



ไอทีจีเนียส เอ็นจิเนียริง

IT GENIUS ENGINEERING CO., LTD.

COURSE ID

DOP-02

COURSE OUTLINE · รายละเอียดหลักสูตรอบรม

DEVOPS / OBSERVABILITY

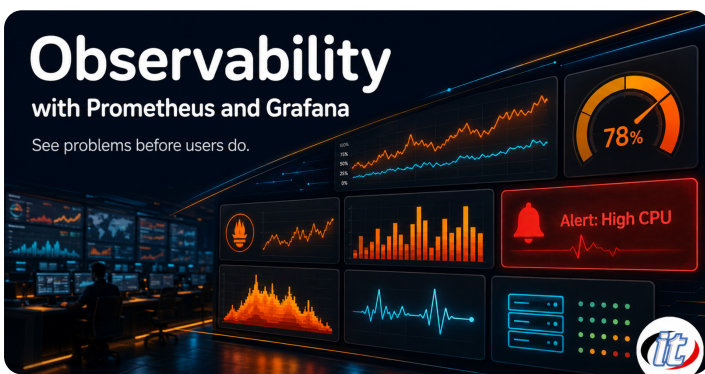
สร้างระบบ Observability ด้วย Prometheus และ Grafana

18 ชั่วโมง

3 วัน

ควรมีพื้นฐาน Linux

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

เมื่อระบบล่มหรือช้า คำถามแรกคือรู้ได้อย่างไรและรู้เร็วแค่ไหน องค์กรจำนวนมากยังรู้ปัญหาจากการที่ลูกค้าโทรมาแจ้ง ไม่ใช่จากระบบเฝ้าระวังของตนเอง Prometheus และ Grafana เป็นชุดเครื่องมือมาตรฐานแบบโอเพนซอร์สที่ทีม DevOps และ SRE ทั่วโลกใช้เก็บ Metric แสดงผลเป็น Dashboard และแจ้งเตือนก่อนที่ผู้ใช้จะได้รับผลกระทบ

หลักสูตรนี้พาผู้เรียนสร้างระบบ Observability ตั้งแต่ศูนย์ ครอบคลุมการติดตั้ง Prometheus และการเก็บ Metric จากเซิร์ฟเวอร์ คอนเทนเนอร์ ฐานข้อมูล และแอปพลิเคชันด้วย Exporter การเขียน PromQL เพื่อค้นหาและคำนวณค่า การออกแบบ Dashboard ที่อ่านง่ายด้วย Grafana การตั้งกฎแจ้งเตือนที่ไม่รบกวนเกินจำเป็นด้วย Alertmanager ไปจนถึงการเก็บข้อมูลระยะยาวและการต่อยอดสู่ Log และ Trace ทุกหัวข้อมือทำจริงบนเครื่อง Lab (อบรม 3 วัน วันละ 6 ชั่วโมง รวม 18 ชั่วโมง ระดับ Intermediate)

เครื่องมือ

Bash Shell

systemd

OpenSSH

Networking

APT / DNF

ราคาค่าอบรม · PUBLIC / IN-HOUSE

10,500 บาท / ท่าน

ราคายังไม่รวมภาษีมูลค่าเพิ่ม (VAT 7%) · รวมเอกสาร Lab Guide ไฟล์ตั้งค่าและ Dashboard ตัวอย่าง อาหารกลางวันและอาหารว่าง และ Certificate · เครื่อง Lab เตรียมให้ในคลาส · รับจัด In-house ขอใบเสนอราคาได้



เพิ่มเพื่อน
LINE

02-570-8449

088-807-9770

@itgenius

www.itgenius.co.th

01 วัตถุประสงค์ Objectives

- เข้าใจแนวคิด Observability และความต่างระหว่าง Metric, Log และ Trace
- ติดตั้งและตั้งค่า Prometheus พร้อมเข้าใจสถาปัตยกรรมการทำงานแบบ Pull
- เก็บ Metric จากเซิร์ฟเวอร์ คอนเทนเนอร์ และฐานข้อมูลด้วย Exporter ต่าง ๆ
- ตั้งค่า Service Discovery ให้ระบบค้นหาเป้าหมายที่ต้องเฝ้าระวังอัตโนมัติ
- เขียน PromQL เพื่อค้นหา คำนวณอัตรา และสรุปข้อมูล Metric
- ออกแบบ Dashboard ด้วย Grafana ให้อ่านง่ายและสื่อสารกับทีมได้
- ฝัง Metric ในแอปพลิเคชันของตนเองและเฝ้าระวังตามหลัก Four Golden Signals
- ตั้งกฎแจ้งเตือนและจัดการการแจ้งเตือนด้วย Alertmanager อย่างมีประสิทธิภาพ
- วางแผนการเก็บข้อมูลระยะยาวและต่อยอดสู่การเก็บ Log และ Trace

