Deep dive into { git }

Michael Kaufmann

Michael Kaufmann

Founder & Managing Director, Xebia Germany

@mike_kaufmann

@wulfland

https://writeabout.net

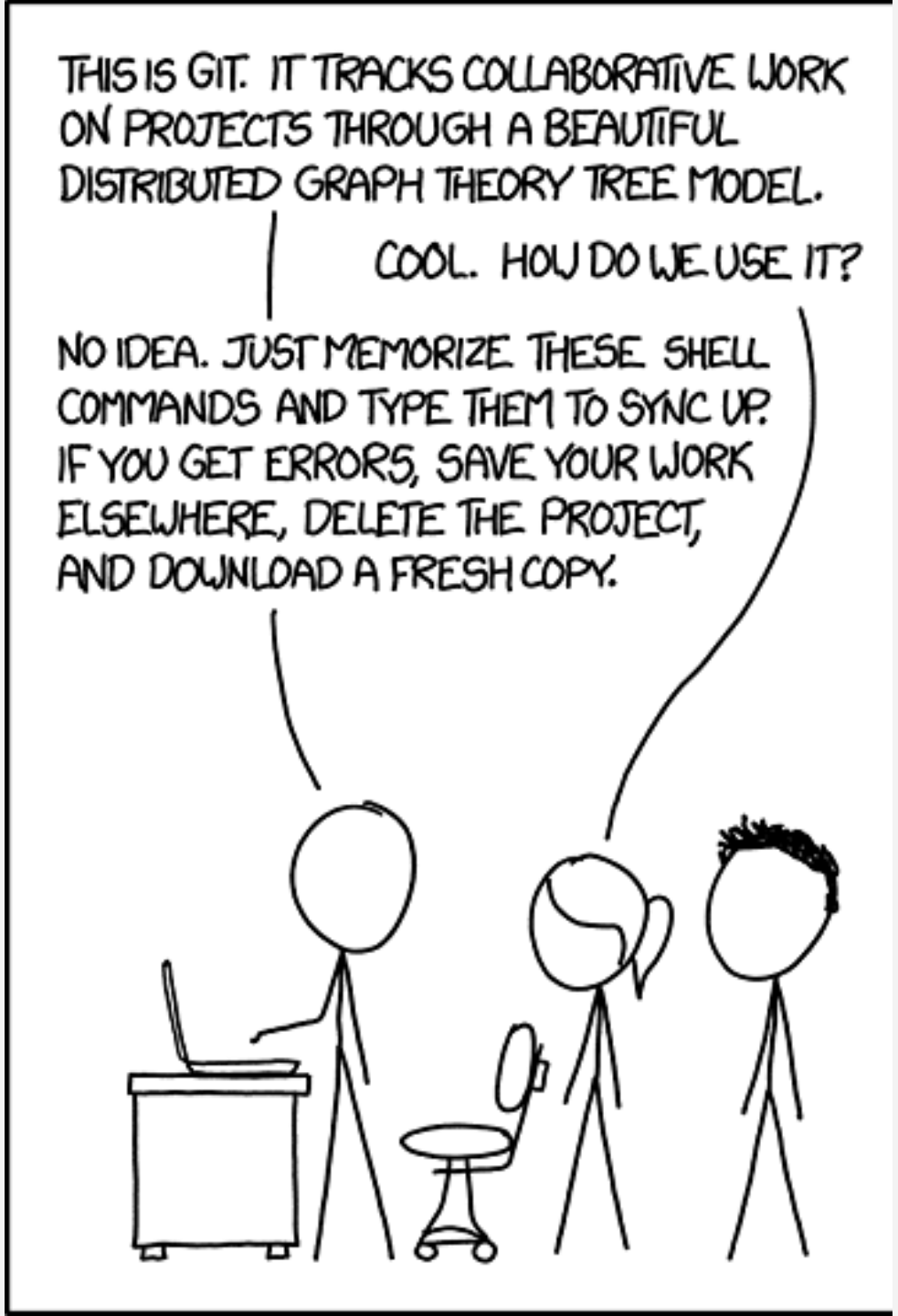
Microsoft REGIONAL DIRECTOR

MVP Microsoft® Most Valuable Professional

- 25 years software developer
- 15 years ALM & DevOps
- 10 years Git and GitHub
- Microsoft RD & MVP

Agenda

- Getting started
- The 3 principles
  - Hash / SHA
  - Snapshots
  - Trees (DAG)
- Branches and tags
- Referencing commits
- Diffs and patches
- Merging
- Resolving merge conflicts
- Remotes
- Branching workflows
- Git aliases
- Reverting changes
- Ammending changes
- Resetting your repository
- Rebase
- Cherry-picking and the reflog
- The stash


THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

Getting started

Getting started

- Local git configs

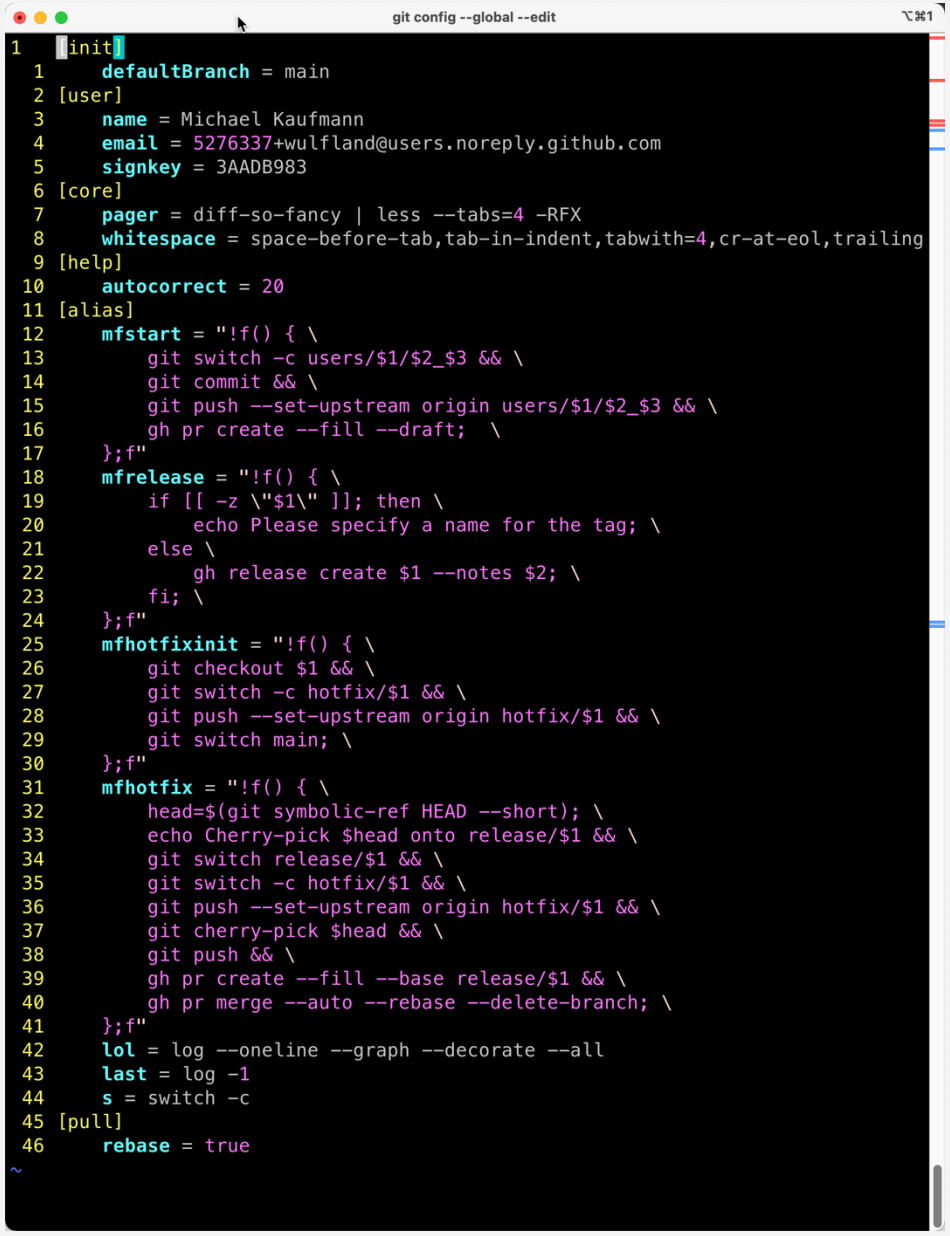

--system


--global


-- local

```
$ git config --global user.email foo@bar
$ git config --global user.name ”Foo Bar”
$ git config --global init.defaultBranch main
$ git config --global help.autocorrect 20
$ git config --global core.editor "code --wait"
$ git config --global alias.last ”log -1"
$ git config --global --edit
```

- Functions ”!f(){ … };f "
- $1, $2, $3, … $@
- Line endings: \
- && or ; to separate commands



https://github.com/wulfland/DeepDive

Check yor git installation

Make sure git is installed and the version is > 2.23:

```
$ git --version
> git version 2.35.1
```

If not, download and install git.

