



ปั้นระบบอัตโนมัติอย่างมือโปรด้วย Jenkins และ GitHub Actions ร่วมกับ n8n



เมื่อไหร่ที่ขั้นตอนการ Deploy ที่ซ้ำซาก, เชื่องช้า และเต็มไปด้วยความเสี่ยง ในโลกการพัฒนาซอฟต์แวร์ ที่ความเร็วคือหัวใจสำคัญ การทำงานแบบเดิมๆ ไม่เพียงแต่ดูเครียดไปเสียเอง แต่ยังทำให้คุณเสียเวลาไปกับงาน ที่ไม่ควรต้องทำด้วยมืออีกต่อไป หลักสูตรนี้จะพาคุณก้าวกระโดดจากการทำงาน Manual สู่โลกของ DevOps เต็มรูปแบบ ที่ทุกอย่างตั้งแต่การ Build, Test, และ Deploy จะถูกจัดการด้วยระบบอัตโนมัติ

ผ่าน Docker Container ซึ่งเป็นมาตรฐานขององค์กรชั้นนำทั่วโลก พร้อมปลดล็อกศักยภาพการทำงานของคุณให้เหนือกว่าที่เคย

เราจะพาคุณลงมือทำจริงกับ 3 เครื่องมือ DevOps ที่เป็นหัวใจหลักของวงการ ใช้พลังของ Jenkins และ GitHub Actions สร้าง CI/CD Pipeline ที่แข็งแกร่งสำหรับแอปพลิเคชันยอดนิยมทั้ง Java Spring Boot, Python Flask, และ Node.js แต่เราจะไปไกลกว่านั้น! คุณจะได้พบกับ n8n "กาวใจอัจฉริยะ" ที่จะเชื่อมต่อทุกบริการเข้าด้วยกัน สร้างระบบแจ้งเตือนอัตโนมัติ, สร้าง PR อัตโนมัติ หรือแม้กระทั่งสร้างบอท AI มาช่วยรีวิวกโค้ดให้คุณ จบคอร์สนี้คุณจะได้ไม่แค่ความรู้ แต่จะได้ทักษะในการ "ปั้น" ระบบอัตโนมัติที่ใช้งานได้จริง พร้อมก้าวสู่การทำงานแบบมืออาชีพอย่างเต็มตัว

วัตถุประสงค์:

- เข้าใจแนวคิดพื้นฐานของ CI/CD และ Workflow Automation
- สามารถติดตั้งและใช้งาน Jenkins สำหรับสร้าง Pipeline ได้
- เรียนรู้การใช้งาน GitHub Actions เพื่อ Build & Deploy อัตโนมัติ
- ใช้ n8n เชื่อมต่อระบบต่างๆ และสร้าง Automation Workflow ได้
- สามารถออกแบบและปรับใช้ระบบ CI/CD ที่เหมาะสมกับงานจริง

กลุ่มเป้าหมาย:

- Web Developers (Frontend/Full-stack) ที่ต้องการเพิ่มทักษะด้าน AI และ LLMs เข้าไปในสายงาน
- JavaScript/TypeScript Developers ที่ต้องการสร้างแอปพลิเคชัน AI ที่ใช้งานได้จริง
- Solution Architects หรือ Tech Leads ที่ต้องการเข้าใจกระบวนการสร้าง AI Chatbot เพื่อนำไปประยุกต์ใช้กับระบบขององค์กร
- ผู้ที่สนใจสร้าง Product AI และต้องการเรียนรู้ Tech Stack ที่ทันสมัยและครบวงจร
- ทีมภายในองค์กร (IT/ผลิตภัณฑ์/ซัพพอร์ต/เซลล์) ที่อยากมีออกถาม-ตอบจากคู่มือ/สเปก/นโยบาย
- อาจารย์/วิทยากร/ที่ปรึกษา ที่ต้องการเวิร์กช็อป RAG แบบนำไปสอนและปรับใช้ได้ทันที

ความรู้พื้นฐาน:

- มีความเข้าใจพื้นฐานการทำงานของ Web Application เล็กน้อย
- มีความเข้าใจในการใช้ Command Line เบื้องต้น
- ใช้งาน Git และ GitHub พื้นฐานได้ (ถ้าไม่มีก็สามารถเรียนได้)
- เคยเขียนโปรแกรมภาษาใดภาษาหนึ่งมาบ้าง (เช่น Python, JavaScript, Java)
- ไม่จำเป็นต้องมีประสบการณ์การใช้งาน Jenkins, GitHub Actions หรือ n8n มาก่อน

ระยะเวลาในการอบรม:

- 18 ชั่วโมง (3 วัน)



วิทยากรผู้สอน:

- อาจารย์สามิตร ไทยม

เนื้อหาการอบรม:

