

# Version Control : Basic Git

## 1. จุดประสงค์

- 1.1. นักศึกษารู้จัก Version Control System (VCS)
- 1.2. นักศึกษาสามารถใช้งานคำสั่ง Git พื้นฐานได้
- 1.3. นักศึกษาสามารถสร้าง Git repository (Github) และใช้งานจริงเบื้องต้นได้

## 2. What is “version control”

Version control คือ ระบบที่จัดการการเปลี่ยนแปลงที่เกิดขึ้นกับไฟล์หนึ่งหรือหลายไฟล์เพื่อที่สามารถเรียกเวอร์ชันใดเวอร์ชันหนึ่งกลับมาดูเมื่อไรก็ได้ เช่น หากกำลังจัดทำเอกสารอะไรสักอย่างอยู่ เมื่อทำเสร็จทำการบันทึก ต่อมามีการแก้ไขตอนหลัง และสุดท้ายต้องกลับไปใช้เวอร์ชันก่อนหน้านี้ ซึ่งการใช้ Version Control System (VCS) เข้ามาช่วย จะทำให้สามารถย้อนกลับไปหาเวอร์ชันก่อน ๆ ได้ นอกจากนั้น VCS ยังช่วยให้สามารถเปรียบเทียบการแก้ไขในอดีต กับปัจจุบัน หรือตรวจสอบการแก้ไขในแต่ละครั้งได้อีกด้วย รวมถึงยังช่วยให้สามารถกู้คืนไฟล์ที่ถูกลบหรือทำความเสียหายโดยไม่ตั้งใจได้อย่างง่ายดาย

## 3. Git

Git คือ Version Control แบบ Distributed เป็นระบบที่ใช้จัดเก็บและควบคุมการเปลี่ยนแปลงที่เกิดขึ้นกับไฟล์ชนิดใดก็ได้ ไม่ว่าจะเป็น Text File หรือ Binary File

### 3.1. การติดตั้ง

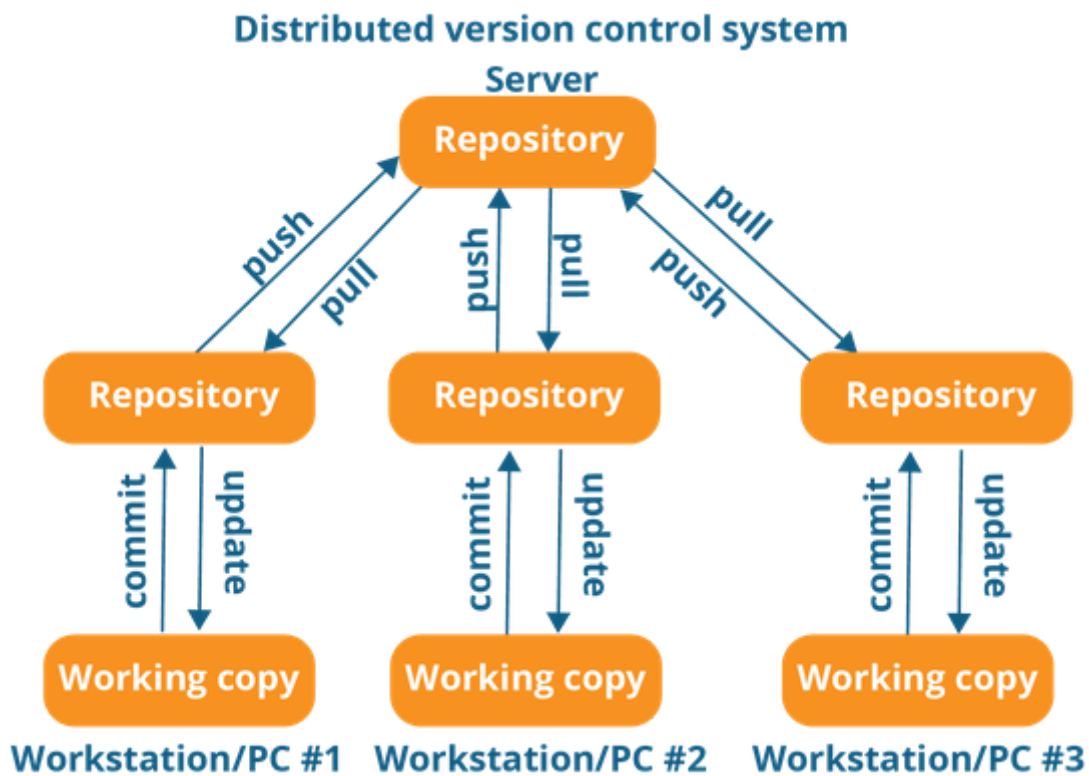
การติดตั้ง Git สามารถดาวน์โหลดได้ที่ <https://git-scm.com/> หลังจากติดตั้งเสร็จแล้ว สามารถตรวจสอบได้ว่าการติดตั้งนั้นเสร็จสมบูรณ์หรือไม่ โดยสามารถใช้คำสั่ง (macOS, Linux ใช้ Terminal ส่วนใน Windows ใช้ Git Bash ที่ติดตั้งมาพร้อมกับ Git)

```
$git --version
```