Check your git config

1. Check your name and email address:

```
$ git config --global user.name
$ git config --global user.email
```

If this does not return the desired values, set the config using these commands:

```
$ git config --global user.name '<your_name>'
$ git config --global user.email '<your_email_address>'
```

2. Set the default branch to `main`:

```
$ git config --global init.defaultBranch main
```

3. Check your default editor

Check your default editor (i.e. by running `git config --global --edit`) If you like the editor, then you are good. If your stuck in vim (click `ESC` `:` `q` `!` `Enter` to exit), then configure the editor of your choice - for example CSCode:

```
$ git config core.editor 'code --wait'
```

The 3 principles

The 3 principles

- Everything in git is a file – and changes are tracked using SHA hash values (SHA-1 or SHA-256)
- Git saves snapshots – and not deltas
- Everything is a rooted tree
  - Commits
  - Files and folders

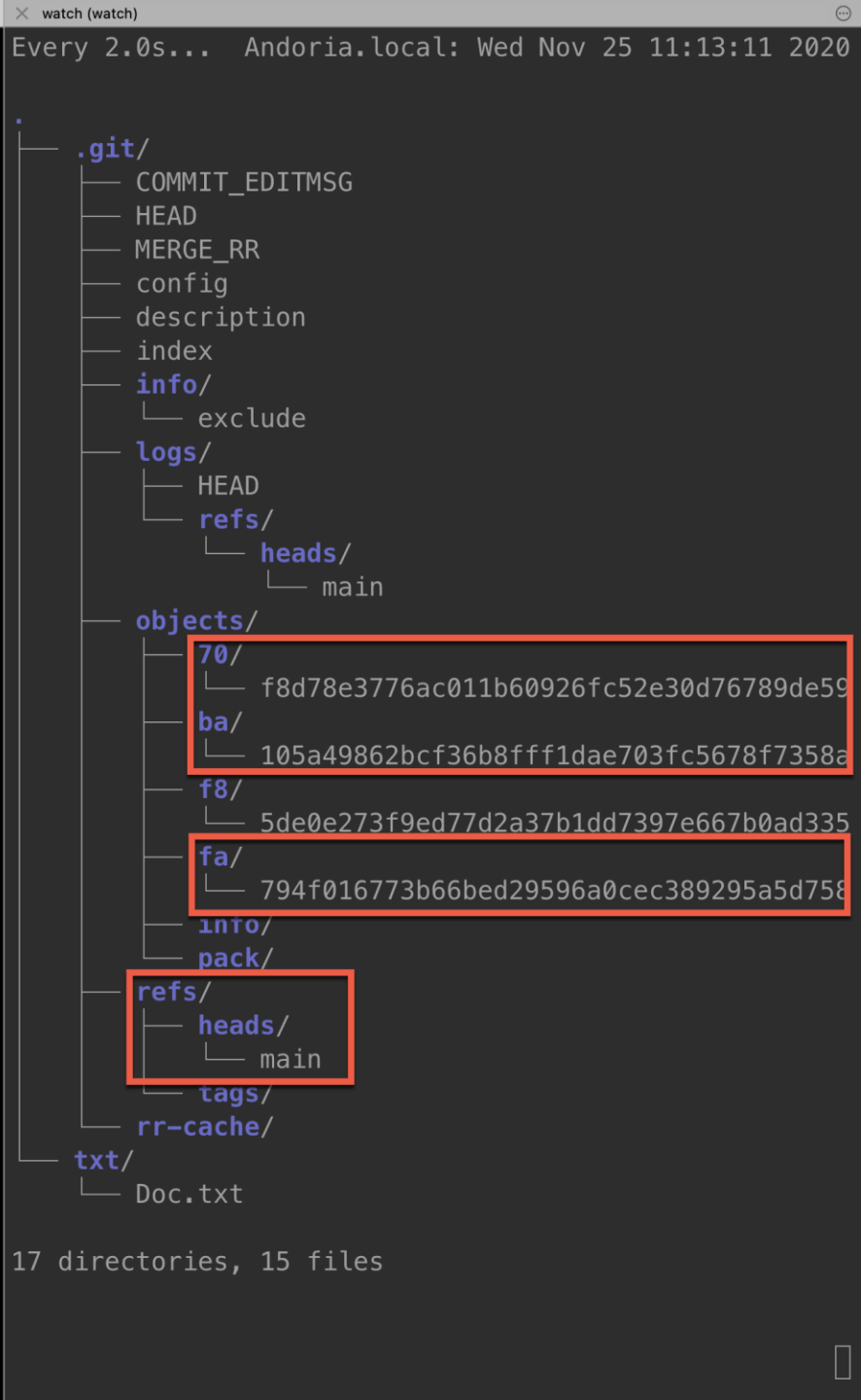
..UnterDerHaube (-zsh)

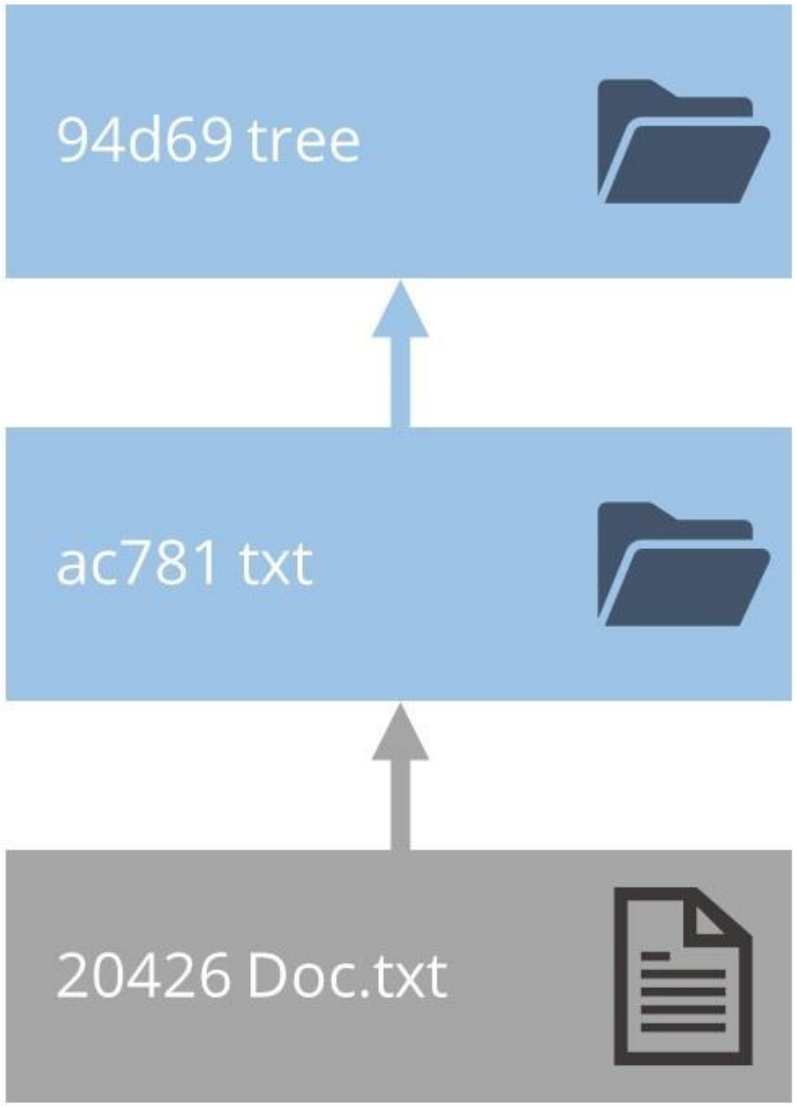
```
~/source/repos/UnterDerHaube                                                    10:56:10
❯ git init
Initialized empty Git repository in /Users/m.kaufmann/source/repos/UnterDerHaube/.git/
~/source/repos/UnterDerHaube  git main ?1                                       10:56:36
❯ git hash-object txt/Doc.txt
f20acb769ad757b2058c8c3c96e7eac2e1258503
~/source/repos/UnterDerHaube  git main ?1                                       10:57:58
❯ vim txt/Doc.txt
~/source/repos/UnterDerHaube  git main ?1                          14s          10:58:21
❯ git hash-object txt/Doc.txt
f85de0e273f9ed77d2a37b1dd7397e667b0ad335
~/source/repos/UnterDerHaube  git main ?1                                       10:58:24
❯ git add txt/Doc.txt
~/source/repos/UnterDerHaube  git main +1                                       10:59:03
❯ cat .git/HEAD
ref: refs/heads/main
~/source/repos/UnterDerHaube  git main +1                                       11:00:27
❯ cat .git/Index
DIRC_@*@@
@_@*@g@@@@@@@>@]@@s@@wH{@9~f{
@5
  txt/Doc.txt@({@7@@@3@>@pC@k@J@%
~/source/repos/UnterDerHaube  git main +1                                       11:00:43
❯ git ls-files --stage
100644 f85de0e273f9ed77d2a37b1dd7397e667b0ad335 0       txt/Doc.txt
~/source/repos/UnterDerHaube  git main +1                                       11:01:05
❯ cat .git/objects/f8/5de0e273f9ed77d2a37b1dd7397e667b0ad335
xK@@OR03b@K@M@Rp)@Sp@@M-QPr@OVRp*@/@@@/@L@@K@Q(@)2@@0@@@L@â@@O@@@%
~/source/repos/UnterDerHaube  git main +1                                       11:03:05
❯ git cat-file -t f85de
blob
~/source/repos/UnterDerHaube  git main +1                                       11:05:52
❯ git cat-file -p f85de
Name: Dr. Emmet "Doc" Brown
Original time: 1985
Nichname: Doc
~/source/repos/UnterDerHaube  git main +1                                       11:05:57
❯
```

watch (watch)

```
Andoria.local: Wed Nov 25 11:06:00 2020

.
├── .git/
│   ├── HEAD
│   ├── config
│   ├── description
│   ├── index
│   ├── info/
│   │   └── exclude
│   ├── objects/
│   │   ├── f8/
│   │   │   └── 5de0e273f9ed77d2a37b1dd7397e667b
0ad335
│   │   ├── info/
│   │   └── pack/
│   └── refs/
│       ├── heads/
│       └── tags/
└── txt/
    └── Doc.txt

10 directories, 7 files
```

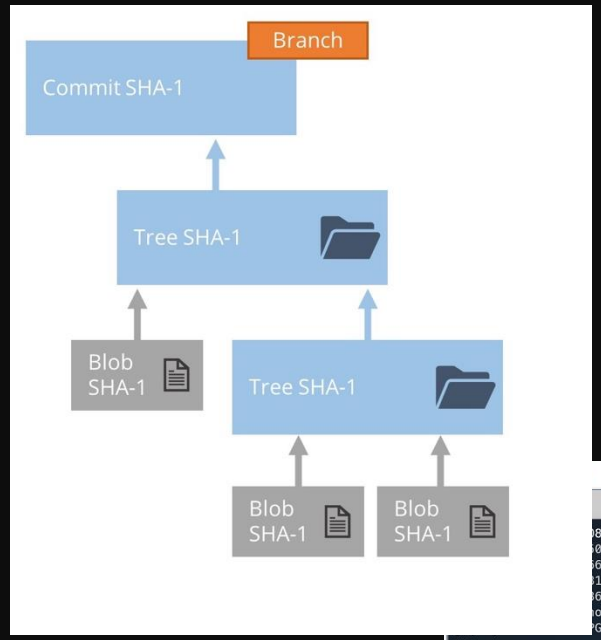

..UnterDerHaube (-zsh)
~/source/repos/UnterDerHaube git main 11:09:37
git cat-file -t fa79
commit
~/source/repos/UnterDerHaube git main 11:09:42
git cat-file -p fa79
tree 70f8d78e3776ac011b60926fc52e30d76789de59
author Michael Kaufmann <wulfland@hotmail.com> 1606298912 +0100
committer Michael Kaufmann <wulfland@hotmail.com> 1606298912 +0100
ErsterCommit
~/source/repos/UnterDerHaube git main 11:09:46
git cat-file -t 70f8
tree
~/source/repos/UnterDerHaube git main 11:10:57
git cat-file -p 70f8
040000 tree ba105a49862bcf36b8fff1dae703fc5678f7358a txt
~/source/repos/UnterDerHaube git main 11:11:01
git ls-tree ba105
100644 blob f85de0e273f9ed77d2a37b1dd7397e667b0ad335 Doc.txt
~/source/repos/UnterDerHaube git main 11:11:19
cat .git/refs/heads/main
fa794f016773b66bed29596a0cec389295a5d758
~/source/repos/UnterDerHaube git main 11:12:48

