สถาบันไอทีจีเนียส
IT GENIUS INSTITUTE

COURSE ID
SVC-02
DEVOPS / CI/CD

COURSE OUTLINE · รายละเอียดหลักสูตรอบรม

Continuous Integration and Delivery with Jenkins

สร้าง CI/CD แบบ Pipeline as Code ด้วย Jenkins, Docker และ GitHub

- 30 ชั่วโมง
- 5 วัน
- พื้นฐาน Git และ Linux 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

ทีมพัฒนาซอฟต์แวร์ที่ยัง Build, Test และ Deploy ด้วยมือมักเจอปัญหาเดิมซ้ำ ๆ ทั้งโค้ดที่ทำงานได้บนเครื่องหนึ่งแต่พังบนอีกเครื่อง การปล่อยเวอร์ชันใหม่ที่ช้าและเสี่ยง และการไล่หาว่าการเปลี่ยนแปลงไหนทำให้ระบบเสีย Jenkins เป็นเครื่องมือ CI/CD แบบโอเพนซอร์สที่องค์กรจำนวนมากใช้อยู่ แต่หลายแห่งยังตั้งค่าแบบ Freestyle Job ผ่านหน้าเว็บทีละจุด ทำให้ติดตามการเปลี่ยนแปลงไม่ได้ ทำซ้ำยาก และดูแลยากเมื่อจำนวนโปรเจกต์เพิ่มขึ้น

หลักสูตรนี้เป็นฉบับปรับปรุงใหม่ที่เน้น Pipeline as Code บน Jenkins LTS รุ่นปัจจุบัน พาผู้เรียนตั้งแต่แนวคิด CI/CD และ Docker ติดตั้งและดูแล Jenkins อย่างปลอดภัย เขียน Jenkinsfile แบบ Declarative ทำ Multibranch Pipeline กับ GitHub ใช้ Agent และ Docker Agent Build และ Test พร้อมรายงานผล สร้างและ Push Docker Image ตั้ง Quality Gate และสแกนความปลอดภัย แยกโค้ดซ้ำเป็น Shared Libraries วางกลยุทธ์ Deploy แบบ Blue/Green จัดการ Jenkins ด้วย Configuration as Code และฝึก Deploy โปรเจกต์ Python และ Java ขึ้นเซิร์ฟเวอร์และ Cloud (อบรม 5 วัน วันละ 6 ชั่วโมง รวม 30 ชั่วโมง ระดับ Intermediate)

เครื่องมือ
- Jk Jenkins LTS
- JF Jenkinsfile
- Dk Docker
- GH GitHub
- SQ SonarQube

ราคาค่าอบรม · PUBLIC / IN-HOUSE

15,900 บาท / ท่าน

สมัครผ่านเว็บไซต์ ลด 10%

ราคายังไม่รวมภาษีมูลค่าเพิ่ม (VAT 7%) · รวมเอกสารประกอบการอบรม Lab Guide ซอร์สโค้ดตัวอย่างครบทุก Workshop และ Certificate · รับจัด In-house ขอใบเสนอราคาได้

เพิ่มเพื่อน LINE

02-570-8449 · 088-807-9770 · @itgenius · www.itgenius.co.th

01 วัตถุประสงค์ Objectives

- เข้าใจแนวคิด Continuous Integration, Continuous Delivery และ Continuous Deployment
- ทำ Container ให้แอปด้วย Docker และจัดการ Image บน Registry ได้
- ติดตั้ง ตั้งค่า และรักษาความปลอดภัยของ Jenkins LTS ได้
- เขียน Pipeline as Code ด้วย Jenkinsfile แบบ Declarative และทำ Multibranch Pipeline กับ GitHub ได้
- ใช้ Agent และ Docker Agent Build และ Test พร้อมเก็บรายงานและ Artifact ได้
- ตั้ง Quality Gate สแกนช่องโหว่ และจัดการ Credentials อย่างปลอดภัยได้
- สร้าง Shared Libraries และจัดการ Jenkins ด้วย Configuration as Code ได้
- Deploy แอปแบบอัตโนมัติด้วยกลยุทธ์ Blue/Green ขึ้นเซิร์ฟเวอร์และ Cloud ได้

02 กลุ่มเป้าหมาย Target Audience

- นักพัฒนาซอฟต์แวร์ที่ต้องการทำ Build, Test และ Deploy แบบอัตโนมัติ
- วิศวกร DevOps และผู้ดูแลระบบที่ดูแล Jenkins ขององค์กร
- หัวหน้าทีมและ QA ที่ต้องการวางกระบวนการปล่อยซอฟต์แวร์ที่เชื่อถือได้
- ครู อาจารย์ และนักศึกษาที่สนใจงานด้าน CI/CD

