



สถาบันไอทีจีเนียส
IT GENIUS INSTITUTE

COURSE ID
DOP-12

COURSE OUTLINE · รายละเอียดหลักสูตรอบรม

DEVOPS / CI/CD

Enterprise CI/CD with GitLab, Jenkins and Kubernetes

วาง Pipeline มาตรฐานองค์กร ตั้งแต่ Quality Gate ถึง GitOps

- 30 ชั่วโมง
- 5 วัน
- พื้นฐาน Git, Linux, Docker และ Kubernetes เบื้องต้น
- ระดับ Intermediate



ระยะเวลา: 5 วัน

30 ชั่วโมง (5 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

องค์กรส่วนใหญ่มีระบบ CI/CD ใช้งานอยู่แล้ว แต่มักเป็น Pipeline ที่แต่ละทีมเขียนกันเอง ไม่มีมาตรฐานกลาง ไม่มีเกณฑ์คุณภาพโค้ดที่ชัดเจน การสแกนความปลอดภัยทำบ้างไม่ทำบ้าง และการ Deploy ขึ้น Production ยังพึ่งคนเพียงไม่กี่คน หลายองค์กรยังใช้ทั้ง GitLab CI/CD และ Jenkins ควบคู่กัน จึงต้องการคนที่เข้าใจทั้งสองระบบและออกแบบให้ทำงานร่วมกันได้อย่างเป็นระบบ

หลักสูตรนี้มองจากมุมมองของผู้วางมาตรฐาน Pipeline ให้หลายทีม พาผู้เรียนออกแบบ CI/CD ด้วย GitLab CI/CD และ GitLab Runner ลงมือทำ Jenkins Pipeline as Code พร้อม Agent บน Kubernetes วาง SonarQube Quality Gate และ Security Scan เป็นด้านบังคับ จัดการ Artifact และ Container Image แล้ว Deploy ขึ้น Kubernetes ด้วย Helm 4 แบบ Rolling, Blue-Green และ Canary ปิดท้ายด้วย GitOps ด้วย Argo CD บนโปรเจกต์จำลอง LedgerLine Payment Services ชุดเดียวตลอดหลักสูตร (อบรม 5 วัน วันละ 6 ชั่วโมง รวม 30 ชั่วโมง ระดับ Intermediate)

เครื่องมือ

GL GitLab

JK Jenkins

SQ SonarQube

TV Trivy

K8 Kubernetes

ราคาค่าอบรม · PUBLIC / IN-HOUSE

13,900 บาท / ท่าน

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

สมัครผ่านเว็บไซต์ ลด 10%



เพิ่มเพื่อน
LINE

☎ 02-570-8449

☎ 088-807-9770

LINE @itgenius

🌐 www.itgenius.co.th



01 วัตถุประสงค์ Objectives

- ออกแบบสถาปัตยกรรม CI/CD ระดับองค์กรที่มีมาตรฐานกลาง นำกลับมาใช้ซ้ำได้ และตรวจสอบย้อนหลังได้
- ใช้ GitLab CI/CD ขั้นสูง เช่น CI/CD Components, Parent-Child Pipeline และ Merge Request Pipeline
- ออกแบบ GitLab Runner Fleet ที่ปลอดภัยและรองรับหลายทีม
- เขียน Jenkins Declarative Pipeline และ Shared Library พร้อมใช้ Dynamic Agent บน Kubernetes
- ติดตั้งและเชื่อม SonarQube กับ Pipeline พร้อมกำหนด Quality Gate ที่เหมาะกับองค์กร
- ผนวก SAST, Dependency Scan, Container Scan, Secret Detection และ SBOM เข้ากับ Pipeline
- Deploy ขึ้น Kubernetes ด้วย Helm 4 แบบ Rolling, Blue-Green และ Canary พร้อม Rollback
- แยก CI กับ CD ด้วยแนวคิด GitOps และ Promote ระบบระหว่าง Environment ด้วย Argo CD