Section 1: พื้นฐานและการติดตั้งเครื่องมือ DevOps CI/CD

- ติดตั้ง Visual Studio Code พร้อมส่วนเสริม (Extension) ที่จำเป็น
- ติดตั้ง Java JDK
- ติดตั้ง Git
- ติดตั้ง Docker Desktop

Section 2: พื้นฐาน Docker

- บทนำสู่ Docker
- พื้นฐานคำสั่งใน Docker
- การสร้าง Docker Image
- การ Deploy Docker Container
- การจัดการ Docker Container

Section 3: พื้นฐาน CI/CD และ Git/GitHub

- CI/CD คืออะไร ทำไมเราถึงต้องใช้
- ทำความรู้จัก Continuous Integration, Continuous Delivery, Continuous Deployment
- ทบทวนการใช้งาน Git & GitHub ที่จำเป็นสำหรับ CI/CD (Branching, Pull Request)
- สร้างโปรเจกต์ตัวอย่างและเตรียมความพร้อมบน GitHub

Section 4: เริ่มต้นกับ Jenkins Server

- Jenkins คืออะไร และมีหน้าที่อะไรใน CI/CD Pipeline
- การติดตั้ง Jenkins Server แบบง่ายๆ ด้วย Docker
- สำรวจหน้า Dashboard และเมนูต่างๆ ที่สำคัญของ Jenkins
- โครงสร้างและการทำงานของ Jenkins Pipeline
- การเขียน Jenkinsfile เบื้องต้น
- Integrate Jenkins กับ GitHub Repository
- สร้าง CI Pipeline แรกด้วย Jenkins เพื่อดึงโค้ดจาก GitHub มาทดสอบ
- Lab: สร้าง Pipeline สำหรับ Build & Test Project



Section 5: สร้าง Workflow อัตโนมัติด้วย Jenkins Pipeline

- เตรียมโครงสร้างโปรเจกต์ Java Spring Boot สำหรับการ deploy บน docker container
- สร้าง CI Pipeline ใน Jenkins เพื่อ deploy Java Spring Boot framework
- เตรียมโปรเจกต์ python Flask API บน docker container
- สร้าง CI Pipeline ใน Jenkins เพื่อ deploy Flask API framework
- เตรียมโปรเจกต์ Node.js Express API framework บน docker container
- สร้าง CI Pipeline ใน Jenkins เพื่อ deploy Node.js Express API framework

Section 6: พื้นฐาน GitHub Actions

- GitHub Actions คืออะไร ต่างจาก Jenkins อย่างไร
- องค์ประกอบหลัก: Workflows, Jobs, Steps, Actions
- เรียนรู้โครงสร้างและองค์ประกอบของไฟล์ YAML สำหรับ Workflow
- ใช้ GitHub Marketplace Actions
- ตั้งค่า GitHub Actions สำหรับ CI/CD ของโปรเจกต์จริง
- สร้าง Workflow ให้ GitHub Actions ทำการ Test อัตโนมัติทุกครั้งที่มีการ Push code

Section 7: สร้าง Workflow อัตโนมัติด้วย GitHub Actions

- สร้าง CI Pipeline ใน GitHub Actions เพื่อ deploy Java Spring Boot framework
- สร้าง CI Pipeline ใน GitHub Actions เพื่อ deploy Flask API framework
- สร้าง CI Pipeline ใน GitHub Actions เพื่อ deploy Node.js Express API framework

Section 8: การเชื่อมต่อกับ n8n

- Workflow Automation คืออะไร? รู้จักกับ n8n
- ติดตั้ง N8N ด้วย Docker และทำความเข้าใจหน้าตาโปรแกรม
- ส่วนประกอบของ N8N: Nodes, Credentials, Workflows
- สร้าง Workflow เชื่อมต่อ GitHub, Jenkins และระบบอื่นๆ
- ตั้งค่าการแจ้งเตือนอัตโนมัติผ่าน Slack/Email/Line/Discord เมื่อ Deploy สำเร็จ
- สร้าง Pull Request (PR) อัตโนมัติสำหรับอัปเดต Dependencies
- สร้างบอท AI ช่วยรีวิวโค้ดเบื้องต้น

Section 9: Integration & Real-World Pipeline

- การนำเครื่องมือทั้งหมดมาทำงานร่วมกัน
- สร้าง Pipeline สมบูรณ์: Push code > GitHub Actions ทำงาน > สั่งให้ n8n ส่งข้อความแจ้งเตือน "เริ่ม Build"
- สร้าง Pipeline ที่ซับซ้อนขึ้น: GitHub Actions สั่ง Trigger ไปที่ Jenkins Job เพื่อทำการ Deploy
- Best Practices และการนำไปใช้จริงในองค์กร
- ทำ Mini Project สร้าง CI/CD Flow ของตนเอง
- สรุปเนื้อหาทั้งหมด, Q&A และแนวทางการนำไปต่อยอด