อมีการสร้างทรัพยากรนั้น ๆ บน AWS ที่จะมีการระบุชื่อของกลุ่มที่สร้าง
- สร้างเว็บแอปพลิเคชันที่มีฟังก์ชันในการดูข้อมูลและลบทรัพยากรให้สะดวกมากยิ่งขึ้น และมีการคำนวณค่าใช้จ่ายของ IAM User นักศึกษา
- อาจารย์สามารถดูทรัพยากรของทุกกลุ่มที่สร้างขึ้นได้โดยจะแสดงผลให้เข้าใจได้ง่าย และสามารถสั่งลบทรัพยากรที่สร้างขึ้นจาก IAM User นักศึกษาได้ทั้งหมดได้ในขั้นตอนเดียว
- เว็บแอปพลิเคชันมีระบบสนับสนุนเพิ่มเติมจาก AWS Management Console เช่น การแจ้งเตือนการสร้าง VPC เกินจำนวน และ Internet Gateway เกินจำนวน เป็นต้น

3.1.3 Use Case Diagram



รูปที่ 3.1 Use Case Diagram

3.2 แผนดำเนินงาน

ตารางที่ 3.1 ปฏิทินแผนดำเนินงาน

ขั้นตอนการดำเนินงาน	ระยะเวลาการดำเนินงาน															
	สิงหาคม 2564				กันยายน 2564				ตุลาคม 2564				พฤศจิกายน 2564			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1. ศึกษาและทดลองทำ auto-tagging สำหรับ AWS Resource																
2. ศึกษาและทดลองการทำ AWS SDK (ใส่ส่วนของการเรียกดูข้อมูล s3 ec2 และลบ instance)																
3. ออกแบบ UI หรือ prototype ของเว็บ																
4. สร้าง API สำหรับการดึงข้อมูล Resource																

และลบ Resource																
5. เริ่มพัฒนาเว็บ แอป																
6. เริ่มทำระบบ แจ้ง เตือน เมื่อมี การสร้าง Resource ที่เกิน ความจำเป็น																
7. เริ่มศึกษาเรื่อง การคำนวณ ค่าใช้จ่ายใน กรณีที่แยก Resource ตาม IAM User																



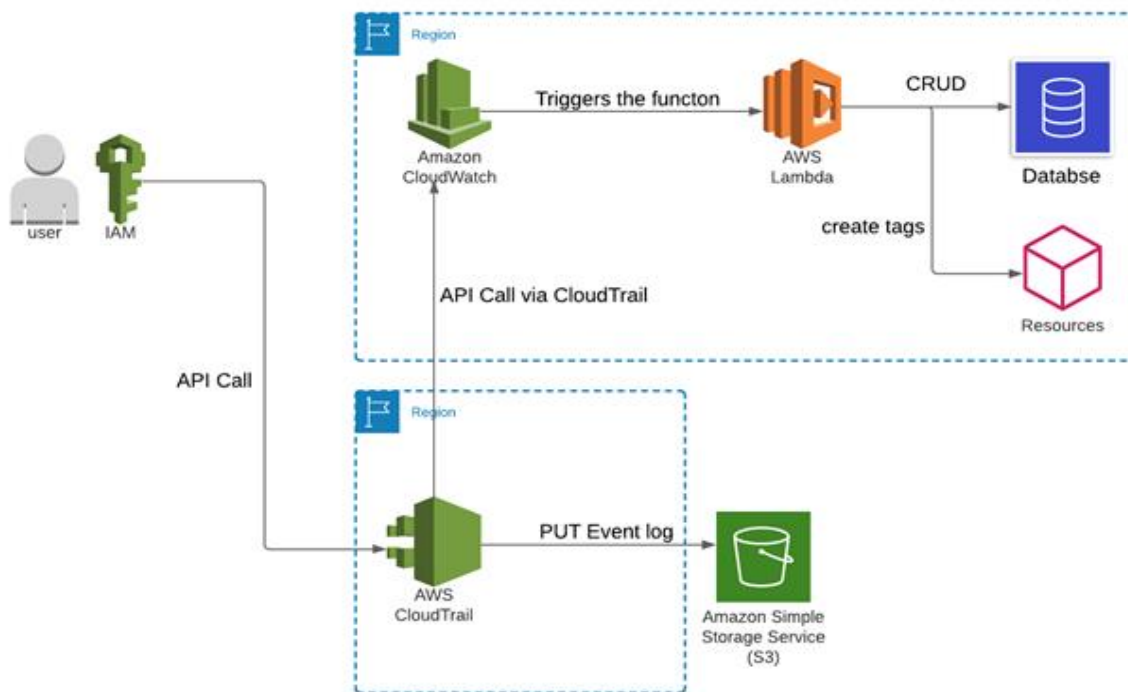
หมายถึง เริ่มต้นทำในเวลาหนึ่งจนสิ้นสุดในอีกเวลาหนึ่ง



