

Federico Cantini  
[federico.cantini@lib4ri.ch](mailto:federico.cantini@lib4ri.ch)

Frank Hösli  
[frank.hoesli@lib4ri.ch](mailto:frank.hoesli@lib4ri.ch)

# Make your life easier with GIT

# GIT basics

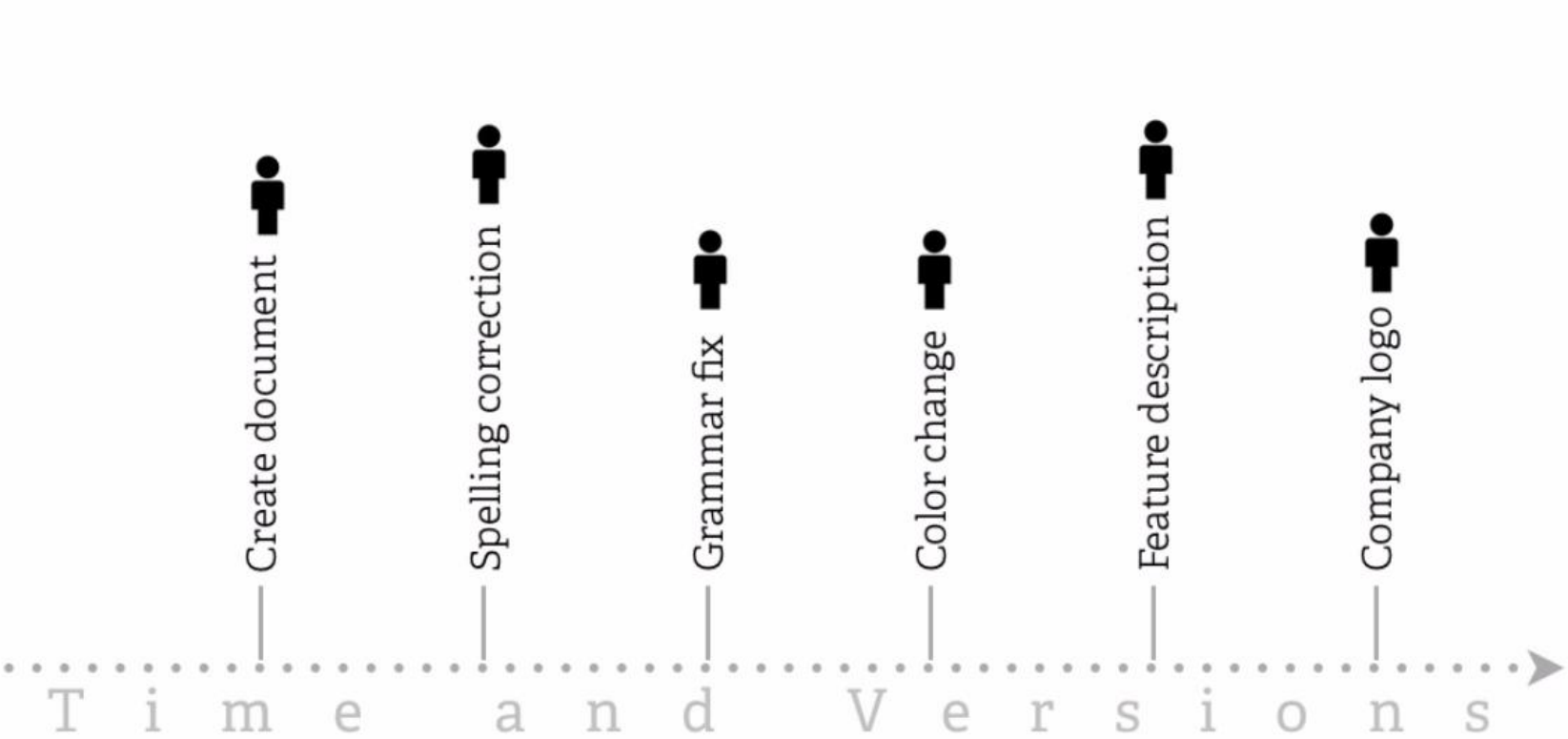
# Motivation

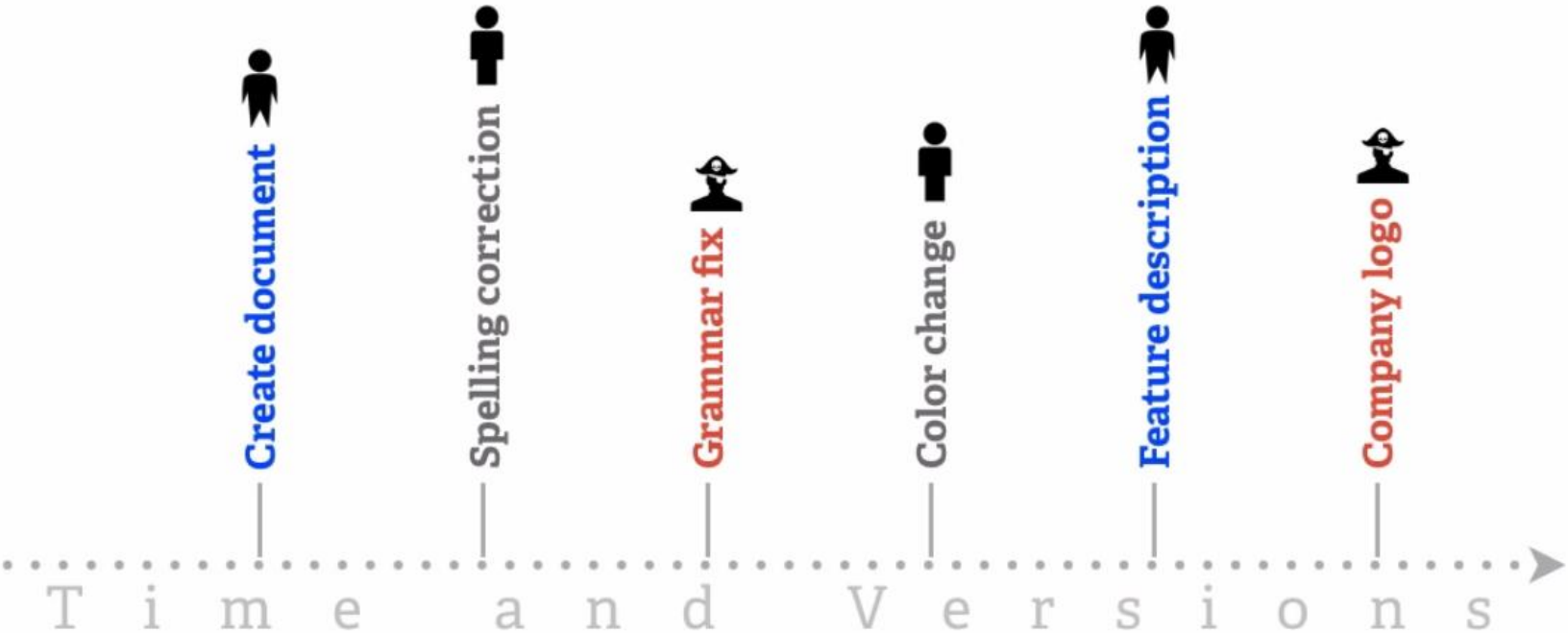
Have you ever:

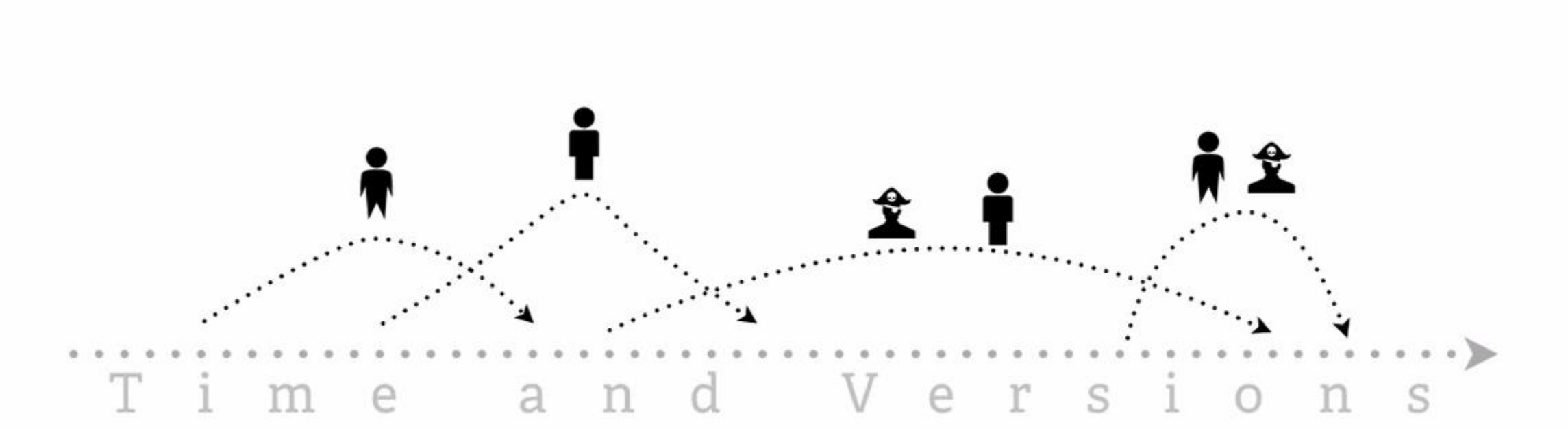
- Made changes you didn't mean to?
- Made some changes, but can't remember what they were?
- Wanted to try something but was worried about breaking something else?
- Went crazy trying to figure out which version of your code produced certain output?
- Tried to set up a convenient workflow for working on code together with others?