03 จุดเด่นของหลักสูตร Highlights

- **Pipeline as Code** Jenkinsfile แบบ Declarative แทน Freestyle Job
- **Jenkins LTS รุ่นปัจจุบัน** Java 21 หรือ 25 และ Pipeline Graph View
- **Docker ตลอดสาย** Docker Agent, Build และ Push Image
- **ปลอดภัยตั้งแต่ต้น** Credentials, Quality Gate และการสแกนช่องโหว่
- **Deploy ของจริง** Python และ Java ขึ้นเซิร์ฟเวอร์และ Cloud

04 ความรู้พื้นฐาน Prerequisites

- เข้าใจการเขียนโปรแกรมและขั้นตอนพัฒนาซอฟต์แวร์เบื้องต้น
- ใช้ Git ได้ เช่น commit, branch, push และ pull
- ใช้คำสั่งพื้นฐานบน Linux ได้
- มีบัญชี GitHub และติดตั้ง Docker Desktop บนเครื่องได้ หากต้องการทำ Lab บน AWS ต้องมีบัญชีของผู้เรียนเอง

05 สิ่งที่ต้องเตรียมและเครื่องมือ Setup & Tools

รายการ	รายละเอียด
อุปกรณ์	Notebook RAM 16 GB ขึ้นไป เชื่อมต่ออินเทอร์เน็ต ติดตั้ง Docker Desktop หรือ VM Linux ได้
CI/CD Server	Jenkins LTS รุ่นล่าสุดบน Docker (Java 21 หรือ 25) พร้อม Pipeline, Git, Docker Pipeline และ Configuration as Code Plugin
Source Control	Git และบัญชี GitHub ของผู้เรียน พร้อม Webhook และ Pull Request
Build และคุณภาพ	Maven, Python และ pytest, SonarQube Community Build และ Trivy สำหรับสแกน Image
ปลายทาง Deploy	เซิร์ฟเวอร์ Linux ที่รัน Docker Compose และ AWS EC2 ผ่านบัญชีของผู้เรียน

หมายเหตุ: ปรับปรุงใหม่ปี 2026 จากหลักสูตร Continuous Integration and Delivery with Jenkins เดิม: เปลี่ยนจากการตั้งค่า Job ผ่านหน้าเว็บมาเป็น Pipeline as Code ด้วย Jenkinsfile แบบ Declarative ทั้งหมด ใช้ Jenkins LTS รุ่นปัจจุบันที่รันบน Java 21 หรือ 25 เพิ่ม Multibranch Pipeline, Docker Agent, Shared Libraries, Configuration as Code, Quality Gate และการสแกนช่องโหว่ ใช้ Pipeline Graph View แทน Blue Ocean ที่เลิกพัฒนาแล้ว และเปลี่ยน Workshop Java จาก Tomcat ที่ตั้งเองมาเป็น Container บน AWS EC2 เครื่องมือหลักเป็นโอเพนซอร์สใช้ได้ฟรี ส่วน AWS ใช้บัญชีของผู้เรียนเอง ซึ่งอาจมีค่าใช้จ่ายหากเกินสิทธิ์ใช้ฟรี เนื้อหาอิงเวอร์ชัน LTS ล่าสุด ณ วันอบรม

06 ภาพรวมการเรียน 5 วัน Big Picture

วัน	หัวข้อหลัก
Day 1-2 Section 1-8	แนวคิด CI/CD, Git Workflow, Docker และการทำ Container ให้แอป, ติดตั้งและดูแล Jenkins LTS อย่างปลอดภัย แล้วเริ่มเขียน Pipeline as Code ด้วย Jenkinsfile แบบ Declarative
Day 3 Section 9-12	Multibranch Pipeline กับ GitHub, Agent และ Docker Agent, Build และ Test พร้อมรายงานผล, Build และ Push Docker Image ด้วย Credentials อย่างปลอดภัย
Day 4-5 Section 13-20	Quality Gate และการสแกนความปลอดภัย, Shared Libraries, กลยุทธ์ Deploy แบบ Blue/Green, Workshop โปรเจกต์ Python และ Java ขึ้น Cloud, Configuration as Code, Monitoring และ Capstone