หมายถึง เริ่มต้นทำในเวลาหนึ่งจนถึงวันสิ้นสุดโครงการ

3.3 หลักการทำงานของระบบ

3.3.1 หลักการทำงานของระบบ Auto Tag



รูปที่ 3.2 Auto-Tagging Solution Workflow

เมื่อ IAM User ทำการสร้าง Resource หรือ ใช้งาน Service ต่างๆ ระบบจะทำการเรียกใช้งานAWS CloudTrail เพื่อทำการตรวจสอบ Event ที่ IAM User ได้กระทำต่อระบบของ AWS และเก็บข้อมูล Event นั้นลงใน AWS S3 และจากนั้น AWS CloudTrail จะทำการเรียกใช้งาน AWS CloudWatch

3.3.2 ลำดับการวางระบบ Auto Tag เบื้องต้น

3.3.2.1 เพิ่มโค้ด Boto 3 ลงใน AWS Lambda

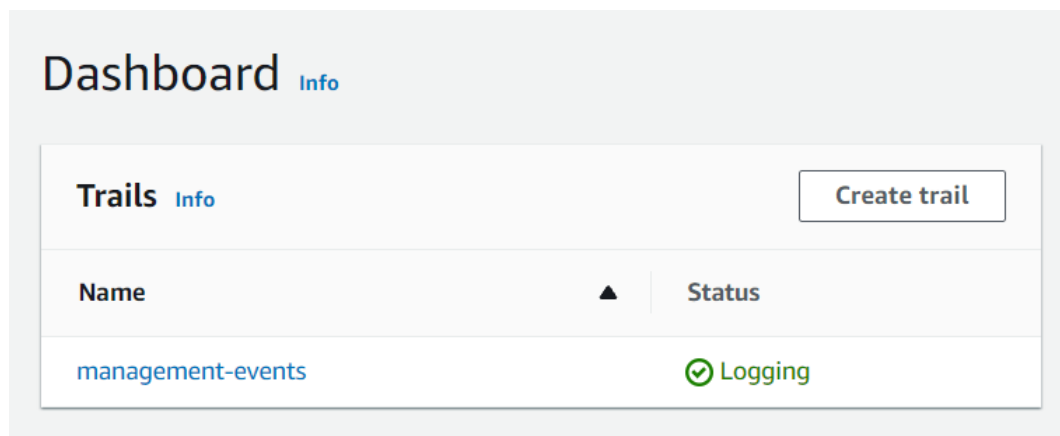
```
1 import json
2 import boto3
3 ec2 = boto3.resource('ec2')
4
5
6 def lambda_handler(event, context):
7     print(json.dumps(event))
8     ids = []
9     instances = []
10    region = event['region']
11    detail = event['detail']
12    eventName = detail['eventName']
13    user_type = detail['userIdentity']['type']
14    arn = detail['userIdentity']['arn']
15    principal = detail['userIdentity']['principalId']
16
17    if user_type == 'IAMUser':
18        user = detail['userIdentity']['userName']
19    else:
20        user = principal.split(':')[1]
21
22    print('Region Name ' + region)
23    print('Event Name ' + eventName)
24    print('User type ' + user_type)
25    print('User Name ' + user)
```

