



Generative AI Apps and Agents on Amazon Bedrock

สร้างแอป RAG และ AI Agent บน AWS ด้วย Amazon Bedrock

● 18 ชั่วโมง

● 3 วัน

● พื้นฐาน Python และ AWS เบื้องต้น

● ระดับ Intermediate



ระยะเวลา: 3 วัน

18 ชั่วโมง (3 วัน วันละ 6 ชม.) · 09:00-16:00 น.



รูปแบบ: Classroom / Hybrid

Onsite หรือ Online ผ่าน Microsoft Teams · Public / In-house



รายละเอียดหลักสูตร

www.itgenius.co.th

การทำต้นแบบแชทบอตด้วย Generative AI ใช้เวลาไม่กี่ชั่วโมง แต่การพาแอปเดี่ยวก้าวขึ้นใช้งานจริงในองค์กรเป็นอีกเรื่องหนึ่ง ทีมพัฒนาต้องตอบให้ได้ว่าจะเลือกโมเดลไหน ค่าใช้จ่ายต่อคำขอเท่าไร คำตอบอ้างอิงเอกสารที่ถูกต้องหรือไม่ จะกันข้อมูลส่วนบุคคลรั่วและ Prompt Injection อย่างไร และเมื่อเพิ่ม AI Agent ที่เรียกระบบอื่นได้เอง จะควบคุมสิทธิ์และติดตามการทำงานของมันเป็นอย่างไร Amazon Bedrock มีบริการที่ช่วยเรื่องเหล่านี้ครบ แต่ต้องรู้ว่าจะประกอบแต่ละส่วนเข้าด้วยกันอย่างไร

หลักสูตรนี้พานักพัฒนาสร้างแอป Generative AI บน AWS ด้วย Python ตั้งแต่ต้นจนจบ เริ่มจากเรียกโมเดลผ่าน Converse API, Streaming, Tool Use และการคุมต้นทุน ต่อด้วยระบบ RAG บน Amazon Bedrock Knowledge Bases พร้อม Reranker และการวัดคุณภาพ ป้องกันความเสี่ยงด้วย Amazon Bedrock Guardrails, IAM และ Logging แล้วสร้าง AI Agent ด้วย Bedrock Agents และ Strands Agents ก่อน Deploy ขึ้น Amazon Bedrock AgentCore พร้อม Memory, Gateway, Identity และ Observability ปิดท้ายด้วย Capstone ผู้ช่วย AI ภายในองค์กร เนื้อหาช่วยฝึกทักษะที่สอดคล้องกับข้อสอบ AWS Certified Generative AI Developer - Professional (AIP-C01) (อบรม 3 วัน วันละ 6 ชั่วโมง รวม 18 ชั่วโมง ระดับ Intermediate)

เครื่องมือ

AWS

Bedrock

Knowledge Bases

AgentCore

Python

ราคาค่าอบรม · PUBLIC / IN-HOUSE

9,900 บาท / ท่าน

ราคายังไม่รวมภาษีมูลค่าเพิ่ม (VAT 7%) · รวมเอกสารประกอบการอบรม Lab Guide โค้ด Python คนทุก Lab โปสเตอร์ Capstone ต้นแบบ และ Certificate · ไม่รวมค่าบริการ AWS ในบัญชีของผู้เรียน · รับผิดชอบ In-house ขอใบเสนอราคาได้

สมัครผ่านเว็บไซต์ ลด 10%



เพิ่มเพื่อน
LINE



01 วัตถุประสงค์ Objectives

- เรียกใช้ Foundation Model บน Amazon Bedrock ด้วย Converse API แบบ Streaming, Multimodal และ Tool Use
- เลือกโมเดลและออกแบบการเรียกใช้ให้คุ้มค่าด้วย Prompt Caching, Batch Inference และ Cross-Region Inference
- จัดการ Prompt เป็นเวอร์ชันและทดสอบคุณภาพก่อนนำขึ้นใช้งาน
- สร้างระบบ RAG ด้วย Bedrock Knowledge Bases และปรับคุณภาพด้วย Chunking, Metadata และ Reranker
- ป้องกันเนื้อหาไม่เหมาะสม ข้อมูลส่วนบุคคล และ Hallucination ด้วย Bedrock Guardrails
- สร้าง AI Agent ด้วย Bedrock Agents และ Strands Agents พร้อมเชื่อม Tool ผ่าน MCP
- Deploy Agent ขึ้น AgentCore Runtime พร้อม Memory, Gateway, Identity และ Policy
- ติดตาม วัดผล และควบคุมค่าใช้จ่ายของแอป GenAI ใน Production ได้