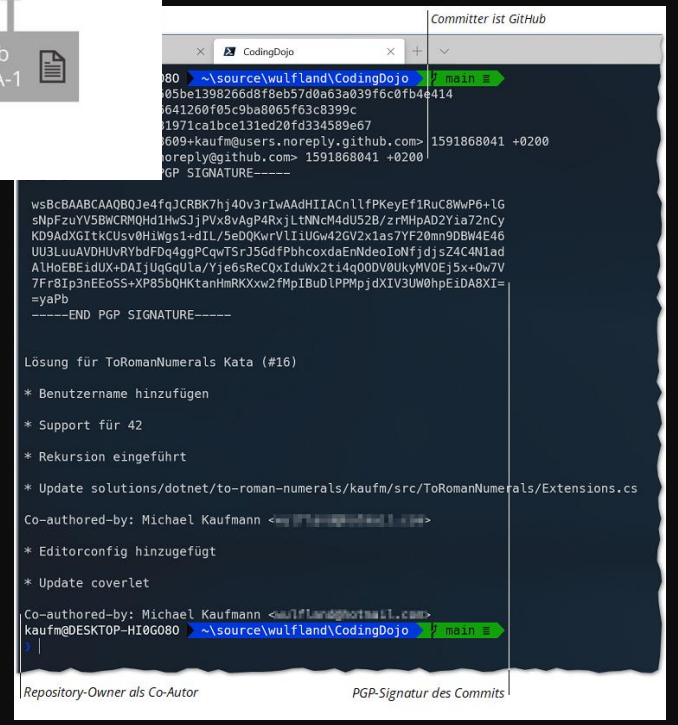

watch (watch)
Every 2.0s... Andoria.local: Wed Nov 25 11:13:11 2020
.
.git/
COMMIT_EDITMSG
HEAD
MERGE_RR
config
description
index
info/
exclude
logs/
HEAD
refs/
heads/
main
objects/
70/
f8d78e3776ac011b60926fc52e30d76789de59
ba/
105a49862bcf36b8fff1dae703fc5678f7358a
f8/
5de0e273f9ed77d2a37b1dd7397e667b0ad335
fa/
794f016773b66bed29596a0cec389295a5d758
info/
pack/
refs/
heads/
main
tags/
rr-cache/
txt/
Doc.txt
17 directories, 15 files