รูปที่ 3.3 โค้ด Boto 3

```
27 if not detail['responseElements']:
28     print("ResponseElement is missing. There could be an error that occurred.")
29     if detail['errorCode']:
30         print('Error Code: ' + detail['errorCode'])
31     if detail['errorMessage']:
32         print('Error Message: ' + detail['errorMessage'])
33     return False
34 else:
35     if eventName == 'RunInstances':
36         items = detail['responseElements']['instancesSet']['items']
37         for item in items:
38             ids.append(item['instanceId'])
39             instances.append(item['instanceId'])
40             base = ec2.instances.filter(InstanceIds=ids)
41             for inst in base:
42                 for vol in inst.volumes.all():
43                     ids.append(vol.id)
44                 for eni in inst.network_interfaces:
45                     ids.append(eni.id)
46             print('Instance IDs: ' + str(instances))
47             print('Total number of instance launched: ' + str(len(instances)))
48         elif eventName == 'CreateVpc':
49             ids.append(detail['responseElements']['vpc']['vpcId'])
50
51     print('ids = %s' % ids)
52     if ids:
53         ec2.create_tags(Resources=ids, Tags=[
54             {'Key': 'owner', 'Value': user},
55             {'Key': 'PrincipalId', 'Value': principal}
56         ])
57
58     return True
```

รูปที่ 3.4 โค้ด Boto 3

3.3.2.2 เปิดใช้งาน AWS CloudTrail



รูปที่ 3.5 อธิบายการเปิดใช้งาน AWS CloudTrail

3.3.2.3 เพิ่ม Event Rule ใน AWS CloudWatch

☒ Event Pattern ⓘ ☐ Schedule ⓘ

Build event pattern to match events by service ▼

Service Name	EC2 ▼
Event Type	AWS API Call via CloudTrail ▼

รูปที่ 3.6 อธิบายการเพิ่ม Event Rule ใน AWS CloudWatch

เลือก Service ที่จะดัก Event เช่น EC2 และเลือก Event Type เป็น AWS API Call via CloudTrail หมายความว่าดัก Event เมื่อ User กระทำต่อ Service ผ่านทาง SDK หรือ AWS Management Console เช่น การสร้าง แก้ไข และ ลบ

```
{
  "source": [
    "aws.ec2"
  ],
  "detail-type": [
    "AWS API Call via CloudTrail"
  ],
  "detail": {
    "eventSource": [
      "ec2.amazonaws.com"
    ],
    "eventName": [
      "RunInstances",
      "CreateVpc"
    ]
  }
}
```

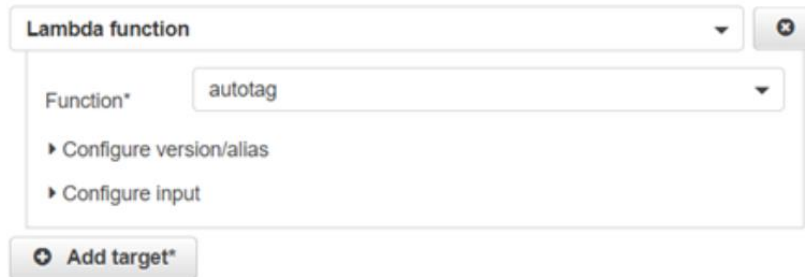
รูปที่ 3.7 อธิบายการเพิ่ม Event ที่ต้องการจะดักผ่าน JSON

ใส่ Event ที่ต้องการจะดักใน key “eventName” โดย Event ทั้งหมดสามารถหาข้อมูลเพิ่มเติม

3.3.2.4 ใต้ Targets trigger function ใน CloudWatch

Targets

Select Target to invoke when an event matches your Event Pattern or when schedule is triggered.



รูปที่ 3.8 อธิบายการ ใต้ Targets Trigger Function ใน CloudWatch

เลือก Lambda Function ที่สร้างในขั้นตอนที่ 3.2.2.1

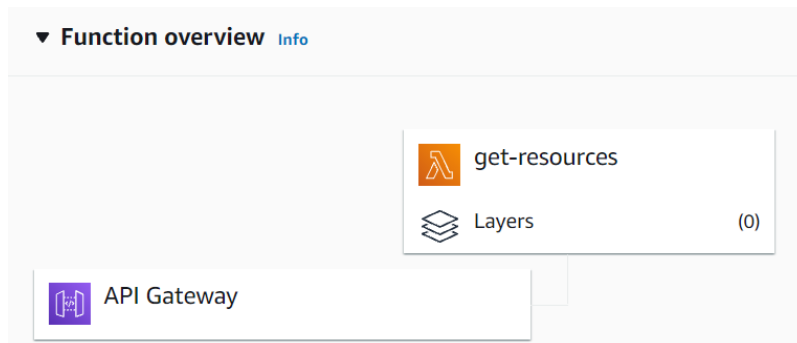
3.3.2 การดึงข้อมูลทรัพยากร

3.3.2.1 ฟังก์ชันโค้ด AWS SDK (Boto3)