02 กลุ่มเป้าหมาย Target Audience

- นักพัฒนา Backend และ Full-stack ที่ต้องสร้างแอป Generative AI บน AWS
- Cloud Engineer และ Solutions Architect ที่ออกแบบระบบ RAG หรือ AI Agent ให้องค์กร
- ทีม AI หรือ Innovation ที่ต้องการพาต้นแบบขึ้นใช้งานจริงอย่างปลอดภัย
- ผู้ที่ต้องการฝึกทักษะที่สอดคล้องกับข้อสอบ AWS Certified Generative AI Developer - Professional
- ผู้ที่ผ่าน AWS AI Practitioner แล้วและต้องการลงมือสร้างแอปจริง

03 จุดเด่นของหลักสูตร Highlights

- **ลงมือเขียนโค้ดทุกวัน** Python และ boto3 ในบัญชี AWS ของผู้เรียน
- **RAG ที่วัดคุณภาพได้** Knowledge Bases, Reranker และ Evaluation
- **ปลอดภัยตั้งแต่ออกแบบ** Guardrails, IAM, การเข้ารหัส และ Logging
- **Agent พร้อม Production** Strands Agents, MCP และ AgentCore
- **คุ้มค่าใช้จริงได้** เลือกโมเดล, Caching, Batch และลบทรัพยากรหลัง Lab

04 ความรู้พื้นฐาน Prerequisites

- เขียน Python ได้ เช่น ฟังก์ชัน, Dictionary, JSON และการติดตั้ง Package
- ใช้งาน AWS Console และเข้าใจ IAM, S3 และ Lambda ในระดับพื้นฐาน
- เข้าใจแนวคิด LLM และ Prompt เบื้องต้น หากผ่าน AWS AI Practitioner มาแล้วจะเรียนได้ราบรื่นขึ้น
- Notebook ที่ติดตั้ง Python 3, VS Code และ AWS CLI ได้ พร้อมบัญชี AWS แบบ Paid Plan

05 สิ่งที่ต้องเตรียมและเครื่องมือ Setup & Tools

รายการ	รายละเอียด
อุปกรณ์	Notebook เชื่อมต่ออินเทอร์เน็ต ติดตั้ง Python 3, VS Code, Git และ AWS CLI
บัญชี AWS	บัญชีของผู้เรียนหรือ Sandbox ขององค์กรแบบ Paid Plan พร้อมสิทธิ์ใช้ Amazon Bedrock
Amazon Bedrock	Converse API, Prompt Management, Knowledge Bases, Guardrails, Agents และ Evaluations
Agent	Strands Agents SDK, Model Context Protocol (MCP) และ Amazon Bedrock AgentCore
บริการประกอบ	Amazon S3, S3 Vectors, AWS Lambda, CloudWatch, CloudTrail และ AWS Budgets

หมายเหตุ: Lab ทั้งหมดทำในบัญชี AWS ของผู้เรียนเอง ซึ่งมีค่าใช้จ่ายตามการใช้งานจริง เช่น ค่า Token ของโมเดล, Vector Store และ AgentCore ที่ใช้ได้เฉพาะบัญชีแบบ Paid Plan หลักสูตรเลือกโมเดลขนาดเล็ก ใช้ S3 Vectors เป็นค่าเริ่มต้น และมีขั้นตอนลบทรัพยากรหลังจบแต่ละ Lab แต่ผู้เรียนควรตั้ง AWS Budgets ไว้ก่อนเริ่ม ต่อยอดจากหลักสูตร RAG & Knowledge Base โดยเน้นบริการ Managed ของ AWS ตั้งแต่ RAG ไปจนถึง Agent ใน Production เนื้อหาสอดคล้องกับทักษะบางส่วนของข้อสอบ AIP-C01 แต่ไม่ใช่หลักสูตรตรวจสอบอย่างเป็นทางการของ AWS และไม่รวมค่าสอบ ค่าบริการและเฟิร์มแวร์ของ Bedrock และ AgentCore เปลี่ยนบ่อย เนื้อหาอ้างอิงเวอร์ชันล่าสุด ณ วันที่อบรม

