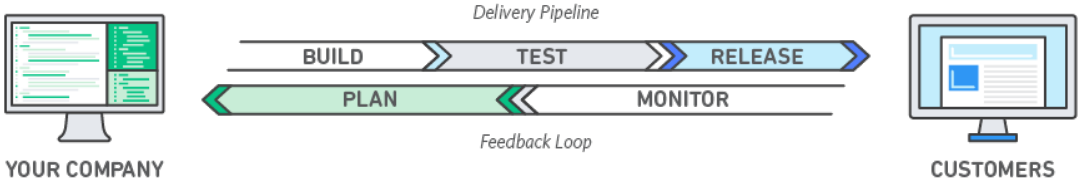
นายศุภพงศ์ สกุลคู รหัสวิชา 322371 รหัสประจำตัว 593021591-7 Sec 2

1. DevOps คืออะไร

DevOps คือการผสมผสานแนวความคิดเชิงวัฒนธรรม แนวทางปฏิบัติ และเครื่องมือต่างๆ ที่ช่วยเพิ่มความสามารถขององค์กรในการส่งมอบแอปพลิเคชันและบริการอย่างรวดเร็ว โดยพัฒนาและปรับปรุงผลิตภัณฑ์ต่างๆ ให้เร็วกว่ากระบวนการการพัฒนาซอฟต์แวร์และการจัดการโครงสร้างพื้นฐานแบบดั้งเดิม ความรวดเร็วนี้ช่วยให้องค์กรสามารถให้บริการแก่ลูกค้าของตนได้ดีขึ้น และสามารถแข่งขันในตลาดได้อย่างมีประสิทธิภาพมากขึ้น



ประโยชน์ของ DevOps ต่อบริษัทที่ใช้ DevOps ในการพัฒนาซอฟต์แวร์

1. สามารถ deploy software ได้มากกว่า 200 เท่า เมื่อเทียบกับการไม่ใช้ DevOps

2. สามารถแก้ปัญหาที่เกิดขึ้นกับระบบได้ไวกว่า 24 เท่า

3. สามารถลดอัตราล้มเหลวของการ change ระบบได้มากกว่า 3 เท่า

4. สามารถลดเวลาในการผลิต software ได้มากกว่า 2,555 เท่า

5. สามารถลดเวลาในการแก้ปัญหาด้าน security ได้มากกว่า 50%

6. สามารถลดเวลาการทำงาน ทำให้พนักงานมีเวลาเพิ่มขึ้นมากกว่า 29 %

7. สามารถลด cost ให้บริษัทได้มากกว่า จาก cost ที่เกิดขึ้นเมื่อระบบมีปัญหา และ cost จากการจ้างบุคลากร

2. CI/CD คืออะไร มีความสัมพันธ์กับ DevOps อย่างไร

2.1 CI คืออะไร

CI (Continuous Integration) คือ กระบวนการรวม source code ของคนในทีมพัฒนาเข้าด้วยกัน และมีการ test ด้วย test script เพื่อให้แน่ใจว่าไม่มี error ในส่วนใดๆ ของโปรแกรมแล้วถึงได้ทำการ commit ไปที่ branch master ต่อไป โดยในการพัฒนานั้น มักใช้ Build Server มาช่วย คือจะเริ่มทำการ Integration กัน ตั้งแต่เมื่อมีการเปลี่ยนแปลง Source Code ที่ Repository กลาง ระบบจะทำการตรวจสอบ Code หลังจากการเปลี่ยนแปลงว่าทำงานร่วมกันได้หรือไม่ก็ตั้งแต่ Compile, Testing