02 กลุ่มเป้าหมาย Target Audience

- นักพัฒนาที่รับผิดชอบหรือดูแล Pipeline ของทีม
- DevOps Engineer และ Platform Engineer ที่ต้องวางมาตรฐาน CI/CD ให้หลายทีม
- Tech Lead และ Software Architect ที่ต้องกำหนดเกณฑ์คุณภาพโค้ดและกระบวนการ Release
- ทีมที่ใช้ Jenkins อยู่และกำลังพิจารณาย้ายหรือใช้ร่วมกับ GitLab CI/CD
- ผู้ดูแลระบบที่ต้องเตรียม Infrastructure ให้ Pipeline Deploy ขึ้น Kubernetes

03 จุดเด่นของหลักสูตร Highlights

- **GitLab และ Jenkins ในหลักสูตรเดียว** โปรเจกต์เดียวกัน เห็นข้อดีข้อจำกัดชัดเจน
- **คุณภาพเป็นด้านบังคับ** SonarQube Quality Gate และ Security Scan
- **Deploy ครบทุกกลยุทธ์** Rolling, Blue-Green และ Canary พร้อม Rollback
- **มาตรฐานกลางขององค์กร** CI/CD Component และ Shared Library
- **ปิดท้ายด้วย GitOps** แยก CI กับ CD และ Promote ด้วย Argo CD

04 ความรู้พื้นฐาน Prerequisites

- เขียนโปรแกรมได้อย่างน้อย 1 ภาษา และเข้าใจขั้นตอน Build และ Test ของภาษานั้น
- ใช้ Git ได้คล่อง รวมถึง Branch, Merge Request และ Tag และใช้ Linux Command Line ได้ในงานประจำ
- ใช้ Docker ได้ในระดับ Build และ Run Image และรู้จัก Kubernetes เบื้องต้น เช่น Pod, Deployment และ Service
- ไม่จำเป็นต้องเคยใช้ Jenkins, SonarQube, Helm หรือ Argo CD มาก่อน

05 สิ่งที่ต้องเตรียมและเครื่องมือ Setup & Tools

รายการ	รายละเอียด
GitLab	GitLab 19 (บัญชี gitlab.com ของผู้เรียนหรือ GitLab Self-managed ขององค์กร), GitLab Runner แบบ Docker และ Kubernetes Executor
CI และคุณภาพโค้ด	Jenkins LTS (Java 21 หรือ 25) พร้อม Kubernetes Plugin และ JCasC, SonarQube Community Build และ SonarScanner
Security Scan	Semgrep (SAST), Trivy (Dependency, Container, SBOM) และ Secret Detection
Deploy	Kubernetes uu kind หรือ k3d, Helm 4, Gateway API และ Argo CD 3.x
อุปกรณ์ผู้เรียน	Notebook RAM 16 GB ขึ้นไป (แนะนำ 32 GB) พร้อม Docker, Git, VS Code, kubectl และ Helm

หมายเหตุ: หลักสูตรนี้ต่างจาก Continuous Integration and Delivery with Jenkins (5 วัน) ที่เน้น Jenkins เป็นหลัก โดยมองในมุมมองมาตรฐานให้หลายทีม ใช้ทั้ง GitLab และ Jenkins พร้อม SonarQube, Security Gate และ GitOps ผู้เรียนใช้ Notebook และบัญชี GitLab ของตนเองหรือขององค์กร Lab รับบนเครื่องผู้เรียนด้วย kind หรือ k3d Security Scanner ในตัวของ GitLab บางรายการต้องใช้ License ระดับ Ultimate จึงสอนเครื่องมือโอเพนซอร์สควบคู่ ฟีเจอร์ของเครื่องมือเปลี่ยนบ่อย เนื้อหาอ้างอิงเวอร์ชันล่าสุด ณ วันอบรม