02 กลุ่มเป้าหมาย Target Audience

- DevOps Engineer และ Site Reliability Engineer ที่ต้องดูแลความเสถียรของระบบ
- ผู้ดูแลระบบและทีม Infrastructure ที่ต้องการระบบเฝ้าระวังที่ใช้งานได้จริง
- นักพัฒนาที่ต้องการเข้าใจพฤติกรรมและประสิทธิภาพของแอปพลิเคชันบน Production
- ทีมที่ใช้ Docker และ Kubernetes และต้องการเฝ้าระวังคอนเทนเนอร์
- หัวหน้าทีมเทคนิคที่ต้องการวางระบบแจ้งเตือนและตัวชี้วัดความพร้อมใช้งาน

03 จุดเด่นของหลักสูตร Highlights

- **สร้างระบบจริงตั้งแต่ศูนย์** ติดตั้ง เก็บ Metric ทำ Dashboard และแจ้งเตือน
- **เขียน PromQL เป็น** ค้นหาและคำนวณค่าที่ต้องการได้เอง
- **Dashboard ที่ใช้งานจริง** ออกแบบให้ทีมอ่านแล้วตัดสินใจได้
- **แจ้งเตือนอย่างมีประสิทธิภาพ** ลด Alert Fatigue ด้วยกฎที่ออกแบบดี
- **ต่อยอด Log และ Trace** เห็นภาพรวม Observability ครบสามด้าน



04 ความรู้พื้นฐาน Prerequisites

- ใช้งาน Linux ในระดับคำสั่งพื้นฐานได้ (แนะนำผ่านหลักสูตร Linux Administration มาก่อน)
- เข้าใจแนวคิดเครือข่ายพื้นฐาน เช่น พอร์ต HTTP และ DNS
- หากเคยใช้ Docker มาบ้างจะช่วยให้ทำ Lab ได้ราบรื่นขึ้น
- ไม่จำเป็นต้องเขียนโปรแกรมเป็น แต่ควรอ่านไฟล์ตั้งค่า YAML ได้

05 สิ่งที่ต้องเตรียมและเครื่องมือ Setup & Tools

รายการ	รายละเอียด
อุปกรณ์	Notebook เชื่อมต่ออินเทอร์เน็ตได้ (แนะนำ RAM 8 GB ขึ้นไป)
เครื่อง Lab	เครื่อง Linux สำหรับติดตั้งระบบเฟิร์มแวร์ (สถาบันเตรียมให้)
เครื่องมือหลัก	Prometheus, Grafana, Alertmanager, Node Exporter และ cAdvisor
สภาพแวดล้อม	Docker และ Docker Compose สำหรับติดตั้งชุดเครื่องมืออย่างรวดเร็ว
เอกสารอ้างอิง	ไฟล์ตั้งค่าตัวอย่าง ชุด PromQL ที่ใช้บ่อย และ Dashboard ตัวอย่างของหลักสูตร

หมายเหตุ: สถาบันเตรียมเครื่อง Lab และสภาพแวดล้อมให้ผู้เรียนแต่ละคนติดตั้งระบบเฟิร์มแวร์ด้วยตนเองตลอดหลักสูตร ผู้เรียนสามารถนำโจทย์ระบบขององค์กรมาปรึกษาได้ โดยไม่ต้องนำ Credential หรือข้อมูลลับของระบบจริงเข้ามาในคลาส เครื่องมือทั้งหมดเป็นโอเพนซอร์สที่นำไปติดตั้งในองค์กรได้โดยไม่มีค่าลิขสิทธิ์

06 ภาพรวมการเรียนรู้ 3 วัน Big Picture

วัน	หัวข้อหลัก
Day 1 Section 1-4	แนวคิด Observability, การติดตั้ง Prometheus, การเก็บ Metric ด้วย Exporter และ Service Discovery
Day 2 Section 5-8	ภาษา PromQL, การทำ Dashboard ด้วย Grafana, การเฟิร์มแวร์ระบบและแอปพลิเคชัน
Day 3 Section 9-12	Alerting และ Alertmanager, การเก็บข้อมูลระยะยาว, Log และ Trace เบื้องต้น และ Capstone



DAY
1

พื้นฐาน Prometheus และการเก็บ Metric

เนื้อหาการอบรม · Section 1-4

รวม

6 ชั่วโมง

1 แนวคิด Observability และ Monitoring

- ความต่างระหว่าง Monitoring และ Observability และเหตุผลที่องค์กรต้องมีทั้งสองอย่าง
- สามเสาหลักของ Observability: Metric, Log และ Trace และการเลือกใช้ให้เหมาะสมกับปัญหา
- ตัวชี้วัดสำคัญ: Four Golden Signals และแนวคิด SLI, SLO และ Error Budget
- ภาพรวมสถาปัตยกรรม Prometheus, Grafana และ Alertmanager ที่ทำงานร่วมกัน