2.2 CD คืออะไร

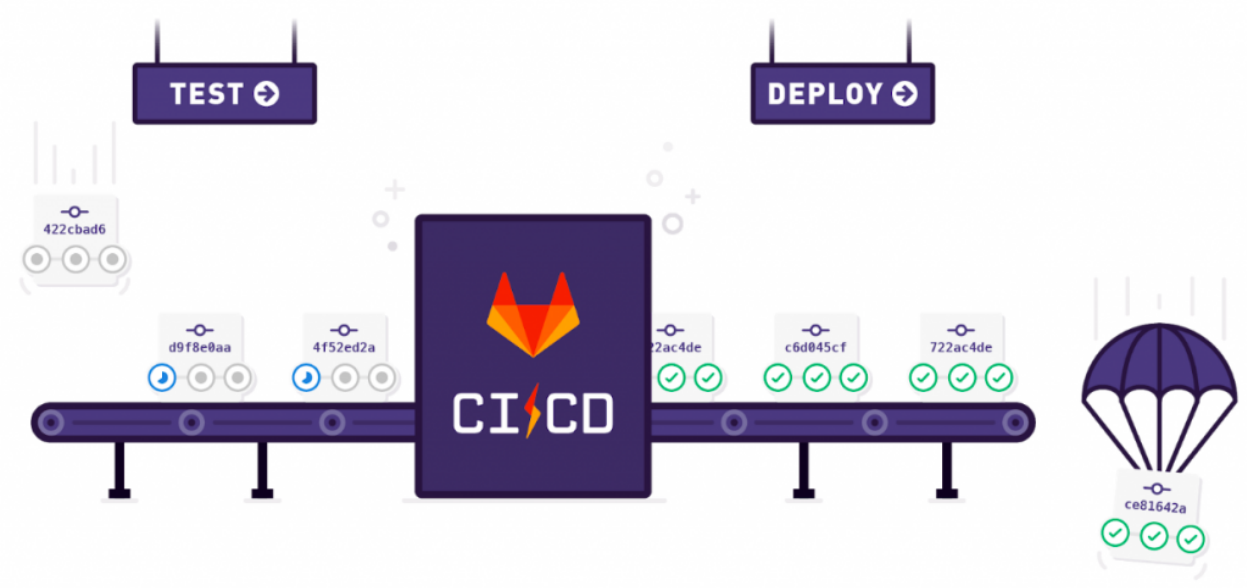
2.2.1 CD (Continuous Deployment) คือ การ Deploy ขึ้น production โดยจะทำทุกขั้นตอน ตั้งแต่ complie build ไปจนถึง deploy ขึ้น production แบบอัตโนมัติทั้งหมด

2.2.2 CD (Continuous Delivery) คือ การทำทุกขั้นตอนคล้ายกันกับ Continuous Deployment ต่างกันตรงที่จะไม่มีการ deploy ขึ้น production ขึ้นในทันที แต่จะเป็นการทำ manual deploy หรือจะเป็นแบบ one click deploy ก็ได้ หลังจาก QA หรือ ฝ่าย Business พอใจในตัว product ที่ทีมทำออกมา และหนึ่งในหน้าที่ของ DevOps Engineer คือ การพัฒนา CI/CD Pipeline Script เพื่อให้ทีมงานมั่นใจว่าโปรแกรมที่เขียนมาไม่มีบั๊กที่ได้เขียน Test เอาไว้แล้วโผล่ขึ้นมา ทุกครั้งที่มีการ Push Code ขึ้นไปเพิ่มเติม จึงต้องมีการรันชุดทดสอบการทำงานของโปรแกรมทั้งหมด ถ้าการทดสอบล้มเหลว Developer จะต้องมาแก้บั๊กให้เรียบร้อย ห้ามมีการส่งต่อขึ้นไปให้ผู้ใช้งาน ส่วนถ้าการทดสอบสำเร็จราบรื่น แบบนั้นจึงค่อยปล่อยให้โปรแกรมไปในขั้นตอนต่อไป ก็คือการจับโปรแกรมใส่ Docker Image แล้วเอาไป Deploy บน Platform ที่ต้องการ จึงกล่าวได้ว่า DevOps กับ CI/CD มีความสัมพันธ์กันอย่างมาก

2.3 CI/CD Pipeline

Code จาก Developer ทุก Push ที่ส่งขึ้นไปที่ Repo จะต้องมีการตรวจสอบความถูกต้องอยู่ตลอด เพื่อให้มั่นใจว่า เราไม่ได้ส่งโปรแกรมที่มีปัญหาไปให้ User วิธีการก็คือ การสร้างระบบที่จะถูก Trigger ให้รันโดยอัตโนมัติจากการ Push Code ขึ้นมา ระบบที่ว่านี้เรียกว่า Continuous Integration and Continuous Delivery (CI/CD) และเพิ่มคำว่า Pipeline เข้าไป เพื่อบอกว่ามันเป็นการทำงานที่ต่อเนื่องการโดยอัตโนมัติ

โดยขั้นตอนเหล่านี้จะรันบนเครื่อง Server ที่สร้างขึ้นมาเพื่อรันงานพวกนี้โดยเฉพาะ (Pipeline Runner) โปรแกรมที่ใช้เป็นหลักเลยก็คือ GitLab CI/CD



อ้างอิงจาก : https://about.gitlab.com/product/continuous-integration/

DevOps จะต้องเขียน Script ว่า หลังจากมี Code ส่งขึ้นมาแล้ว จะต้องให้มันรันคำสั่งอะไรยังไงบ้าง โดยกระบวนการทำ CI/CD ก็มักจะมีขั้นตอนที่คล้ายๆ กัน ดังนี้