06 ภาพรวมการเรียนรู้ 5 วัน Big Picture

วัน	หัวข้อหลัก
Day 1 Section 1-5	สถาปัตยกรรม CI/CD ระดับองค์กร, DORA Metrics, GitLab CI/CD ขั้นสูง, Parent-Child Pipeline, CI/CD Components และ GitLab Runner Fleet
Day 2 Section 6-10	Jenkins uu Kubernetes ด้วย JCasC, Declarative Pipeline, Multibranch, Shared Library, Dynamic Pod Agent และการใช้ร่วมกับ GitLab
Day 3 Section 11-15	SonarQube, Quality Profile และ Quality Gate uu New Code, SAST ด้วย Semgrep, Dependency และ Container Scan ด้วย Trivy, SBOM และ Secret
Day 4 Section 16-20	Versioning และ Registry, Helm 4 Chart มาตรฐาน, Rolling, Blue-Green และ Canary ด้วย Gateway API, Protected Environment และ Rollback
Day 5 Section 21-25	แนวคิด GitOps, Argo CD 3.x, Promote ระหว่าง Dev, Staging และ Production, วัดผลด้วย DORA และ Master Workshop รับมือ Release ที่ล้มเหลว



DAY
1

Enterprise CI/CD Architecture และ GitLab CI/CD ขั้นสูง

เนื้อหาการอบรม · Section 1-5

รวม
6 ชั่วโมง

1 สถาปัตยกรรม CI/CD ระดับองค์กร

- องค์ประกอบของ Pipeline ที่ดี: Build, Test, Quality Gate, Security Scan, Package, Deploy และ Verify
- Branching Strategy (Trunk-based กับ GitFlow) และ Release Strategy ที่สอดคล้องกัน
- ตัวชี้วัด DORA: Deployment Frequency, Lead Time, Change Failure Rate และ Time to Restore
- ภาพรวมโปรเจกต์ LedgerLine Payment Services: payment-api (Python) และ notification-service (Node.js)

2 GitLab CI/CD ขั้นสูง

- rules, workflow และ Merge Request Pipeline สำหรับควบคุมว่า Job ใดรันเมื่อใด
- Variables, Protected Variables และขอบเขตของ CI/CD Job Token รวมถึงความเปลี่ยนแปลงใน GitLab 19
- Environment และ Deployment History สำหรับตรวจสอบย้อนหลัง
- Lab: เขียน .gitlab-ci.yml ของ payment-api ให้รันต่างกันตาม Branch และ Merge Request

3 Lab: Pipeline หลาย Service ที่เร็วขึ้น

- Parent-Child Pipeline และ Multi-project Pipeline สำหรับระบบหลาย Service
- needs และ DAG เพื่อลดเวลาารอระหว่าง Stage
- Cache กับ Artifacts ใช้ต่างกันอย่างไร และการตั้ง Key ให้ Cache ใช้ได้จริง
- Lab: แยก Pipeline ของสอง Service แล้ววัดเวลาก่อนและหลังปรับ

4 Lab: มาตรฐานกลางด้วย CI/CD Components

- สร้าง CI/CD Component และเผยแพร่ใน CI/CD Catalog ภายในองค์กร
- กำหนด Input, การ Version Component และการเรียกใช้จากหลายโปรเจกต์
- แนวทางบังคับใช้ Pipeline มาตรฐานโดยไม่ปิดทางให้ทีมปรับเพิ่มเอง
- Lab: สร้าง Component กลางสำหรับ Build และ Test ที่ใช้ได้ทั้งสอง Service

5 Workshop Day 1: GitLab Runner Fleet

- เลือก Instance, Group และ Project Runner ตามโครงสร้างองค์กร
- Docker Executor กับ Kubernetes Executor และการกำหนด Resource ต่อ Job
- ความปลอดภัยของ Runner: Protected Runner, แยก Runner สำหรับงาน Deploy และความเสี่ยงของ Privileged Mode
- Workshop: ติดตั้ง GitLab Runner แบบ Kubernetes Executor และแยก Runner ที่มีสิทธิ์ Deploy ด้วย Tag และ Protected Branch