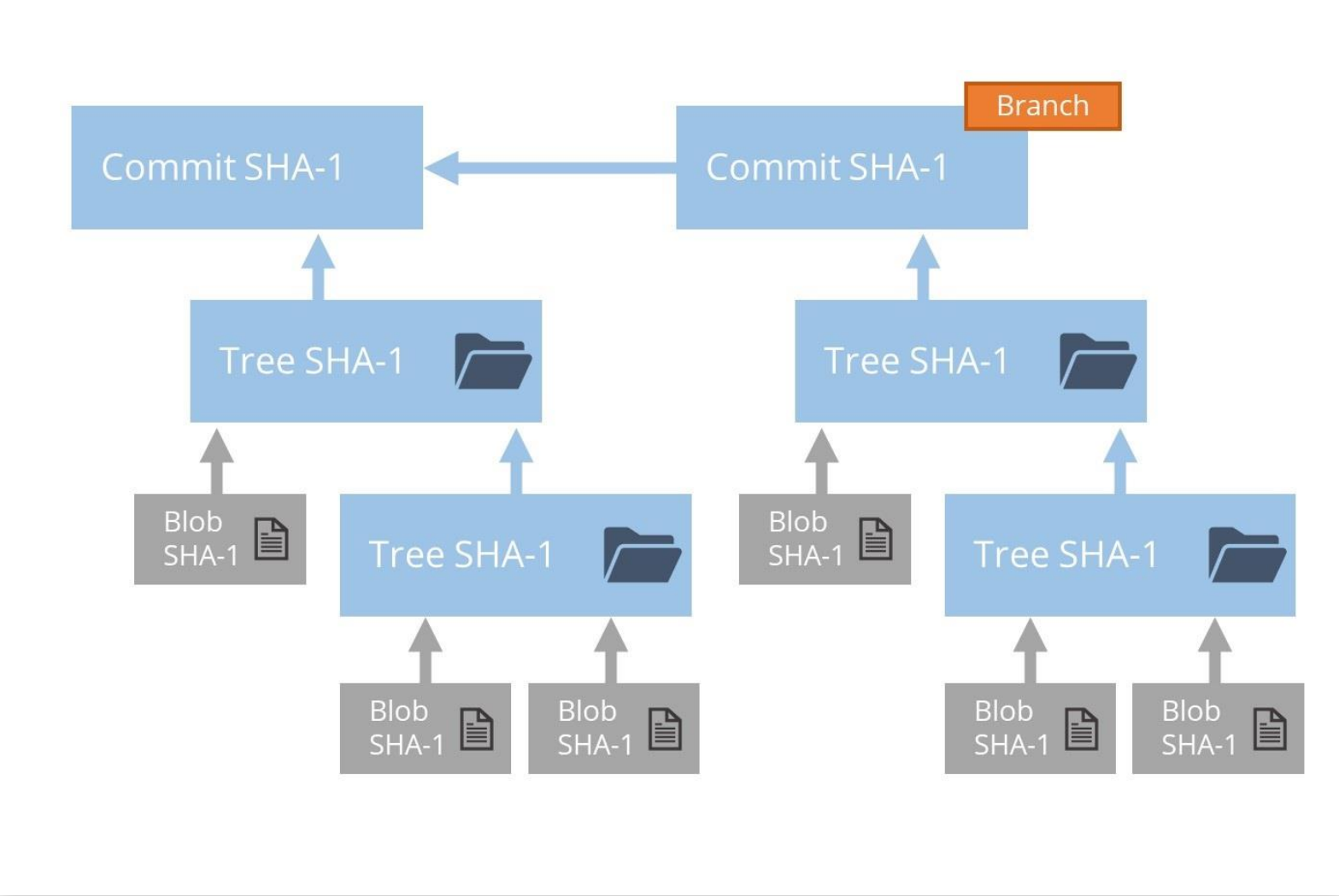
The structure of a tree

The structure of a commit





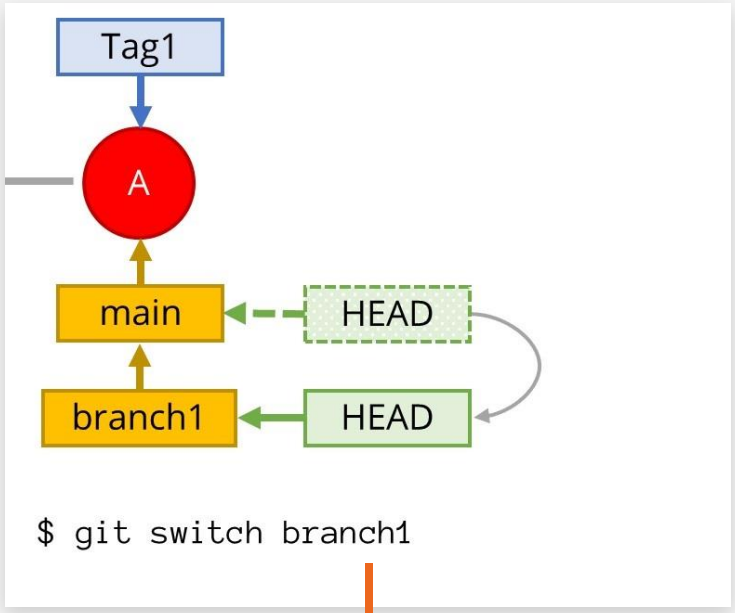
Everything is a rooted tree

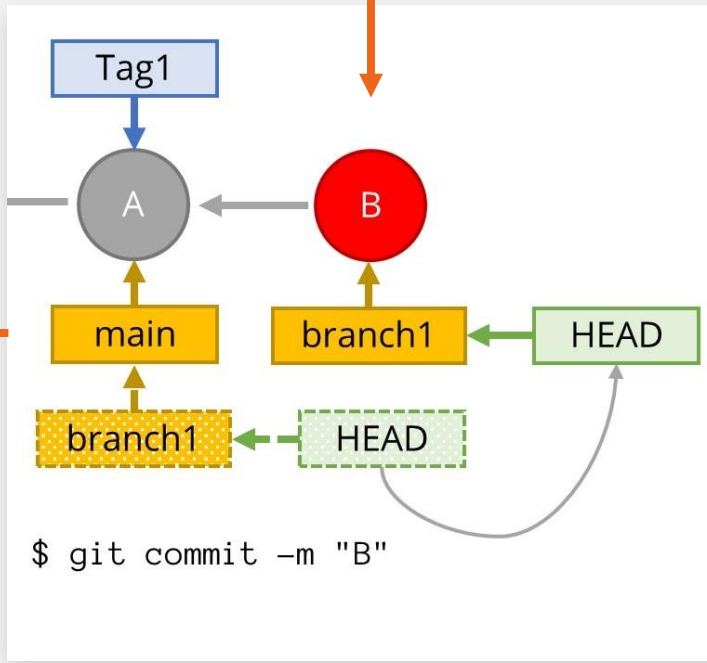


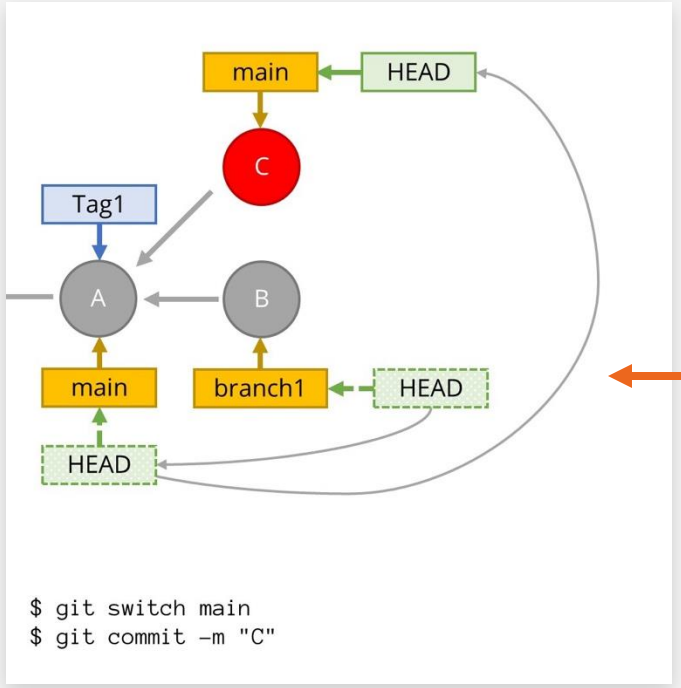
Branches and tags

Branches

- Pointers to a commit
- The branch that HEAD points to
  - will be the parent commit for a new commit
  - Will pointto the new commit (git update-ref)

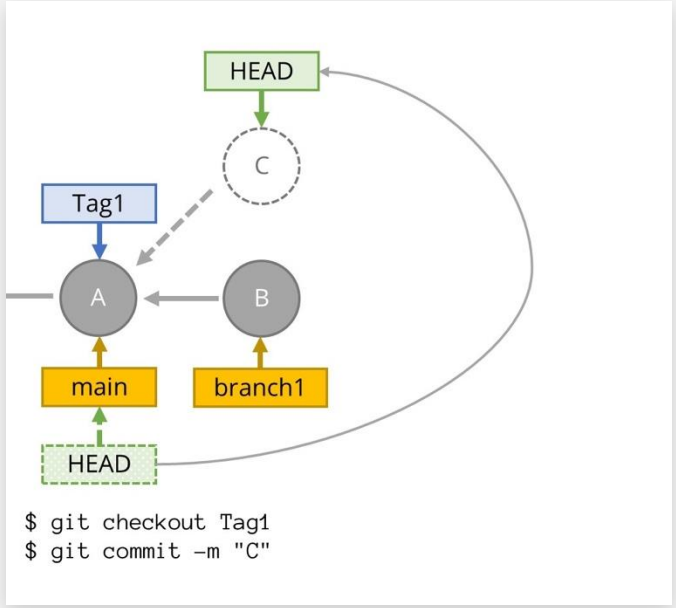






Tags

- Fix pointers to a commit
- Annotated tags – object of type tag that points to a commit (git tag –a)
- Detached HEAD mode



Referencing commits

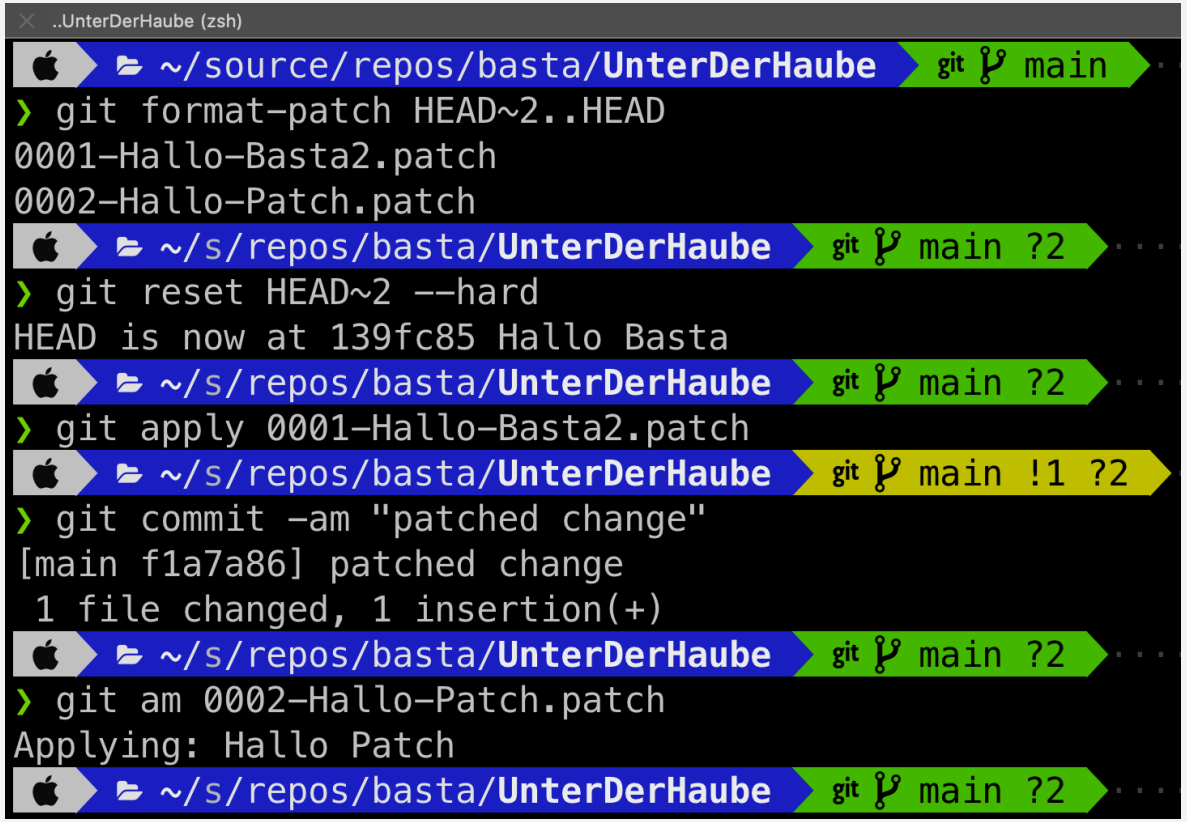
Referencing commits

- **Absolute references**
  - The SHA-1 value **c36554656ee78362ced084e5114b0a51001e2f57**
  - The first part (at least 4 – must be unique) **c365**
  - **HEAD**
  - **Tag**
  - **Branch**
- **Relative in time**
  - **HEAD@{3}**
  - **main@{two.weeks.ago}**
  - **HEAD@{last.month}**
  - **main@{yesterday}**
- **Relative to another reference**
  - **^** = points to the parent commit
  - **^^** = points to the parent of the parent
  - **^2** = points to the second parent of the same commit (merge commits)
  - **~** = points to the first parent of the commit
  - **~2** = ~~ = points to the parent of the parent
- **Ranges**
  - Double dot: **main..experiment** (reachable from experiment but not main)
  - **experiment..main**
  - Tripple dot: **main...experiment** (either of the two references but not both **--left-right**)

Diffs and patches

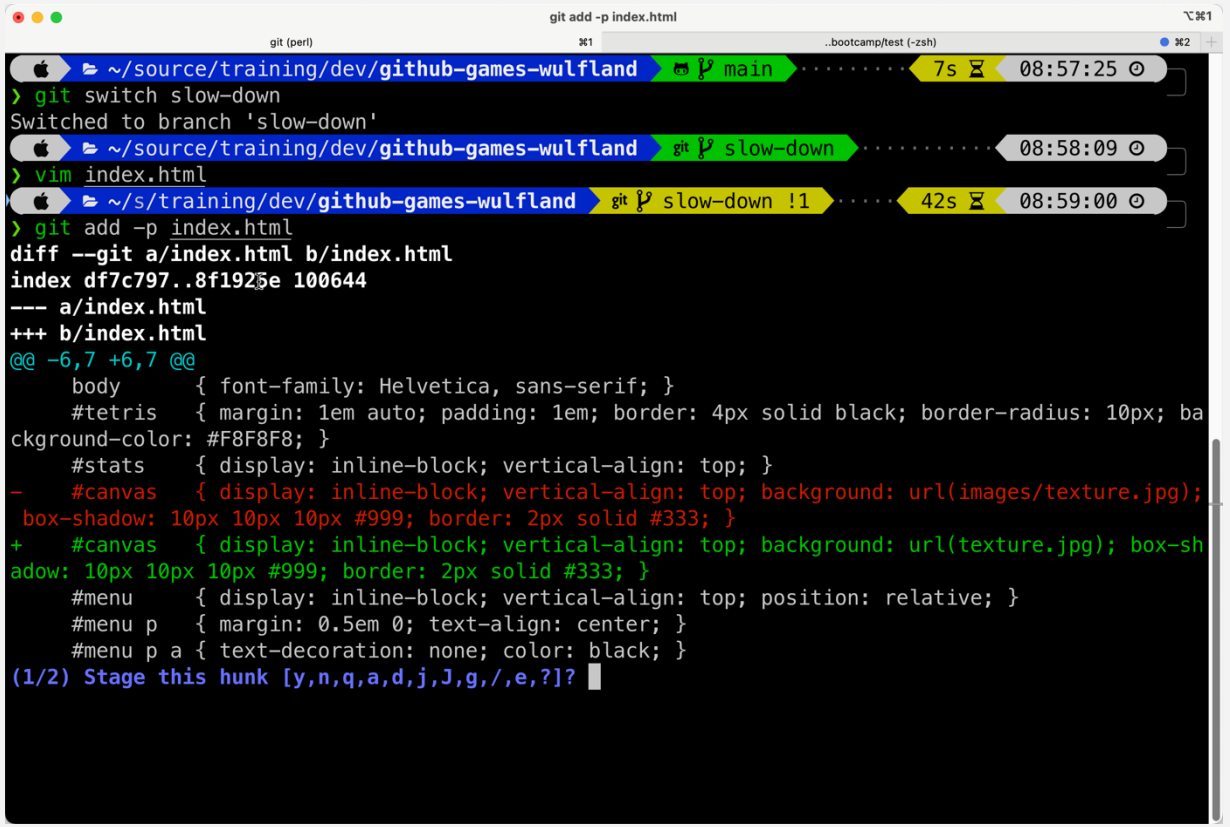
Diffs and patches

```
$ git diff
$ git diff --staged
$ git diff <REF-1> <REF-2>
$ git diff HEAD^!

$ git format-patch HEAD~2..HEAD
$ git reset --hard HEAD~2

$ git apply 0001*.patch

$ git commit -am
$ git am 0002*.patch
```