2 การติดตั้งและตั้งค่า Prometheus

- สถาปัตยกรรมแบบ Pull และ Time Series Database ของ Prometheus
- การติดตั้ง Prometheus ด้วย Docker Compose และโครงสร้างไฟล์ prometheus.yml
- แนวคิด Metric, Label และ Time Series และการตั้งชื่อ Metric ที่ดี
- Lab: ติดตั้ง Prometheus และเฝ้าระวังตัวมันเองเป็นเป้าหมายแรก

3 การเก็บ Metric ด้วย Exporter

- การเก็บ Metric ของเซิร์ฟเวอร์ด้วย Node Exporter (CPU, Memory, Disk, Network)
- การเฝ้าระวังคอนเทนเนอร์ด้วย cAdvisor และการเก็บ Metric ของฐานข้อมูลและเว็บเซิร์ฟเวอร์
- ชนิดของ Metric: Counter, Gauge, Histogram และ Summary และการเลือกใช้
- Lab: ติดตั้ง Exporter หลายตัวและตรวจสอบว่า Prometheus เก็บข้อมูลได้ครบ

4 Service Discovery และการจัดการเป้าหมาย

- การกำหนดเป้าหมายแบบ Static และข้อจำกัดเมื่อระบบขยายตัว
- Service Discovery จากไฟล์ DNS และคลาวด์ และการใช้กับ Kubernetes
- การใช้ Relabeling เพื่อจัดระเบียบ Label และคัดกรองเป้าหมาย
- Lab: ตั้งค่า Service Discovery ให้เป้าหมายใหม่ถูกเฝ้าระวังอัตโนมัติ



DAY
2

PromQL และ Grafana Dashboard

เนื้อหาการอบรม · Section 5-8

รวม

6 ชั่วโมง

5 ภาษา PromQL เบื้องต้น

- การเลือกข้อมูลด้วย Selector และการกรองด้วย Label Matcher
- ฟังก์ชันที่ใช้บ่อย: rate, increase, sum, avg, max และการใช้ by และ without
- การคำนวณ Percentile จาก Histogram และการหาค่าที่ผิดปกติ
- Lab: เขียน PromQL ตอบคำถามจริง เช่น อัตราการเกิดข้อผิดพลาดและการใช้ทรัพยากร

6 Grafana และการสร้าง Dashboard

- การติดตั้ง Grafana และการเชื่อมต่อ Prometheus เป็น Data Source
- ชนิดของ Panel และการเลือกกราฟให้เหมาะกับข้อมูลแต่ละแบบ
- การใช้ตัวแปร (Variable) ทำ Dashboard ที่เลือกเซิร์ฟเวอร์หรือบริการได้
- Lab: สร้าง Dashboard เฝ้าระวังเซิร์ฟเวอร์และคอนเทนเนอร์ของตนเอง

7 การออกแบบ Dashboard ที่ใช้งานได้จริง

- หลักการออกแบบ Dashboard ให้อ่านแล้วรู้ทันทีว่าระบบปกติหรือมีปัญหา
- การจัดกลุ่มข้อมูลตามผู้ใช้งาน: Dashboard สำหรับผู้บริหาร ทีม Ops และนักพัฒนา
- การนำเข้า Dashboard สำเร็จรูปจากชุมชนและการปรับให้เหมาะกับองค์กร
- Lab: ปรับปรุง Dashboard ให้สื่อสารสถานะระบบได้ในหน้าจอเดียว

8 การเฝ้าระวังแอปพลิเคชัน

- การฝัง Metric ในแอปพลิเคชันด้วย Client Library และการเปิด Endpoint ให้ Prometheus ดึง
- การวัดตามหลัก Four Golden Signals: Latency, Traffic, Errors และ Saturation
- การเฝ้าระวังจากภายนอกด้วย Blackbox Exporter เพื่อจำลองมุมมองผู้ใช้
- Lab: เพิ่ม Metric ในแอปพลิเคชันตัวอย่างและสร้าง Dashboard ติดตามผล



DAY
3

Alerting การเก็บข้อมูล และ Capstone

เนื้อหาการอบรม · Section 9-12

รวม

6 ชั่วโมง

9 การตั้งกฎแจ้งเตือน (Alerting Rules)