DAY
2

Jenkins Pipeline as Code uu Kubernetes

เนื้อหาการอบรม · Section 6-10

รวม
6 ชั่วโมง

6 สถาปัตยกรรม Jenkins และ Configuration as Code

- Controller, Agent, Executor และ Plugin Ecosystem
- ข้อจำกัด Java ของ Jenkins LTS ปัจจุบัน (Java 21 หรือ 25) และการวางแผนอัปเดต
- ติดตั้ง Jenkins uu Kubernetes และตั้งค่าให้ทำงานได้ด้วย Jenkins Configuration as Code (JCasC)
- Lab: ติดตั้ง Jenkins ด้วย JCasC พร้อม Credentials และ Plugin ที่กำหนดไว้ในไฟล์

7 Lab: Declarative Pipeline และ Jenkinsfile

- โครงสร้าง pipeline, agent, stages, steps และ post
- Parameters, Environment, Credentials และ when Condition
- Multibranch Pipeline และการเชื่อม Jenkins กับ GitLab Repository ผ่าน Webhook
- Lab: สร้าง Multibranch Pipeline ของ notification-service

8 Lab: Shared Library สำหรับมาตรฐานกลาง

- โครงสร้าง Shared Library (vars และ src) และการ Version Library
- สร้าง Step กลางสำหรับ Build, Test และ Publish ให้ทุกทีมเรียกใช้
- ทดสอบและปล่อย Library เวอร์ชันใหม่โดยไม่กระทบ Pipeline เดิม
- Lab: ย้ายขั้นตอนซ้ำใน Jenkinsfile ไปเป็น Step กลางใน Shared Library

9 Lab: Dynamic Agent uu Kubernetes

- Kubernetes Plugin และ Pod Template ที่มีหลาย Container ต่อ Build
- จัดการ Workspace, Cache และ Resource ของ Agent
- สิทธิ์ของ Agent uu Cluster และการแยก Namespace สำหรับงาน CI
- Lab: รับ Build และ Test ของ notification-service uu Dynamic Pod Agent

10 Workshop Day 2: GitLab CI/CD กับ Jenkins

- เปรียบเทียบด้าน Pipeline Syntax, การดูแลรักษา, ความปลอดภัย และ Ecosystem
- แนวทางใช้ร่วมกัน เช่น GitLab เป็น Source Control และ Jenkins เป็น CI
- แนวทางย้ายจาก Jenkins ไป GitLab CI/CD ที่ละโปรเจกต์
- Workshop: ส่งสถานะ Build จาก Jenkins กลับไปแสดงบน Merge Request ของ GitLab



DAY
3

Code Quality และ Security Gates

เนื้อหาการอบรม · Section 11-15

รวม
6 ชั่วโมง

11 SonarQube สำหรับองค์กร

- เปรียบเทียบ SonarQube Server (Edition ต่าง ๆ และรุ่น 2026.1 LTA), SonarQube Community Build และ SonarQube Cloud
- ติดตั้ง SonarQube พร้อม PostgreSQL บน Docker หรือ Kubernetes
- แนวคิด Bugs, Vulnerabilities, Security Hotspots, Code Smells, Coverage และ Duplication
- Lab: ติดตั้ง SonarQube Community Build และวิเคราะห์โปรเจกต์แรก

12 Lab: Quality Profile และ Quality Gate

- ปรับ Quality Profile ตามภาษาและมาตรฐานขององค์กร
- แนวคิด New Code และ Quality Gate ที่เข้ากับโค้ดใหม่โดยไม่ขวางงานเดิม
- ส่งผล Coverage จาก pytest และ Test ของ Node.js เข้า SonarQube
- Lab: ตั้ง Quality Gate สำหรับ New Code ของ LedgerLine

13 Lab: เชื่อม SonarQube กับ Pipeline

- SonarScanner ใน GitLab CI/CD และการรอผล Quality Gate ก่อนไปขึ้นถัดไป
- SonarQube Plugin ใน Jenkins และ waitForQualityGate
- การแสดงผลบน Merge Request ตามความสามารถของ Edition ที่ใช้
- Lab: เชื่อม payment-api บน GitLab และ notification-service บน Jenkins

14 Lab: Security Scanning ใน Pipeline

- SAST ด้วย Semgrep และภาพรวม GitLab SAST
- Dependency Scanning และ Container Image Scanning ด้วย Trivy
- สร้าง SBOM และตรวจ Secret ที่หลุดในโค้ด
- Lab: เพิ่ม SAST, Dependency Scan, Container Scan และ SBOM ให้ทั้งสอง Service

