สถาบันไอทีจีเนียส
IT GENIUS INSTITUTE

COURSE ID
OTC-11
DEVOPS / KUBERNETES

COURSE OUTLINE · รายละเอียดหลักสูตรอบรม

GitOps with Argo CD

วางระบบ Continuous Delivery บน Kubernetes ให้ Git เป็นแหล่งความจริงเพียงแห่งเดียว

- 18 ชั่วโมง
- 3 วัน
- พื้นฐาน Git, Docker และ Kubernetes 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

GitOps คือแนวทางทำ Continuous Delivery ที่ใช้ Git เป็นแหล่งความจริงเพียงแห่งเดียวของระบบ ทุกการเปลี่ยนแปลงของแอปและ Infrastructure เกิดผ่าน Commit และ Pull Request ที่ตรวจสอบย้อนหลังได้ แล้วให้ตัวควบคุมใน Cluster ดึงค่าจาก Git มาปรับระบบให้ตรงอยู่เสมอ Argo CD เป็นเครื่องมือ GitOps สำหรับ Kubernetes ที่องค์กรใช้กันแพร่หลาย และรุ่น 3 ได้ปรับค่าเริ่มต้นหลายอย่างให้ปลอดภัยและเร็วขึ้นกว่ารุ่นก่อน

หลักสูตรนี้พาผู้เรียนวางระบบ GitOps ด้วย Argo CD 3 ตั้งแต่ติดตั้งบน Kubernetes Cluster สร้าง Application จาก YAML, Helm และ Kustomize ตั้ง Sync Policy, Sync Wave และ Hook จัดการ Project, RBAC และ Private Repository จากนั้นขยายไปหลายสภาพแวดล้อมและหลาย Cluster ด้วย ApplicationSet จัดการ Secret อย่างปลอดภัย และเชื่อม CI Pipeline ด้วย GitHub Actions ให้การ Deploy เกิดขึ้นจาก Git อัตโนมัติ ปิดท้ายด้วย Workshop วางระบบ GitOps ครบวงจร (อบรม 3 วัน วันละ 6 ชั่วโมง รวม 18 ชั่วโมง ระดับ Intermediate)

เครื่องมือ
- AR Argo CD 3
- K8 Kubernetes
- H Helm
- KZ Kustomize
- GH GitHub Actions

ราคาค่าอบรม · PUBLIC / IN-HOUSE

9,500 บาท / ท่าน

สมัครผ่านเว็บไซต์ ลด 10%

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

เพิ่มเพื่อน LINE

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

01 วัตถุประสงค์ Objectives

- เข้าใจหลักการ GitOps และความต่างจากการ Deploy แบบ Push ด้วย CI
- ติดตั้งและตั้งค่า Argo CD 3 บน Kubernetes พร้อมใช้งานทั้ง Web UI และ CLI
- สร้าง Application จาก YAML, Helm และ Kustomize และจัดโครงสร้าง Repository ตามสภาพแวดล้อม
- ตั้ง Sync Policy, Sync Wave, Hook และ Diffing ให้เหมาะกับงานจริง
- ควบคุมสิทธิ์ด้วย Project, RBAC และ SSO และเชื่อม Private Repository อย่างปลอดภัย
- Deploy หลายสภาพแวดล้อมและหลาย Cluster ด้วย ApplicationSet
- จัดการ Secret แบบ GitOps โดยไม่เก็บรหัสผ่านไว้ใน Git ตรง ๆ
- เชื่อม CI Pipeline กับ Argo CD และแก้ปัญหาที่พบบ่อยได้

02 กลุ่มเป้าหมาย Target Audience

- DevOps Engineer และ Platform Engineer ที่ดูแลการ Deploy บน Kubernetes
- ผู้ดูแลระบบและทีม Infrastructure ที่ต้องการให้การเปลี่ยนแปลงตรวจสอบย้อนหลังได้
- นักพัฒนาที่ต้องการเข้าใจเส้นทางของโค้ดตั้งแต่ Commit จนขึ้นระบบจริง
- ทีมที่ใช้ Argo CD รุ่น 2 อยู่และต้องการอัปเดตแนวทางเป็นรุ่น 3
- นักศึกษาและผู้สนใจที่มีพื้นฐาน Git และ Kubernetes

03 จุดเด่นของหลักสูตร Highlights

- **อัปเดตเป็น Argo CD 3** ค่าเริ่มต้นใหม่และแนวทางปัจจุบัน
- **Helm และ Kustomize** จัด Repository ตามสภาพแวดล้อม
- **Sync อย่างมีระเบียบ** Prune, Self-Heal, Sync Wave และ Hook
- **หลาย Cluster** ApplicationSet และ Project กับ RBAC
- **CI ต่อ CD ครบวงจร** GitHub Actions อัปเดต Image Tag อัตโนมัติ

