



สถาบันไอทีจีเนียส
IT GENIUS INSTITUTE

COURSE ID
DOP-10

COURSE OUTLINE · รายละเอียดหลักสูตรอบรม

DEVOPS / CI/CD

DevOps for Full Stack Developers

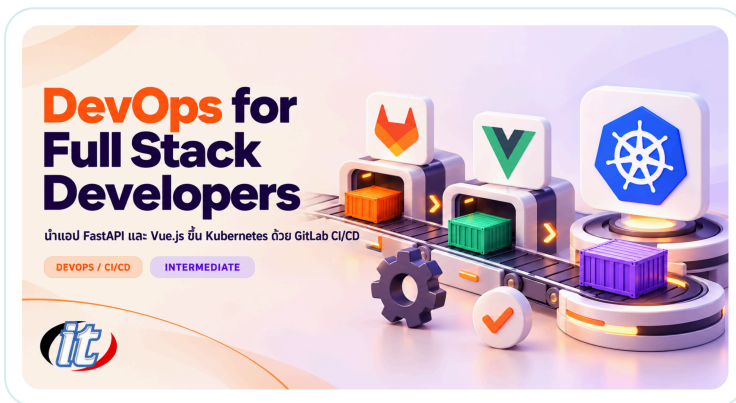
นำแอป FastAPI และ Vue.js ขึ้น Kubernetes อัตโนมัติด้วย GitLab CI/CD

● 24 ชั่วโมง

● 4 วัน

● พื้นฐาน Python, JavaScript และ Git

● ระดับ Intermediate



ระยะเวลา: 4 วัน

24 ชั่วโมง (4 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

ทีมพัฒนาถูกคาดหวังให้ส่งมอบซอฟต์แวร์ได้เร็วและบ่อยขึ้น นักพัฒนา Full Stack จึงไม่ได้รับผิดชอบเพียงการเขียนโค้ด แต่ต้องเข้าใจว่าโค้ดของตนถูก Build, Test และนำขึ้น Kubernetes อย่างไร ปัญหาที่พบบ่อยคือรันบนเครื่องได้แต่ขึ้นระบบไม่ได้ Pipeline ช้าและเปราะบาง และนักพัฒนาต้องรอทีม Ops ทุกครั้งที่ต้องการ Deploy

หลักสูตรนี้มองจากมุมมองของนักพัฒนา เราผู้เรียนนำแอป FastAPI (Backend) และ Vue 3 กับ Vite (Frontend) มาแพ็กเป็น Container ด้วย Docker สร้าง Pipeline อัตโนมัติด้วย GitLab CI/CD และ GitLab Runner เก็บ Image ใน GitLab Container Registry แล้ว Deploy ขึ้น Kubernetes ด้วย Helm 4 และ Gateway API พร้อม Environment, Review App ต่อ Merge Request และ Rollback ทั้งหมดด้วยแอปบนโปรเจกต์ PulseDesk Service Portal ชุดเดียว จนได้ Pipeline ต้นแบบที่ Push โค้ดแล้วระบบขึ้น Kubernetes ได้เองครบวงจร (อบรม 4 วัน วันละ 6 ชั่วโมง รวม 24 ชั่วโมง ระดับ Intermediate)

เครื่องมือ

GL GitLab CI/CD

FA FastAPI

Vu Vue.js

Dk Docker

K8 Kubernetes

ราคาค่าอบรม · PUBLIC / IN-HOUSE

9,900 บาท / ท่าน

ราคายังไม่รวมภาษีมูลค่าเพิ่ม (VAT 7%) · รวมเอกสารประกอบการอบรม Lab Guide รายวัน โค้ดตั้งต้นของแอป ไฟล์ Pipeline และ Helm Chart และ Certificate · รับจัด In-house ขอใบเสนอราคาได้

สมัครผ่านเว็บไซต์ ลด 10%



เพิ่มเพื่อน
LINE

☎ 02-570-8449

☎ 088-807-9770

💬 @itgenius

🌐 www.itgenius.co.th



01 วัตถุประสงค์ Objectives

- เข้าใจแนวคิด DevOps และบทบาทของนักพัฒนาในวงจร Build, Test, Release และ Deploy
- แพ็กแอป FastAPI และ Vue.js เป็น Container Image ที่เล็ก ปลอดภัย และพร้อมใช้งานจริง
- เขียน .gitlab-ci.yml ที่มีโครงสร้างดี ใช้ Cache, Artifacts, rules และ needs ได้เหมาะสม
- ติดตั้ง ลงทะเบียน และเลือก Executor ของ GitLab Runner ได้ตามบริบทงาน
- เขียน Manifest ของ Kubernetes ที่นักพัฒนาต้องใช้ และเปิดบริการด้วย Gateway API
- ใช้ Helm 4 จัดการการ Deploy หลายสภาพแวดล้อมได้
- สร้าง Pipeline ที่ Deploy ขึ้น Kubernetes อัตโนมัติ พร้อม Environment, Review App และ Rollback