### 3.2. ความรู้เบื้องต้น

Git เปรียบเสมือนฐานข้อมูลแบบ Distributed เป็นการจัดเก็บกระจายไปในหลาย ๆ ที่ โดยหลัก ๆ จะแบ่งออกเป็น 2 ส่วน ดังนี้

1. **local** คือ ส่วนที่จัดเก็บในเครื่องที่เราทำงาน ซึ่งหลังจากเสร็จและต้องการแชร์ไฟล์กับคนอื่น จะต้อง push ขึ้น remote เพื่อให้คนอื่นได้รับไฟล์นั้น ๆ ด้วย
2. **remote** คือ Server กลาง ที่ทุกคนจะใช้งานร่วมกัน ทั้งในการดึงไฟล์ และอัปเดตไฟล์

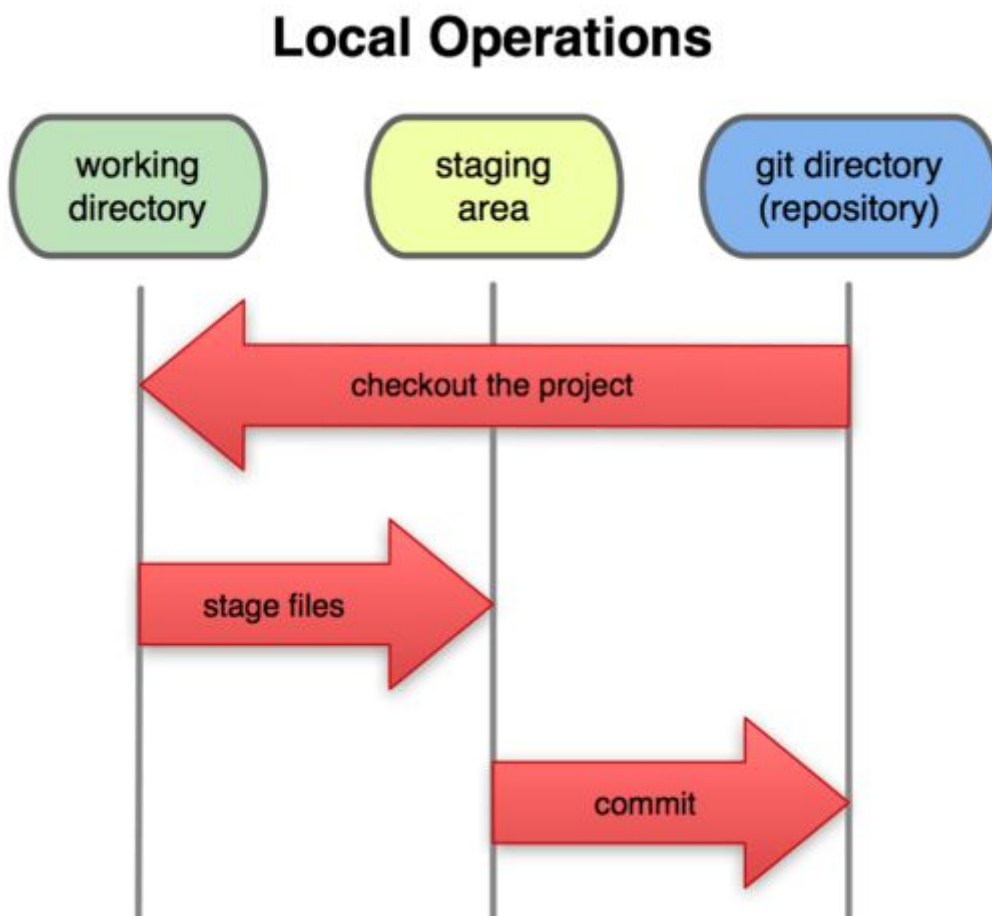


รูปที่ 1 Distributed version control system

### 3.3. 3 สถานะของ Git

Git แบ่งสถานะออกเป็น 3 สถานะ

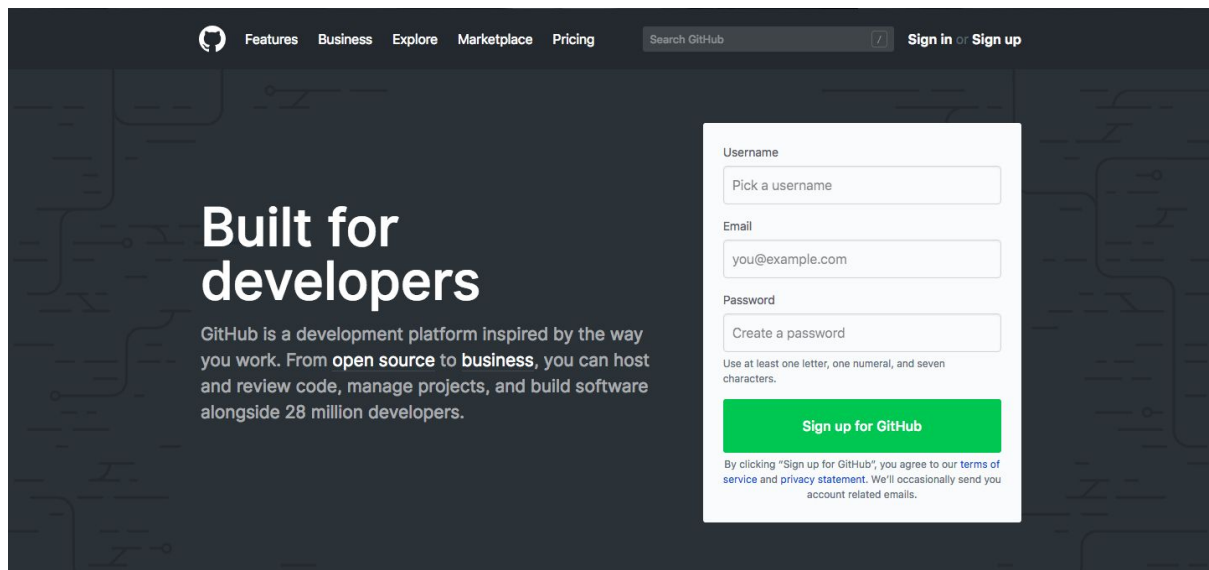
1. **Committed** (ถูกจัดเก็บในฐานข้อมูล local เรียบร้อย)
2. **Modified** (ไฟล์ที่มีการแก้ไข แต่ยังไม่ถูกจัดเก็บในฐานข้อมูล)
3. **Staged** (ไฟล์ที่มีการแก้ไข และถูกทำเครื่องหมายไว้เพื่อเตรียมทำการ committed ในขั้นต่อไป)



รูปที่ 2 Local Operations

## 4. เริ่มต้นใช้งาน Git

1. ให้นักศึกษาร่างบัญชี Github (<https://github.com/>) เพื่อใช้งาน Git Server



รูปที่ 3 Github.com



2. ให้นักศึกษาร่าง Repository ชื่อ *Lab-ITF* ดังรูปที่ 4

**Create a new repository**

A repository contains all the files for your project, including the revision history.