# Version control

- Version control is a system that records changes to a file or set of files over time
- It allows to:
  - revert selected files back to a previous state
  - revert the entire project back to a previous state
  - compare changes over time
  - work concurrently with others
  - see who last modified something and when
- If you break things you can (usually) easily recover







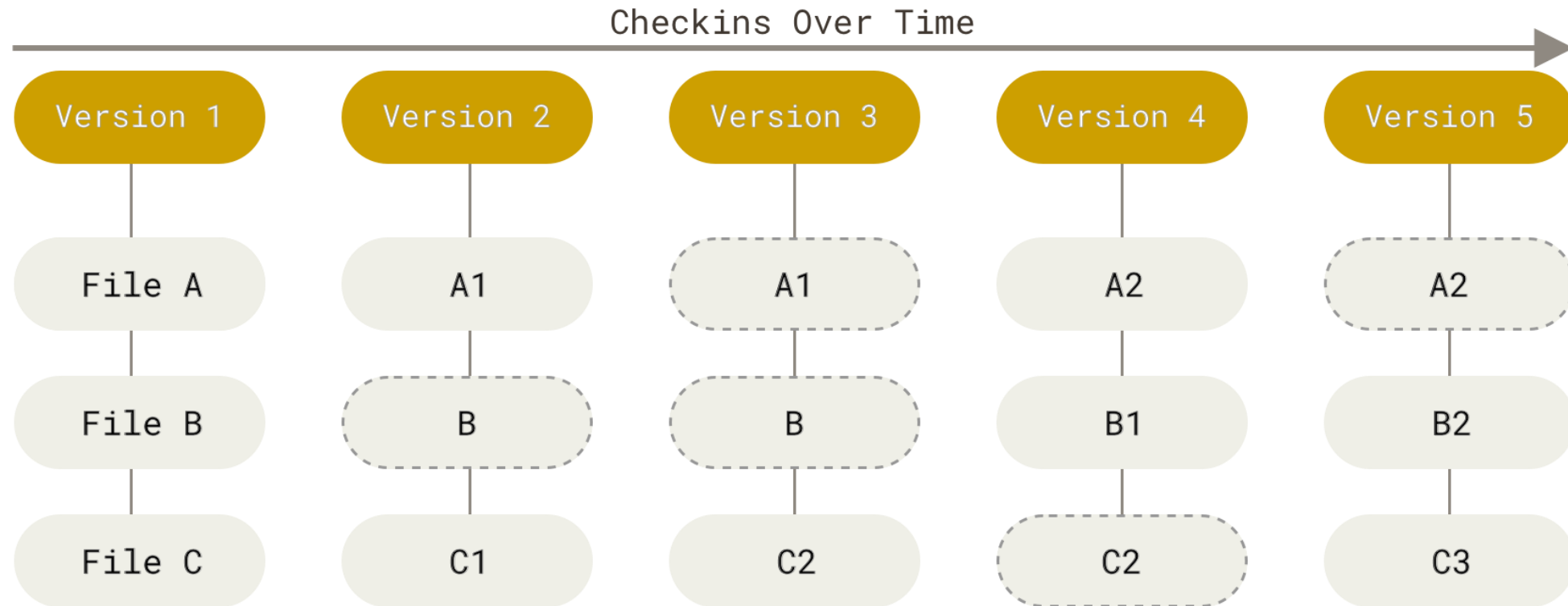


<https://git-scm.com>

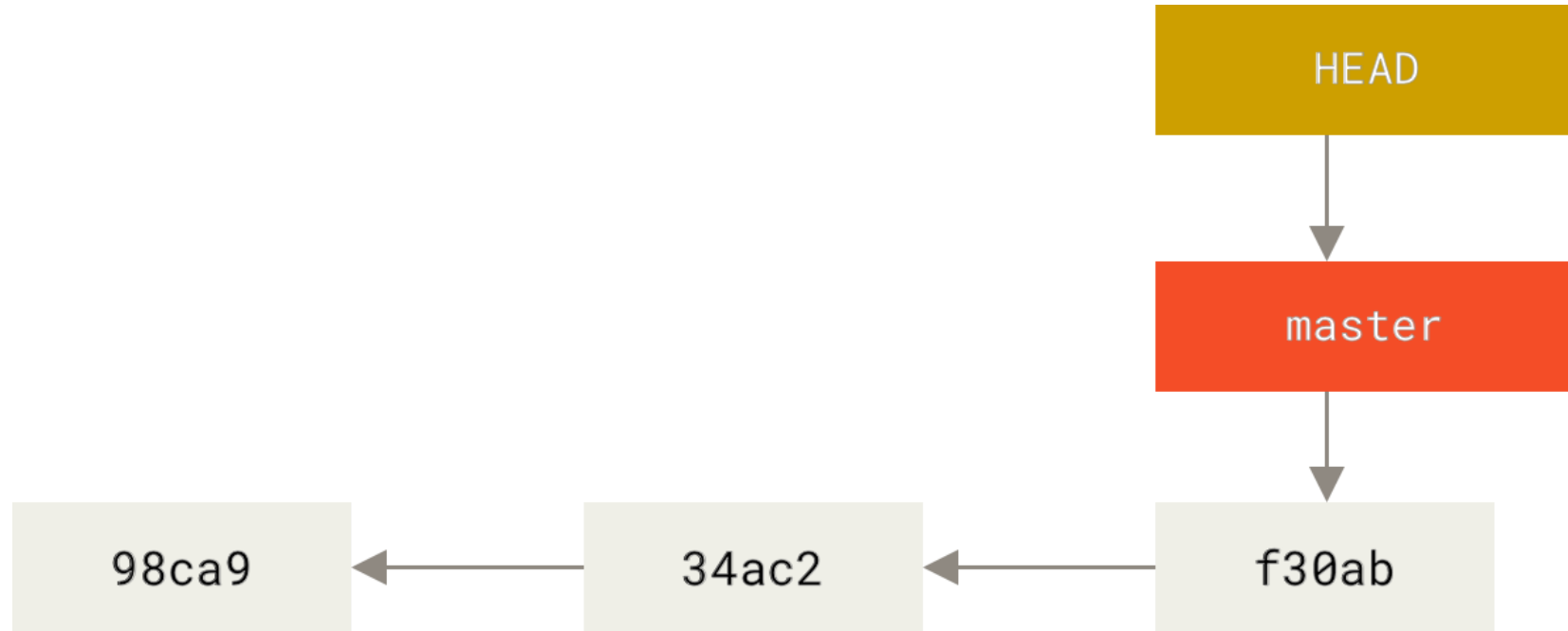
- First release April 2005
- Started by the team developing the Linux kernel
- De facto standard Version Control System
- Free/open source software (GPL v.2 license) <https://git-scm.com/about/free-and-open-source>
- Available for GNU/Linux, macOS and Windows <https://git-scm.com/downloads>
- Works with any programming language (any text file indeed and possibly other file types)
- Command line interface
- GUI clients available <https://git-scm.com/downloads/guis>
- **git ≠ GitHub ≈ GitLab ≈ Codeberg ≈ ...**