02 กลุ่มเป้าหมาย Target Audience

- นักพัฒนา Full Stack ที่ใช้ Python และ JavaScript หรือ TypeScript และต้องการ Deploy งานของตนเองได้
- นักพัฒนา Backend หรือ Frontend ที่กำลังย้ายระบบขึ้น Kubernetes
- Tech Lead ที่ต้องการวางมาตรฐาน Pipeline ให้ทีมพัฒนา
- ทีมที่ใช้ GitLab อยู่แล้วและต้องการใช้ GitLab CI/CD ให้เต็มประสิทธิภาพ

03 จุดเด่นของหลักสูตร Highlights

- มุมมองของนักพัฒนา สิ่งที่ต้องทำได้เองเพื่อนำโค้ดขึ้นระบบ
- อัปเดตเครื่องมือปัจจุบัน GitLab 19, Kubernetes 1.37 และ Helm 4
- ใช้ Gateway API แทน Ingress NGINX ที่หยุดการดูแลแล้ว
- Pipeline ที่ใช้ได้จริง Test, Build, Scan, Deploy และ Rollback
- Lab ยืดหยุ่น gitlab.com หรือ Self-managed กับ kind

04 ความรู้พื้นฐาน Prerequisites

- เขียน Python ได้ในระดับพื้นฐาน และเคยใช้ Web Framework อย่างน้อย 1 ตัว
- เขียน JavaScript หรือ TypeScript ได้ในระดับพื้นฐาน
- ใช้ Git ได้ทั้ง branch, commit, push และ merge request และใช้ Linux Command Line พื้นฐานได้
- โน้ตบุ๊ก RAM 16 GB ขึ้นไป (แนะนำ 32 GB) พร้อมสิทธิ์ Administrator และบัญชี gitlab.com หรือ GitLab ขององค์กร

05 สิ่งที่ต้องเตรียมและเครื่องมือ Setup & Tools

รายการ	รายละเอียด
แอปพลิเคชัน	Python 3.14 ขึ้นไป, FastAPI, uv, pytest, Vue 3.5, Vite 8, Vitest และ PostgreSQL 18
Container	Docker, Docker Compose และ Nginx สำหรับ Serve Frontend
CI/CD	GitLab 19 (gitlab.com หรือ Self-managed), GitLab Runner และ GitLab Container Registry
Kubernetes	Kubernetes 1.37 uu kind หรือ k3d, kubectl, Gateway API, Helm 4 และ GitLab agent for Kubernetes
Security Scan	Trivy สำหรับสแกนช่องโหว่ของ Container Image ใน Pipeline

หมายเหตุ: หลักสูตรนี้มองจากฝั่งทีมพัฒนาแอป ส่วนมุมมองทีม Platform ที่ใช้ Jenkins, SonarQube และ Pipeline ระดับองค์กร อยู่ในหลักสูตร Enterprise CI/CD with GitLab, Jenkins and Kubernetes และหากต้องการลงลึกเรื่อง Docker เรียน Docker in Practice ได้ ผู้เรียนใช้บัญชี gitlab.com แบบฟรีหรือ GitLab ขององค์กร และรัน Runner กับ Kubernetes บนโน้ตบุ๊กของตนเองฟีเจอร์บางตัว เช่น Protected Environments และ Security Scanning ในตัวของ GitLab ต้องใช้แผน Premium หรือ Ultimate หลักสูตรจึงสาธิตเป็นภาพรวมและใช้ Trivy แทน เนื้อหาจะอิงรุ่นล่าสุด ณ วันอบรม