Diffs and patches

```
$ git add -p | --patch
```

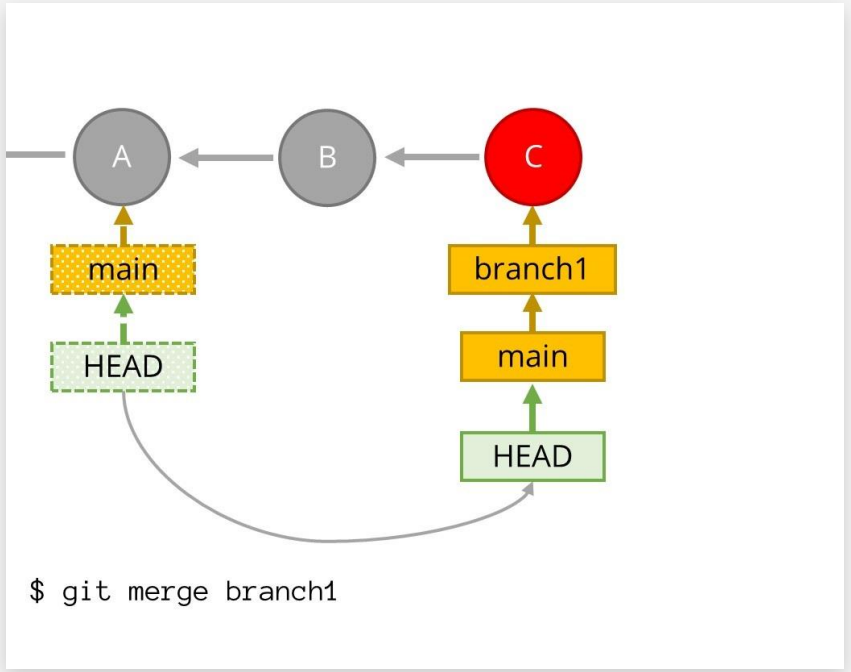


Merging

Fast-forward merge

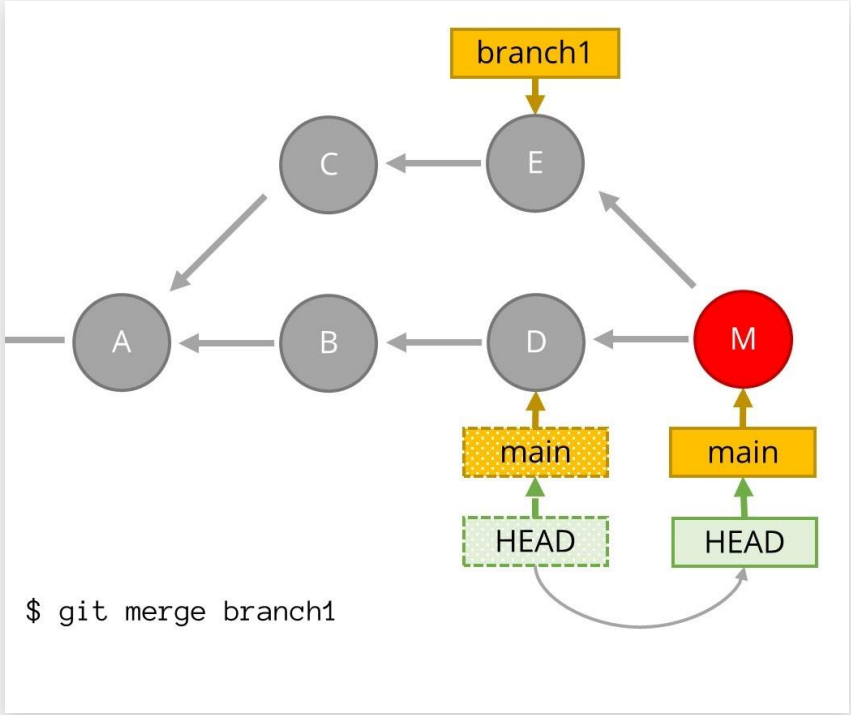
```
$ git switch main
$ git merge [--ff] branch1
> Updating 5efe25f..02fdfff
> Fast-forward
> C.txt | 1 +
> 1 file changed, 1 insertion(+)
> create mode 100644 C.txt
```

```
$ git merge --ff-only
$ git merge --no-ff
```



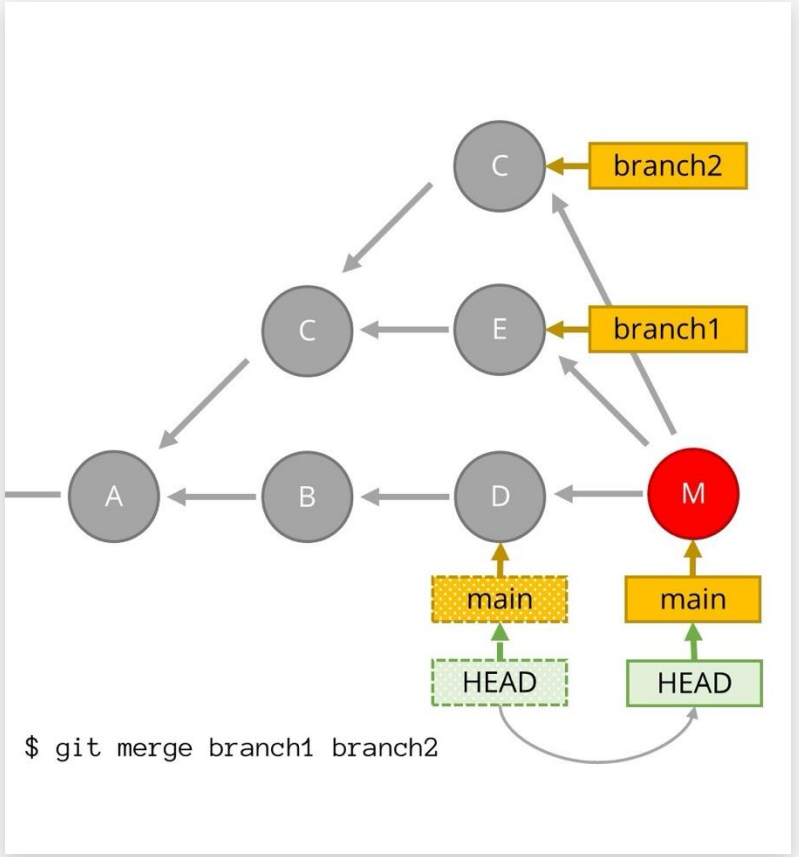
Merge commit

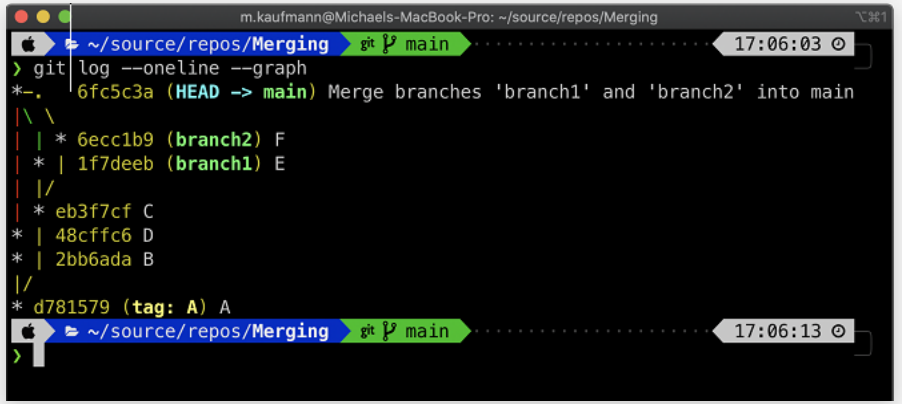
```
$ git merge branch1
> Merge made by the 'recursive' strategy.
> C.txt | 2 ++
> 1 file changed, 2 insertions(+)
> create mode 100644 C.txt
```



