

Introduktion till git och github (för statistiker)

Måns Magnusson

2016

Overview

Versionshantering

Git

Github, GitLab, BitBucket ...

Grunderna i git

Syfte

- ▶ Grunden för modern versionshantering

Syfte

- ▶ Grunden för modern versionshantering
- ▶ Använda git i “skarpt läge”

Syfte

- ▶ Grunden för modern versionshantering
- ▶ Använda git i “skarpt läge”
- ▶ Egen fördjupning vid behov

Syfte

- ▶ Hur versionhanterar ni idag?
- ▶ Om ni samarbetar flera i samma projekt?
- ▶ Om ni ändrar i varandras kod?

Vad är versionshantering?

[What is version control?]

Varför versionhantera?

- ▶ Samarbete i samma projekt, i samma kod

Varför versionhantera?

- ▶ Samarbete i samma projekt, i samma kod
- ▶ Spara olika versioner

Varför versionhantera?

- ▶ Samarbete i samma projekt, i samma kod
- ▶ Spara olika versioner
- ▶ Återställa tidigare versioner

Varför versionhantera?

- ▶ Samarbete i samma projekt, i samma kod
- ▶ Spara olika versioner
- ▶ Återställa tidigare versioner
- ▶ Spårbarhet (av kod och data)

Varför versionhantera?

- ▶ Samarbete i samma projekt, i samma kod
- ▶ Spara olika versioner
- ▶ Återställa tidigare versioner
- ▶ Spårbarhet (av kod och data)
- ▶ Best practice

Best practice for scientific computing

Best practice for scientific computing (Wilson et al., 2014):

Best practice for scientific computing

Best practice for scientific computing (Wilson et al., 2014):

- ▶ use a version control system (3b)

Best practice for scientific computing

Best practice for scientific computing (Wilson et al., 2014):

- ▶ use a version control system (3b)
- ▶ put everything that has been created manually in version control (3c)

Best practice for scientific computing

Best practice for scientific computing (Wilson et al., 2014):

- ▶ use a version control system (3b)
- ▶ put everything that has been created manually in version control (3c)
- ▶ use pre-merge code reviews (8a)

Best practice for scientific computing

Best practice for scientific computing (Wilson et al., 2014):

- ▶ use a version control system (3b)
- ▶ put everything that has been created manually in version control (3c)
- ▶ use pre-merge code reviews (8a)
- ▶ use an issue tracking tool (8c)

Vad är git?

- ▶ Versionshanteringssystem

Vad är git?

- ▶ Versionshanteringssystem
- ▶ Skapades 2005 för Linux

Vad är git?

- ▶ Versionshanteringssystem
- ▶ Skapades 2005 för Linux
- ▶ Distribuerat och lokalt

Vad är git?

- ▶ Versionshanteringssystem
- ▶ Skapades 2005 för Linux
- ▶ Distribuerat och lokalt
- ▶ Kommandoradbaserat men... SourceTree (eller R-Studio) gör livet lättare

Vad är git?

- ▶ Versionshanteringssystem
- ▶ Skapades 2005 för Linux
- ▶ Distribuerat och lokalt
- ▶ Kommandoradbaserat men... SourceTree (eller R-Studio) gör livet lättare
- ▶ Passar stora och små projekt

Vad är git?

- ▶ Versionshanteringssystem
- ▶ Skapades 2005 för Linux
- ▶ Distribuerat och lokalt
- ▶ Kommandoradbaserat men... SourceTree (eller R-Studio) gör livet lättare
- ▶ Passar stora och små projekt
- ▶ På väg att bli standard för dataanalys och statistisk programmering

Varför git?

- ▶ (Relativt) enkelt att börja använda

Varför git?

- ▶ (Relativt) enkelt att börja använda
- ▶ Kan användas av enskilda medarbetare

Varför git?

- ▶ (Relativt) enkelt att börja använda
- ▶ Kan användas av enskilda medarbetare
- ▶ Påverkar inte mappstrukturer och filer

Varför git?

- ▶ (Relativt) enkelt att börja använda
- ▶ Kan användas av enskilda medarbetare
- ▶ Påverkar inte mappstrukturer och filer
- ▶ Öppen källkod

Varför git?

- ▶ (Relativt) enkelt att börja använda
- ▶ Kan användas av enskilda medarbetare
- ▶ Påverkar inte mappstrukturer och filer
- ▶ Öppen källkod
- ▶ Distribuerat

Varför git?