06 ภาพรวมการเรียนรู้ 4 วัน Big Picture

วัน	หัวข้อหลัก
Day 1 Section 1-5	แนวคิด DevOps จากมุมมองนักพัฒนา, Twelve-Factor App, แพ็ก FastAPI และ Vue.js เป็น Container แบบ Multi-stage และ Non-root, รันทั้งระบบด้วย Docker Compose
Day 2 Section 6-10	เขียน .gitlab-ci.yml, ตั้งตั้งและลงทะเบียน GitLab Runner, Cache, Artifacts, needs และ CI/CD Components, Test และ Coverage uu Merge Request, Build, Scan และ Push Image
Day 3 Section 11-15	Kubernetes ที่นักพัฒนาต้องใช้, ConfigMap, Secret และ Probe, เปิดบริการด้วย Gateway API, Migration ด้วย Job และจัดการการ Deploy หลาย Environment ด้วย Helm 4
Day 4 Section 16-20	GitLab agent for Kubernetes, Environments และ Manual Approval, Review App ต่อ Merge Request, Rollback และ Master Workshop Pipeline ตั้งแต่ Commit ถึง Production



DAY
1

DevOps Foundations และ Container ของ FastAPI กับ Vue.js

เนื้อหาการอบรม · Section 1-5

รวม
6 ชั่วโมง

1 DevOps สำหรับนักพัฒนา

- วงจร Plan, Code, Build, Test, Release, Deploy, Operate และ Monitor
- หลัก Twelve-Factor App ที่ทำให้แอปพร้อมสำหรับ Container และ Kubernetes
- Continuous Integration, Continuous Delivery และ Continuous Deployment ต่างกันอย่างไร
- บทบาทของนักพัฒนากับทีม Platform ในการนำโค้ดขึ้นระบบ

2 Lab: รู้จักแอป PulseDesk

- โครงสร้างโปรเจกต์ FastAPI และการจัดการ Dependency ด้วย uv
- โครงสร้างโปรเจกต์ Vue 3.5 กับ Vite 8 และการตั้งค่า API URL ผ่าน Environment
- Health Endpoint, การอ่าน Config จาก Environment Variables และ Structured Logging
- Lab: รันแอปบนเครื่องและสร้าง Ticket ทดสอบครบ Flow

3 Lab: Containerizing FastAPI

- Dockerfile แบบ Multi-stage สำหรับ Python และการเลือก Slim Base Image
- รันด้วย Non-root User และตั้งค่า Uvicorn สำหรับ Production
- ใช้ .dockerignore และ Layer Caching ให้ Build เร็วขึ้น
- Lab: Build Image ของ pulsedesk-api แล้วเปรียบเทียบกับขนาดก่อนและหลังปรับแต่ง

4 Lab: Containerizing Vue.js

- Build Static Files ด้วย Vite แล้ว Serve ผ่าน Nginx
- ตั้งค่า Nginx สำหรับ SPA (History Mode) และการส่ง Request ไป API
- จัดการ Config ของ Frontend ตอน Runtime แทนการ Build ใหม่ทุก Environment
- Lab: Build Image ของ pulsedesk-web ที่ใช้ Image เดียวได้ทุก Environment

5 Workshop: Docker Compose สำหรับ Local Development

- รวม API, Web และ PostgreSQL 18 ใน Compose เดียว พร้อม Health Check และ Volume
- รัน Migration และ Seed ข้อมูลจำลองตอนเริ่มระบบ
- แก้ปัญหาที่พบบ่อย เช่น Service เริ่มก่อนฐานข้อมูลพร้อม
- Workshop: รัน PulseDesk ทั้งระบบด้วยคำสั่งเดียวและทดสอบครบ Flow



DAY
2

GitLab CI/CD และ GitLab Runner

เนื้อหาการอบรม · Section 6-10

รวม
6 ชั่วโมง

6

พื้นฐาน GitLab CI/CD

- โครงสร้าง .gitlab-ci.yml: stages, jobs, script, image และ variables
- Branch Pipeline, Merge Request Pipeline และ Tag Pipeline
- ควบคุมการทำงานด้วย rules และ workflow
- Lab: Pipeline แรกของ PulseDesk ที่ทำงานเมื่อ Push และเปิด Merge Request

7

Lab: GitLab Runner

- สถาปัตยกรรมของ Runner และประเภท Instance, Group และ Project Runner
- สร้าง Runner ผ่าน UI และลงทะเบียนด้วย Runner Authentication Token
- เปรียบเทียบ Shell, Docker และ Kubernetes Executor และการเลือกใช้ด้วย Tags
- Lab: ลงทะเบียน Runner แบบ Docker Executor บนเครื่องของผู้เรียน