06 ภาพรวมการเรียนรู้ 3 วัน Big Picture

วัน	หัวข้อหลัก
Day 1 Section 1-5	เตรียมบัญชีและสิทธิ์ IAM, เรียกโมเดลด้วย Converse API แบบ Streaming และ Multimodal, เลือกโมเดลและคุมต้นทุนด้วย Prompt Caching และ Batch, จัดการ Prompt อย่างเป็นระบบ และให้โมเดลเรียกใช้ Tool
Day 2 Section 6-10	Embedding และ Vector Store บน AWS, สร้าง RAG ด้วย Bedrock Knowledge Bases, เพิ่มคุณภาพด้วย Hybrid Search, Reranker และ Evaluation, ป้องกันด้วย Bedrock Guardrails และวาง Security กับ Governance
Day 3 Section 11-15	แนวคิด AI Agent และ Bedrock Agents, เขียน Agent ด้วย Strands Agents และ MCP, Deploy ขึ้น AgentCore Runtime พร้อม Memory, Gateway และ Identity, ติดตามผลด้วย Observability แล้วปิดท้ายด้วย Capstone



DAY
1

เรียกใช้ Foundation Model อย่างมืออาชีพ

เนื้อหาการอบรม · Section 1-5

รวม
6 ชั่วโมง

1 Lab: เตรียมบัญชี AWS และ Amazon Bedrock

- ภาพรวมบริการ Generative AI บน AWS: Bedrock, AgentCore และ SageMaker AI ต่างกันอย่างไร
- เลือก Region, ตรวจสอบสิทธิ์ใช้โมเดล และทำความเข้าใจ Cross-Region Inference Profile
- สร้าง IAM Role และ Policy แบบสิทธิ์น้อยที่สุดสำหรับนักพัฒนาและแอป
- ตั้งค่า AWS CLI, boto3 และ Credential บนเครื่องอย่างปลอดภัย
- Lab: ตั้ง AWS Budgets และเรียกโมเดลครั้งแรกจาก Python

2 Lab: เรียกโมเดลด้วย Converse API

- โครงสร้าง Converse API: System Prompt, Messages และ Inference Configuration
- ส่งรูปภาพและเอกสารเข้าโมเดลแบบ Multimodal
- ตอบแบบ Streaming ด้วย ConverseStream เพื่อประสบการณ์ที่ลื่นไหล
- จัดการ Throttling และ Error ด้วย Retry แบบ Exponential Backoff
- Lab: สร้าง API ส่งเอกสารที่รองรับทั้งข้อความและไฟล์ PDF

3 เลือกโมเดลและควบคุมต้นทุน

- เปรียบเทียบตระกูลโมเดลใน Bedrock ตามคุณภาพ ความเร็ว ภาษา และราคา
- คำนวณต้นทุนต่อคำจาก Input และ Output Token
- ลดต้นทุนด้วย Prompt Caching, Batch Inference และการเลือกโมเดลขนาดเล็ก
- On-demand เทียบกับ Provisioned Throughput และการวางแผน Model Routing
- Lab: วัดคุณภาพ ความเร็ว และต้นทุนของ 3 โมเดลกับงานเดียวกัน

4 Lab: Prompt Engineering และ Prompt Management

- เขียน System Prompt, Few-shot และกำหนดรูปแบบ Output เป็น JSON
- เก็บ Prompt เป็น Template พร้อมตัวแปรและเวอร์ชันด้วย Bedrock Prompt Management
- ต่อหลายขั้นตอนเป็น Workflow ด้วย Amazon Bedrock Flows
- ทดสอบ Prompt แบบ Regression ก่อนเปลี่ยนเวอร์ชันใน Production
- Lab: จัดการ Prompt ของงานจัดหมวดหมู่คำร้องลูกค้าเป็นเวอร์ชัน

5 Lab: Tool Use และ Function Calling

- นิยาม Tool ด้วย JSON Schema และให้โมเดลตัดสินใจเรียกใช้
- เขียน Loop รับคำขอเรียก Tool รับฟังก์ชัน แล้วส่งผลกลับให้โมเดล
- ตรวจสอบ Parameter และจัดการ Error ของ Tool ไม่ให้แอปล่ม
- ใช้ AWS Lambda เป็น Tool ที่เชื่อมระบบภายในองค์กร
- Lab: ผู้ช่วยตอบสถานะคำสั่งซื้อจากระบบจำลอง