Octopus merge

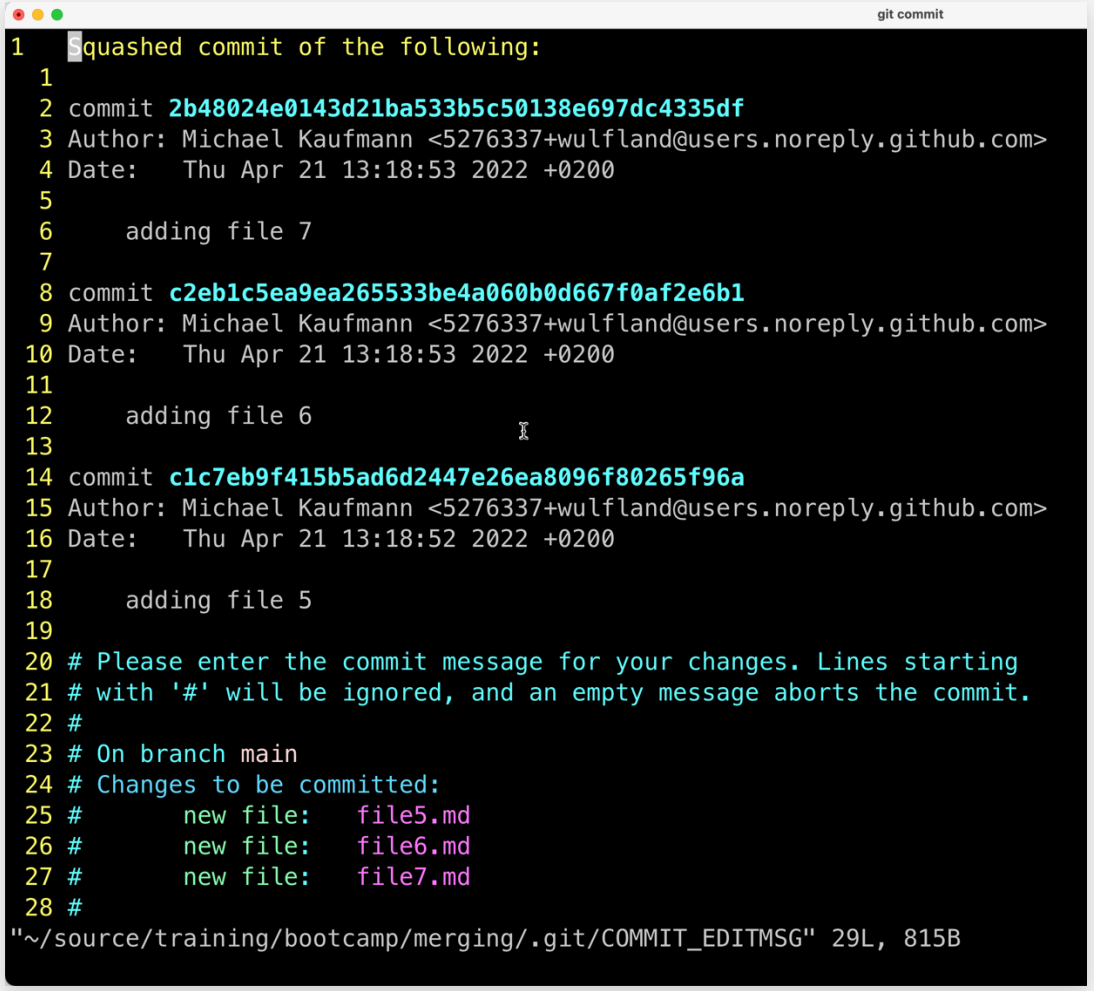
```
$ git merge branch1 branch2
> Trying simple merge with branch1
> Trying simple merge with branch2
> Merge made by the 'octopus' strategy.
>  C.txt | 2 ++
>  F.txt | 1 +
>  2 files changed, 3 insertions(+)
>  create mode 100644 C.txt
>  create mode 100644 F.txt
```





Squash merge

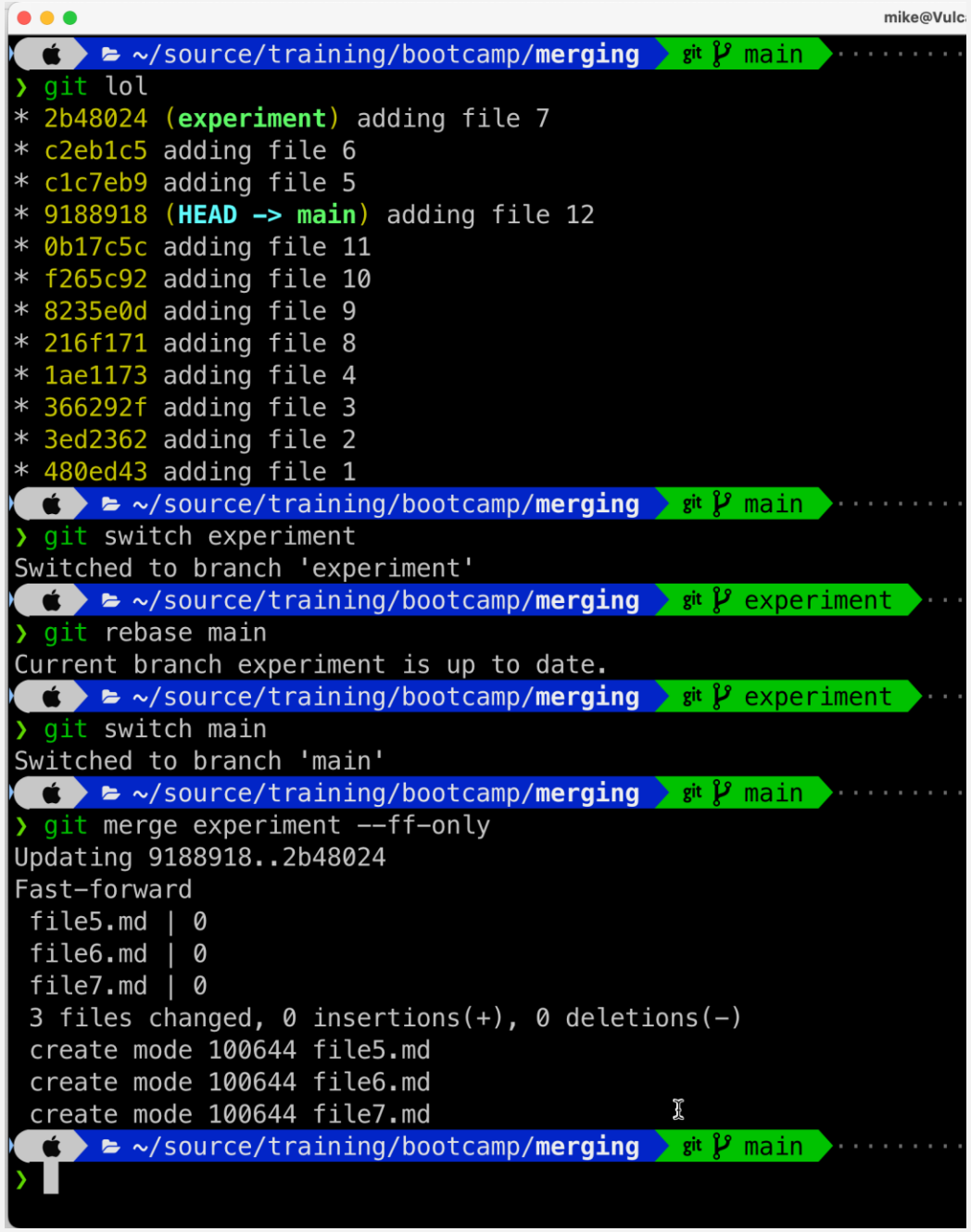
```
$ git merge --squash <branch>
$ git commit
```



Rebase and fast-forward merge

```
$ git switch experiment
$ git rebase main
$ git switch main
$ git merge experiment --ff-only
```