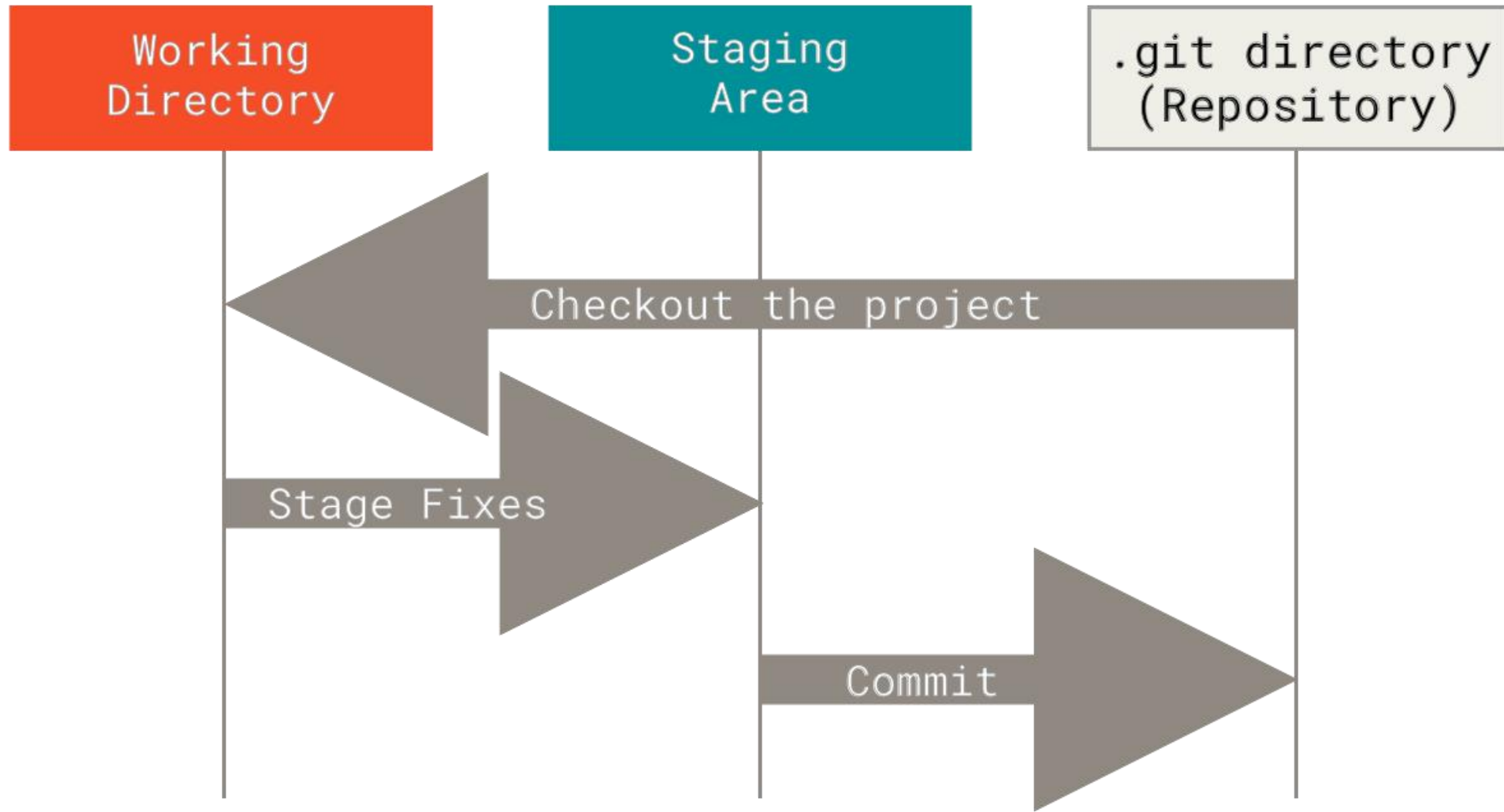
"Git logo" by [Jason Long](#) is licensed under [CC BY 3.0](#)

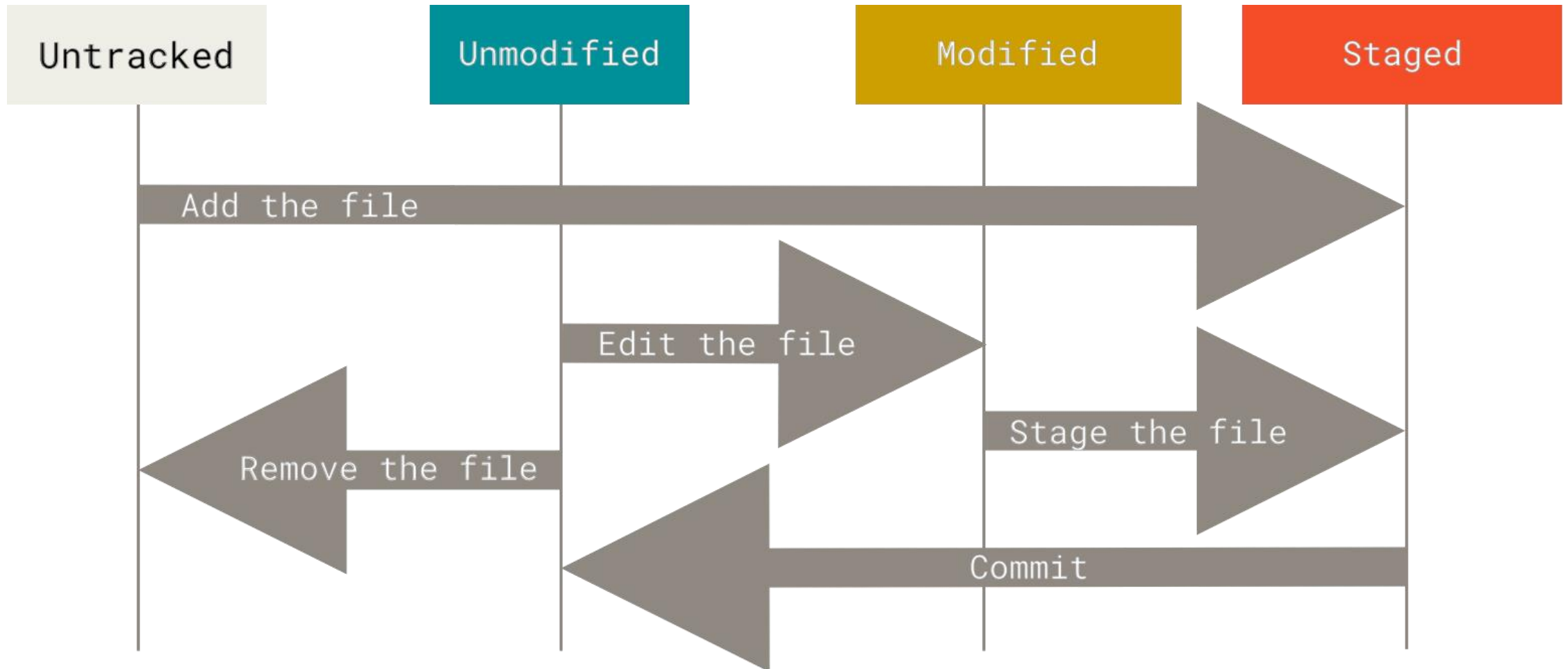
# Locally track changes



# Using GIT locally







```
$ git config --global user.name "John Doe"
$ git config --global user.email johndoe@example.com
```

*Windows (Notepad++):*

```
$ git config --global core.editor "'C:/Program Files/Notepad++/notepad++.exe' -multiInst -notabbar -nosession -noPlugin"
```

*Windows (Notepad):*

```
$ git config core.editor notepad
```

*macOS:*

```
$ git config --global core.editor "open --wait-apps --new -e"
```

```
$ git init <repository name>
$ git status
$ git add .
```

```
$ git diff
```

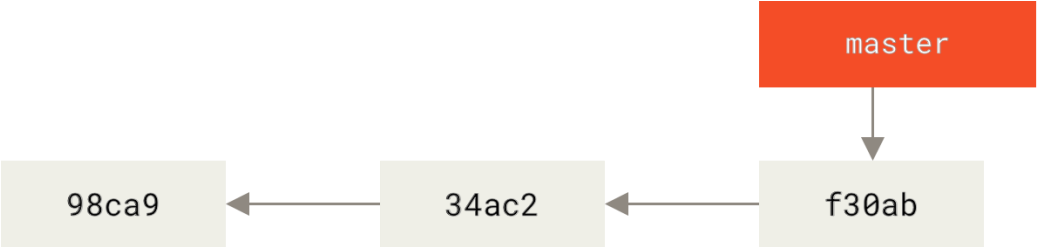
```
$ git log --all --graph
```

```
$ git commit -m "<message>"
$ git commit --amend
```

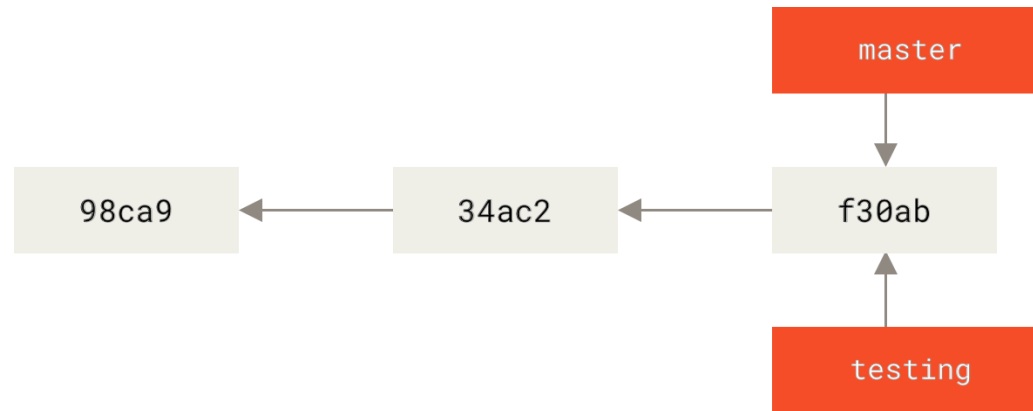
```
$ git reset <ID> --hard
$ git checkout <ID>
```