15 Workshop Day 3: นโยบายความปลอดภัยของ Pipeline

- กำหนดระดับความรุนแรงที่ยอมรับได้และการจัดการ False Positive อย่างมีหลักฐาน
- ป้องกัน Supply Chain: Pin เวอร์ชัน Image และ Dependency และจำกัดสิทธิ์ Job Token
- Workshop: ส่งโค้ดที่มีช่องโหว่และ Secret แล้วดู Pipeline หยุดตามนโยบาย
- แก้ไขจนผ่าน Quality Gate และ Security Gate แล้วบันทึกผลลง Workbook



DAY
4

Artifact Management และ Kubernetes Deployment Strategies

เนื้อหาการอบรม · Section 16-20

รวม
6 ชั่วโมง

16 Artifact, Versioning และ Registry

- Semantic Versioning, การตั้ง Image Tag และหลัก Build Once, Deploy Many
- GitLab Container Registry และ Package Registry พร้อม Cleanup Policy
- Promote Artifact ระหว่าง Environment โดยไม่ Build ใหม่
- Lab: ตั้งกลยุทธ์ Versioning และ Tag Image ของทั้งสอง Service

17 Lab: Helm 4 สำหรับมาตรฐานการ Deploy

- โครงสร้าง Chart มาตรฐานที่ใช้ได้หลาย Service และการจัด values ตาม Environment
- Library Chart และการตรวจ Chart ด้วย helm lint และ helm template ใน Pipeline
- ความเปลี่ยนแปลงสำคัญใน Helm 4 เช่น Server-side Apply สำหรับการติดตั้งใหม่
- Lab: สร้าง Helm Chart มาตรฐานของ LedgerLine

18 Lab: Rolling Update และ Blue-Green

- Rolling Update: maxSurge, maxUnavailable และ Readiness Probe
- Blue-Green: สลับ Traffic ระหว่างสองเวอร์ชันและ Rollback กันที
- เลือกกลยุทธ์ตามความเสี่ยงของแต่ละ Release
- Lab: Deploy notification-service แบบ Blue-Green แล้วสลับกลับ

19 Lab: Canary ด้วย Gateway API

- แบ่ง Traffic ตามน้ำหนักด้วย HTTPRoute ของ Gateway API
- เกณฑ์ตัดสินใจไปต่อหรือถอย เช่น Error Rate และ Latency
- ขยายสัดส่วน Traffic เป็นขั้นจากใน Pipeline
- Lab: Deploy payment-api แบบ Canary 10% และเพิ่มสัดส่วนทีละขั้น

20 Workshop Day 4: Deploy อย่างปลอดภัยจาก Pipeline

- เชื่อม Pipeline กับ Cluster ด้วย GitLab agent for Kubernetes และ Credentials ของ Jenkins
- Protected Environments, Manual Approval และ Deployment Freeze
- Smoke Test หลัง Deploy และ Rollback อัตโนมัติเมื่อไม่ผ่าน
- Workshop: ปลอ่ย Release ที่ Smoke Test ไม่ผ่านแล้วตรวจว่า Rollback ทำงานถูกต้อง

DAY
5

GitOps ด้วย Argo CD และ Master Workshop

เนื้อหาการอบรม · Section 21-25

รวม
6 ชั่วโมง

21 แนวคิด GitOps

- หลักการ Declarative, Versioned, Pulled Automatically และ Continuously Reconciled
- แยก Application Repository กับ Environment (Config) Repository
- ข้อดีข้อจำกัดของ Push-based Deploy จาก Pipeline กับ Pull-based Deploy ด้วย GitOps
- ออกแบบโครงสร้าง Environment Repository ของ LedgerLine

22 Lab: Argo CD 3.x

- ติดตั้ง Argo CD และสถาปัตยกรรมโดยสรุป
- Application, Project, Sync Policy และ Self-heal
- Promote ระหว่าง Dev, Staging และ Production ผ่าน Merge Request ใน Environment Repository
- Lab: ให้ Argo CD Deploy ทั้งสอง Service ไป Dev และ Staging