8

Pipeline ที่เร็วและดูง่าย

- Cache สำหรับ Dependency ของ Python และ Node.js กับ Artifacts สำหรับส่งผลลัพธ์ระหว่าง Job
- needs สำหรับรัน Job แบบ DAG ไม่ต้องรอทั้ง Stage
- ลดโค้ดซ้ำด้วย extends, include และ CI/CD Components
- Lab: วัดเวลา Pipeline ก่อนและหลังใช้ Cache และ needs

9

Lab: Test และ Quality ใน Pipeline

- รัน pytest และ Vitest พร้อมรายงานผล Test บน Merge Request
- แสดง Coverage บน Merge Request
- Lint และ Format Check ทั้ง Python และ JavaScript
- Lab: ทำให้ Merge Request ที่ Test ไม่ผ่านถูกล็อกก่อน Merge

10

Lab: Build, Scan และ Push Container Image

- Build Image ใน Pipeline ด้วย Docker-in-Docker หรือ Buildah และข้อควรระวังด้านความปลอดภัย
- Push เข้า GitLab Container Registry ด้วย Predefined Variables
- ตั้ง Tag ตาม Commit SHA และ Version
- สแกนช่องโหว่ด้วย Trivy และให้ Pipeline ล้มเมื่อพบช่องโหว่ระดับสูง
- Workshop: Pipeline ของ PulseDesk ที่ Lint, Test, Build, Scan และ Push ครบทั้งสอง Image



DAY
3

Kubernetes และ Helm สำหรับนักพัฒนา

เนื้อหาการอบรม · Section 11-15

รวม
6 ชั่วโมง

11 Lab: Kubernetes ที่นักพัฒนาต้องรู้

- สถาปัตยกรรม Cluster โดยสรุป และการทำงานผ่าน kubectl
- Pod, Deployment, ReplicaSet และ Service
- เตรียม Cluster สำหรับ Lab ด้วย kind หรือ k3d
- Lab: Deploy pulsedesk-api ด้วย Manifest ที่เขียนเอง

12 Config, Secret และ Health Check

- ConfigMap และ Secret สำหรับ Config ของ FastAPI และ Vue.js
- Liveness, Readiness และ Startup Probe ที่สอดคล้องกับ Health Endpoint ของแอป
- Resource Requests และ Limits และผลการ Schedule
- Lab: เพิ่ม Probe และ Resource ให้ PulseDesk แล้วดูพฤติกรรมเมื่อแอปไม่พร้อม

13 Lab: เปิดบริการด้วย Gateway API

- ทำไมต้องเปลี่ยนจาก Ingress: Ingress NGINX หยุดการดูแลตั้งแต่มีนาคม 2026
- Gateway และ HTTPRoute และการกำหนด Path สำหรับ Web และ API
- ติดตั้ง Gateway Controller ใน Cluster ของ Lab
- Lab: เปิดใช้งาน PulseDesk ผ่าน Gateway API

14 ฐานข้อมูลและ Stateful Workload

- เลือกใช้ PostgreSQL ภายนอก Cluster หรือรันใน Cluster สำหรับ Lab
- PersistentVolumeClaim และการเก็บข้อมูลให้คงอยู่
- รัน Migration ด้วย Kubernetes Job ก่อนปล่อยแอปเวอร์ชันใหม่
- Lab: รัน Migration ของ PulseDesk ด้วย Job

15 Lab: Helm 4 สำหรับนักพัฒนา

- โครงสร้าง Chart, Templates และ values
- แยก values ตาม Environment: dev, staging และ production
- คำสั่ง install, upgrade และ rollback และความเปลี่ยนแปลงสำคัญใน Helm 4 เช่น Server-side Apply
- Workshop: แปลง Manifest ของ PulseDesk เป็น Helm Chart แล้ว Deploy ขึ้น Cluster



DAY
4

Continuous Deployment ที่ Kubernetes และ Master Workshop

เนื้อหาการอบรม · Section 16-20

รวม
6 ชั่วโมง

16 Lab: เชื่อม GitLab กับ Kubernetes

- GitLab agent for Kubernetes และการให้สิทธิ์ Pipeline เข้าถึง Cluster อย่างปลอดภัย
- ใช้ Kubernetes Executor ให้ Job ของ Runner รันเป็น Pod ใน Cluster
- จัดการ Secret ของ Pipeline ด้วย Masked และ Protected Variables
- Lab: ติดตั้ง Agent และ Deploy ด้วย Helm จาก Pipeline ครั้งแรก