DAY 1

CI/CD, Git และ Docker

เนื้อหาการอบรม · Section 1-4

รวม
6 ชั่วโมง

1 แนวคิด CI/CD และ DevOps

- Continuous Integration, Continuous Delivery และ Continuous Deployment ต่างกันอย่างไร
- ประโยชน์ของ CI/CD ต่อความเร็วและความเสถียรของการปล่อยซอฟต์แวร์
- ขั้นตอนของ Pipeline: Checkout, Build, Test, Scan, Package และ Deploy
- ภาพรวมเครื่องมือ CI/CD และจุดที่ Jenkins เหมาะสม
- ตัวอย่างกรณีการใช้ CI/CD ในองค์กร

2 Lab: Git Workflow สำหรับ CI

- Branching Strategy แบบ Trunk-based และ Feature Branch
- Pull Request, Code Review และ Status Check
- Tag และ Semantic Versioning สำหรับการปล่อยเวอร์ชัน
- ตั้งค่า Repository บน GitHub พร้อม Branch Protection
- Lab: เตรียม Repository ของแอปตัวอย่างสำหรับทั้งหลักสูตร

3 Lab: Docker สำหรับงาน CI/CD

- Image, Container, Layer และ Registry
- เขียน Dockerfile ที่ดี: Multi-stage Build, .dockerignore และ Non-root User
- จัดการ Container, Volume และ Network
- รันหลายบริการด้วย Docker Compose
- Lab: รันแอปพร้อมฐานข้อมูลด้วย Docker Compose

4 Lab: Dockerize แอปพลิเคชัน

- ทำ Container ให้แอป Python และแอป Java ที่ Build ด้วย Maven
- ตั้งชื่อและติด Tag ของ Image ด้วย Git Commit และเลขเวอร์ชัน
- Push Image ขึ้น Docker Hub หรือ GitHub Container Registry
- ลดขนาด Image และเลือก Base Image ที่ปลอดภัย
- Lab: Build, Tag และ Push Image ของแอปตัวอย่างด้วยมือก่อนทำให้อัตโนมัติ

DAY 2

ติดตั้ง Jenkins และ Pipeline as Code

เนื้อหาการอบรม · Section 5-8

รวม **6 ชั่วโมง**

5 Lab: ติดตั้ง Jenkins LTS

- สถาปัตยกรรม Jenkins: Controller, Agent, Executor และ Plugin
- ติดตั้ง Jenkins LTS บน Docker พร้อม Volume เก็บข้อมูล และข้อกำหนด Java 21 หรือ 25
- Setup Wizard, ผู้ดูแลระบบคนแรก และ Plugin ที่แนะนำ
- ทำความรู้จักหน้าจอ Jenkins และ Pipeline Graph View
- Lab: เปิดใช้งาน Jenkins พร้อมสำหรับทั้งหลักสูตร

6 Lab: ดูแลและรักษาความปลอดภัยของ Jenkins

- จัดการ Plugin อัปเดต และตรวจคำเตือนด้านความปลอดภัย
- ผู้ใช้ กลุ่ม และการกำหนดสิทธิ์แบบ Matrix หรือ Role-based
- เก็บรหัสผ่าน Token และ SSH Key ใน Jenkins Credentials
- ไม่รัน Build บน Controller และตั้งค่าความปลอดภัยที่สำคัญ
- Lab: สร้างผู้ใช้ตามบทบาทและทดสอบสิทธิ์

7 Lab: Jenkinsfile แรกแบบ Declarative

- Freestyle Job กับ Pipeline as Code ต่างกันอย่างไร และทำไมควรเก็บ Pipeline ไว้ใน Git
- โครงสร้าง pipeline, agent, stages, steps และ post
- Pipeline Job แบบ Pipeline script from SCM
- ใช้ Snippet Generator และ Replay ช่วยเขียนและทดลอง
- Lab: Pipeline แรกที่ Checkout, Build และ Test แอปตัวอย่าง

8 Lab: Declarative Pipeline เชิงลึก

- environment, parameters และ options เช่น timeout และ retry
- เงื่อนไข when ตาม Branch, Tag และ Parameter
- Stage แบบ parallel และ matrix สำหรับทดสอบหลายสภาพแวดล้อม
- ขั้นตอนอนุมัติด้วย input ก่อน Deploy
- Lab: Pipeline ที่ทดสอบแบบขนานและรอการอนุมัติก่อนขึ้น Production