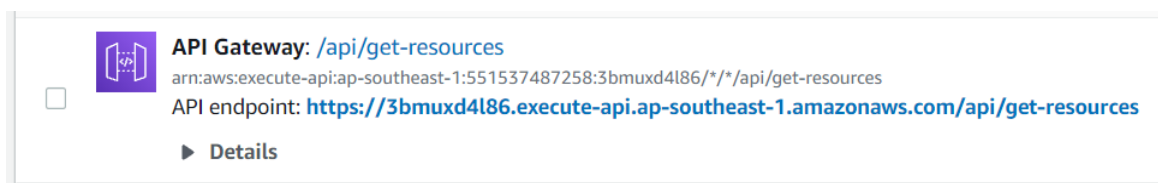
```
1 from boto3.session import Session
2 import os
3
4 session = Session(
5     aws_access_key_id=os.environ['ACCESS_KEY_ID'],
6     aws_secret_access_key=os.environ['SECRET_ACCESS_KEY'],
7     region_name=os.environ['REGION_NAME']
8 )
9
10 def lambda_handler(event, context):
11
12     client = session.client('resourcegroupstaggingapi')
13     resources = {}
14
15     if 'queryStringParameters' in event:
16         resources = client.get_resources(
17             TagFilters=[
18                 {
19                     'Key': 'owner',
20                     'Values': event['queryStringParameters']['owner'].split(',')
21                 }
22             ],
23         )
24     else:
25         resources = client.get_resources()
26
27     return resources
28
```

รูปที่ 3.9 ฟังก์ชันโค้ด AWS SDK (boto3)