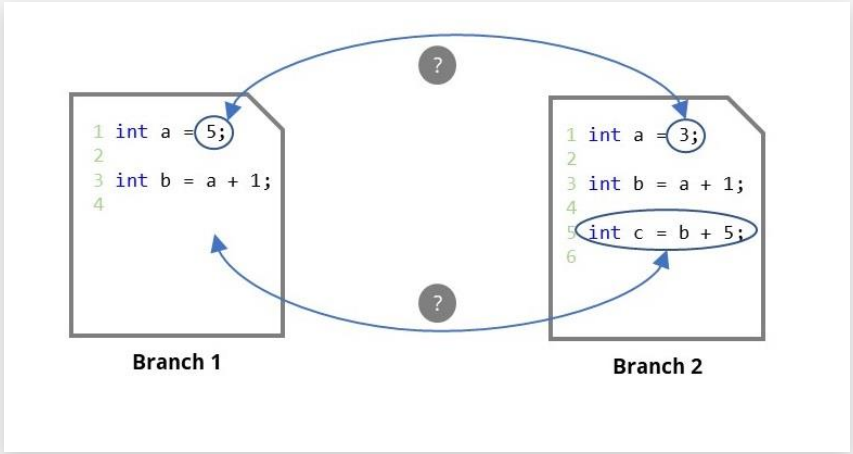
```
$ git pull --rebase
```

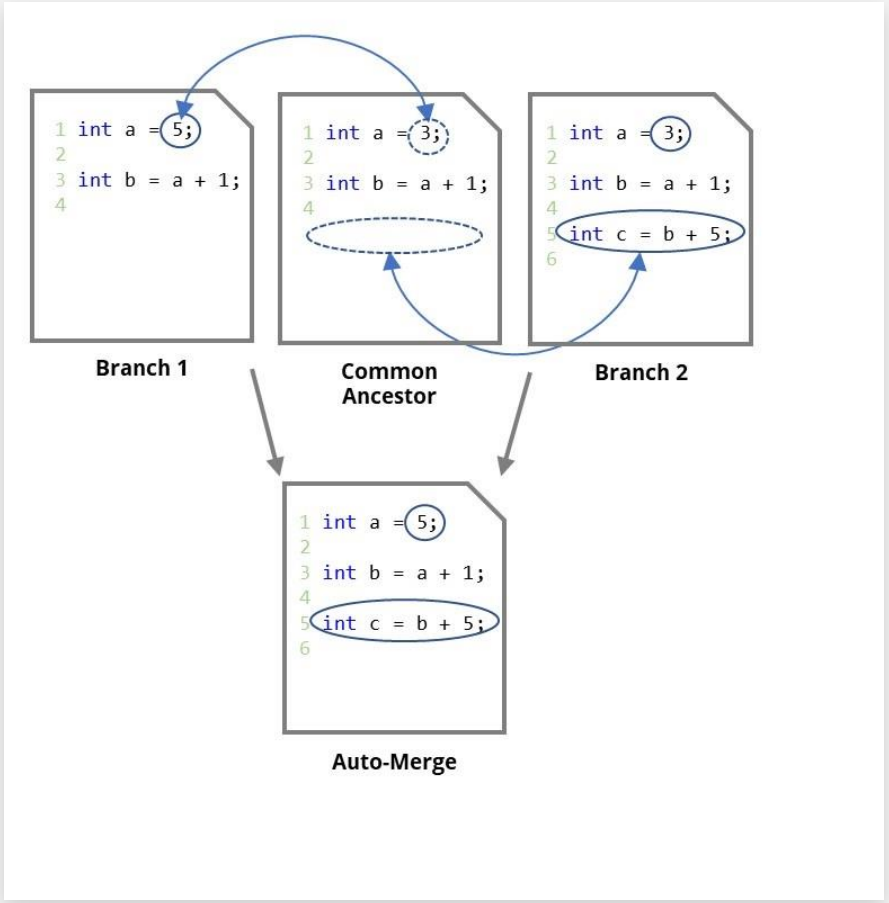


Resolving merge conflicts

Three-way merge

- Three-way merge = using the common anchestor



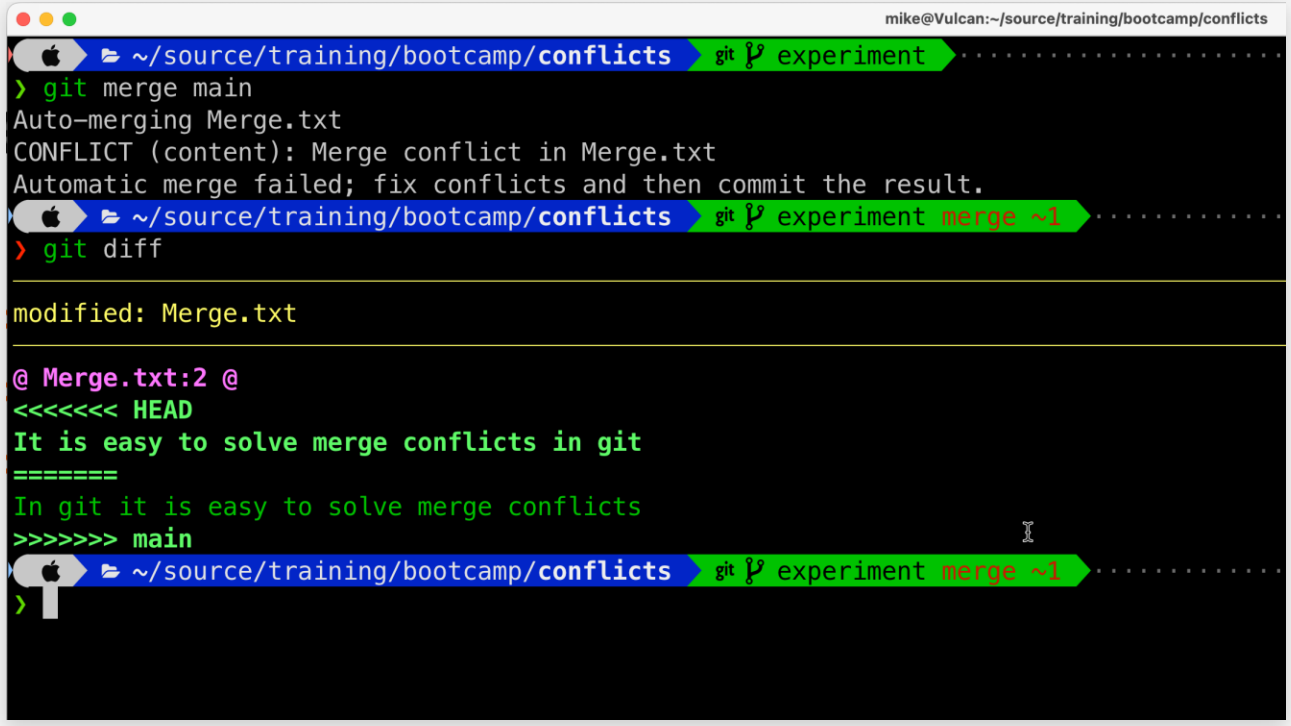


Resolving merge conflicts

```
> Auto-merging Merge.txt
> CONFLICT (content): Merge conflict in Merge.txt
> Automatic merge failed; fix conflicts and then commit the result.
```

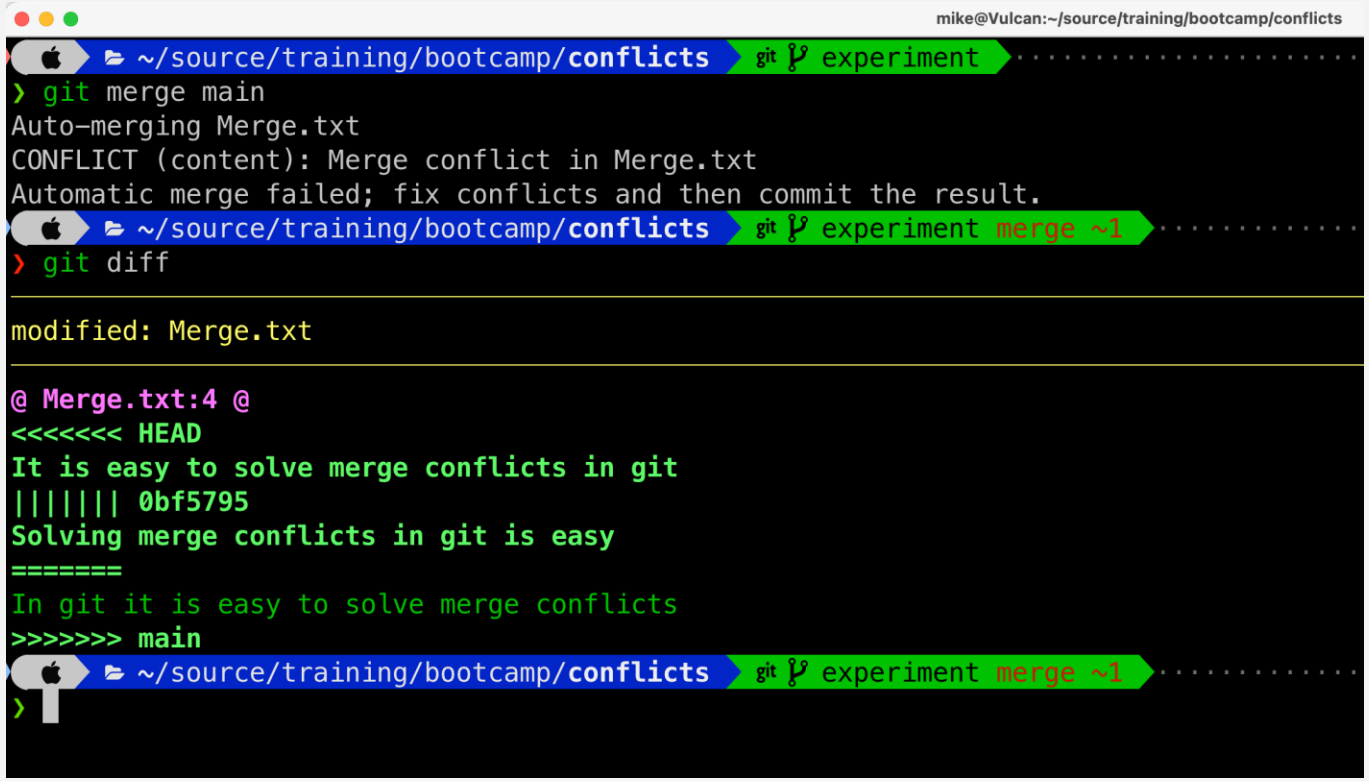
› <<<<<<<, =======, and >>>>>>>

```
<<<<<<<
HEAD (ours)
=======
MERGE_HEAD (incoming, theirs)
>>>>>>>
```



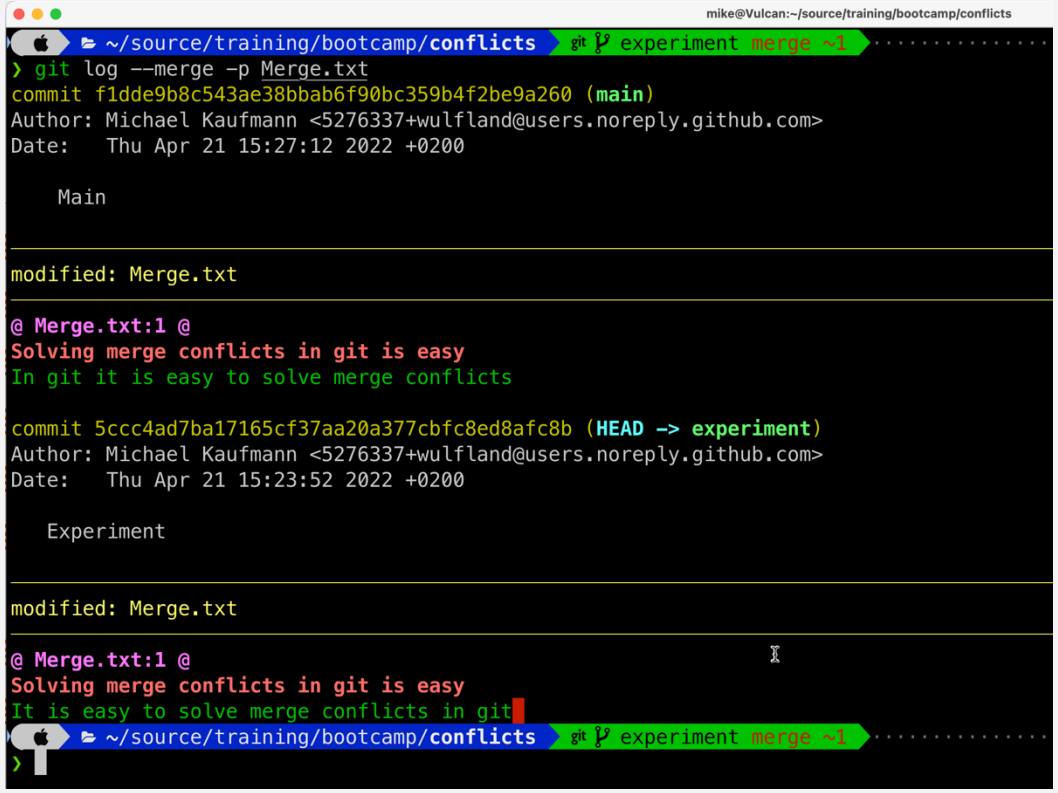
Resolving merge conflicts

```
$ git config --global merge.conflictStyle diff3
```