2.3.1 Build ขั้นตอนนี้คือการเตรียมโปรแกรมให้พร้อมสำหรับการใช้งาน ขั้นตอนนี้จะโหลด Dependency ต่างๆ แล้วรัน Code สำหรับ Build โปรแกรมให้พร้อมสำหรับการ Test

2.3.2 Test นำโปรแกรมที่ได้จากขั้นตอนก่อนหน้ามารัน Script เพื่อทดสอบการทำงาน การ Test นี้มันก็คือ Code ทั่วไปนี่ล่ะ เพียงแต่ทำหน้าที่ในการตรวจสอบการทำงานของโปรแกรมของเรา รันให้ครบทุกอันให้มั่นใจว่าโปรแกรมยังทำงานได้ปรกติดีอยู่แม้จะมีการเปลี่ยนแปลง Source Code ไป ถ้ามี Test ที่ไม่ผ่านขึ้นมา ไม่ต้องเสียใจ ควรยินดี เพราะนั่นคือสัญญาณว่า Test ที่เราเขียนขึ้นมามันมีประโยชน์ นอกจากนั้นเรายังจับบั๊กได้ก่อนจะไปถึง User ด้วย แต่ในสถาการณ์ปกติจะต้อง Test ให้ผ่านทั้งหมด ซึ่งเมื่อผ่านแล้วก็จะไปที่ขั้นตอนต่อไป

2.3.3 Package ขั้นตอนนี้คือการเอาโปรแกรมมาจับใส่กล่องเตรียมพร้อมสำหรับการนำออกไปใช้งาน กล่องที่ว่าก็คือ Docker Image นั่นเอง ถ้าไม่เห็นภาพให้ลองจินตนาการว่า คือ การ Build ให้ได้ไฟล์ .exe ขึ้นมาที่จะเอาไปรันที่ไหนก็ได้นั่นเอง ทีนี้เราก็พร้อมสำหรับการนำโปรแกรมนี้ไปรันที่ไหนก็ได้ที่มี Docker

จากนั้นจึงส่ง Docker Image ที่ได้ไปไว้บน Host ที่เรียกว่า Docker Registry ซึ่งเราจะได้ URL มาหนึ่งอัน สำหรับนำไปใช้งาน

2.3.4 Deploy สุดท้ายคือการเอาโปรแกรมไปรันบน Server จริง โดยเราติดต่อเข้าไปหา Server ที่เราใช้งาน อาจจะเป็น Kubernetes หรืออื่นๆ ก็แล้วแต่ เมื่อต่อเข้าไปแล้ว ก็บอกว่าให้เอา Docker Image URL ที่ได้มาจากขั้นตอนก่อนหน้า เอามาใช้ใน Deployment ที่ต้องการ ทีนี้ระบบ Container Orchestration ก็จะทำการ Download Docker Image ลงมา สร้าง Container และทำการรัน เมื่อทุกอย่างเรียบร้อยดีแล้ว มันจะลบ Container เวอร์ชั่นเก่าโดยอัตโนมัติ เป็นอันสิ้นสุดกระบวนการ CI/CD

3 หากองค์กรต้องการนำเอา DevOps และ CI/CD เข้ามาเป็นส่วนเสริมในกระบวนการพัฒนาซอฟต์แวร์จะต้องทำอย่างไรบ้าง

DevOps มีรูปแบบการทำงานค่อนข้างจะตายตัว ซึ่งจะพยายามทำให้ Process ต่างๆ ทำงานไปได้โดยอัตโนมัติ ไม่ต้องมีคนคอยเข้าไปกด Deploy เอง ยกเว้นในส่วนของการการวางแผน การพัฒนา และดูผลลัพธ์ ในส่วนนี้ยังจำเป็นต้องอาศัยสมองคนต่อยู่ ถ้าหากต้องการนำ DevOps ไปใช้ในการพัฒนาซอฟต์แวร์แล้วนั้น จะมีขั้นตอน ดังนี้

