

Deal with (g)it

Top 5 tips to make the best out of (g)it



#1 Sign your commits

Make visible you are the real owner of a commit.



Code Issues 4 Pull requests Actions Projects Security ...

main ▾

Commits on Jan 10, 2022

README more like read deez nuts
torvalds committed on Jan 10, 2022

4 f0326ef

Commits on Dec 24, 2021

Rename files
torvalds committed on Dec 24, 2021

16 8f991f9

Commits on Dec 5, 2021



Follow

torvalds Linus Torvalds

Portland, OR
Committed to this repository

13 0bb33e3

iously skeptical

torvalds committed on Nov 23, 2021

19 8c95bc3

when the whitespace sus
torvalds committed on Nov 22, 2021

12 5eaa2b0

Commits on Oct 31, 2021

UwU
torvalds committed on Oct 31, 2021

27 dbed716

Commits on Oct 30, 2021

sussy little baka uwu
torvalds committed on Oct 30, 2021

118 4f503a0

Are these commit from Linus?

- this is a real github repo
- these are real commits

GitHub: Amog-OS/AmogOS

Fellow linux redditors may know this already

Anyone can be Linus!

- git config is not verifying anything

So you can set up your local config ...

```
git config --global user.name "torvalds"  
git config --global user.email "torvalds@osdl.org"
```

... and be Linus:

```
touch proof  
git stage proof  
git commit -m "I am Linus!"
```

... at least that's what it's going to look like

Commits on Jan 24, 2023

chore: Remove PR template	Verified	1ed42c0	🔗
chore(issues): Add new bug report template	Verified	cf9a9ec	🔗
chore(issues): Add new feature request template	Verified	834ac2d	🔗

Commits on Jan 23, 2023

ci: replace github release orb with github cli	Verified	8b4e8a1	🔗
------------------------------------------------	--------------------------	-------------------------	-------------------

Commits on Nov 8, 2022

docs: Add FOSSA badge	Verified	55958	🔗
-----------------------	--------------------------	-----------------------	-------------------

Commits on Oct 14, 2022

chore(sonar): Add config	Verified	refad6d	🔗
docs: Add sonarcloud badges	Verified	e7adce8	🔗
chore: Align tests with go conventions	Verified	08fd2b8	🔗

Commits on Sep 18, 2022

docs: Add dependabot badge	Verified	753af63	🔗
----------------------------	--------------------------	-------------------------	-------------------

Verified commits to the rescue!

- signed with [GPG](#)
- visible on all git hosting platforms
- checkable with the git cli

GPG stands for GNU Privacy Guard and is a free tool to sign and encrypt things

Setting it up is easy - 3 steps to victory

1. create a [GPG key \(Tutorial\)](#)
2. add the key to your user profile on the git hosting platform (e.g. [GitHub](#))
3. configure your local git installation:

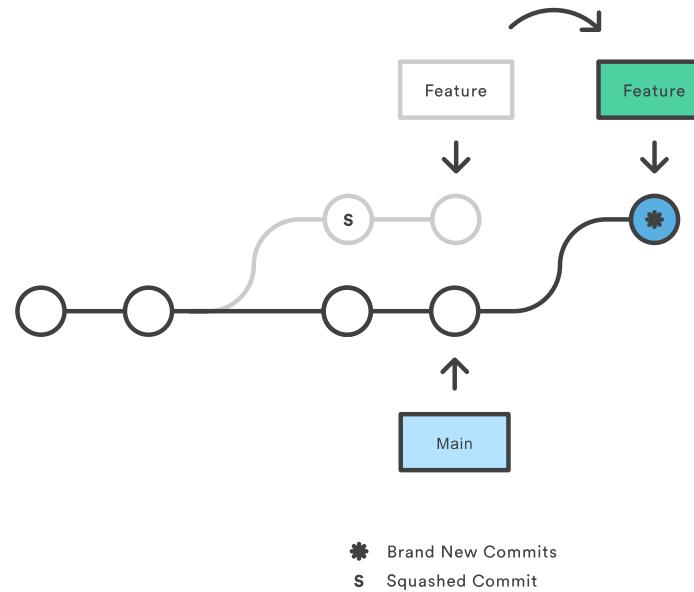
```
# set gpg key  
git config --global user.signingkey <key-id>  
  
# autosign commits  
git config --global commit.gpgsign true
```

Alternative: Use [GPG Key Creation Assistant](#) - created by me

#2 Keep your history clean

| accountants don't use erasers or they end up in jail

But luckily we are not accountants!