- ▶ (Relativt) enkelt att börja använda
- ▶ Kan användas av enskilda medarbetare
- ▶ Påverkar inte mappstrukturer och filer
- ▶ Öppen källkod
- ▶ Distribuerat
- ▶ Snabbt

Varför git?

- ▶ (Relativt) enkelt att börja använda
- ▶ Kan användas av enskilda medarbetare
- ▶ Påverkar inte mappstrukturer och filer
- ▶ Öppen källkod
- ▶ Distribuerat
- ▶ Snabbt
- ▶ Säkert/moget

Varför git?

- ▶ (Relativt) enkelt att börja använda
- ▶ Kan användas av enskilda medarbetare
- ▶ Påverkar inte mappstrukturer och filer
- ▶ Öppen källkod
- ▶ Distribuerat
- ▶ Snabbt
- ▶ Säkert/moget
- ▶ Standard i R-världen - hantering av R-paket på github

Varför git?

- ▶ (Relativt) enkelt att börja använda
- ▶ Kan användas av enskilda medarbetare
- ▶ Påverkar inte mappstrukturer och filer
- ▶ Öppen källkod
- ▶ Distribuerat
- ▶ Snabbt
- ▶ Säkert/moget
- ▶ Standard i R-världen - hantering av R-paket på github
- ▶ Bas i modern (statistisk) mjukvaruutveckling

Vad är git remote?

- ▶ Central lagring av versionshanterad kod, data m.m.

Vad är git remote?

- ▶ Central lagring av versionshanterad kod, data m.m.
- ▶ Distribuera kod mellan datorer

Vad är git remote?

- ▶ Central lagring av versionshanterad kod, data m.m.
- ▶ Distribuera kod mellan datorer
- ▶ github, GitLab, BitBucket...

Vad är git remote?

- ▶ Central lagring av versionshanterad kod, data m.m.
- ▶ Distribuera kod mellan datorer
- ▶ github, GitLab, BitBucket...
- ▶ Elementära hemsidor för projekt

Vad är git remote?

- ▶ Central lagring av versionshanterad kod, data m.m.
- ▶ Distribuera kod mellan datorer
- ▶ github, GitLab, BitBucket...
- ▶ Elementära hemsidor för projekt
- ▶ Issue trackers / Wikis / Diskussionsmöjligheter

Varför git remote?

- ▶ Underlättar samarbeten (kraftigt)

Varför git remote?

- ▶ Underlättar samarbeten (kraftigt)
- ▶ Distribuera kod mellan datorer

Varför git remote?

- ▶ Underlättar samarbeten (kraftigt)
- ▶ Distribuera kod mellan datorer
- ▶ Säkerhet och backup

Varför git remote?

- ▶ Underlättar samarbeten (kraftigt)
- ▶ Distribuera kod mellan datorer
- ▶ Säkerhet och backup
- ▶ Administrera projekt

Varför git remote?

- ▶ Underlättar samarbeten (kraftigt)
- ▶ Distribuera kod mellan datorer
- ▶ Säkerhet och backup
- ▶ Administrera projekt
- ▶ The burn it all down workflow

Section 4

Grunderna i git

Acknowledgments

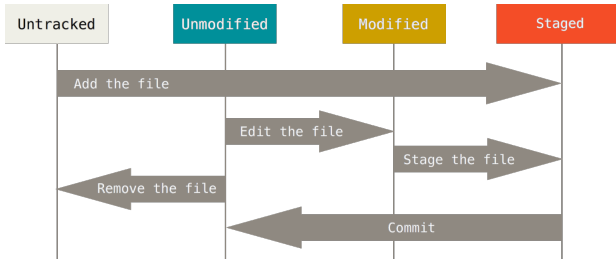
Bilder är tagna från:

- ▶ The git book [\[här\]](#)
- ▶ Atlassian git tutorial [\[här\]](#)