---

**Owner**      **Repository name**

 n3n / ITF-Lab 

Great repository names are short and memorable. Need inspiration? How about **verbose-carnival**.

**Description (optional)**

ITF-Lab 2018


---

☒ **Public**  
Anyone can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

---

☐ **Initialize this repository with a README**  
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

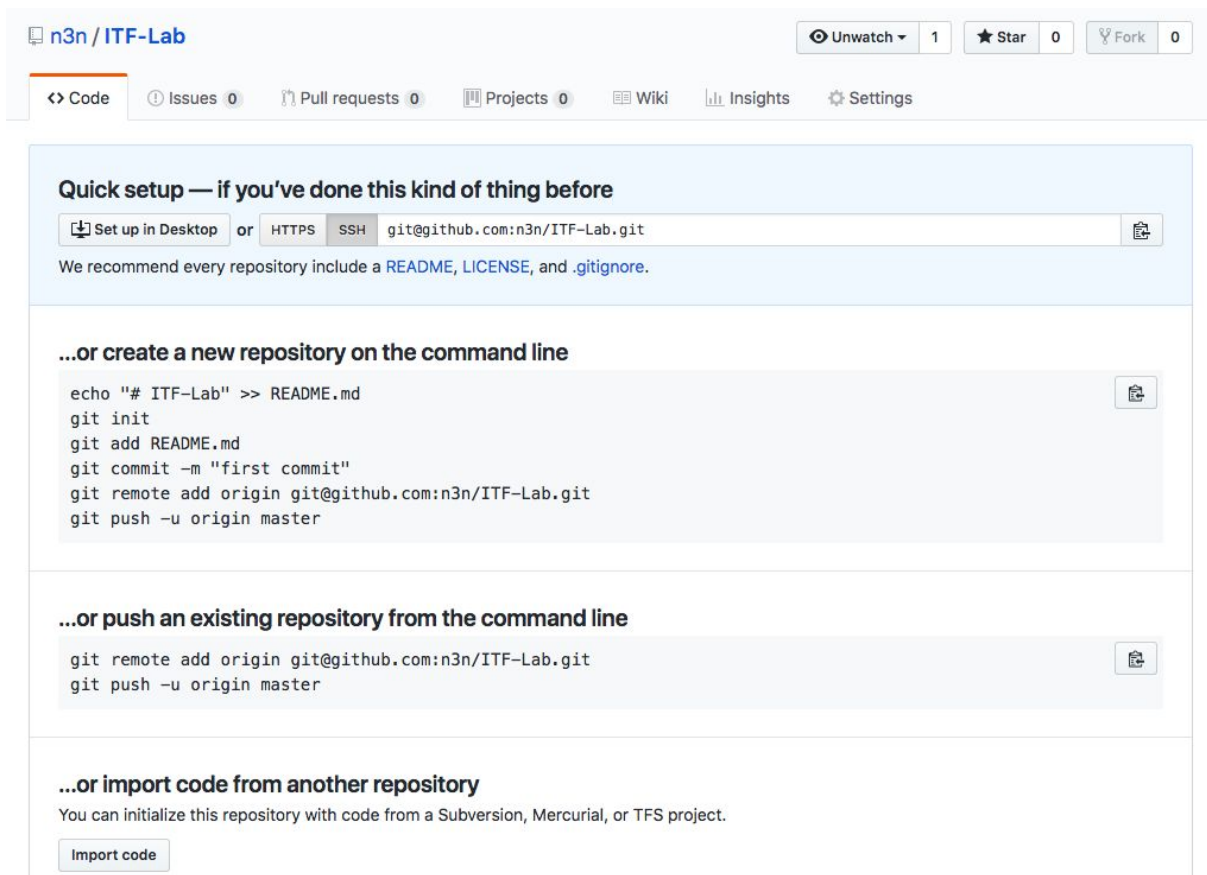
Add .gitignore: **None** | Add a license: **None** 

