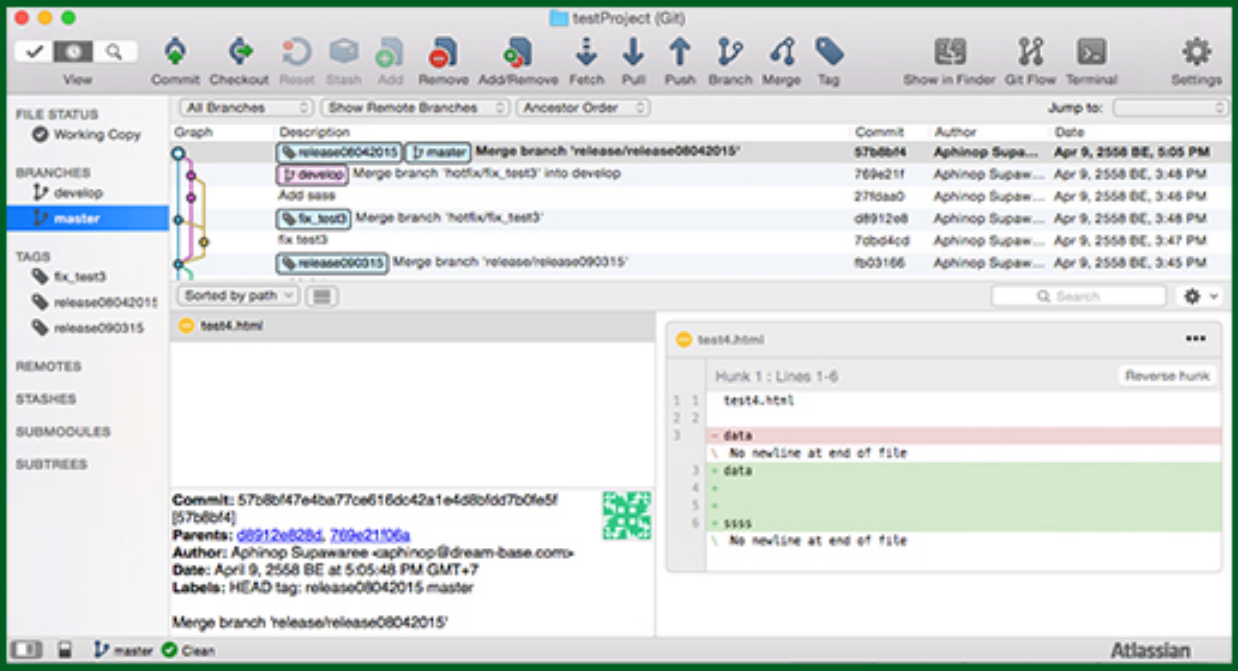
GIT เอาอยู่



ถ้าคุณยังทำสิ่งเหล่านี้ในการเก็บ source code อยู่ จงเปลี่ยนมันซะ

- ✓ copy file แล้วเอามารวมกันโดยใช้ Thumb Drive
- ✓ เก็บ Source Code ด้วย Thumb Drive
- ✓ เมื่อเขียน code แล้วอยากย้อนกลับแต่ทำไม่ได้

โดย อภินพ ศุภวารี

บทที่ 1 Overview

เชื่อว่าหลายๆคนในวงการ โปรแกรมเมอร์น่าจะเคยได้ยิน หรือเคยใช้ Git มาบ้างไม่มากก็น้อย ซึ่งปัจจุบัน git เป็นที่นิยมเป็นอย่างมากในวงการโปรแกรมเมอร์บ้านเรา บริษัทหลายๆแห่งเริ่มนำมาให้โปรแกรมเมอร์นำมาใช้พัฒนาโปรแกรม เพื่อให้โปรแกรมเมอร์ในทีมสามารถทำงานร่วมกันได้ง่ายขึ้น ไม่จำเป็นที่จะต้องนำแฟรชไดร์มาก๊อปปี้ไฟล์กันไปมา เรามาทำความรู้จัก git กันเลยดีกว่า