23 Lab: เชื่อม CI กับ GitOps

- ให้ Pipeline อัปเดต Image Tag ใน Environment Repository แทนการ Deploy ตรง
- ตรวจสอบสถานะ Sync และแจ้งเตือนเมื่อเกิด Drift
- สิทธิ์ของ Pipeline ต่อ Environment Repository และการตรวจทานก่อน Production
- Lab: ปรับ Pipeline ให้ CI สิ้นสุดที่การเปิด Merge Request ไปยัง Environment Repository

24 วัดผลและขยายผลในองค์กร

- เก็บตัวชี้วัด DORA จาก Pipeline และ Deployment History
- แผนนำ Pipeline มาตรฐานไปใช้กับทีมอื่นทีละขั้น
- Governance: ใครงาน Component, Shared Library และ Quality Gate
- Checklist นกวน Pipeline Blueprint ตั้งแต่ Commit จนถึง Production

25 Master Workshop: Release ที่ล้มเหลว

- ผู้เรียนรวมทีมรับมือสถานการณ์ต่อเนื่องบน Pipeline ที่สร้างสะสมตั้งแต่วันที่ 1-4
- สถานการณ์จำลอง: ช่องโหว่ใหม่ใน Dependency, Quality Gate ไม่ผ่าน และ Canary ที่มี Error Rate สูง
- วิเคราะห์ต้นเหตุ แก้ไข และ Promote ไป Production ผ่าน Argo CD
- นำเสนอแผน Rollout Pipeline มาตรฐานสำหรับองค์กร

07 เรียนจบแล้วได้อะไร Outcomes

- ออกแบบ CI/CD ระดับองค์กรได้ มาตรฐานกลางและวัดผลด้วย DORA
- ใช้ GitLab CI/CD ขั้นสูงได้คล่อง Components, Parent-Child และ Runner Fleet
- ใช้ Jenkins อย่างเป็นระบบ JCasC, Shared Library และ Pod Agent
- บังคับใช้คุณภาพและความปลอดภัย SonarQube, Semgrep, Trivy และ SBOM
- Deploy อย่างมั่นใจ Helm 4, Blue-Green, Canary และ Rollback
- ใช้ GitOps ได้จริง Promote ระหว่าง Environment ด้วย Argo CD

08 สิ่งที่คุณเรียนจะได้รับ Deliverables

- เอกสารประกอบการอบรมครบทั้ง 5 วัน และ Lab Guide รายวัน
- แอปตัวอย่าง LedgerLine ชุด CI/CD Component, Jenkins Shared Library และ Helm Chart ที่ทดสอบใน Lab แล้ว
- Pipeline Blueprint และ Checklist สำหรับนำไปเป็นมาตรฐานกลางของทีม
- ประกาศนียบัตรจบหลักสูตร (Certificate of Completion)

09 ข้อมูลผู้สอน Instructor

อ.สาธิต ไทยม วิทยากรผู้มีประสบการณ์วางระบบ CI/CD และ Kubernetes ในงานจริง สอนผ่านโปรเจกต์ต่อเนื่องที่ผู้เรียนยกระดับ Pipeline ของสอง Service ให้อยู่ใต้มาตรฐานเดียวกันทีละวัน อธิบายเหตุผลเบื้องหลังแต่ละการตัดสินใจ แล้วให้ผู้เรียนพิสูจน์ด้วยการทำให้ Pipeline หยุดและกู้คืนเอง เพื่อให้กลับไปวางมาตรฐานให้ทีมของตนเองได้อย่างมั่นใจ

ราคาค่าอบรม 13,900 บาท/ท่าน · สนใจ IN-HOUSE ขอใบเสนอราคา

13,900 บาท / ท่าน (ยังไม่รวม VAT 7%)

สมัครผ่านเว็บไซต์ ลด 10%

☎ 02-570-8449

LINE @itgenius

☎ 088-807-9770

✉ contact@itgenius.co.th



เพิ่มเพื่อน
LINE