---

**Create repository**

รูปที่ 4 Create repository

### 3. หลังจากเสร็จสิ้นจะได้หน้าต่างดังรูปที่ 5



รูปที่ 5 “ITF-Lab” repository

### 4. ให้นักศึกษาเปิดโปรแกรม Terminal หรือ Git bash สำหรับคนที่ใช้ Windows และใช้คำสั่งเพื่อโคลน Repository ลงบนเครื่อง

```
$git clone https://github.com/{username}/ITF-Lab.git
```

\* ให้เปลี่ยน username เป็นของนักศึกษา

### 5. ให้นักศึกษาสร้างไฟล์ *README.md* พร้อมทั้งเนื้อหาภายในไฟล์ดังนี้

```
# ITF-Lab 2018
```

6. ให้นักศึกษาตรวจสอบสถานะ Git โดยใช้คำสั่งดังนี้

```
$git status
```

7. ให้ทำการ **stage** และ **commit** ไฟล์ในข้อที่ 5 และ **push** ขึ้น remote repository โดยมีคำสั่งดังนี้

```
$git add README.md      # ทำการ stage ไฟล์ที่ชื่อ README.md
$git status              # ตรวจสอบสถานะไฟล์
$git commit -m "first commit ITF-Lab"    # ทำการ commit ไฟล์ พร้อมทั้งข้อความบรรยาย
$git status              # ตรวจสอบสถานะไฟล์
$git push -u origin master  # ทำการ push ไฟล์ขึ้นบน remote server
```

8. หลังจากเสร็จสิ้นข้อที่ 7 ให้นักศึกษาริเฟรชหน้าเว็บ Repository เพื่อสังเกตความเปลี่ยนแปลง

9. ให้ทำการสร้างไฟล์ชื่อ **app.py** และใส่เนื้อหาภายในไฟล์ดังนี้

```
message = "Hello, ITF-Lab"
print(message)
```

10. ให้ทำการ **stage** และ **commit** ไฟล์ในข้อที่ 9 และ **push** ขึ้น remote repository โดยมีคำสั่งดังนี้


```
$git add app.py
$git commit -m "Add app.py with print message"
$git push
```

11. ให้ทำการแก้ไขไฟล์ในข้อที่ 9 เป็นดังนี้

```
# Comments
message = "Hello, ITF-Lab 2018!!!"
print(message)
```

12. ให้ทำการ **stage** และ **commit** ไฟล์ในข้อที่ 11 และ **push** ขึ้น remote repository

13. เมื่อเสร็จสิ้นทุกขั้นตอนแล้วให้ตรวจสอบ commit ล่าสุดบน remote repository ว่าถูกต้อง เพื่อให้แน่ใจว่าไฟล์ใน local และ remote ตรงกันแล้ว

14. ให้นักศึกษาทดลองแก้ไขไฟล์บน Github โดยให้คลิกเข้าไฟล์ *README.md* บนเว็บ และคลิกรูป  ที่อยู่ด้านขวา โดยให้มีเนื้อหาในไฟล์ดังนี้

```
# ITF-Lab 2018
```

```
## Author
```

```
* ชื่อ-นามสกุล รหัสนักศึกษา
```