Squash your feature branches

- powerful feature allowing you to combine multiple commits
- supported by git hosts such as GitHub/GitLab on PRs

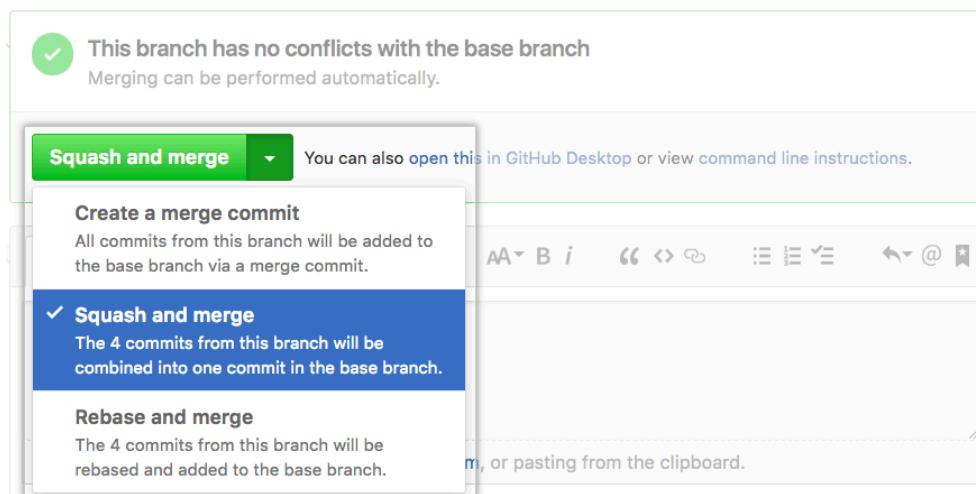
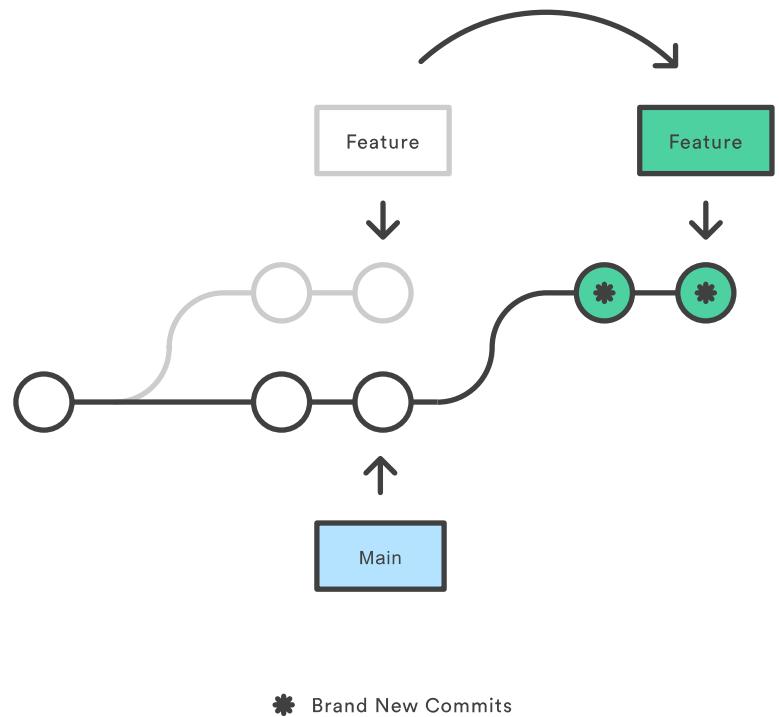


Image for git squash: atlassian.com



Rebase your changes

- local repo is not up to date
- main branch has evolved faster than feature branch
- smaller changes most of the time don't need to be merged

#3 Simplify your daily life

Git and your shell has more to offer than just builtin commands!