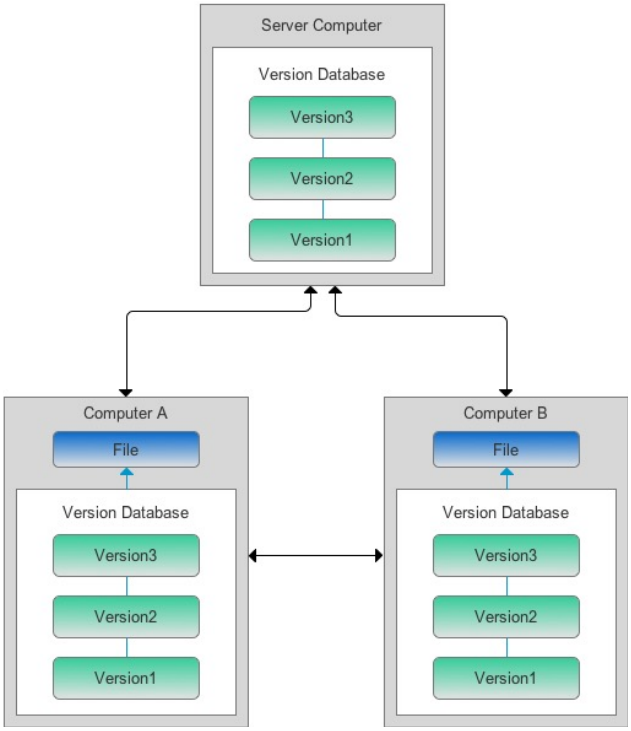
Git คืออะไร

Git คือ Version control ชนิดหนึ่ง ซึ่งเป็นระบบที่ทำหน้าที่จัดเก็บไฟล์ต่างๆที่มีการเปลี่ยนแปลง ในโปรเจคของเราไม่ว่าจะมีเพียงไฟล์เดียว หรือมีหลายไฟล์ก็ตามเพื่อให้สามารถเรียกเวอร์ชั่นใดเวอร์ชั่นหนึ่งที่เราต้องการกลับมาใช้งาน หรือนำกลับมาดูเมื่อไหร่ก็ได้

ตัวอย่างเช่น เมื่อมีการแก้ไขไฟล์ต่างๆ และเรานำไปเก็บไว้ใน Git เมื่อวันหนึ่งมีข้อผิดพลาดกับไฟล์ที่เราทำอยู่ เราสามารถดึงข้อมูลของไฟล์นั้นเพื่อย้อนกลับไปยังไฟล์ที่เราเคย นำไปเก็บไว้ใน Git นั้นกลับมาใช้ได้ อีกทั้งยังสามารถดูได้ว่าเราเคยแก้ไข หรือใครแก้ไขอะไรในไฟล์นั้นไว้บ้าง

การทำงานของ Git

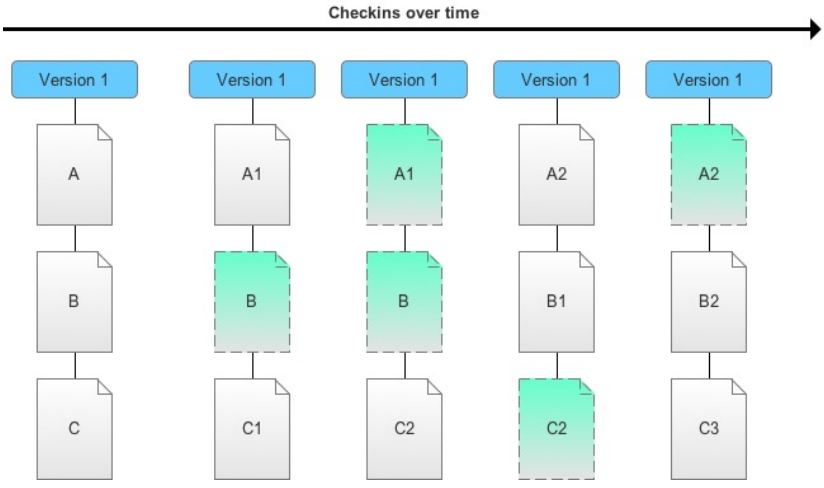
Git เป็น Version control แบบ Distributed Version Control Systems (DVCSs) คือแต่ละ client จะได้ข้อมูลทั้งหมดของ repository มาเลย หมายความว่าถึงเครื่องเซิร์ฟเวอร์จะเสีย แต่ client ก็จะยังสามารถทำงานร่วมกันต่อไปได้ และ repository ของ client ก็ยังสามารถถูกก๊อปปี้กลับไปยังเซิร์ฟเวอร์เพื่อกูข้อมูลกลับคืนก็ได้เช่นกัน สรุปก็คือเมื่อมีการ checkout แต่ละครั้งหมายถึง การทำสำรองข้อมูลทั้งหมดออกมาแบบเต็ม ๆ นั่นเอง