หลังจากนั้น Commit changes ให้เสร็จสิ้น

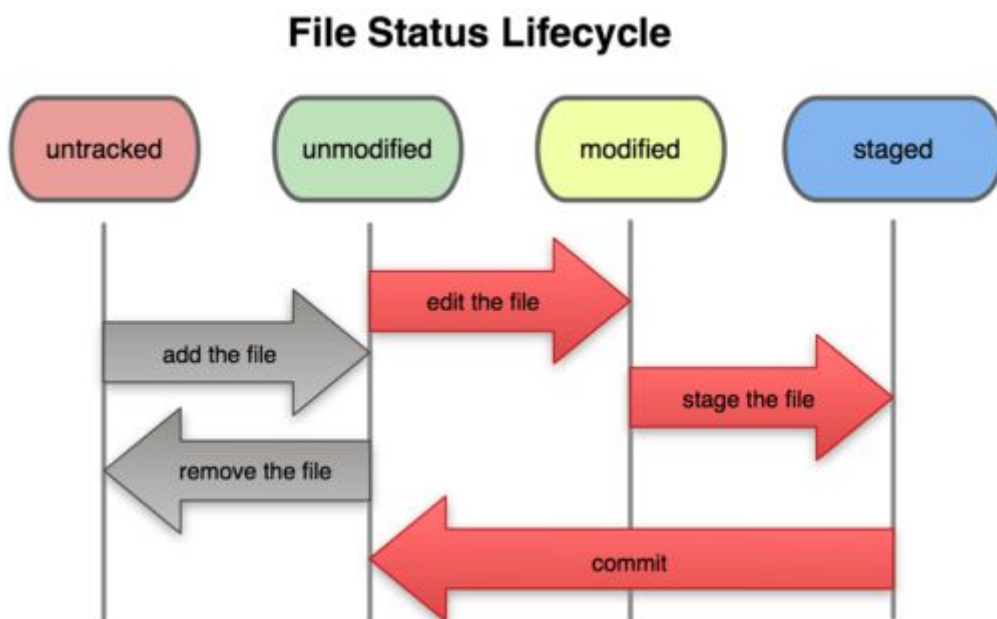
15. ให้นักศึกษาอัปเดตไฟล์ใน local โดยใช้คำสั่ง

```
$git pull
```

เพื่อดึงข้อมูลบน remote ลงมาบนเครื่อง หลังจากนั้นให้สังเกตการเปลี่ยนแปลงของไฟล์

\* หากนักศึกษาคนไหน ทำการแก้ไขในเครื่อง local และยังไม่ได้ทำการ stage จะไม่สามารถ pull ได้ หรือ หากใครทำการแก้ไขไฟล์ *README.md* ก่อนหน้าที่จะแก้ไขไฟล์บน remote หลังจาก pull อาจมีการ Conflict ได้