Add helper tools

- any command that's prefixed with `git-` can be a git command
- `git-semver-tag` becomes `git semver-tag`

Create alias in your git config

- entire shell command
- another subcommand

e.g. here are mine:

```
[alias]
p = pull
squash-all = "!f(){ git reset $(git commit-tree HEAD^{tree} -m \"${1:-A new start}\");};f"
sync = "f() { git pull --rebase && git push; };f"
```

Alias entire git commands

- for the extreme lazy
- add to your shellrc

```
# use gpush  
alias gpush = "git push"
```

```
# use commit "my message"  
alias commit = "git stage . && git commit -m"
```

#4 Use branches and tags wisely

Nothing is immutable by default.

Choose the right branching concept

- trunk based development is a valid choice
- build your workflow around the real world, not some theory
- committing to the main branch is not (always) a sin
- be creative and think out of the box
- don't see branching concepts as a religion

Tags are not immutable

- tags can be changed and deleted, making them also work as pointers
- that can break dependency pins
- only unique thing in git are commit hashes

#5 Rely on conventions

Who doesn't love conventions?

Commit with respect to conventional commits

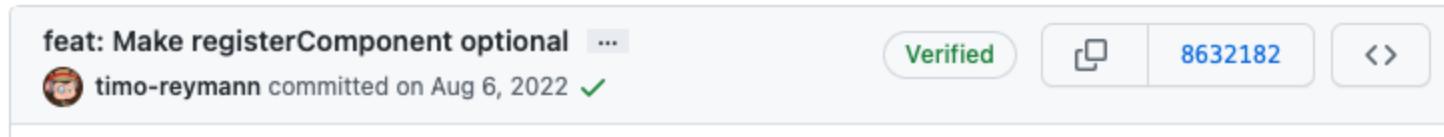
- be specific about your impact
- short and concise description
- explain details in the body

The format is super simple:

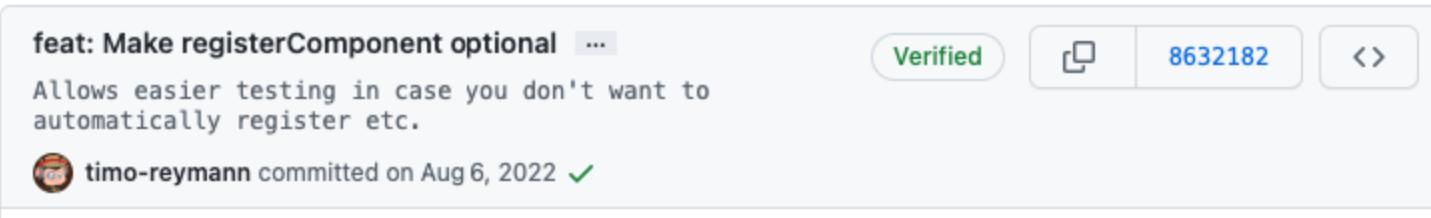
```
<type>[(optional scope)]: <description>  
[optional body]  
[optional footer(s)]
```

Use the commit body

- the commit can have a body, use it when required
- collapses by default in web UIs and toolings
 - when viewing the history:



- after clicking on the dots:



Use Conventional Comments on PRs

- no more undertone in your comments
- clear scope and less words to type and read
- take time for praise as well

The format, again, is super simple:

```
<label> [(decorations)]: <subject>
```

```
[discussion]
```

Recap

- sign your commits with GPG
- keep the history clean, dont hesitate to use erasers
- make your daily life easier with git and shell alias as well as third party packages and scripts
- be aware tags and branches are mutable all the way, only reliable thing are commit hashes
- use conventions for commits and PRs, don't forget about the message body



Q&A

Questions, concerns, ideas? - Now is the time

That's (g)it

Slides (Source):

[github/timo-reymann/slides-deal-with-git](https://github.com/timo-reymann/slides-deal-with-git)

Slides (HTML):

deal-with-git.slides.timo-reymann.de

Twitter: [@timo_reymann](https://twitter.com/timo_reymann)

GitHub: [timo-reymann](https://github.com/timo-reymann)

In case of fire



- ➊ 1. git commit
- ➋ 2. git push
- ➌ 3. leave building