DAY
2

RAG, Guardrails และความปลอดภัย

เนื้อหาการอบรม · Section 6-10

รวม
6 ชั่วโมง

6 Lab: Embedding และ Vector Store บน AWS

- Embedding ทำอย่างไร และการเลือกโมเดล Embedding กับจำนวนมิติ
- ตัวเลือก Vector Store: S3 Vectors, OpenSearch Serverless และ Aurora PostgreSQL กับ pgvector
- เปรียบเทียบต้นทุนขั้นต่ำ ความเร็ว และความสามารถของแต่ละตัวเลือก
- Similarity Search, Metadata Filter และ Hybrid Search
- Lab: สร้าง Embedding จากเอกสารภาษาไทยและค้นหาความคล้ายด้วย Python

7 Lab: RAG ด้วย Amazon Bedrock Knowledge Bases

- สร้าง Knowledge Base จาก Amazon S3 และเลือก Vector Store
- Chunking แบบ Fixed-size, Hierarchical และ Semantic เลือกอย่างไร
- Parsing เอกสารที่มีตารางและรูปภาพด้วย Foundation Model หรือ Bedrock Data Automation
- Retrieve เทียบกับ RetrieveAndGenerate พร้อมแหล่งอ้างอิงในคำตอบ
- Lab: แดตบอตตอบคำถามจากคู่มือองค์กรพร้อม Citation

8 ปรับคุณภาพและวัดผล RAG

- ใส่ Metadata ให้เอกสารเพื่อกรองตามแผนก วันที่ หรือสิทธิ์
- เพิ่มความแม่นยำด้วย Reranker Model และ Query Reformulation
- Sync ข้อมูลใหม่แบบ Incremental เมื่อเอกสารต้นทางเปลี่ยน
- วัดคุณภาพ RAG ด้วย Amazon Bedrock Evaluations และ LLM-as-a-judge
- Lab: เทียบผลก่อนและหลังปรับ Chunking และ Reranker ด้วยชุดคำถามทดสอบ

9 Lab: Amazon Bedrock Guardrails

- Content Filter, Denied Topic และ Word Filter สำหรับแอปองค์กร
- Sensitive Information Filter สำหรับปิดบังหรือบล็อกข้อมูลส่วนบุคคล
- Contextual Grounding Check และ Automated Reasoning Check an Hallucination
- ใช้ ApplyGuardrail API กับโมเดลใดก็ได้ และป้องกัน Prompt Injection
- Lab: ใส่ Guardrail ให้แดตบอต RAG แล้วทดสอบด้วยข้อความโจมตี

10 Security และ Governance ของแอป GenAI

- IAM แบบสิทธิ์น้อยที่สุดสำหรับโมเดล Knowledge Base และ Agent
- เข้ารหัสด้วย AWS KMS และเชื่อมต่อแบบส่วนตัวผ่าน VPC Endpoint
- Model Invocation Logging, CloudTrail และการเก็บ Log อย่างระวังข้อมูลส่วนบุคคล
- การใช้ข้อมูลองค์กรกับโมเดลอย่างสอดคล้อง PDPA และนโยบายภายใน
- Workshop: Checklist ความปลอดภัยก่อนเปิดแอป GenAI ให้ผู้ใช้จริง

DAY
3

AI Agent ตั้งแต่ต้นแบบถึง Production

เนื้อหาการอบรม · Section 11-15

รวม
6 ชั่วโมง

11 Lab: แนวคิด AI Agent และ Bedrock Agents

- Agent ต่างจาก Chatbot อย่างไร: Reasoning, Tool, Memory และ Planning
- รูปแบบ ReAct, Workflow แบบกำหนดขั้นตอน และ Multi-agent
- Bedrock Agents แบบ Managed กับ Action Group และ Knowledge Base
- เลือกระหว่าง Agent แบบ Managed กับการเขียนเองด้วย Framework
- Lab: สร้าง Bedrock Agent ที่ค้นหาคู่มือและเปิดคำร้องผ่าน Lambda

12 Lab: เขียน Agent ด้วย Strands Agents

- โครงสร้าง Strands Agents SDK: Model, System Prompt และ Tool
- สร้าง Tool ด้วย Python Decorator และเรียกใช้ Tool จาก MCP Server
- รูปแบบ Multi-agent: Agents as Tools, Graph และ Swarm
- กำหนดเงื่อนไขหยุด Timeout และขอบเขตสิทธิ์ให้ Agent
- Lab: Agent วิเคราะห์ข้อมูลยอดขายที่เรียก Tool หลายตัว