- การเขียน Alerting Rule ด้วย PromQL และการกำหนดระยะเวลา for ให้ไม่แจ้งเตือนพรีำเพรื่อ
- การจัดระดับความรุนแรงและการใส่ข้อมูลที่ต้องใช้แก้ปัญหาไว้ในการแจ้งเตือน
- แนวคิดการแจ้งเตือนตามอาการที่ผู้ใช้รู้สึก (Symptom-based Alerting)
- Lab: เขียนกฎแจ้งเตือนสำหรับเซิร์ฟเวอร์และบริการที่เฝ้าระวังอยู่

10 Alertmanager และการส่งการแจ้งเตือน

- การติดตั้งและตั้งค่า Alertmanager และการเชื่อมกับ Prometheus
- การจัดกลุ่ม (Grouping) การระงับ (Silence) และการยับยั้ง (Inhibition) เพื่อลดเสียงรบกวน
- การส่งการแจ้งเตือนเข้าอีเมล, LINE, Slack หรือช่องทางขององค์กร
- Lab: ตั้งค่าส่งการแจ้งเตือนเข้าช่องทางจริงและทดสอบสถานการณ์ระบบล่ม

11 การเก็บข้อมูลระยะยาวและประสิทธิภาพ

- การกำหนดระยะเวลาเก็บข้อมูลและการประเมินพื้นที่จัดเก็บที่ต้องใช้
- แนวทางการเก็บข้อมูลระยะยาวและการทำ Federation เมื่อระบบขยายตัว
- การปรับแต่งประสิทธิภาพ: ความถี่ในการดึงข้อมูล จำนวน Label และปัญหา Cardinality สูง
- การสำรองข้อมูลและการดูแลระบบเฝ้าระวังให้พร้อมใช้งานเสมอ

12 Log, Trace และ Capstone

- การเก็บ Log ด้วย Loki และการดู Log ควบคู่กับ Metric ใน Grafana
- แนวคิด Distributed Tracing และ OpenTelemetry สำหรับตามรอยคำขอข้ามบริการ
- โจทย์รวมยอด: วางระบบ Observability ครบชุดสำหรับบริการหนึ่งตั้งแต่เก็บ Metric ถึงแจ้งเตือน
- Workshop: นำเสนอ Dashboard และกฎแจ้งเตือนที่ออกแบบ พร้อมรับ Feedback



07 เรียนจบแล้วได้อะไร Outcomes

- **ติดตั้งระบบเฝ้าระวังได้เอง** Prometheus, Grafana และ Alertmanager
- **เก็บ Metric ได้ครบทุกชั้น** เซิร์ฟเวอร์ คอนเทนเนอร์ และแอปพลิเคชัน
- **เขียน PromQL เป็น** ตอบคำถามเรื่องประสิทธิภาพได้ด้วยตัวเอง
- **ออกแบบ Dashboard ที่ใช้งานจริง** ทิมอ่านแล้วตัดสินใจได้ทันที
- **ตั้งการแจ้งเตือนที่มีคุณภาพ** รู้ปัญหาก่อนลูกค้าโทรมาแจ้ง
- **ต่อยอด Log และ Trace ได้** เห็นภาพ Observability ครบสามด้าน

08 สิ่งที่คุณเรียนจะได้รับ Deliverables

- เอกสารประกอบการอบรมฉบับสมบูรณ์ และ Lab Guide การติดตั้งระบบเฝ้าระวังทีละขั้นตอน
- ไฟล์ตั้งค่า Prometheus, Alertmanager และ Docker Compose ที่ใช้ในคลาส
- Dashboard ตัวอย่างสำหรับ Grafana และชุด PromQL และกฎแจ้งเตือนที่ใช้บ่อย
- ประกาศนียบัตรจบหลักสูตร (Certificate of Completion) และช่องทางสอบถามวิทยากรหลังจบหลักสูตร

09 ข้อมูลผู้สอน Instructor

อ.สามิตร โทยม ผู้สอนและผู้เชี่ยวชาญด้านโครงสร้างพื้นฐานและ DevOps ที่มีประสบการณ์กว่า 15 ปี วางระบบเฝ้าระวังและระบบอัตโนมัติให้องค์กรจำนวนมาก จุดเด่นคือการอธิบายให้เข้าใจง่าย เชื่อมโยงกับเหตุการณ์ระบบล่มจริงที่เคยเจอ และพาผู้เรียนสร้างระบบเฝ้าระวังจนใช้งานได้ด้วยตัวเอง

ราคาค่าอบรม 10,500 บาท/ท่าน · สนใจ IN-HOUSE ขอใบเสนอราคา

10,500 บาท / ท่าน (ยังไม่รวม VAT 7%)

☎ 02-570-8449

LINE @itgenius

☎ 088-807-9770

✉ contact@itgenius.co.th



เพิ่มเพื่อน
LINE