04 ความรู้พื้นฐาน Prerequisites

- ใช้งาน Git ได้ เช่น Commit, Branch, Merge และ Pull Request บน GitHub หรือ GitLab
- เข้าใจ Docker และ Container Image เบื้องต้น
- รู้จัก Kubernetes พื้นฐาน เช่น Pod, Deployment, Service และใช้คำสั่ง kubectl ได้
- โน้ตบุ๊กที่ติดตั้ง Docker Desktop หรือ Docker Engine ได้ RAM 16 GB ขึ้นไป พร้อมสิทธิ์ติดตั้งโปรแกรม และบัญชี GitHub ของตนเอง

05 สิ่งที่ต้องเตรียมและเครื่องมือ Setup & Tools

รายการ	รายละเอียด
อุปกรณ์	Notebook Windows, macOS หรือ Linux RAM 16 GB ขึ้นไป เชื่อมต่ออินเทอร์เน็ต
Kubernetes	Cluster บนเครื่องด้วย kind หรือ k3d ผ่าน Docker และ kubectl
GitOps	Argo CD 3 (Web UI และ argocd CLI), Helm และ Kustomize
Git และ CI	บัญชี GitHub ของผู้เรียน, GitHub Actions และ Container Registry
Secret และ Monitoring	Sealed Secrets หรือ External Secrets Operator, Prometheus และ Grafana

หมายเหตุ: หลักสูตรนี้ปรับปรุงใหม่ปี 2026 จากฉบับเดิมที่สอนบน Argo CD รุ่น 2 เนื้อหาเปลี่ยนเป็น Argo CD 3 ทั้งหมด รวมค่าเริ่มต้นใหม่อย่างการติดตาม Resource ด้วย Annotation และการติดตั้งแบบ Server-side Apply เพิ่ม ApplicationSet, การจัดการ Secret, SSO, Notification และการแก้ปัญหา และเปลี่ยนตัวอย่าง CI หลักเป็น GitHub Actions โดยยังอธิบายแนวทางเดียวกันสำหรับ GitLab CI ทุกเครื่องมือเป็น Open Source ใช้ได้ฟรี ผู้เรียนใช้บัญชี GitHub ของตนเอง และรัน Cluster บนเครื่องของตนเอง เนื้อหาอ้างอิงรุ่นล่าสุดในวันอบรม

06 ภาพรวมการเรียน 3 วัน Big Picture

วัน	หัวข้อหลัก
Day 1 Section 1-5	แนวคิด GitOps และ OpenGitOps, เตรียม Kubernetes Cluster บนเครื่อง, ติดตั้ง Argo CD 3 และ CLI, สร้าง Application ทั้งแบบ YAML, Web UI และ CLI แล้ว Deploy ด้วย Helm และ Kustomize
Day 2 Section 6-10	จัดโครงสร้าง Repository ตามสภาพแวดล้อม, Sync Policy กับ Prune และ Self-Heal, Sync Wave และ Hook, Diffing Customization, Project, RBAC, SSO และการเชื่อม Private Repository อย่างปลอดภัย
Day 3 Section 11-15	ApplicationSet กับหลาย Cluster, จัดการ Secret แบบ GitOps, ต่อ CI Pipeline ด้วย GitHub Actions ให้อัปเดต Image Tag อัตโนมัติ, Notification และ Monitoring แล้วปิดท้ายด้วย Workshop GitOps ครบวงจร

DAY 1

แนวคิด GitOps และเริ่มต้น Argo CD 3

เนื้อหาการอบรม · Section 1-5

รวม
6 ชั่วโมง

1 GitOps คืออะไร และทำไมต้องใช้

- หลัก OpenGitOps: Declarative, Versioned, Pulled Automatically และ Continuously Reconciled
- Push Deployment ด้วย CI เทียบกับ Pull Deployment แบบ GitOps
- แยก CI กับ CD และบทบาทของ Git, Registry และ Cluster
- ภาพรวมเครื่องมือ GitOps และจุดเด่นของ Argo CD
- Lab: เตรียม Repository ตัวอย่างบน GitHub ของตนเอง

2 Lab: เตรียม Kubernetes Cluster บนเครื่อง

- สร้าง Cluster ด้วย kind หรือ k3d ผ่าน Docker
- ทบทวน kubectl, Namespace, Deployment และ Service
- ติดตั้ง Ingress Controller สำหรับเข้าถึงแอปจากเบราว์เซอร์
- จัดการหลาย Context ใน kubeconfig
- Lab: Deploy แอปตัวอย่างด้วย kubectl apply แบบเดิมเพื่อเปรียบเทียบ