3.3.2.2 สร้าง API Gateway เพื่อเรียกใช้งานฟังก์ชัน

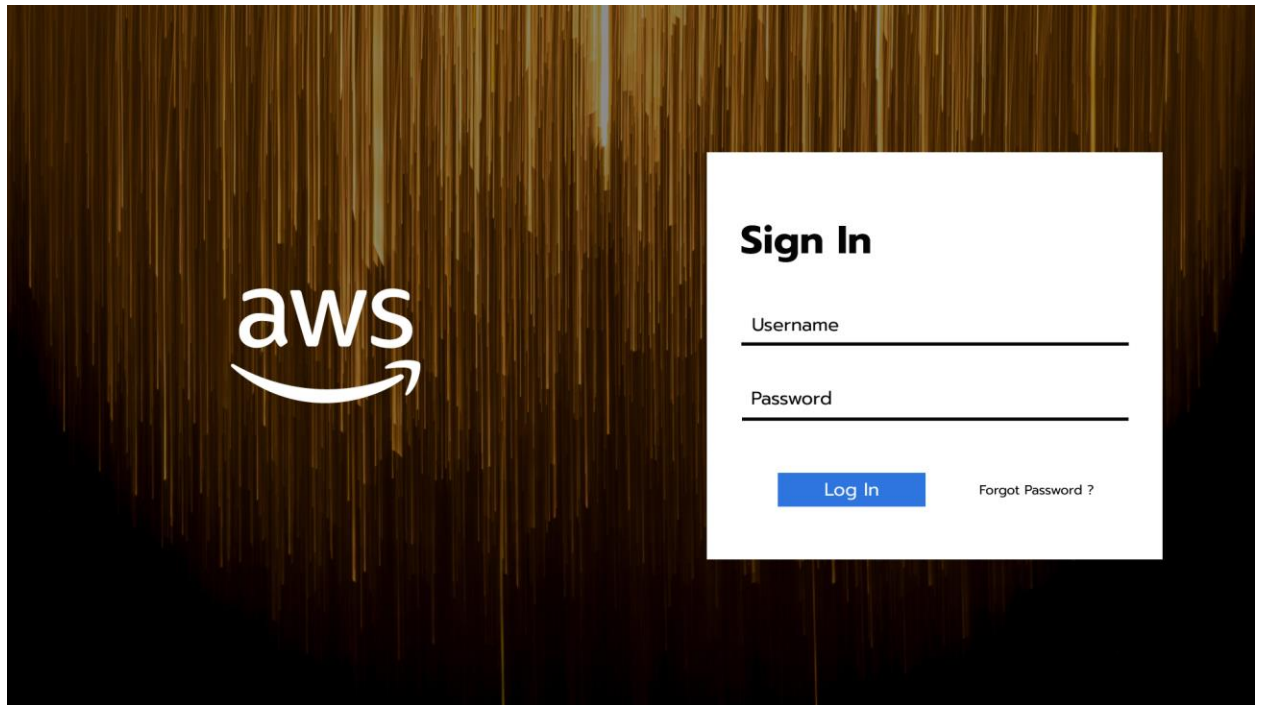


รูปที่ 3.10 การสร้าง API Gateway เพื่อเรียกใช้งานฟังก์ชัน (1)

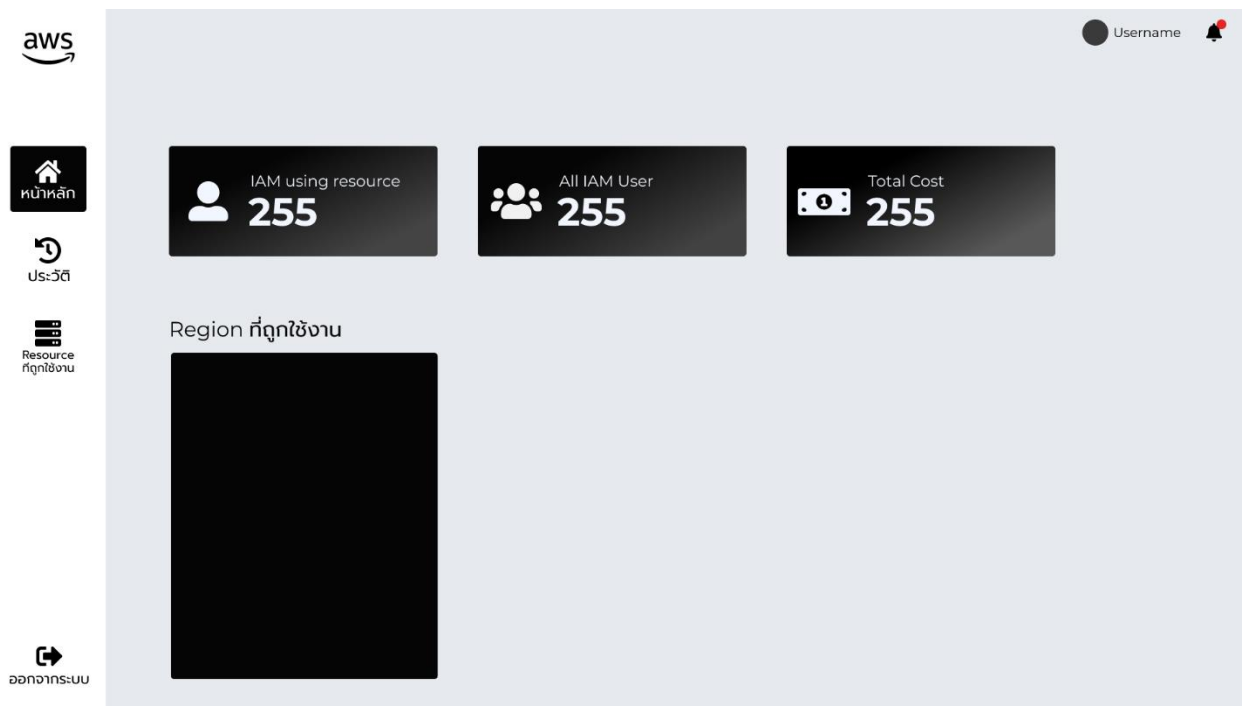


รูปที่ 3.11 การสร้าง API Gateway เพื่อเรียกใช้งานฟังก์ชัน (2)