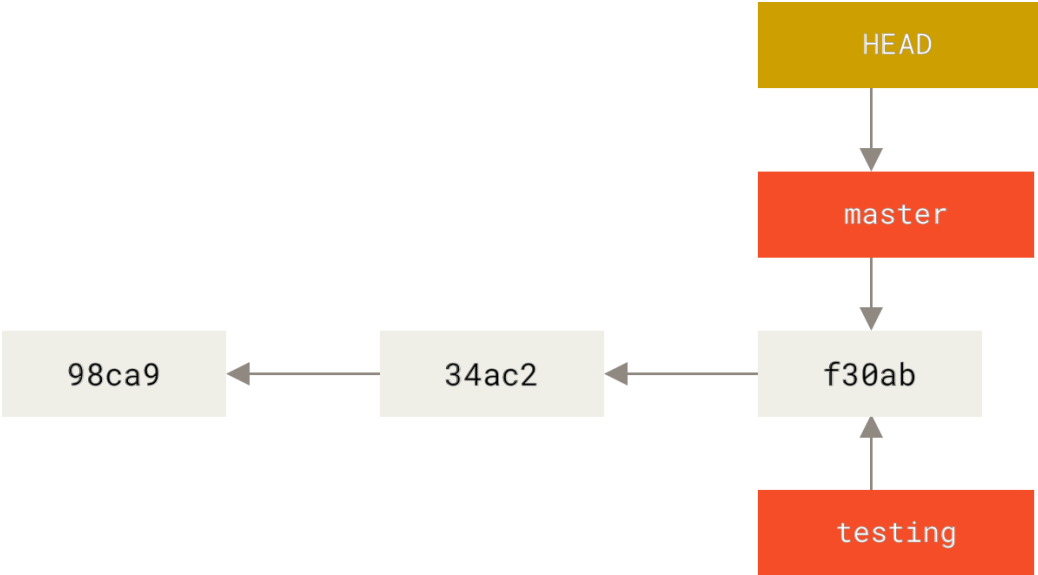
```
$ git tag -a v<version number> -m "<version description>"
```

# Branching

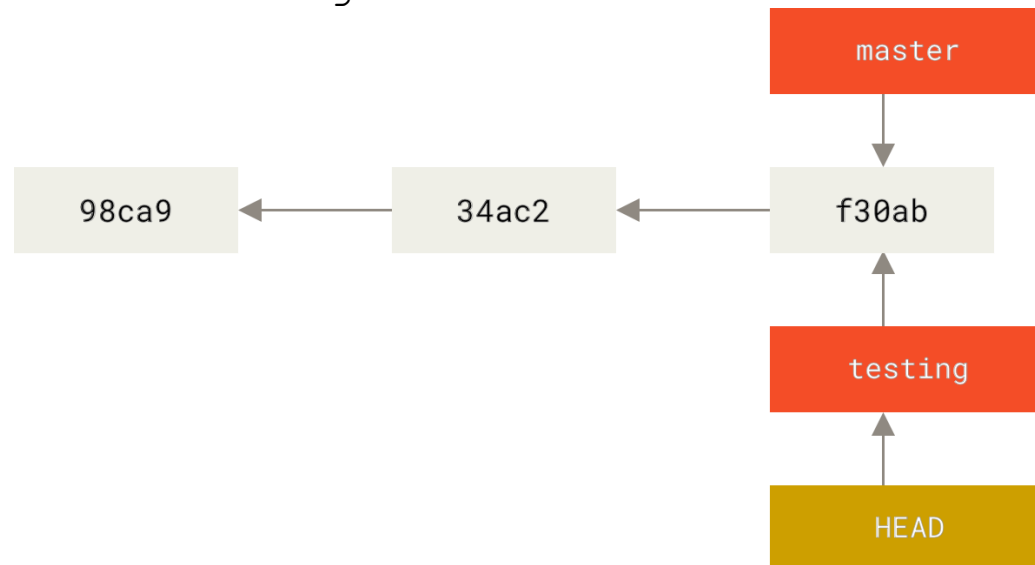


```
$ git branch -c testing
```

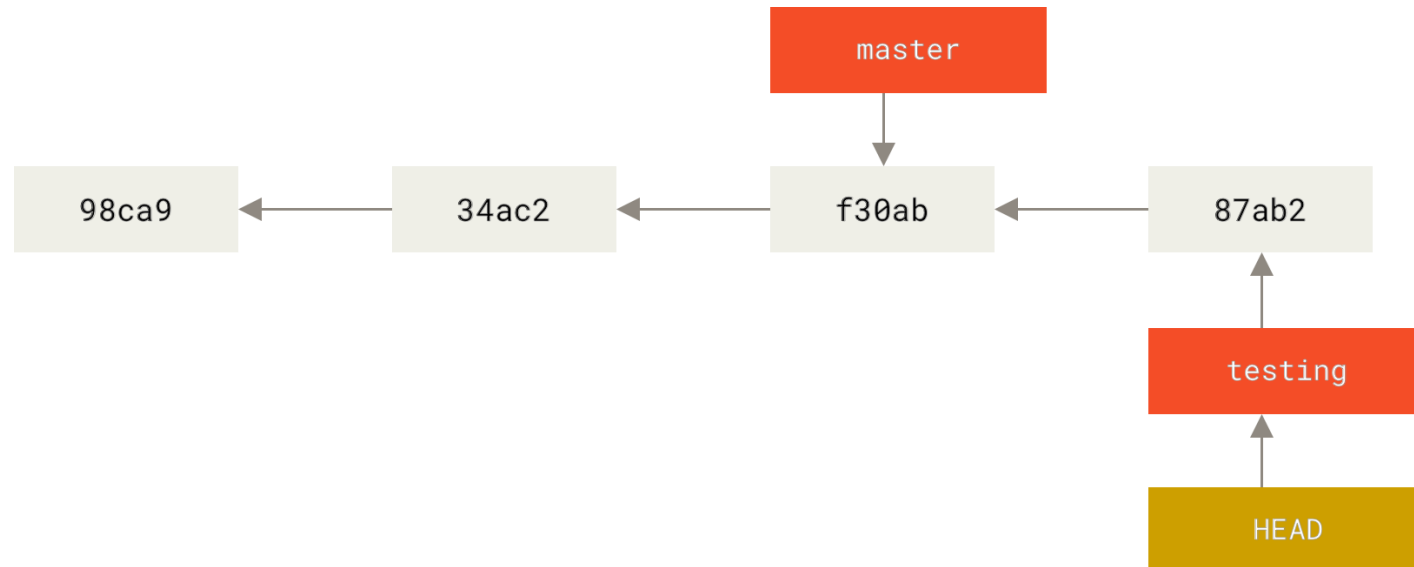


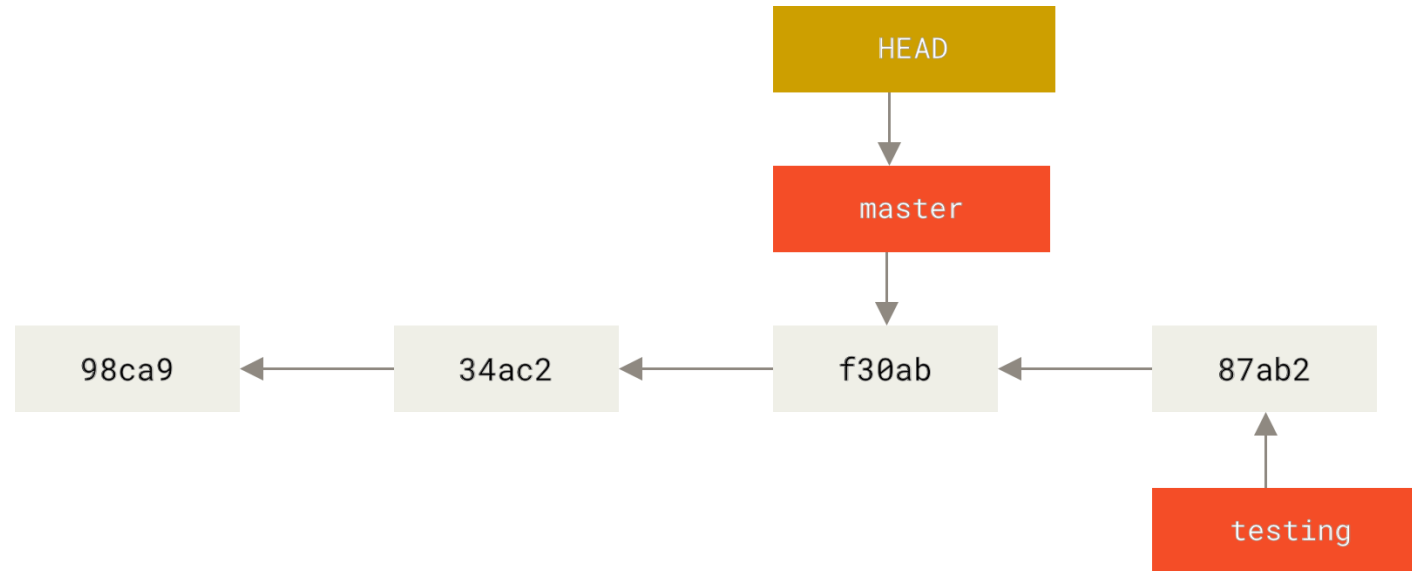


```
$ git checkout testing
or
$ git switch testing
```



```
$ git commit -m 'Make a change'
```