รูปที่ 6 แสดงวงจรสถานะของไฟล์ภายใน Git จากการใช้คำสั่งต่าง ๆ

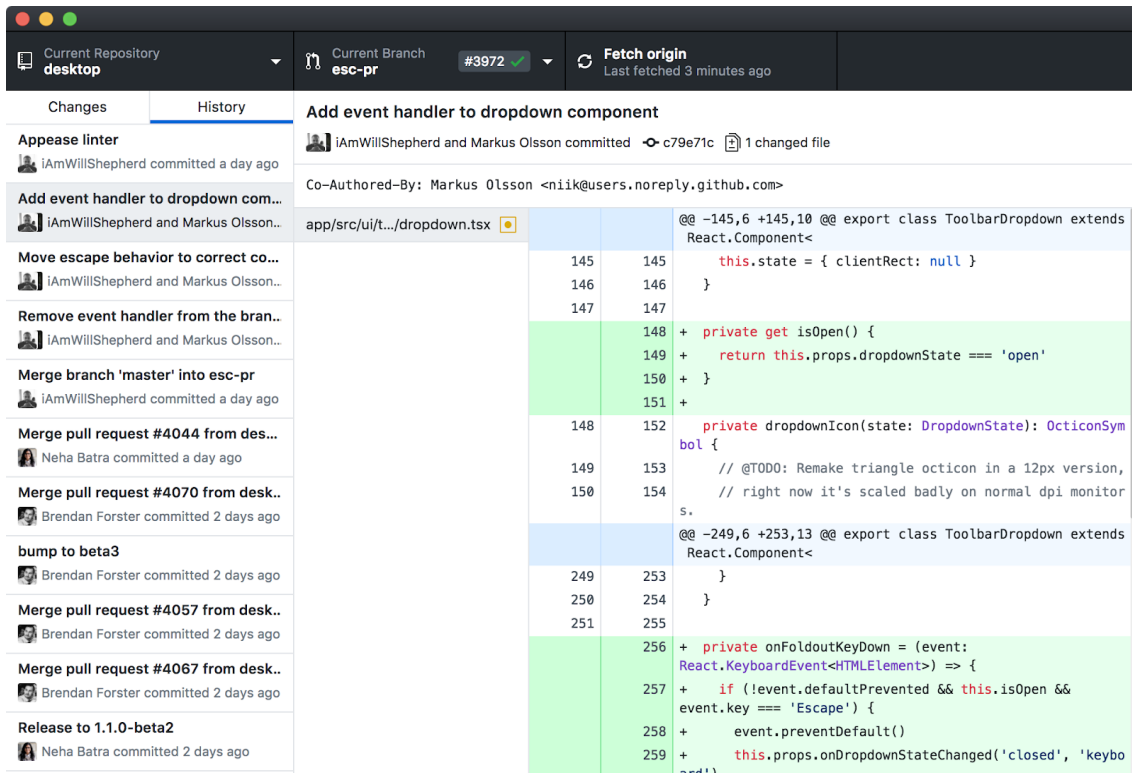


รูปที่ 6 File Status Lifecycle

## 5. Git GUI Clients

ในบทที่ 4 ได้เรียนรู้การใช้งาน Git พื้นฐานในรูปแบบ Command Line กันแล้ว ซึ่งจริง ๆ Git ก็มีในรูปแบบโปรแกรม ที่มีหน้าต่างสวยงาม เข้าใจง่ายเช่นกัน โดยในที่นี้ได้ยกตัวอย่างมาด้วยกัน 3 โปรแกรม โดยเป็นโปรแกรมฟรีทั้งหมด ไม่เสียค่าใช้จ่ายใด ๆ ทั้งสิ้น

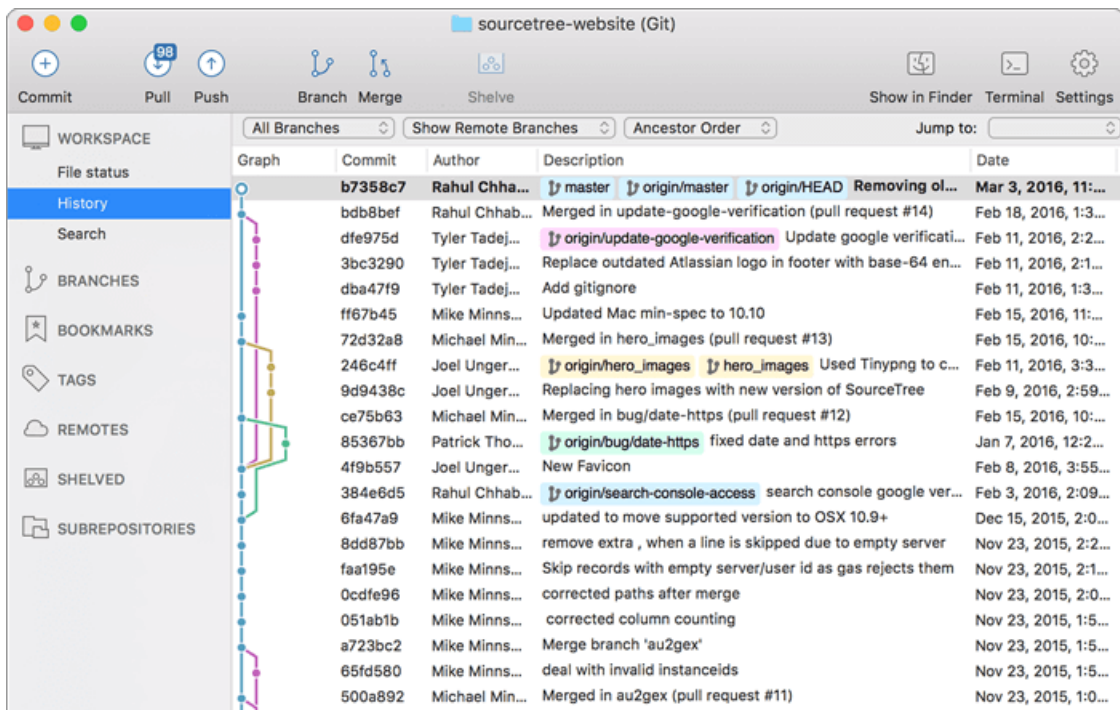
### 5.1. GitHub Desktop (<https://desktop.github.com/>)