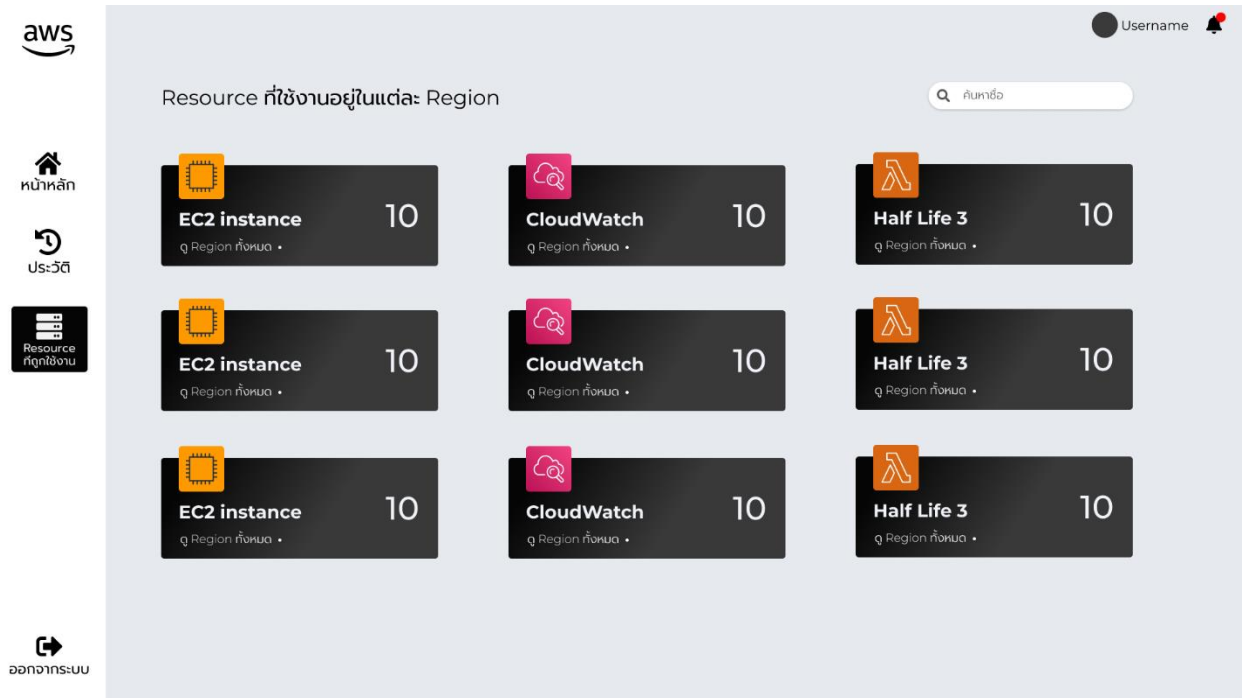
3.3.3 Prototype



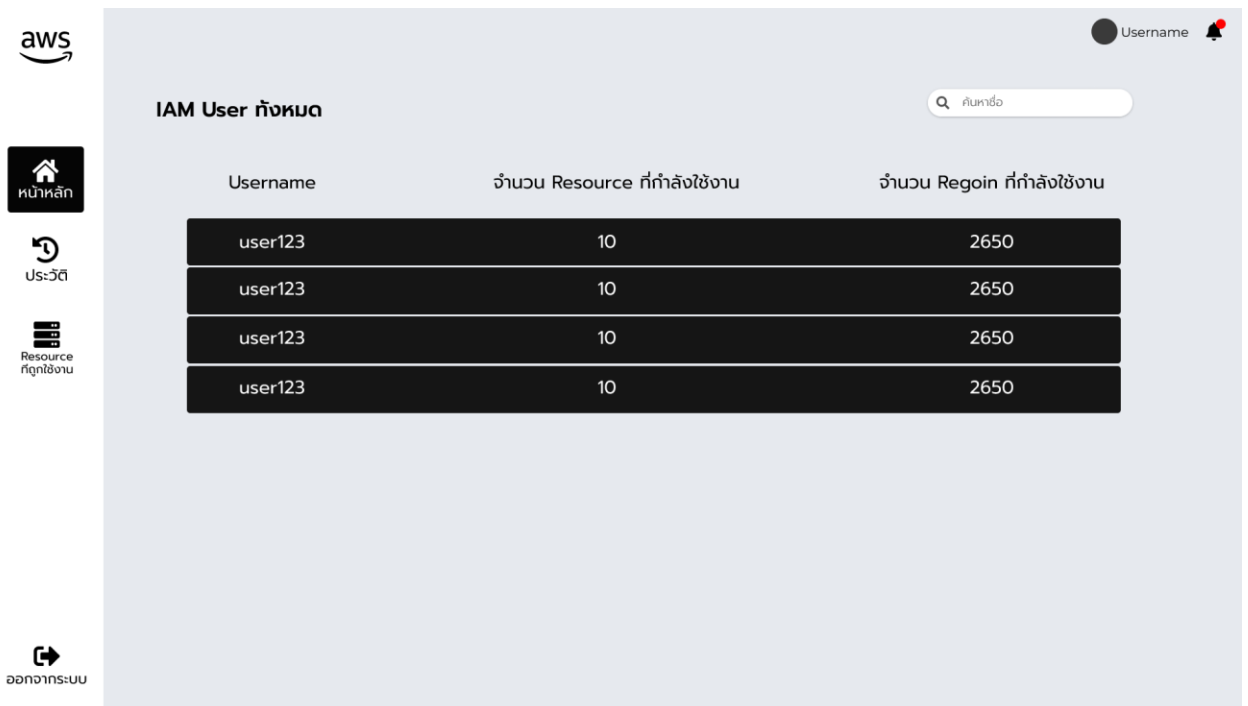
รูปที่ 3.12 หน้า Log in



รูปที่ 3.13 หน้า Dashboard ของอาจารย์ผู้สอน



รูปที่ 3.14 หน้าแสดงผล Resource ที่กำลังใช้งานในแต่ละ Region



รูปที่ 3.15 หน้าแสดง IAM User ทั้งหมด

หน้าหลัก

ประวัติ

Resource ที่ถูกใช้งาน

ออกจากระบบ

EC2 instance

id	เจ้าของ Resource	เวลาที่ถูกรับ		
rs01	10	2650		
rs02	10	2650		
rs03	10	2650		
rs04	10	2650		

รูปที่ 3.16 หน้ารายละเอียดของ Resource

หน้าหลัก

ประวัติ

Resource ที่ถูกใช้งาน

ออกจากระบบ

IAM User ที่ใช้งาน Resource อยู่

ค้นหา

Username	จำนวน Resource ที่กำลังใช้งาน	จำนวน Region ที่กำลังใช้งาน
user123	10	2650
user123	10	2650
user123	10	2650
user123	10	2650

รูปที่ 3.17 หน้าแสดงผล IAM User ที่ใช้งาน Resource

บรรณานุกรม

- [1] Bill Dry. **“Automatically tag new AWS resources based on identity or role.”** [Online].
Available: <https://aws.amazon.com/blogs/mt/auto-tag-aws-resources/>
[ที่เข้าถึง 4 สิงหาคม 2564]
- [2] devahoy. **“getting-started-with-nextjs.”** [Online].
Available: <https://devahoy.com/blog/2020/03/getting-started-with-nextjs/>
[ที่เข้าถึง 6 สิงหาคม 2564]
- [3] Next.JS. **“Next.JS Documentation”** [Online].