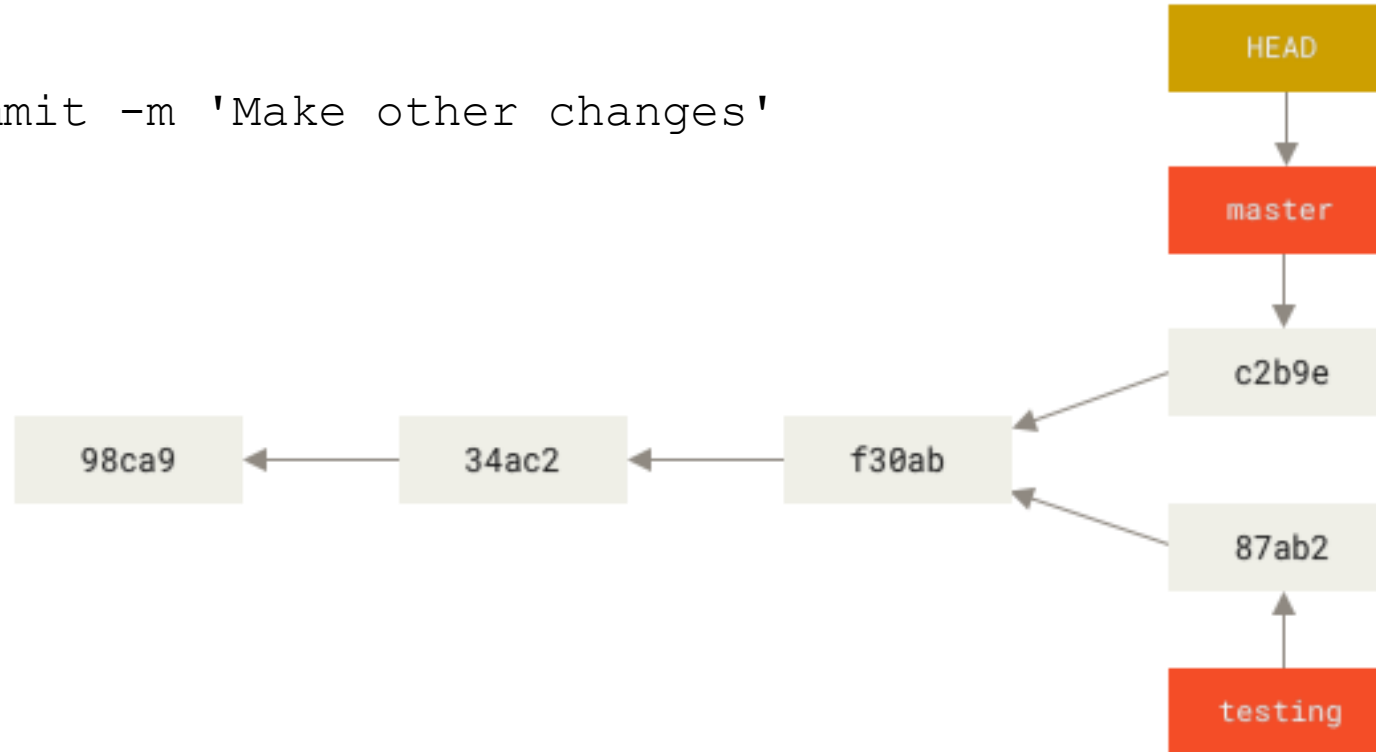


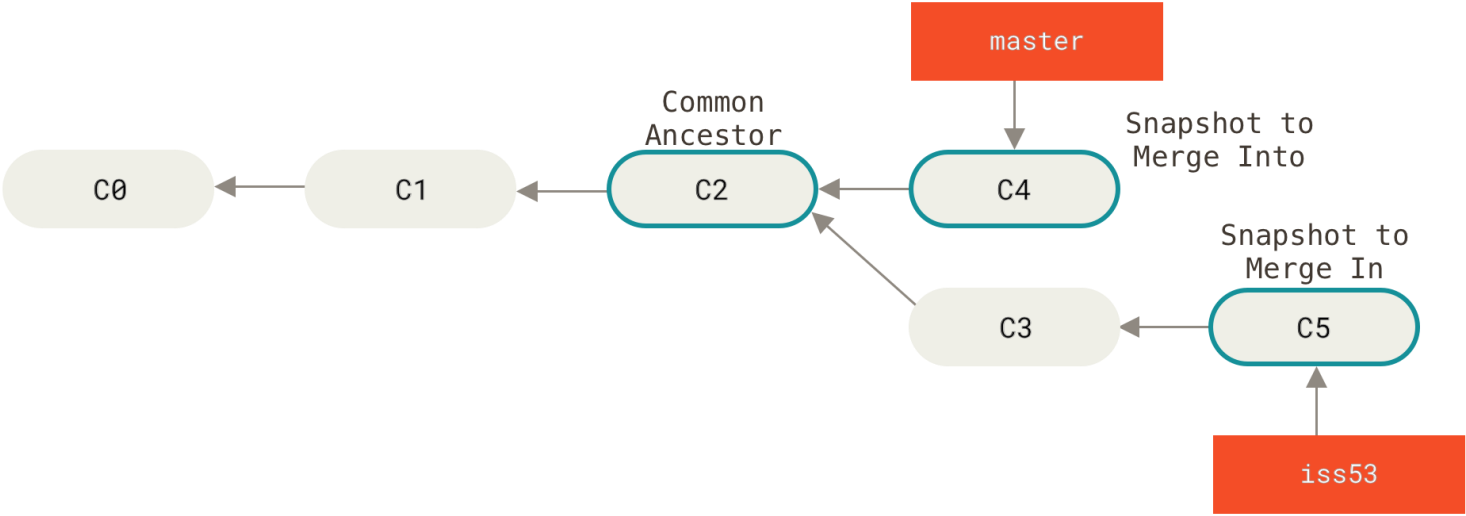


```

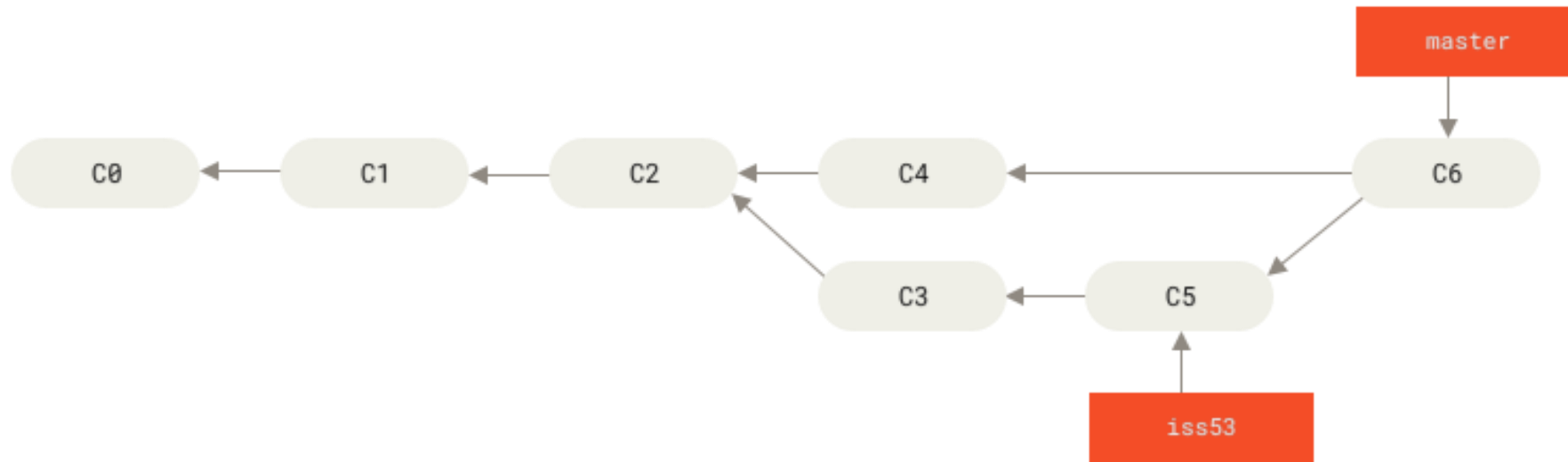
$ git checkout master
or
$ git switch master
    
```

