



Docker in Practice

ใช้ Docker อย่างมืออาชีพ ตั้งแต่ Image ที่ดีจนถึงระบบที่ปลอดภัย

● 18 ชั่วโมง

● 3 วัน

● พื้นฐาน Command Line และเขียนโปรแกรม

● 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

Container เป็นรูปแบบมาตรฐานของการส่งมอบซอฟต์แวร์ ตั้งแต่เครื่องของนักพัฒนา CI/CD Pipeline ไปจนถึง Kubernetes บน Production แต่หลายทีมยังใช้ Docker แคร่ระดับรันไทม์ Image จึงใหญ่ Build ช้า มีช่องโหว่ติดมากับ Base Image รัน Container ด้วยสิทธิ์ root และแก้ปัญหาได้ยากเมื่อระบบมีหลาย Service ขณะเดียวกัน Docker เองก็เปลี่ยนไปมาก เช่น Docker Engine 29 ที่ใช้ containerd image store เป็นค่าเริ่มต้นสำหรับการติดตั้งใหม่ และ Docker Compose v5 ที่ส่งงาน Build ให้ Docker Bake

หลักสูตรนี้พาผู้เรียนใช้ Docker Engine 29, Docker Desktop, BuildKit และ Buildx, Docker Compose v5 รวมถึง Docker Scout และ Trivy ในแบบที่ทีมใช้งานจริง ตั้งแต่กลไกภายในของ Container การเขียน Dockerfile ที่ได้ Image เล็กและ Build เร็ว การจัดการ Network และ Volume การประกอบระบบหลาย Service ด้วย Compose ไปจนถึง Security, Registry, Resource Limit และการ Debug อย่างเป็นระบบ ทั้งหมดต่อยอดบนระบบตัวอย่าง BrewTrack Coffee Ordering ชุดเดียว จนพร้อมส่งเข้า CI/CD และต่อยอดสู่ Kubernetes (อบรม 3 วัน วันละ 6 ชั่วโมง รวม 18 ชั่วโมง ระดับ Intermediate)

เครื่องมือ

Dk Docker

BK BuildKit

Cp Compose

Tv Trivy

k8 Kubernetes

ราคาค่าอบรม · PUBLIC / IN-HOUSE

9,500 บาท / ท่าน

สมัครผ่านเว็บไซต์ ลด 10%

ราคายังไม่รวมภาษีมูลค่าเพิ่ม (VAT 7%) · รวมเอกสารประกอบการอบรม Lab Guide รายวัน Dockerfile และไฟล์ Compose ของระบบตัวอย่างครบทุก Lab และ Certificate · รับจัด In-house ขอใบเสนอราคาได้



เพิ่มเพื่อน
LINE

01 วัตถุประสงค์ Objectives

- เข้าใจกลไกภายในของ Container ทั้ง Namespace, cgroups, Image Layer และ containerd
- เขียน Dockerfile ที่ได้ Image เล็ก Build เร็ว และดูแลง่ายด้วย Multi-stage Build และ BuildKit
- Build Image หลายสถาปัตยกรรม (amd64 และ arm64) ด้วย Buildx ได้
- จัดการ Network, Volume และ Configuration ของ Container ได้อย่างถูกต้อง
- ประกอบระบบหลาย Service ด้วย Docker Compose พร้อม Healthcheck, Profile และ Watch
- เสริมความปลอดภัยด้วย Non-root, Secret และการสแกนช่องโหว่ของ Image
- วางมาตรฐาน Registry, Tag และ Versioning ของ Image สำหรับทีม
- กำหนด Resource Limit, Logging และ Debug ปัญหาของ Container ได้อย่างเป็นระบบ

02 กลุ่มเป้าหมาย Target Audience

- นักพัฒนา Backend, Frontend และ Full Stack ที่ต้องการใช้ Docker ในงานจริง
- DevOps Engineer และ System Engineer ที่ดูแล Container ขององค์กร
- Tech Lead ที่ต้องการวางมาตรฐานการใช้ Container ของทีม
- ผู้ที่ใช้ Docker ระดับพื้นฐานแล้วและต้องการยกระดับสู่การใช้งานแบบมืออาชีพ
- ผู้ที่เตรียมต่อยอดไปสู่ CI/CD และ Kubernetes