3 Lab: ติดตั้ง Argo CD 3

- สถาปัตยกรรม: API Server, Repo Server, Application Controller และ ApplicationSet Controller
- ทางเลือกการติดตั้ง: Manifest แบบ Non-HA, HA, Core และ Helm Chart
- ติดตั้งด้วย kubectl apply แบบ Server-side และดึงรหัสผ่าน Admin เริ่มต้น
- เข้า Web UI ด้วย Port-forward หรือ Ingress และติดตั้ง argocd CLI
- Lab: ล็อกอิน เปลี่ยนรหัสผ่าน และสำรวจหน้าจอหลัก

4 Application แรกและวงจร Sync

- โครงสร้าง Application: source, destination และ project
- สร้าง Application ด้วย YAML แบบ Declarative, Web UI และ CLI
- สถานะ Sync และ Health และการอ่าน Resource Tree
- ติดตาม Resource ด้วย Annotation ซึ่งเป็นค่าเริ่มต้นใน Argo CD 3
- Lab: แก้ค่าใน Git แล้วดู Argo CD ปรับ Cluster ให้ตรง

5 Lab: Deploy ด้วย Helm และ Kustomize

- Argo CD ตรวจชนิด Source อย่างไร: Directory, Helm และ Kustomize
- Helm Chart จาก Git หรือ Helm Repository พร้อม values ไฟล์แยกตามสภาพแวดล้อม
- Kustomize Base และ Overlay สำหรับ dev, staging และ prod
- Application ที่ใช้หลาย Source ร่วมกัน
- Lab: Deploy แอปเดียวกันสามสภาพแวดล้อมด้วย Overlay

DAY 2

Sync ขั้นสูง ความปลอดภัย และการจัดการสิทธิ์

เนื้อหาการอบรม · Section 6-10

รวม
6 ชั่วโมง

6 จัดโครงสร้าง Repository และการติดตามเวอร์ชัน

- แยก Repository ของโค้ดแอปกับ Repository ของค่า Config
- Monorepo เทียบกับหลาย Repository และรูปแบบ App of Apps
- targetRevision: Branch, Tag, Commit SHA และช่วงเวอร์ชันของ Helm Chart
- Promote จาก dev ไป prod ด้วย Pull Request
- Lab: วางโครงสร้าง Repository สำหรับสามสภาพแวดล้อม

7 Lab: Sync Policy และ Sync Options

- Automated Sync, Prune และ Self-Heal ต่างกันอย่างไร
- Sync Options ที่ใช้บ่อย เช่น CreateNamespace, ServerSideApply และ PruneLast
- Sync Window สำหรับกำหนดช่วงเวลาที่อนุญาตให้ Deploy
- Rollback จากประวัติ และเหตุผลที่ควรย้อนผ่าน Git
- Lab: ทดลองแก้ Resource ใน Cluster ตรง ๆ แล้วดู Self-Heal ทำงาน

8 Sync Wave, Hook และ Diffing Customization

- Sync Phase: PreSync, Sync, PostSync และ SyncFail
- Sync Wave สำหรับกำหนดลำดับ เช่น ฐานข้อมูลก่อนแอป
- Hook สำหรับ Migration ฐานข้อมูลและ Smoke Test
- ignoreDifferences สำหรับฟิลด์ที่ระบบอื่นแก้ เช่น replicas จาก HPA
- Lab: Job ทำ Migration ก่อน Deploy เวอร์ชันใหม่

9 Lab: Project, RBAC และ SSO

- AppProject: จำกัด Source Repository, Destination และชนิด Resource
- Project Role และ Token สำหรับงานอัตโนมัติ
- RBAC ใน argocd-rbac-cm และการแยกสิทธิ์ทีมพัฒนากับทีมปฏิบัติการ
- เชื่อม SSO ผ่าน OIDC หรือ Dex กับผู้ให้บริการตัวตนขององค์กร
- Lab: สร้าง Project ให้ทีมที่ Deploy ได้เฉพาะ Namespace ของตน

10 Private Repository และ Credential

- เชื่อม Private Git Repository ผ่าน HTTPS Token และ SSH Key
- Credential Template สำหรับหลาย Repository ในองค์กรเดียวกัน
- Helm Repository และ OCI Registry แบบ Private
- เก็บ Credential เป็น Kubernetes Secret แบบ Declarative
- Lab: Deploy จาก Private Repository ด้วย Deploy Key

DAY 3

หลาย Cluster, CI Pipeline และ Workshop

เนื้อหาการอบรม · Section 11-15

รวม
6 ชั่วโมง

11 Lab: ApplicationSet และหลาย Cluster

- เพิ่ม Remote Cluster ด้วย argocd cluster add และ Cluster Secret
- ApplicationSet Generator: List, Cluster และ Git Directory
- Matrix Generator สำหรับหลายแอปคูณหลาย Cluster
- Go Template ใน ApplicationSet และการป้องกันการลบแอปโดยไม่ตั้งใจ
- Lab: สร้างแอปให้ทุกสภาพแวดล้อมอัตโนมัติจากโฟลเดอร์ใน Git