DAY 3

Multibranch, Agent และ Docker Pipeline

เนื้อหาการอบรม · Section 9-12

รวม
6 ชั่วโมง

9 Lab: Multibranch Pipeline กับ GitHub

- สร้าง Multibranch Pipeline ที่ค้นหา Branch และ Pull Request อัตโนมัติ
- เชื่อม GitHub ด้วย Credentials และตั้ง Webhook ให้ Build ทุกครั้งที่ Push
- ส่งสถานะ Build กลับไปที่ Pull Request
- กำหนดพฤติกรรมต่างกันระหว่าง Branch หลัก Branch ย่อย และ Pull Request
- Lab: เปิด Pull Request แล้วดู Pipeline ทำงานและรายงานผล

10 Lab: Agent และ Docker Agent

- เพิ่ม Agent ผ่าน SSH หรือแบบ Inbound และกำหนด Label
- ติดตั้ง Docker บน Jenkins Agent และเชื่อมกับ Controller
- ใช้ agent { docker } ให้แต่ละ Stage รันใน Container ของตัวเอง
- Cache ของ Dependency และ Workspace เพื่อให้ Build เร็วขึ้น
- Lab: ใช้ Container ต่างกันสำหรับ Build ฝั่ง Java และฝั่ง Python

11 Lab: Build, Test และรายงานผล

- Build ด้วย Maven และ pip พร้อมรัน Unit Test
- เก็บรายงาน JUnit และแสดงแนวโน้มผลการทดสอบ
- วัด Code Coverage และแสดงผลใน Jenkins
- archiveArtifacts, stash และ unstash ระหว่าง Stage
- Lab: ทำให้ Pipeline ล้มเมื่อ Test ไม่ผ่านและแจ้งสาเหตุให้ชัด

12 Lab: Build และ Push Docker Image ใน Pipeline

- Build Image ใน Pipeline และติด Tag ด้วย Git Commit
- Login Registry ด้วย withCredentials โดยไม่ให้รหัสผ่านโผล่ใน Log
- Push Image ขึ้น Docker Hub หรือ GitHub Container Registry
- สแกนช่องโหว่ของ Image ด้วย Trivy และกำหนดเกณฑ์ให้ Pipeline ล้ม
- Lab: Pipeline ที่ Build, สแกน และ Push Image อัตโนมัติ

DAY 4

คุณภาพ ความปลอดภัย และการ Deploy

เนื้อหาการอบรม · Section 13-16

รวม
6 ชั่วโมง

13 Lab: Quality Gate และความปลอดภัยของ Pipeline

- วิเคราะห์โค้ดด้วย SonarQube และรอผล Quality Gate ใน Pipeline
- ตรวจ Dependency ที่มีช่องโหว่และไลบรารีที่ล้าสมัย
- หลักการ Least Privilege สำหรับ Credentials และ Agent
- ป้องกันการรันโค้ดไม่น่าไว้ใจจาก Pull Request ภายนอก
- Lab: ทำให้ Pipeline หยุดเมื่อคุณภาพโค้ดไม่ผ่านเกณฑ์

14 Lab: Shared Libraries

- ทำไมต้องแยกขั้นตอนที่ใช้ซ้ำออกจาก Jenkinsfile
- โครงสร้าง Shared Library: vars, src และ resources
- เขียน Custom Step และ Pipeline Template สำหรับหลายโปรเจกต์
- กำหนดเวอร์ชันของ Library และนำไปใช้ด้วย @Library
- Lab: ย้ายขั้นตอน Build และ Push Image ไปเป็น Shared Library

15 Lab: กลยุทธ์การ Deploy

- Deploy ไปยังเซิร์ฟเวอร์ผ่าน SSH และ Docker Compose
- แยกสภาพแวดล้อม Dev, Staging และ Production พร้อมขั้นอนุมัติ
- Blue/Green Deployment และการสลับทราฟฟิก
- Health Check หลัง Deploy และการ Rollback อัตโนมัติ
- Lab: Deploy แบบ Blue/Green และทดลอง Rollback

16 Workshop: Deploy โปรเจกต์ Python

- วางแผน Pipeline สำหรับแอป Python ที่รันใน Container
- Test ด้วย pytest, Build Image และสแกนช่องโหว่
- Deploy Container ไปยังเซิร์ฟเวอร์ที่เป็น Jenkins Agent
- Workshop: Pipeline ครบวงจรจาก Push โค้ดถึงแอปทำงานบนเซิร์ฟเวอร์