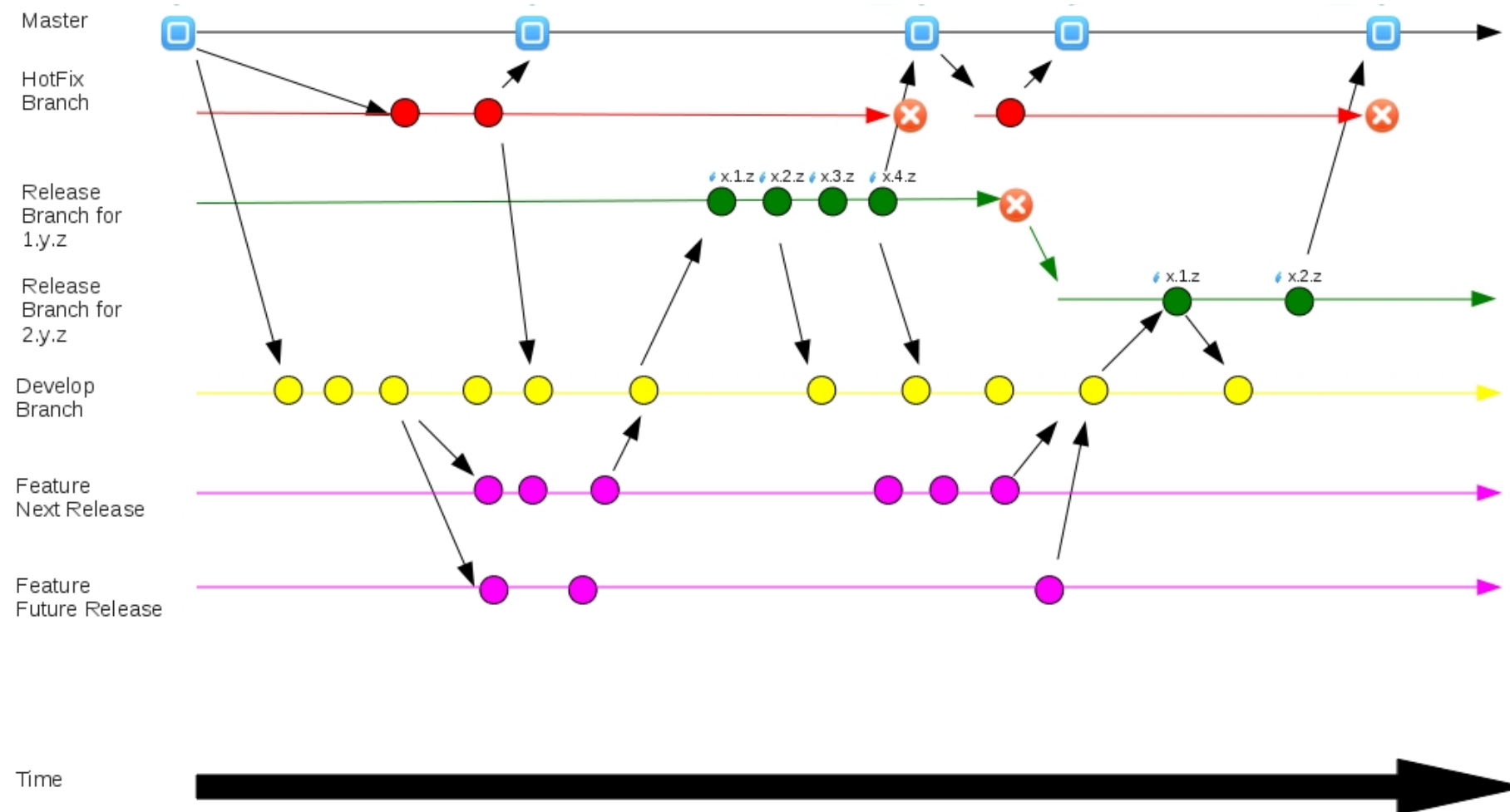
```
$ git commit -m 'Make other changes'
```





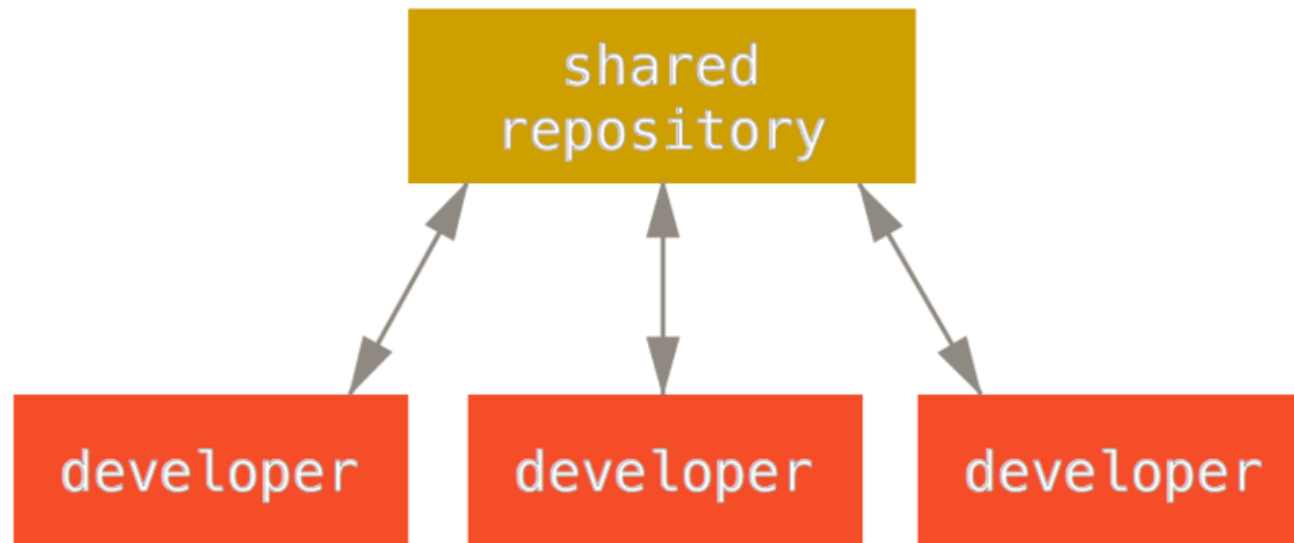
```
$ git merge iss53
```





"GIT Branch" by [miguelpdl](#) is licensed under [CC BY-NC-SA 2.0](#)

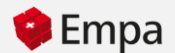
# GIT on the server



## Git based platforms

*On the internet*

- Codeberg <https://codeberg.org/>
- GitLab <https://about.gitlab.com>
- Gitea <https://about.gitea.com/>
- GitHub <https://github.com/>
- Others...



<https://www.empa.ch/group/s909/gitlab>

<https://gitlab.eawag.ch/>

<https://gitlabext.wsl.ch>

[compute@empa.ch](mailto:compute@empa.ch)

[stuart.dennis@eawag.ch](mailto:stuart.dennis@eawag.ch)

## SSH keys

The **Secure Shell Protocol (SSH Protocol)** is a [cryptographic network protocol](#) for operating [network services](#) securely over an [unsecured network](#). Its most notable applications are [remote login](#) and [command-line](#) execution.

SSH uses [public-key cryptography](#) to [authenticate](#) the remote computer and allow it to authenticate the user, if necessary.