3.1 Verify การตรวจสอบอยู่อย่างสม่ำเสมอว่าโปรแกรมที่พัฒนาขึ้นมา ไม่ว่าจะเพิ่มฟีเจอร์อะไรเข้ามาจะไม่ไปกระทบหรือไปก่อบั๊กทำให้การใช้งานเดิมพังเสียหายนั้นเป็นเรื่องสำคัญมาก โดยเฉพาะโปรแกรมขนาดใหญ่ที่ร่วมกันทำกับทีมงานจำนวนมาก เราจะมั่นใจได้อย่างไรว่า Code ที่ Developer X ส่งขึ้นมาจะเข้ากันได้ดีกับการทำงานเดิม วิธีการที่จะรู้ได้เร็วที่สุดคือ เราจะต้องมีการทำ Testing อย่างครอบคลุมในทุกๆ Components ของโปรแกรม แล้วทำให้มันรันโดยอัตโนมัติทุกครั้งก่อนจะปล่อยโปรแกรมออกไป เมื่อใดที่ Test Fail ก็จะรู้ได้ทันทีว่ามีปัญหาเกิดขึ้นแล้ว ให้หยุดการส่งงานที่มีปัญหาไปก่อน สามารถเข้าไปหาสาเหตุและแก้ไขปัญหาได้อย่างทันท่วงที

3.2 Package การจะส่งโปรแกรมขึ้นไปรันบน Server เพื่อความสะดวกรวดเร็ว จะต้องมีการเอาโปรแกรมไปใส่ใน Technology ที่ออกแบบมาเพื่อนำโปรแกรมไปรันได้อย่างราบลื่น ซึ่งในปัจจุบันนี้เราใช้ Docker เวลาจะใช้ เราจะสร้าง Docker Image ที่เอาตัวโปรแกรมพร้อมสำหรับการรันใส่อยู่ในนั้น แล้วเอา Docker Image นี้ไป Deploy ที่ไหนก็ได้สามารถรัน Docker ได้ โปรแกรมก็จะรันขึ้นมาพร้อมใช้งาน

3.3 Release เมื่อมี Docker Image พร้อมแล้ว ก็พร้อมสำหรับการนำโปรแกรมไปรันบน Deployment Platform ที่ต้องการ ตอนนี้กำลังทำของ Development หรือทำหรือ Production อยู่ จะต้องเลือกไป Deploy ให้ถูกที่

3.4 Configure โปรแกรมที่พัฒนานั้น จำเป็นอย่างมาก ที่จะต้องตั้งค่าได้ ตัวอย่างการตั้งค่าที่ควรมีเลย ก็เช่น การเลือก Database ตั้งค่า Email เป็นต้นยิ่งถ้ามี Service แยกอื่นๆ ที่ต้องใช้แล้ว ยิ่งสำคัญ เพราะ Development Server และ Production Server ควรอยู่แยกกันอย่างเด็ดขาด เราจะต้องหาทางสร้าง Docker Image มา 1 อัน แล้วเราจะใช้มัน Deploy ลงบน Environments ที่ต่างกันให้ได้ วิธีการที่จะทำได้ก็คือการทำให้โปรแกรมของเราตั้งค่าได้เนี่ยแหละ คือโปรแกรมที่ไปรันเป็นตัวเดียวกัน แต่เปิดด้วยการตั้งค่าที่แตกต่างกัน ทำให้ได้การทำงานที่แตกต่างกัน โดยปรกติจะใช้ไฟล์ .env และ Environment Variable ในการตั้งค่ากันเป็นมาตรฐานที่ใช้กันบนทุก Platform

3.5 Monitor สุดท้ายที่ขาดไม่ได้เลยคือการตรวจสอบว่าโปรแกรมมันขึ้นไปแล้วทำงานได้ปรกติ มีสุขภาพแข็งแรงดี การดู Log ว่าโปรแกรมเราทำงานมีปัญหาตรงไหน มีความสำคัญอย่างมากในการแก้ไขบั๊ก ยิ่งในตอนนี้ที่มักจะออกแบบให้โปรแกรมมันสามารถทำให้ Horizontal Scale ได้ (การสร้างเครื่องขึ้นมาหลายๆ เครื่องให้มาแบ่ง Load ทำงานแยกกันได้) จำเป็นอย่างยิ่งที่จะต้องมี Centralized Logging System รวบรวม Log เอามาไว้ในที่เดียว เพราะการ Remote เข้าไปอ่าน Log ในแต่ละเครื่องเป็นไปไม่ได้อีกแล้ว

4. เอกสารอ้างอิง

[1] Kriangkrai Chaonithi. (2562). *DevOps CI/CD.* สืบค้น 24 กุมภาพันธ์ 2564, จาก https://www.nupress.grad.nu.ac.th/behavior

[2] Pariwat Saknimitong. (2560). *DevOps คืออะไร.* สืบค้น 24 กุมภาพันธ์ 2564, จาก https://medium.com/@pariwat_s/learn-devops-ตอนที่-2-devops-คืออะไร-18ac48d73625

นายศุภพงศ์ สกุลคู รหัสวิชา 322371 รหัสประจำตัว 593021591-7 Sec 2