03 จุดเด่นของหลักสูตร Highlights

- อัปเดตตาม **Docker ปัจจุบัน** Docker Engine 29 และ Docker Compose v5
- **เข้าใจถึงกลไกภายใน** ไม่ใช่แค่จำคำสั่ง จึงแก้ปัญหาได้เอง
- **Best Practice ที่ใช้ได้ทันที** Multi-stage, Cache, Non-root, Healthcheck
- **Security ตั้งแต่ Image แรก** สแกนช่องโหว่ SBOM และลด Attack Surface
- **ปูทางสู่ Kubernetes** จาก Compose สู่ CI/CD และ Cluster

04 ความรู้พื้นฐาน Prerequisites

- เขียนโปรแกรมได้อย่างน้อย 1 ภาษา และเข้าใจการทำงานของ Web Application
- ใช้ Command Line (Linux Shell หรือ PowerShell) และ Git ได้ในระดับพื้นฐาน
- เข้าใจพื้นฐานเครือข่าย เช่น IP Address, Port, DNS และ HTTP
- มี RAM 16 GB ขึ้นไป เปิด Virtualization แล้ว และมีสิทธิ์ติดตั้งโปรแกรม

05 สิ่งที่ต้องเตรียมและเครื่องมือ Setup & Tools

รายการ	รายละเอียด
Container Runtime	Docker Engine 29 และ containerd บน Linux หรือ Docker Desktop (Windows ผ่าน WSL2)
Build	BuildKit, Docker Buildx และ Docker Bake สำหรับ Multi-platform Build
Multi-container	Docker Compose v5 พร้อม Healthcheck, Profiles, Watch และ Init Container
Security และ Registry	Docker Scout, Trivy, Docker Hub, Private Registry และ Container Registry ของ GitHub หรือ GitLab
ระบบตัวอย่าง	REST API (Node.js หรือ FastAPI), Web Frontend, Nginx, PostgreSQL และ Redis บน VS Code

หมายเหตุ: หลักสูตรนี้ต่อยอดจาก Basic Docker โดยลงลึกเรื่อง BuildKit, Security, Registry และการใช้ Compose กับระบบจริง ทุก Lab รันบนโหนดบู๊ทของผู้เรียน Docker Desktop ใช้ฟรีได้สำหรับบุคคลและองค์กรขนาดเล็ก แต่องค์กรขนาดใหญ่ต้องมี Subscription แบบเสียค่าบริการ ผู้เรียนจึงเลือกใช้ Docker Engine บน WSL2 หรือ Linux แทนได้ ความสามารถบางส่วนของ Docker Scout ขึ้นกับแผนการใช้งาน หลักสูตรจึงใช้ Trivy ซึ่งเป็นโอเพนซอร์สควบคุม ส่วน Kubernetes เป็นการสาธิตเพื่อปูทาง ชื่อและความสามารถของเครื่องมือเปลี่ยนเร็ว เนื้อหาจะอิงรุ่นล่าสุด ณ วันอบรม

06 ภาพรวมการเรียนรู้ 3 วัน Big Picture

วัน	หัวข้อหลัก
Day 1 Section 1-5	กลไกภายในของ Container และ Docker Engine 29, คำสั่ง CLI ที่ใช้จริง, Dockerfile Best Practices, Multi-stage Build, BuildKit Cache และ Secret, Build หลายสถาปัตยกรรมด้วย Buildx
Day 2 Section 6-10	Network, Volume และ Configuration แยกตาม Environment, ประกอบระบบหลาย Service ด้วย Docker Compose v5 พร้อม Healthcheck, Init Container, Profile และ Compose Watch
Day 3 Section 11-15	Non-root และ Read-only Container, สแกนช่องโหว่และ SBOM ด้วย Docker Scout และ Trivy, Registry และ Tag, Resource Limit, Debug และต่อยอดสู่ CI/CD และ Kubernetes



DAY
1

กลไกภายในของ Container และการสร้าง Image ที่ดี

เนื้อหาการอบรม · Section 1-5

รวม
6 ชั่วโมง

1 Lab: Container Internals

- Container เทียบกับ Virtual Machine และสิ่งที่ Container แชร์กับ Host
- Linux Namespace, cgroups และ Union Filesystem
- สถาปัตยกรรม Docker CLI, Docker Engine, containerd และ runc
- ความเปลี่ยนแปลงใน Docker Engine 29: containerd image store เป็นค่าเริ่มต้นสำหรับการติดตั้งใหม่
- Lab: สำรวจ Namespace, cgroups และ Image Layer ของ Container ด้วยคำสั่งจริง