รูป 1-1. Distributed Version Control Systems (DVCSs)

การเก็บข้อมูลเป็น Snapshot

วิธีที่ Git มองข้อมูลต่างๆ นั้น Git จะมองว่าข้อมูลเหล่านั้นเป็นเสมือนภาพถ่าย(snapshot) ของไฟล์ต่างๆภายในโปรเจคที่มีการเปลี่ยนแปลง ในทุกๆครั้งที่ได้ทำการ commit หรือบันทึกสถานะของโปรเจคลงใน Git ระบบของ Git จะทำการ snapshot ไฟล์ทั้งหมดในขณะนั้น และทำการบันทึกข้อมูลการอ้างอิงไปยัง snapshot เหล่านั้น ส่วนไฟล์ไหนที่ไม่ได้มีการเปลี่ยนแปลง Git ก็จะไม่ทำการบันทึกไฟล์นั้นใหม่ แต่จะการอ้างอิงไปยังไฟล์เดิมเท่านั้น



รูป 1-2. Git เก็บข้อมูลของโปรเจคเป็น snapshot

การทำงานเกือบทุกอย่าง จะเป็นการการทำงานในเครื่องตัวเอง

การใช้งาน Git นั้นส่วนใหญ่จะเป็นการทำงานภายในเครื่องของตัวเอง โดยจะเป็นการใช้ข้อมูลต่างๆจากเครื่องของเรา ไม่จำเป็นที่จะต้องใช้ข้อมูลจากเน็ตเวิร์กเลย จากข้อนี้ทำให้ Git มีความเร็วในการทำงานดีกว่า version control แบบอื่น เนื่องจากจะมีข้อมูลการเปลี่ยนแปลงทั้งหมดของโปรเจคอยู่ในเครื่องของเราเลย โดยที่ไม่ต้องไปดึงข้อมูลมาจากที่อื่น ทำให้สามารถใช้งานได้ทันที ดังนั้นในกรณีที่ไม่สามารถใช้เน็ตเวิร์ก หรือเครือข่ายอินเทอร์เน็ตได้ เราก็ยังสามารถทำงานได้เหมือนปกติ

Git มีความเที่ยงตรง

ทุกอย่างที่ Git ทำการบันทึกไว้จะถูกทำการ Checksum แล้วนำมาเป็นตัวอ้างอิง ดังนั้นเราไม่สามารถแก้ไขข้อมูลของไฟล์และไดเร็กทอรี่โดยที่ Git จะไม่รู้ ดังนั้นไม่มีทางที่เราจะทำข้อมูลสูญหายระหว่างการย้าย หรือรับ ส่วนไฟล์ที่เกิดการเสียหาย Git ก็จะสามารถรับรู้ได้เช่นกัน