DAY 5

Cloud, Configuration as Code และ Capstone

เนื้อหาการอบรม · Section 17-20

รวม
6 ชั่วโมง

17 Workshop: Deploy โปรเจกต์ Java ขึ้น AWS

- ภาพรวมการ Deploy โปรเจกต์ Java ที่ Build ด้วย Maven
- เตรียม EC2 และผู้ใช้ IAM ที่มีสิทธิ์เท่าที่จำเป็นในบัญชี AWS ของผู้เรียน
- เก็บ Key ของ AWS ใน Jenkins Credentials อย่างปลอดภัย
- Pipeline ที่ Build, Test, สร้าง Image และ Deploy ไปยัง EC2 อัตโนมัติ
- ปิดและลบทรัพยากรหลังทดลองเพื่อควบคุมค่าใช้จ่าย

18 Lab: Jenkins Configuration as Code

- ตั้งค่า Jenkins ด้วยไฟล์ YAML ผ่าน Configuration as Code Plugin
- ติดตั้ง Plugin และสร้าง Job ตั้งต้นแบบอัตโนมัติ
- รัน Jenkins ด้วย Docker Compose ที่สร้างใหม่ได้ทุกเมื่อ
- สำรองข้อมูล กู้คืน และแนวทางอัปเกรด Jenkins LTS อย่างปลอดภัย
- Lab: สร้าง Jenkins ใหม่ทั้งเครื่องจากไฟล์ใน Git

19 Lab: Monitoring และการแก้ปัญหา Pipeline

- อ่าน Console Log และ Pipeline Graph View เพื่อหาจุดที่ล้ม
- ปัญหาที่พบบ่อย: สิทธิ์ Docker, Workspace เต็ม และ Credentials ผิด
- เก็บ Metrics ด้วย Prometheus Plugin และแสดงผลบน Grafana
- แจ้งผล Build ทางอีเมลหรือแชตของทีม
- Lab: แก้ Pipeline ที่เตรียมไว้ให้เสียหลายจุด

20 Workshop: Capstone CI/CD ครบวงจร

- ออกแบบ Pipeline สำหรับแอปของผู้เรียนหรือแอปตัวอย่างตั้งแต่ Pull Request ถึง Production
- รวม Test, Quality Gate, สแกนความปลอดภัย, Docker Image และ Blue/Green Deploy
- ใช้ Shared Library และ Configuration as Code ให้ทำซ้ำได้
- นำเสนอผลงาน สรุป Best Practices และแนวทางนำไปใช้ในองค์กร

07 เรียนจบแล้วได้อะไร Outcomes

- **เข้าใจ CI/CD อย่างเป็นระบบ** ตั้งแต่ Pull Request ถึง Production
- **ดูแล Jenkins ได้** ติดตั้ง ความปลอดภัย และ JCasC
- **เขียน Jenkinsfile ได้** Declarative, Multibranch และ Shared Library
- **ใช้ Docker ใน Pipeline ได้** Docker Agent, Build และ Push Image
- **คุมคุณภาพและความปลอดภัยได้** Quality Gate และการสแกนช่องโหว่
- **Deploy อัตโนมัติได้** Blue/Green ขึ้นเซิร์ฟเวอร์และ AWS

08 สิ่งที่ผู้เรียนจะได้รับ Deliverables

- เอกสารประกอบการอบรมครบทั้ง 5 วัน และ Lab Guide ทีละขั้นตอน
- ตัวอย่าง Jenkinsfile, Shared Library และไฟล์ Configuration as Code ที่นำไปใช้ต่อได้
- ซอร์สโค้ดแอปตัวอย่าง Python และ Java พร้อม Dockerfile และ Docker Compose
- ประกาศนียบัตรจบหลักสูตร (Certificate of Completion) และช่องทางสอบถามวิทยากรหลังจบหลักสูตร

09 ข้อมูลผู้สอน Instructor

อ.สามิตร โกยม ผู้สอนและผู้ปฏิบัติงานด้าน DevOps ที่วางระบบ CI/CD ด้วย Jenkins และ Docker ในงานจริง เน้นให้ผู้เรียนสร้าง Pipeline ของตัวเองตั้งแต่ต้นจนถึงการ Deploy อธิบายเหตุผลของแต่ละขั้น และฝึกแก้ปัญหาที่พบบ่อยในองค์กร เพื่อให้นำรูปแบบที่ได้ไปใช้กับโปรเจกต์จริงได้ทันที

ราคาค่าอบรม 15,900 บาท/ท่าน · สนใจ IN-HOUSE ขอใบเสนอราคา

15,900 บาท / ท่าน (ยังไม่รวม VAT 7%)

สมัครผ่านเว็บไซต์ ลด 10%

02-570-8449 · 088-807-9770

@itgenius · contact@itgenius.co.th

เพิ่มเพื่อน LINE