2 Lab: Docker CLI ที่ใช้งานจริง

- วงจรชีวิตของ Container: create, start, stop, restart และ rm
- docker exec, logs, inspect, stats และ events
- จัดการ Image ด้วย pull, tag, history และ prune
- Lab: รับและตรวจสอบ Service ของ BrewTrack ที่ละตัวจาก Official Image

3 Dockerfile Best Practices

- ลำดับคำสั่งเพื่อใช้ Layer Cache อย่างมีประสิทธิภาพ
- .dockerignore, COPY เทียบกับ ADD และ ENTRYPOINT เทียบกับ CMD
- เลือก Base Image: Official, Slim, Alpine และ Distroless
- ARG, ENV, LABEL และการทำให้ Build ซ้ำได้ผลเหมือนเดิม
- Lab: เขียน Dockerfile แรกของ REST API แล้ววัดขนาดและเวลา Build

4 Lab: Multi-stage Build และ BuildKit

- แยก Build Stage กับ Runtime Stage เพื่อลดขนาด Image
- Cache Mount สำหรับ Package Manager เพื่อลดเวลา Build ซ้ำ
- Build Secret ที่ไม่ค้างอยู่ใน Layer ของ Image
- Build Target สำหรับ Development และ Production จาก Dockerfile เดียว
- Workshop: ปรับ Image ของ API และ Frontend แล้วเทียบขนาดและเวลาก่อนและหลัง

5 Lab: Buildx และ Multi-platform Image

- Builder Instance และ Driver ของ Buildx
- Build Image สำหรับ linux/amd64 และ linux/arm64 ในคำสั่งเดียว
- ภาพรวมของ Docker Bake สำหรับจัดการ Build หลาย Target
- Lab: Build Image ของ BrewTrack แบบ Multi-platform แล้วตรวจ Manifest ที่ได้



DAY
2

Networking, Storage และ Docker Compose

เนื้อหาการอบรม · Section 6-10

รวม
6 ชั่วโมง

6 Lab: Docker Networking

- Network Driver: bridge, host, none และ overlay ในระดับแนวคิด
- User-defined Bridge Network และ DNS ภายในระหว่าง Container
- Port Publishing และการแยก Network ของ Frontend กับ Backend
- ภาพรวม Firewall Backend แบบ nftables ที่ยังเป็น Experimental ใน Docker Engine 29
- Lab: แยก Network ให้ฐานข้อมูลเข้าถึงได้เฉพาะจาก API

7 Lab: Volumes และ Data Persistence

- Volume, Bind Mount และ tmpfs ต่างกันอย่างไร และเลือกใช้เมื่อไร
- Backup และ Restore ข้อมูลใน Volume
- ปัญหาสิทธิ์ไฟล์ (UID/GID) ระหว่าง Host และ Container
- Lab: เก็บข้อมูล PostgreSQL ใน Volume แล้วทดลอง Backup และ Restore

8 Configuration และ Environment

- Environment Variable, ไฟล์ .env และลำดับความสำคัญของค่า
- แยกค่าตั้งของ Development, Staging และ Production
- ส่ง Secret ให้ Container โดยไม่ฝังไว้ใน Image
- Lab: ย้ายค่าตั้งทั้งหมดของ BrewTrack ออกจาก Image ไปไว้ที่ Environment

9 Lab: Docker Compose v5

- โครงสร้างไฟล์ Compose: services, networks, volumes, configs และ secrets
- Healthcheck และ depends_on แบบมีเงื่อนไขเพื่อคุมลำดับการเริ่มทำงาน
- Init Container สำหรับงานเตรียมระบบ เช่น Database Migration
- Profiles สำหรับ Service เสริม และการรวมไฟล์ Compose หลายไฟล์
- การ Build ใน Compose v5 ที่ทำงานผ่าน Docker Bake

10 Developer Workflow ด้วย Compose Watch

- Compose Watch แบบ sync, rebuild และ sync+restart
- ตั้งค่า Hot Reload ให้ API และ Frontend
- คำสั่งที่ใช้บ่อย: up, down, logs, ps, exec และ config
- Workshop: ประกอบ Frontend, API, PostgreSQL, Redis และ Nginx ใน Compose เดียว พร้อม Watch และ Profile เครื่องมือจัดการฐานข้อมูล



DAY
3

Security, Operations และ Master Workshop

เนื้อหาการอบรม · Section 11-15

รวม
6 ชั่วโมง

11 Lab: Container Security

- รับ Container แบบ Non-root และ Read-only Filesystem
- ลด Linux Capabilities และป้องกันการยกระดับสิทธิ์
- Rootless Mode ในระดับแนวคิด
- ลด Attack Surface ด้วย Minimal หรือ Distroless Image และอัปเดต Base Image เสมอ
- Lab: ปรับ Image ทุกตัวของ BrewTrack ให้รับแบบ Non-root