12 จัดการ Secret แบบ GitOps

- ทำไมห้ามเก็บรหัสผ่านใน Git แม้เป็น Private Repository
- Sealed Secrets: เข้ารหัสก่อน Commit
- External Secrets Operator ดึงค่าจาก Vault หรือ Secret Manager ของ Cloud
- เปรียบเทียบข้อดีข้อจำกัดของแต่ละแนวทาง
- Lab: Deploy แอปที่ใช้รหัสผ่านฐานข้อมูลผ่าน Sealed Secrets

13 Lab: เชื่อม CI Pipeline กับ Argo CD

- ลำดับงาน CI/CD: Test, Build Image, Push Registry แล้วอัปเดตค่าใน Git
- GitHub Actions สร้าง Image และแก้ Image Tag ใน Repository Config อัตโนมัติ
- แนวทางเดียวกันบน GitLab CI และรู้จัก Argo CD Image Updater
- Preview Environment จาก Pull Request
- Lab: Push โค้ดแล้วดูเวอร์ชันใหม่ขึ้น Cluster โดยไม่ต้องสั่ง kubectl

14 Notification, Monitoring และการแก้ปัญหา

- Argo CD Notifications ส่งสถานะ Sync ไป Slack, Teams หรือ Email
- Metrics ของ Argo CD กับ Prometheus และ Dashboard ใน Grafana
- อาการที่พบบ่อย: OutOfSync ไม่จบ, Health ค้าง และ Sync ล้มเหลว
- สำรองค่าด้วย argocd admin export และแนวทางอัปเกรดรุ่น
- Checklist ก่อนใช้ Argo CD กับระบบจริง

15 Workshop: วางระบบ GitOps ครบวงจร

- โจทย์: แอปเว็บพร้อมฐานข้อมูลที่ Deploy สามสภาพแวดล้อม
- ออกแบบ Repository, Project และ ApplicationSet ให้ทีมใช้ร่วมกัน
- ต่อ CI ให้ Promote เวอร์ชันผ่าน Pull Request และจัดการ Secret อย่างปลอดภัย
- นำเสนอผลงาน สรุปแนวทางนำไปใช้ในองค์กร และใช้ AI Assistant ช่วยเขียน Manifest อย่างมีการตรวจทาน

07 เรียนจบแล้วได้อะไร Outcomes

- **เข้าใจ GitOps** Pull Deployment และ Reconciliation
- **ติดตั้งและดูแล Argo CD 3** Web UI, CLI และการอัปเกรด
- **Deploy ได้หลายรูปแบบ** YAML, Helm และ Kustomize
- **คุม Sync อย่างปลอดภัย** Prune, Self-Heal, Wave และ Hook
- **ขยายสู่หลาย Cluster** ApplicationSet, Project และ RBAC
- **ต่อ CI กับ CD ได้** GitHub Actions และ Secret Management

08 สิ่งที่ผู้เรียนจะได้รับ Deliverables

- เอกสารประกอบการอบรมฉบับสมบูรณ์ครบทั้ง 3 วัน และ Lab Guide ทีละขั้นตอน
- Repository ตัวอย่างของแอปและค่า Config สำหรับสามสภาพแวดล้อม ครบทุก Lab
- ไฟล์ Workflow ของ GitHub Actions และ Manifest ของ ApplicationSet สำหรับนำไปปรับใช้
- ประกาศนียบัตรจบหลักสูตร (Certificate of Completion) และช่องทางสอบถามวิทยากรหลังจบหลักสูตร

09 ข้อมูลผู้สอน Instructor

อ.สามิตร โกยม ผู้สอนและวิศวกรที่วางระบบ CI/CD และดูแล Kubernetes ในงานจริง จุดเด่นคือการสอนจาก Repository ตัวอย่างชุดเดียวที่ค่อย ๆ ต่อเติมจากแอปเดียวจนเป็นหลายสภาพแวดล้อมและหลาย Cluster อธิบายเหตุผลของแต่ละการตั้งค่า และให้ผู้เรียนลงมือแก้ปัญหาที่พบบ่อยด้วยตนเองจนวางระบบ GitOps ได้จริง

ราคาค่าอบรม 9,500 บาท/ท่าน · สนใจ IN-HOUSE ขอใบเสนอราคา

9,500 บาท / ท่าน (ยังไม่รวม VAT 7%)

สมัครผ่านเว็บไซต์ ลด 10%

02-570-8449 · 088-807-9770

@itgenius · contact@itgenius.co.th

เพิ่มเพื่อน LINE