รูปที่ 7 GitHub Desktop

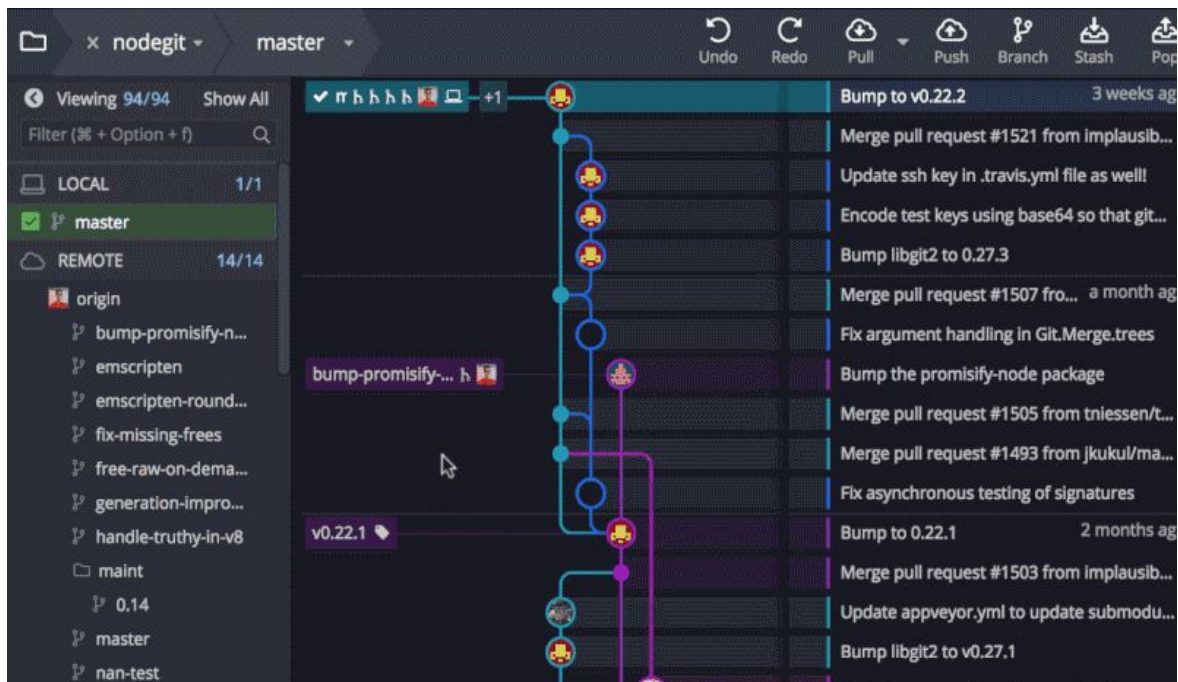


## 5.2. Sourcetree (<https://www.sourcetreeapp.com/>)



รูปที่ 8 Sourcetree

## 5.3. GitKraken (<https://www.gitkraken.com/>)



รูปที่ 9 GitKraken

## ศึกษาเพิ่มเติม

- <https://education.github.com/git-cheat-sheet-education.pdf>
- <https://medium.com/@pakin/git-คืออะไร-git-is-your-friend-c609c5f8efea>
- <https://git-scm.com/book/th/v1/>
- <https://guides.github.com/introduction/git-handbook/>
- <https://www.atlassian.com/git/tutorials>
- <http://education.github.com/>