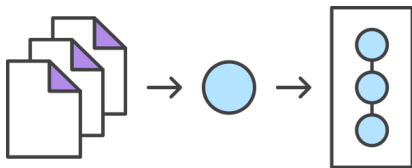
Repository



Arbetsflödet i git

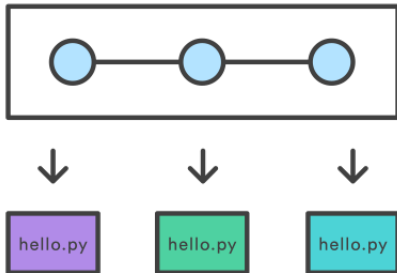


Commit



Commit

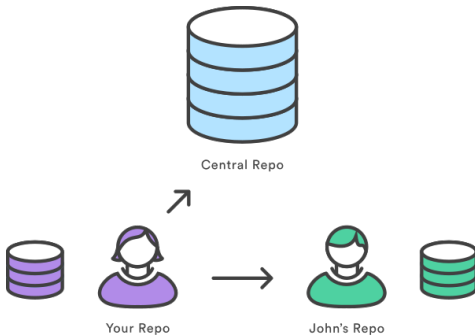
Recording Snapshots (Git)



Commit message

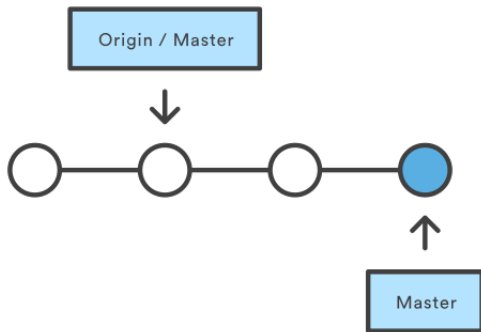
If applied, this commit will [your commit message here]

Remote



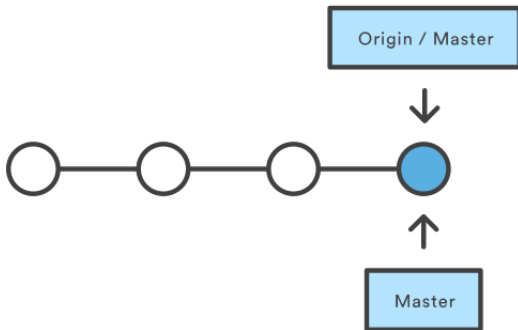
Local till remote - Push

Before Pushing

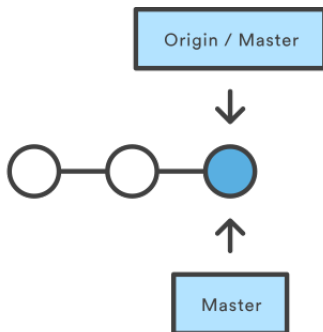


Local till remote - Push

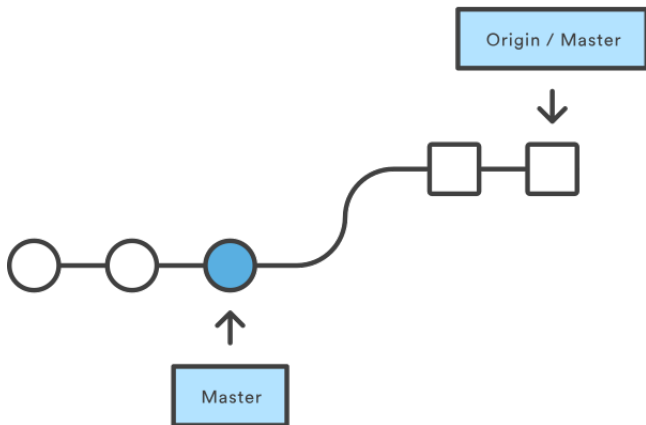
After Pushing



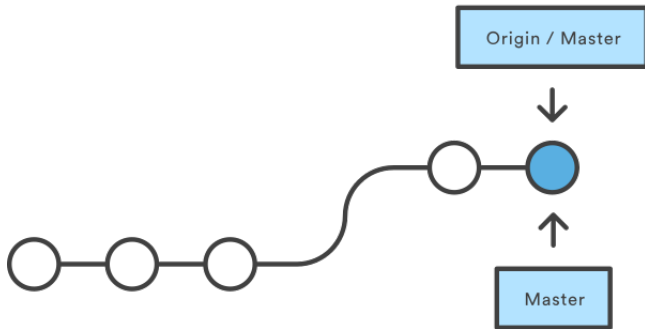
Remote till local - Pull



Remote till local - Pull



Remote till local - Pull



References

Wilson, G., Aruliah, D., Brown, C. T., Hong, N. P. C., Davis, M., Guy, R. T., Haddock, S. H., Huff, K. D., Mitchell, I. M., Plumbley, M. D., et al., 2014. Best practices for scientific computing. PLoS Biol 12 (1), e1001745.