12 Lab: Image Scanning และ Supply Chain

- สแกนช่องโหว่ด้วย Docker Scout และ Trivy พร้อมการอ่านผล CVE
- SBOM และ Provenance Attestation ของ Image
- กำหนดเกณฑ์ไม่ให้ Image ที่มีช่องโหว่ระดับร้ายแรงผ่านไปใช้งาน
- Lab: สแกนแล้วแก้ไขจนผ่านเกณฑ์ที่ทีมตั้งไว้

13 Registry และ Versioning

- Docker Hub, Private Registry และ Container Registry ของ GitHub หรือ GitLab
- กลยุทธ์ Tag: Semantic Version, Git SHA และความเสถียรของ latest
- ยืนยันตัวตนกับ Registry อย่างปลอดภัยด้วย Credential Helper
- Lab: Tag และ Push Image ขึ้น Private Registry ตามกลยุทธ์ Versioning ของทีม

14 Lab: Resource, Logging และ Debugging

- CPU และ Memory Limit, Restart Policy และ OOM Kill
- Logging Driver และการหมุนเวียนไฟล์ Log
- ขั้นตอน Debug: inspect, logs, events, exec และ docker debug สำหรับ Image ที่ไม่มี Shell
- Lab: แก้ปัญหาจำลอง เช่น Container ถูก OOM Kill, Service ต่อฐานข้อมูลไม่ได้ และสิทธิ์ Volume ไม่ถูกต้อง

15 จาก Docker สู่ CI/CD, Kubernetes และ Master Workshop

- Build และ Push Image ใน CI/CD Pipeline
- เทียบแนวคิด Compose กับ Kubernetes: Pod, Deployment และ Service
- สาธิตการรับ Image เดียวกันบน Kubernetes ของ Docker Desktop หรือ kind
- Master Workshop: เก็บงาน BrewTrack ให้ปลอดภัย มี Resource Limit และ Log Rotation พร้อมส่งเข้า Pipeline

07 เรียนจบแล้วได้อะไร Outcomes

- เข้าใจ Container จริง ตั้งแต่ Kernel จนถึง Docker Engine 29
- เขียน Dockerfile มีอาชีพ Image เล็ก Build เร็ว หลายสถาปัตยกรรม
- ประกอบระบบหลาย Service Compose v5 พร้อม Healthcheck และ Watch
- Security ตั้งแต่ต้น Non-root, สแกนช่องโหว่ และ SBOM
- พร้อมใช้งานกับทีม Registry, Tag และ Resource Limit
- แก้ปัญหาได้เป็นระบบ Debug Container ที่ลุ่มได้ด้วยตนเอง

08 สิ่งที่คุณเรียนจะได้รับ Deliverables

- เอกสารประกอบการอบรมฉบับสมบูรณ์ครบทั้ง 3 วัน และ Lab Guide รายวัน
- Dockerfile และไฟล์ Compose ของระบบ BrewTrack Coffee Ordering ครบทุก Lab
- Checklist การเขียน Dockerfile, Security และขั้นตอน Debug สำหรับใช้เป็นมาตรฐานของทีม
- ประกาศนียบัตรจบหลักสูตร (Certificate of Completion) และช่องทางสอบถามวิทยากรหลังจบหลักสูตร

09 ข้อมูลผู้สอน Instructor

อ.สามิตร โคม ผู้สอนและนักพัฒนาที่ใช้ Container เป็นเครื่องมือหลักในการพัฒนาและส่งมอบซอฟต์แวร์ ทั้งบนเครื่องนักพัฒนา CI/CD และ Kubernetes จุดเด่นคือการอธิบายกลไกเบื้องหลังให้เข้าใจง่าย พาผู้เรียนวัดผลก่อนและหลังปรับปรุงทุกครั้ง และฝึกแก้ปัญหาจากสถานการณ์จำลองที่พบบ่อยในงานจริง เพื่อให้กลับไปวางมาตรฐานการใช้ Docker ของทีมได้อย่างมั่นใจ

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

9,500 บาท / ท่าน (ยังไม่รวม VAT 7%)

สมัครผ่านเว็บไซต์ ลด 10%

☎ 02-570-8449

LINE @itgenius

☎ 088-807-9770

✉ contact@itgenius.co.th



เพิ่มเพื่อน
LINE