Git ทำเพียงแค่เพิ่มข้อมูล

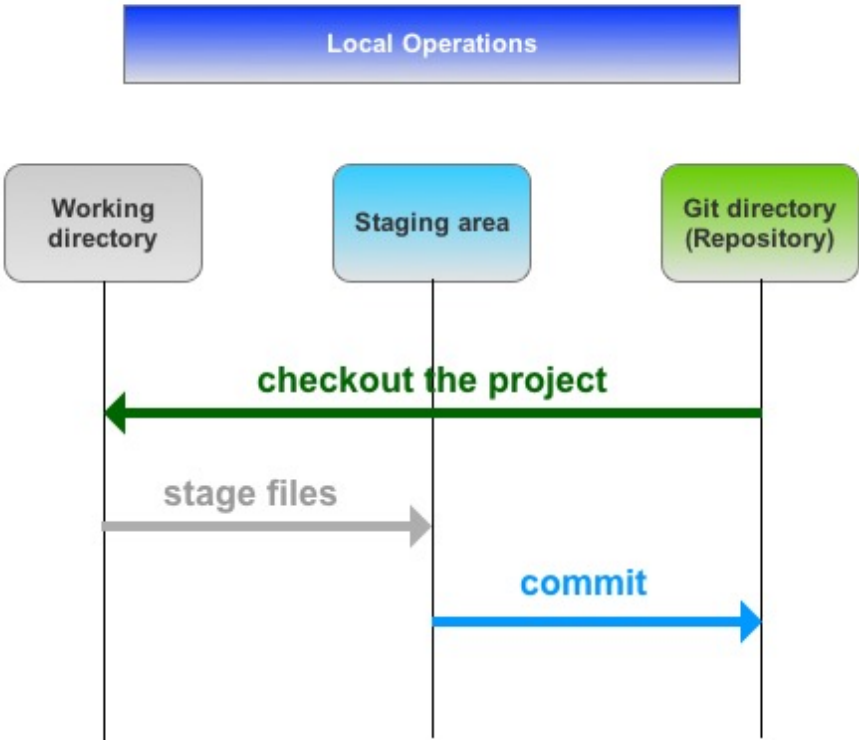
เมื่อเกิดการแก้ไขข้อมูล Git จะทำการเพิ่มข้อมูลทั้งหมดนั้นเข้าไปในฐานข้อมูลของ Git เท่านั้น และเมื่อเรา commit ข้อมูลการแก้ไขลงใน snapshot ของ Git แล้วเรายังสามารถ push ฐานข้อมูลของเราไปเก็บไว้ที่อื่นได้อีกด้วย

การทำงานสามสถานะ

สถานะของไฟล์ในระบบ Git นี้จะมีสถานะอยู่ 3 สถานะ คือ committed(ยืนยันแล้ว), modified(ถูกแก้ไข) และ staged(อยู่ในขั้นตอน) ความหมายของแต่ละสถานะมีดังนี้

- Committed หมายถึงข้แมูลที่ถูกบันทึกลงในฐานข้อมูลในเครื่องของเราเรียบร้อยแล้ว
- Modified หมายถึงไฟล์ของคุณได้ถูกแก้ไขแล้วแต่ยังไม่ได้ทำการ commit
- Staged หมายถึงไฟล์ในเวอร์ชั่นปัจจุบันที่ถูกแก้ไข ที่เราได้ทำการเลือกเพื่อรอที่จะ commit ใน snapshot ต่อไป

จากสถานะทั้ง 3 นี้จึงทำให้โปรเจคที่ใช้ Git มี 3 ส่วน คือ Git directory, Working directory และ Staging area



รูป 1-3. การทำงานของ Working directory, Staging area และ Git directory

Git directory เป็นที่สำหรับใช้เก็บ metadata และ object ต่างๆของฐานข้อมูลสำหรับโปรเจคของเรา ซึ่งเป็นส่วนที่สำคัญที่สุดของ Git และจะเป็นส่วนที่ถูกคัดลอกออกมา เมื่อเราทำการ clone ข้อมูลมาจากเครื่องอื่น

Working directory เป็นเวอร์ชั่นหนึ่งของไฟล์ในโปรเจคที่เราสามารถนำมาแก้ไข หรือใช้งานได้ ซึ่งข้อมูลเหล่านี่คือข้อมูล ที่ถูกดึงออกมาจากฐานข้อมูลซึ่งถูกบีบอัดไว้ใน Git directory แล้วเก็บไว้ในดิสก์

Staging area คือไฟล์ที่ทำหน้าที่เก็บข้อมูล ส่วนที่เราจะทำการ commit ในครั้งต่อไป บางครั้งจะเรียกว่า index แต่ใน Git จะเรียกว่า staging area ซึ่งไฟล์นี้จะอยู่ภายใน Git ของเรา

ขั้นตอนการทำงานพื้นฐานของ Git

1. เมื่อมีการแก้ไขไฟล์ใน working directory ของเรา แล้วเมื่อเราต้องการที่จะ commit ไฟล์นั้นให้เราทำการเลือกไฟล์(stage) เหล่านั้นที่เราต้องการนำขึ้น Git
2. ไฟล์ที่ถูกเลือกจะถูกใส่ snapshot ลงไปใน staging area ของเรา
3. เมื่อทำการ commit(ยืนยัน) ไฟล์ที่อยู่ใน staging area จะถูกนำไปเก็บลงใน git directory

---