13 Lab: Deploy ด้วย Amazon Bedrock AgentCore

- ภาพรวม AgentCore: Runtime, Memory, Gateway, Identity, Policy และ Observability
- Deploy Agent ขึ้น AgentCore Runtime ด้วย AgentCore CLI
- เพิ่ม Memory ระยะสั้นและระยะยาวให้ Agent จำรับทผู้ใช้
- แปลง Lambda และ API ภายในเป็น MCP Tool ผ่าน AgentCore Gateway
- ยืนยันตัวตนและจำกัดการเรียก Tool ด้วย AgentCore Identity และ Policy

14 Observability, Evaluation และการดูแลระบบ

- ติดตาม Trace ของ Agent ที่ละชั้นด้วย AgentCore Observability และ CloudWatch
- วัดคุณภาพ Agent ด้วย AgentCore Evaluations และชุดทดสอบของทีม
- ตั้ง Alarm สำหรับ Error, Latency และการใช้ Token ที่ผิดปกติ
- แนวทาง CI/CD และ Rollback เมื่อเปลี่ยนโมเดลหรือ Prompt
- Lab: หาสาเหตุที่ Agent ตอบผิดจาก Trace แล้วแก้ไข

15 Capstone: ผู้ช่วย AI ภายในองค์กร

- ออกแบบสถาปัตยกรรมที่รวม RAG, Guardrails และ Agent เข้าด้วยกัน
- สร้างผู้ช่วยที่ตอบจากเอกสารและทำงานผ่าน Tool ได้จริง
- ประเมินคุณภาพ ความปลอดภัย และต้นทุนต่อการใช้งาน
- นำเสนอผลงานและเก็บบทเรียนที่ใช้กับ Domain ของข้อสอบ AIP-C01
- ลงทรัพยากรทั้งหมดและสรุป Checklist ก่อนขึ้น Production

07 เรียนจบแล้วได้อะไร Outcomes

- **เรียกใช้โมเดลได้คล่อง** Converse API, Streaming และ Tool Use
- **ออกแบบให้คุ้มค่า** เลือกโมเดล, Caching และ Batch
- **สร้าง RAG ที่แม่นยำ** Knowledge Bases, Reranker และ Evaluation
- **ปกป้องแอปได้** Guardrails, IAM, KMS และ Logging
- **สร้าง Agent ได้จริง** Bedrock Agents, Strands Agents และ MCP
- **พร้อมขึ้น Production** AgentCore และ Observability

08 สิ่งที่คุณเรียนจะได้รับ Deliverables

- เอกสารประกอบการอบรมฉบับสมบูรณ์ครบทั้ง 3 วัน และ Lab Guide ทีละขั้นตอน
- โค้ด Python ควบทุก Lab ตั้งแต่ Converse API, RAG, Guardrails ถึง Agent
- โครงการ Capstone ผู้ช่วย AI ภายในองค์กรสำหรับต่อยอด
- ขั้นตอนลดทรัพยากรและ Checklist ความปลอดภัยก่อนขึ้น Production
- ประกาศนียบัตรจบหลักสูตร (Certificate of Completion) และช่องทางสอบถามวิทยากรหลังจบหลักสูตร

09 ข้อมูลผู้สอน Instructor

อ.สามิตร โกยม วิทยากรผู้พัฒนาแอป Generative AI บน AWS ในงานจริง ทั้งระบบถามตอบจากเอกสารองค์กร ผู้ช่วยที่เชื่อมระบบภายใน และ AI Agent ที่ต้องคุบสิทธิ์และค่าใช้จ่าย สถาบันไอทิจีนียัสเน้นการสอนแบบเขียนโค้ดไปพร้อมกัน อธิบายเหตุผลของการเลือกแต่ละบริการ และวัดผลทั้งคุณภาพ ความปลอดภัย และต้นทุนทุกครั้งที่ปรับระบบ เพื่อให้ผู้เรียนนำแนวทางไปใช้กับโปรเจกต์ขององค์กรได้ทันที

ราคาค่าอบรม 9,900 บาท/ท่าน · สนใจ IN-HOUSE ขอใบเสนอราคา

9,900 บาท / ท่าน (ยังไม่รวม VAT 7%)

สมัครผ่านเว็บไซต์ ลด 10%

☎ 02-570-8449

LINE @itgenius

☎ 088-807-9770

✉ contact@itgenius.co.th



เพิ่มเพื่อน
LINE