*<https://docs.github.com/en/authentication/connecting-to-github-with-ssh>*

```
$ git branch -m master main

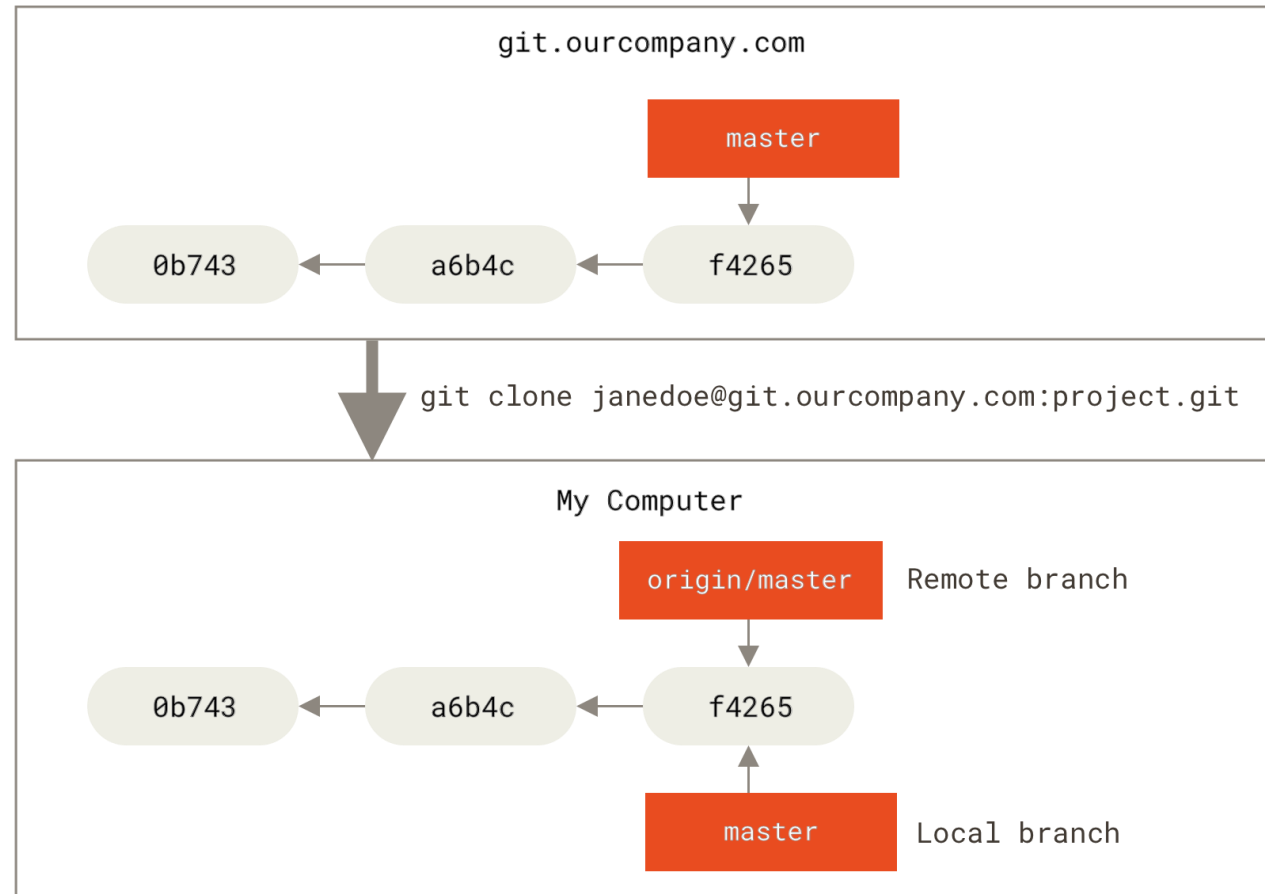
$ git remote add origin <remote URL>
$ git remote set-url origin <remote URL>

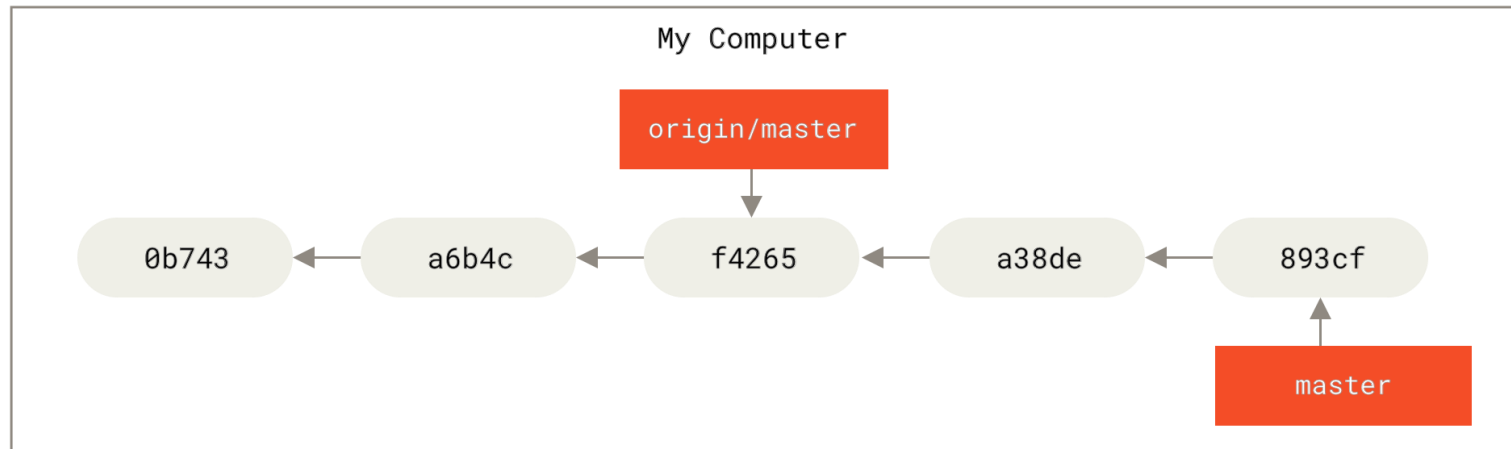
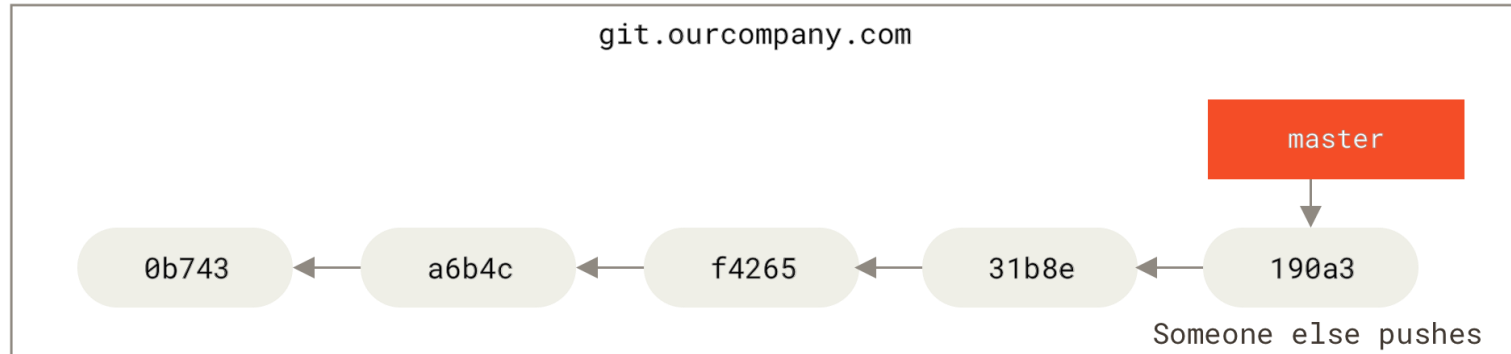
$ git remote show origin

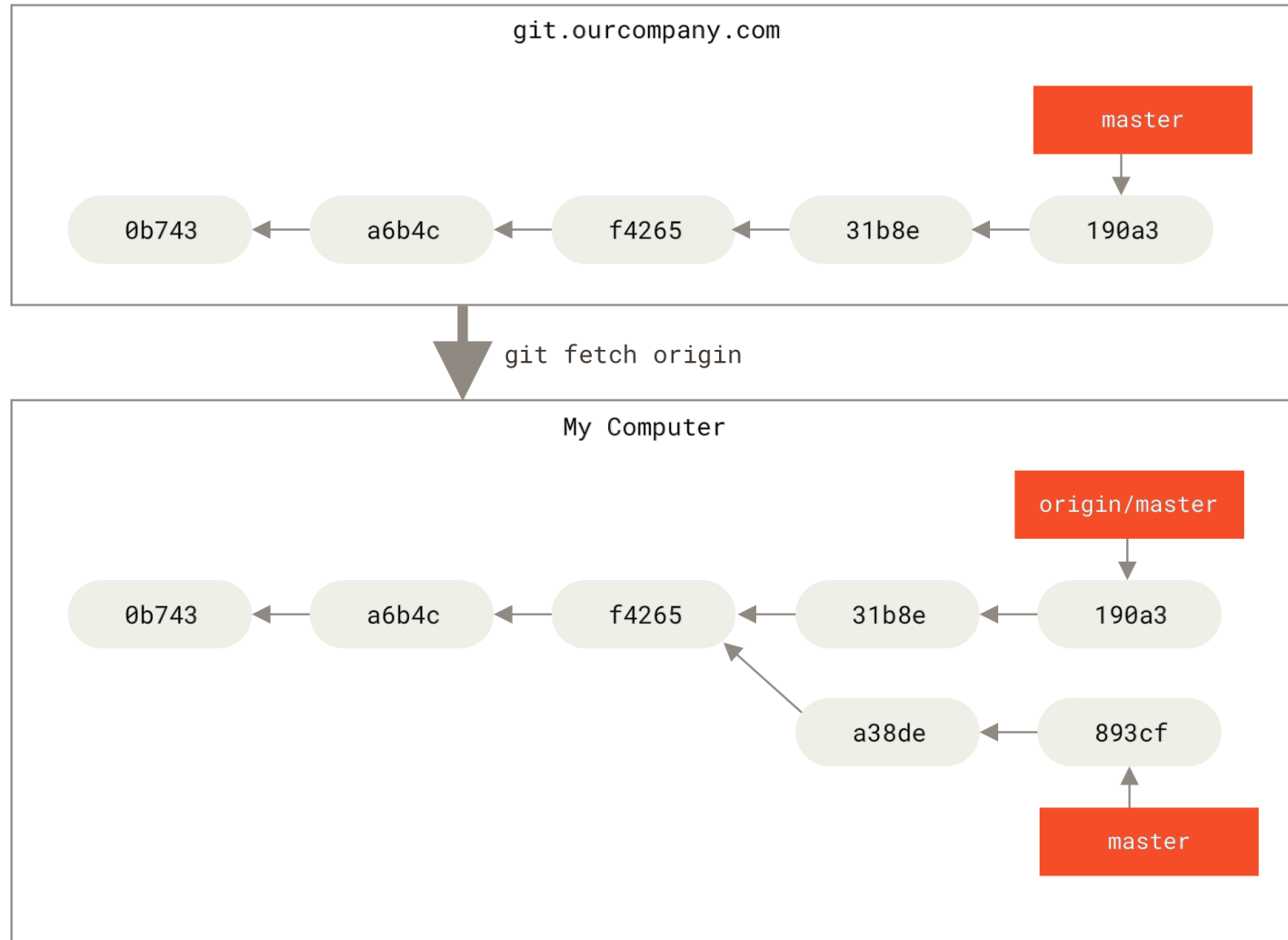
$ git push -u origin <branch name> (first time a new branch is pushed)

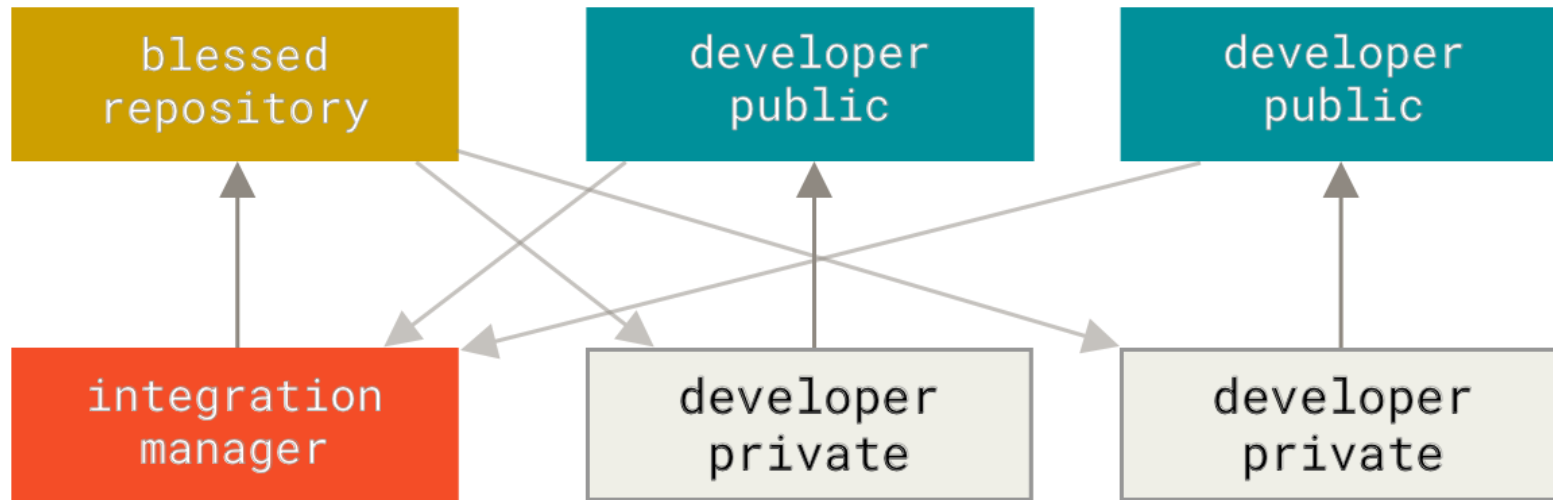
$ git push origin <branch name> (when the branch already exists in the remote)

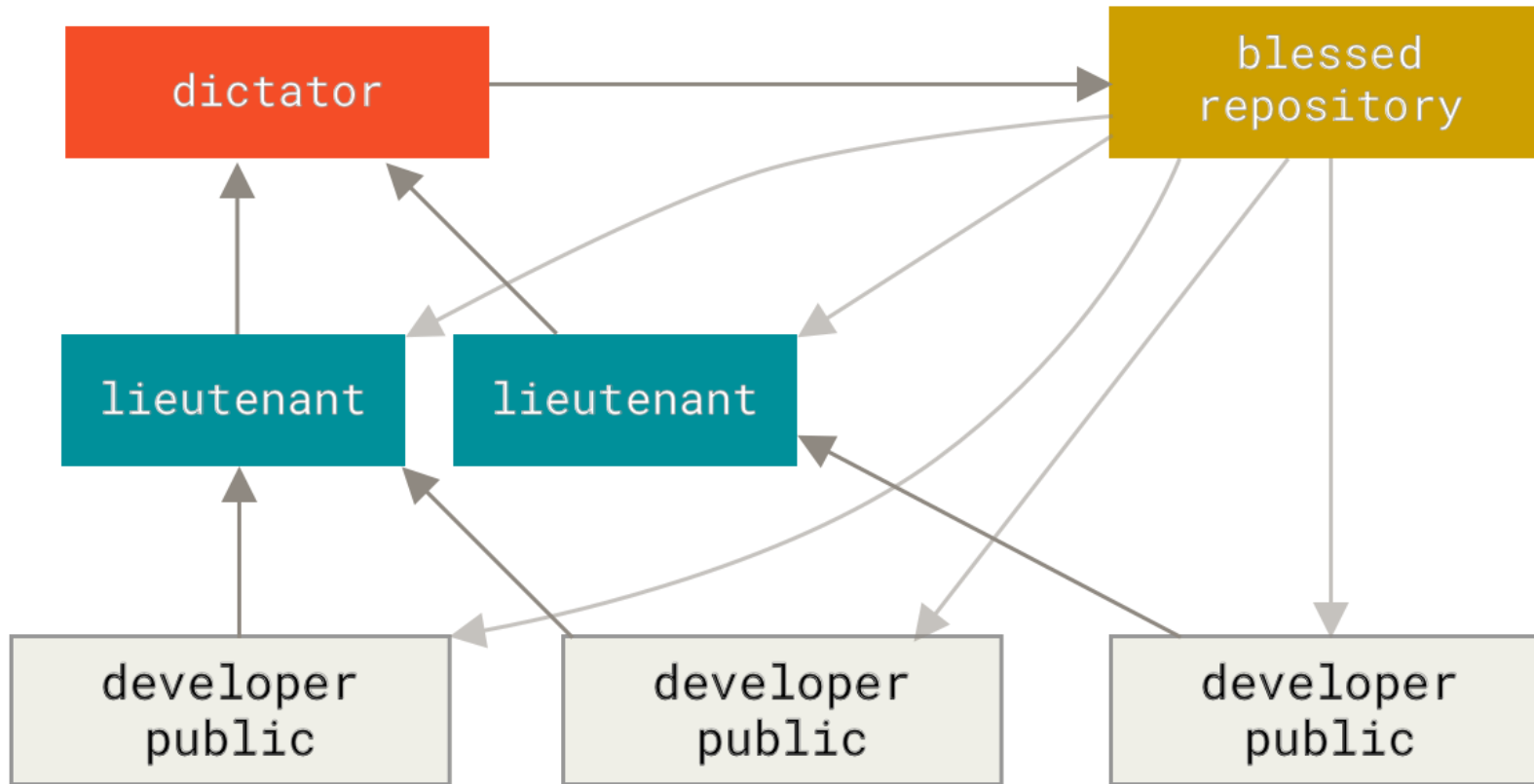
$ git push origin <tag name>
```











# **GIT resources**

- Git website download page (Git, GUI clients) <https://git-scm.com/downloads>
- Git website documentation page  
(Reference manual, “Pro Git” book, videos) <https://git-scm.com/doc>
- GIT cheat sheet <https://git-scm.com/cheat-sheet>
- GitHub Git cheat sheets <https://training.github.com/>
- GitHub authentication documentation  
(2F auth, SSH, etc...) <https://docs.github.com/en/authentication>

Unless otherwise noted, images are from [Pro Git Book](https://git-scm.com/book) (<https://git-scm.com/book>) and are licensed under [CC BY-NC-SA 3.0](https://creativecommons.org/licenses/by-nc-sa/3.0/).

This presentation is licensed under [CC BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/).

# Lib4RI – Excellent Services for Excellent Research.

<https://www.lib4ri.ch>  
[info@lib4ri.ch](mailto:info@lib4ri.ch)  
T: + 41 58 765 57 00