17 Environments และ Deployment Strategy

- GitLab Environments สำหรับ Staging และ Production
- Manual Job ก่อนขึ้น Production และภาพรวม Protected Environments
- Rolling Update ของ Kubernetes และการตรวจสอบความพร้อมก่อนรับ Traffic
- Lab: Deploy ไป Staging อัตโนมัติ และขึ้น Production ด้วยการกดยืนยัน

18 Lab: Review App ต่อ Merge Request

- สร้าง Namespace และ URL เฉพาะของแต่ละ Merge Request
- ลบ Review App อัตโนมัติเมื่อ Merge หรือปิด Merge Request
- ให้ผู้ตรวจงานทดลองใช้ฟีเจอร์ใหม่ก่อน Merge
- Lab: เปิด Merge Request แล้วได้ Review App ของ PulseDesk ทันที

19 Rollback และการตรวจสอบหลัง Deploy

- Rollback ด้วย Helm และด้วยการ Re-deploy Image เวอร์ชันก่อนหน้า
- Smoke Test หลัง Deploy
- อ่าน Log และ Event ของ Kubernetes เพื่อหาสาเหตุปัญหา
- Lab: ปลดเวอร์ชันที่ Probe ไม่ผ่านแล้วดูว่า Rolling Update ปกป้องระบบอย่างไร

20 Master Workshop: Pipeline ตั้งแต่ Commit ถึง Production

- ออกแบบ Pipeline ฉบับสมบูรณ์ของ PulseDesk: Test, Build, Scan และ Deploy ผ่าน Helm
- Staging อัตโนมัติ Review App ต่อ Merge Request และ Production ด้วยการยืนยัน
- จำลองเหตุการณ์ปล่อยเวอร์ชันที่มีปัญหาและ Rollback ภายในเวลาที่กำหนด
- นำเสนอ Pipeline ของทีมและสรุปแนวทางนำไปใช้กับโปรเจกต์จริง

07 เรียนจบแล้วได้อะไร Outcomes

- เข้าใจ DevOps จากมุมมองนักพัฒนา โค้ดเดินทางจาก Commit ถึง Production
- แพ็ก Container มีอาชีพ FastAPI และ Vue.js ที่เล็กและปลอดภัย
- เขียน GitLab CI/CD คล่อง Pipeline เร็ว ดูแลง่าย ใช้ซ้ำได้
- Deploy ขึ้น Kubernetes เอง Manifest, Helm Chart และ Gateway API
- ส่งมอบอย่างปลอดภัย Environment, Review App และ Rollback
- มี Pipeline ต้นแบบ นำไปปรับใช้กับโปรเจกต์ของทีมได้

08 สิ่งที่คุณเรียนจะได้รับ Deliverables

- เอกสารประกอบการอบรมฉบับสมบูรณ์ครบทั้ง 4 วัน และ Lab Guide รายวัน
- โค้ดต้นฉบับของ PulseDesk (FastAPI และ Vue.js) พร้อม Dockerfile และไฟล์ Compose
- ไฟล์ .gitlab-ci.yml ฉบับสมบูรณ์ Manifest ของ Kubernetes และ Helm Chart แยกตาม Environment
- ประกาศนียบัตรจบหลักสูตร (Certificate of Completion) และช่องทางสอบถามวิทยากรหลังจบหลักสูตร

09 ข้อมูลผู้สอน Instructor

อ.สามิตร ไทยม ผู้สอนและนักพัฒนาที่สร้างแอปแบบ Full Stack และนำขึ้น Kubernetes ผ่าน CI/CD ในงานจริง จุดเด่นคือการสอนผ่านโปรเจกต์ต่อเนื่องที่คุณเรียนพาแอปของตนเองเดินทางจากเครื่องนักพัฒนาไปถึง Production ที่ละขั้น อธิบายเหตุผลของแต่ละขั้นตอนให้เข้าใจง่าย และให้คุณเรียนลองทำให้ระบบพังแล้วกลับมาเอง เพื่อให้กลับไปวางมาตรฐาน Pipeline ของทีมได้อย่างมั่นใจ

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

9,900 บาท / ท่าน (ยังไม่รวม VAT 7%)

สมัครผ่านเว็บไซต์ ลด 10%

☎ 02-570-8449

☎ 088-807-9770

📞 @itgenius

✉ contact@itgenius.co.th



เพิ่มเพื่อน
LINE