Available: <https://nextjs.org/docs>
[ที่เข้าถึง 8 สิงหาคม 2564]
- [4] Oliver Berger. **“CloudTrail Event Names – A Comprehensive List.”** [Online]. Available:
<https://www.gorillastack.com/blog/real-time-events/cloudtrail-event-names/>
[ที่เข้าถึง 11 สิงหาคม 2564]
- [5] AWS. **“CloudTrail log file examples.”** [Online]. Available:
<https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-log-file-examples.html> [ที่เข้าถึง 11 สิงหาคม 2564]
- [6] Boto3. **“ResourceGroupsTaggingAPI.”** [Online]. Available:
https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/resourcegroup_taggingapi.html#ResourceGroupsTaggingAPI.Client.get_resources
[ที่เข้าถึง 25 สิงหาคม 2564]
- [7] Boto3. **“EC2.”** [Online]. Available:
<https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/ec2.html> [ที่เข้าถึง 25 สิงหาคม 2564]

[8] Twin.macro. “**twin.macro Next.js**” [Online]. Available:

<https://github.com/ben-rogerson/twin.examples/tree/master/next-styled-components>

[ที่เข้าถึง 26 สิงหาคม 2564]

[9] TailwindCSS. “**Tailwind CSS Documentation**” [Online]. Available:

<https://tailwindcss.com/docs>

[ที่เข้าถึง 26 สิงหาคม 2564]