



Term Project Report: FurEveryHome

กลุ่ม Riyal or Fakay?

รายชื่อสมาชิก

นายพนพล	จุ่นเจิม	6509650146
นางสาวขวัญดาว	ชั้นบุรี	6509650245
นางสาวปุณณภา	ตรีรัตน์ดิลกกุล	6509650559
นายธีรภัทร	เกิดไพบูลย์	6509650468

เสนอ

ผศ.ดร.ประภาพร รัตนธารง

รายงานฉบับนี้เป็นส่วนหนึ่งของรายวิชา

คพ.360 หัวข้อเลือกสรรด้านวิศวกรรมซอฟต์แวร์ ภาคเรียนที่ 1 ปีการศึกษา 2567

คณะวิทยาศาสตร์และเทคโนโลยี สาขาวิทยาการคอมพิวเตอร์

มหาวิทยาลัยธรรมศาสตร์

2. ข้อมูล Repository

Project Repository :

https://github.com/Kwandao6509650245/CS360_Project.git

3. รายงานผลเกี่ยวกับเวลาที่ใช้ตั้งแต่ push code จนนำแอปพลิเคชันขึ้นให้บริการบน EC2

a. แบบ Manual (การทดสอบโค้ด การติดตั้ง Server)

- เวลา Integrate/Build Code: 9.43 นาที
- เวลา Deploy Code ขึ้น Server: 11.04 นาที
- รวมเวลาทั้งหมด: 20.47 นาที

การทดสอบและติดตั้งเซิร์ฟเวอร์แบบ Manual ต้องทำขั้นตอนทั้งหมดด้วยตนเอง ตั้งแต่การทดสอบโค้ดไปจนถึงการตั้งค่าระบบเซิร์ฟเวอร์ ซึ่งอาจใช้เวลาเนื่องจากต้องตั้งค่าและแก้ปัญหาที่เกิดขึ้นระหว่างการติดตั้ง เสี่ยงต่อการเกิด human error

b. แบบ Semi-Automate (มีการใช้ Bash script ช่วยการติดตั้ง Server)

- เวลา Integrate/Build Code: 5.17 นาที
- เวลา Deploy Code ขึ้น Server: 1.11 นาที
- รวมเวลาทั้งหมด: 6.28 นาที

การใช้ Bash script ช่วยในการติดตั้งเซิร์ฟเวอร์ทำให้กระบวนการติดตั้ง dependencies เป็นไปอย่างสะดวก รวดเร็วขึ้น โดยไม่ต้องตั้งค่าติดตั้งทีละขั้นตอน แต่ยังคงต้องทำการตรวจสอบ

c. แบบ Automate (ดำเนินการด้วย CI/CD Pipeline ที่สร้างขึ้น)

- เวลา Integrate/Build Code: 3.32 นาที
- เวลา Deploy Code ขึ้น Server: 0.39 นาที
- รวมเวลาทั้งหมด: 4.11 นาที

This screenshot shows the GitHub Actions interface for a CI workflow named "CI - Test Source Code and Docker Image" with the job "edit images name #30". The workflow is triggered by a push and has a status of "Success". The summary shows a total duration of 3m 32s. The workflow file, "ci-docker.yml", is displayed, showing a sequence of jobs: "build-docker-backend" (2m 14s) followed by "test-docker-image-backend" (52s), and "build-docker-frontend" (2m 8s) followed by "test-docker-image-frontend" (1m 5s). The left sidebar shows the workflow file and run details.

This screenshot shows the GitHub Actions interface for a CD workflow named "CD - Deploy to EC2 with Docker" with the job "edit images name #66". The workflow is triggered by a push and has a status of "Success". The summary shows a total duration of 39s. The workflow file, "cd.yml", is displayed, showing a single job: "check-image-and-deploy" (27s). The left sidebar shows the workflow file and run details. An annotation at the bottom indicates a warning: "check-image-and-deploy: ubuntu-latest pipelines will use ubuntu-24.04 soon. For more details, see https://github.com/ac... Show more".

- การใช้ CI/CD Pipeline ทำให้กระบวนการทั้งการ Build และ Deploy เป็นไปโดยอัตโนมัติ ลดเวลาในการทำงาน และข้อผิดพลาดที่อาจเกิดจากขั้นตอนด้วยมือ และมีการ automate test อยู่เสมอ

4. อภิปรายเกี่ยวกับประโยชน์ของกระบวนการ CI/CD

กระบวนการ CI/CD (Continuous Integration / Continuous Deployment) ช่วยลดเวลาในการพัฒนาและติดตั้งแอปพลิเคชัน โดยสามารถทำให้โค้ดที่เขียนเสร็จแล้วสามารถนำไปทดสอบและติดตั้งขึ้นเซิร์ฟเวอร์ได้โดยอัตโนมัติ กระบวนการนี้มีข้อดีหลายประการ ดังนี้

- **ลดเวลาในการ Deploy:** จากที่ต้องทำการติดตั้งเซิร์ฟเวอร์ด้วยมือหลายขั้นตอน ในกรณีที่ใช้ CI/CD Pipeline เวลาที่ใช้ในการ Deploy ลดลงอย่างมาก และเป็นการติดตั้งแบบ automate เมื่อมีการอัปเดตซอฟต์แวร์จึงเป็นไปได้อย่างสะดวกและรวดเร็วขึ้น
- **ลดข้อผิดพลาด:** การใช้ Pipeline ช่วยให้กระบวนการต่าง ๆ มีความสม่ำเสมอและสามารถลดข้อผิดพลาดที่เกิดจาก human error
- **ความถูกต้องของโค้ด:** กระบวนการ CI/CD สามารถทำการทดสอบโค้ดทุกครั้งที่มีการเปลี่ยนแปลง ทำให้มั่นใจได้ว่าโค้ดที่ได้รับการ Deploy เป็นโค้ดที่ผ่านการทดสอบแล้ว

จากการพัฒนาโปรเจกต์นี้ การสร้าง CI/CD Pipeline มีประโยชน์มาก เนื่องจากช่วยประหยัดเวลาในระยะยาว ทำให้สามารถโฟกัสกับการพัฒนาและปรับปรุงแอปพลิเคชันได้มากขึ้น ดังนั้น จึงควรทำ CI/CD ในโครงการนี้ เพราะจะทำให้กระบวนการ Deploy มีประสิทธิภาพสูงสุด ลดข้อผิดพลาดที่เกิดจากการกระทำด้วยมือ ช่วยลดระยะเวลาในขั้นตอนต่างๆ และสามารถจัดการกับการพัฒนาแอปพลิเคชันในระยะยาวได้ดียิ่งขึ้น