Resolving merge conflicts

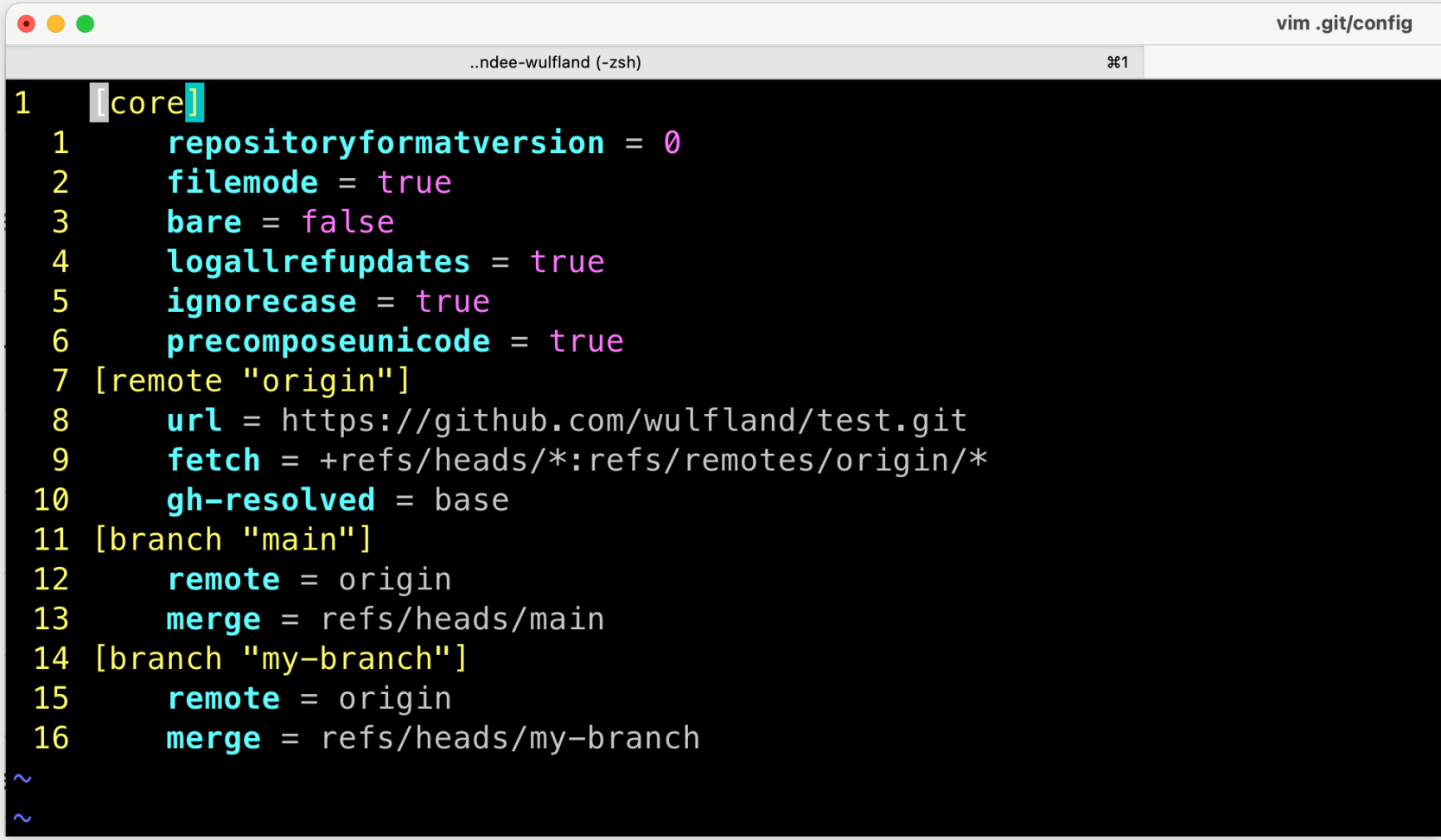
```
$ git merge --abort
$ git diff
$ git log --merge -p <filename>
$ git show :1:<filename> (common anchestor)
$ git show :2:<filename> (HEAD)
$ git show :3:<filename> (MERGE_HEAD)

$ git add <filename>
$ git merge --continue
```



Remotes

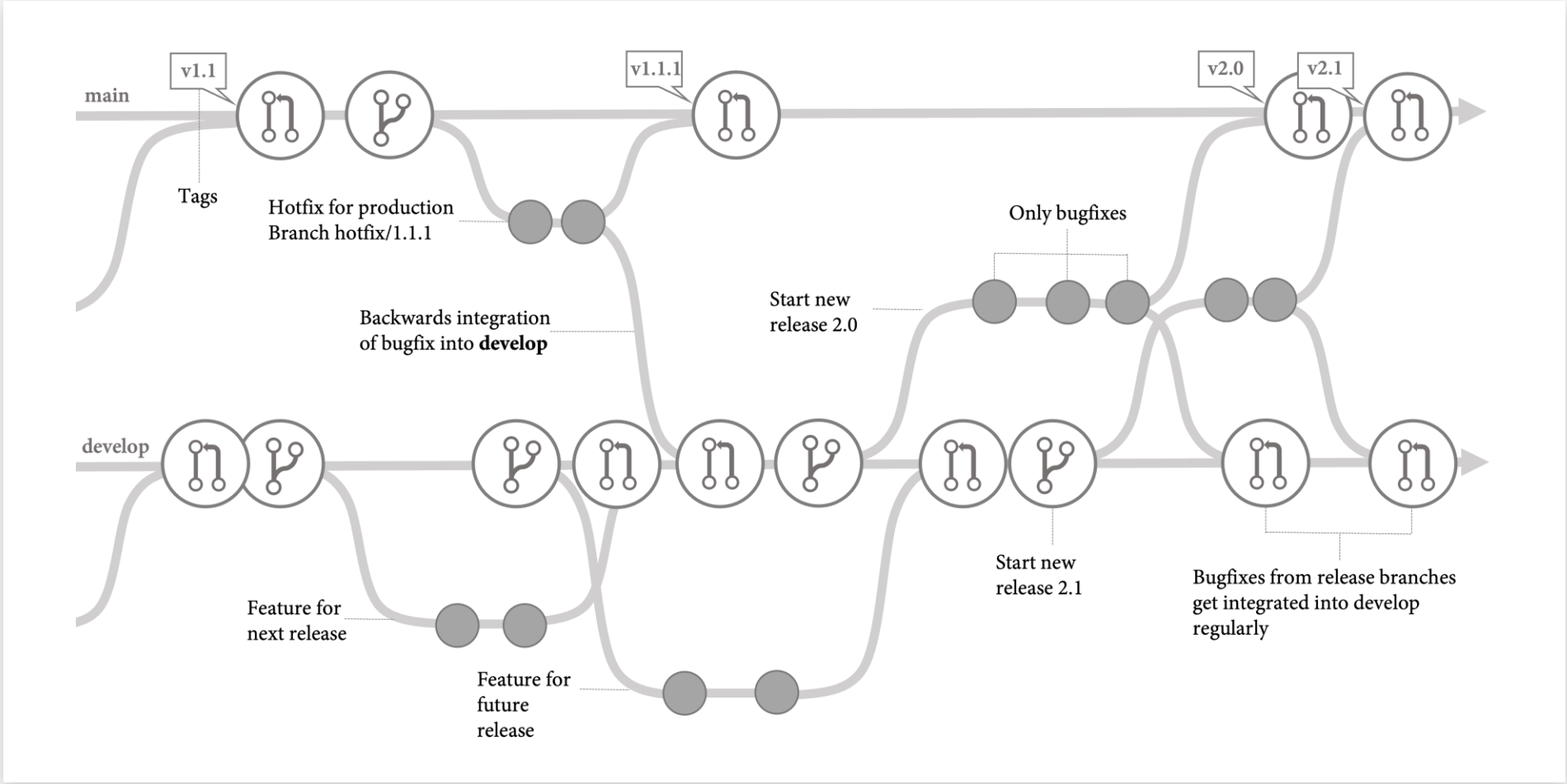
Remotes

```
$ git remote add
$ git push -u
$ git branch -vv
```



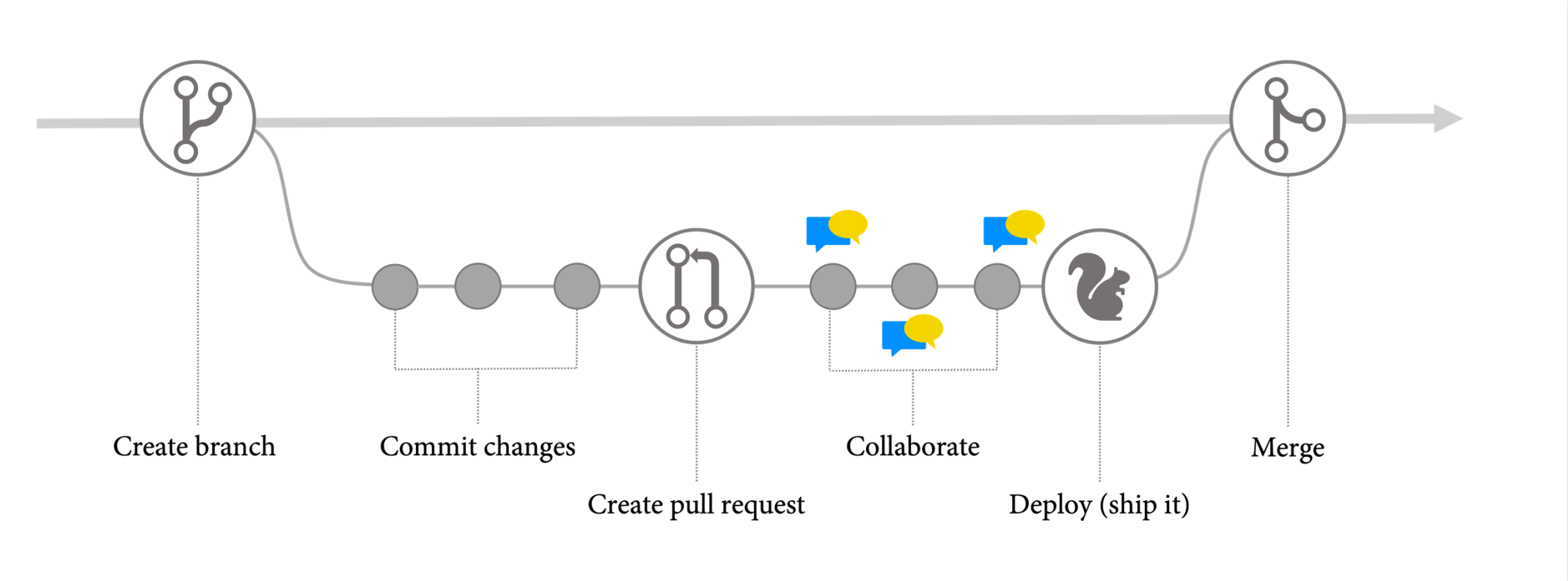
Branching workflows

- **Git flow** (https://nvie.com/posts/a-successful-git-branching-model)



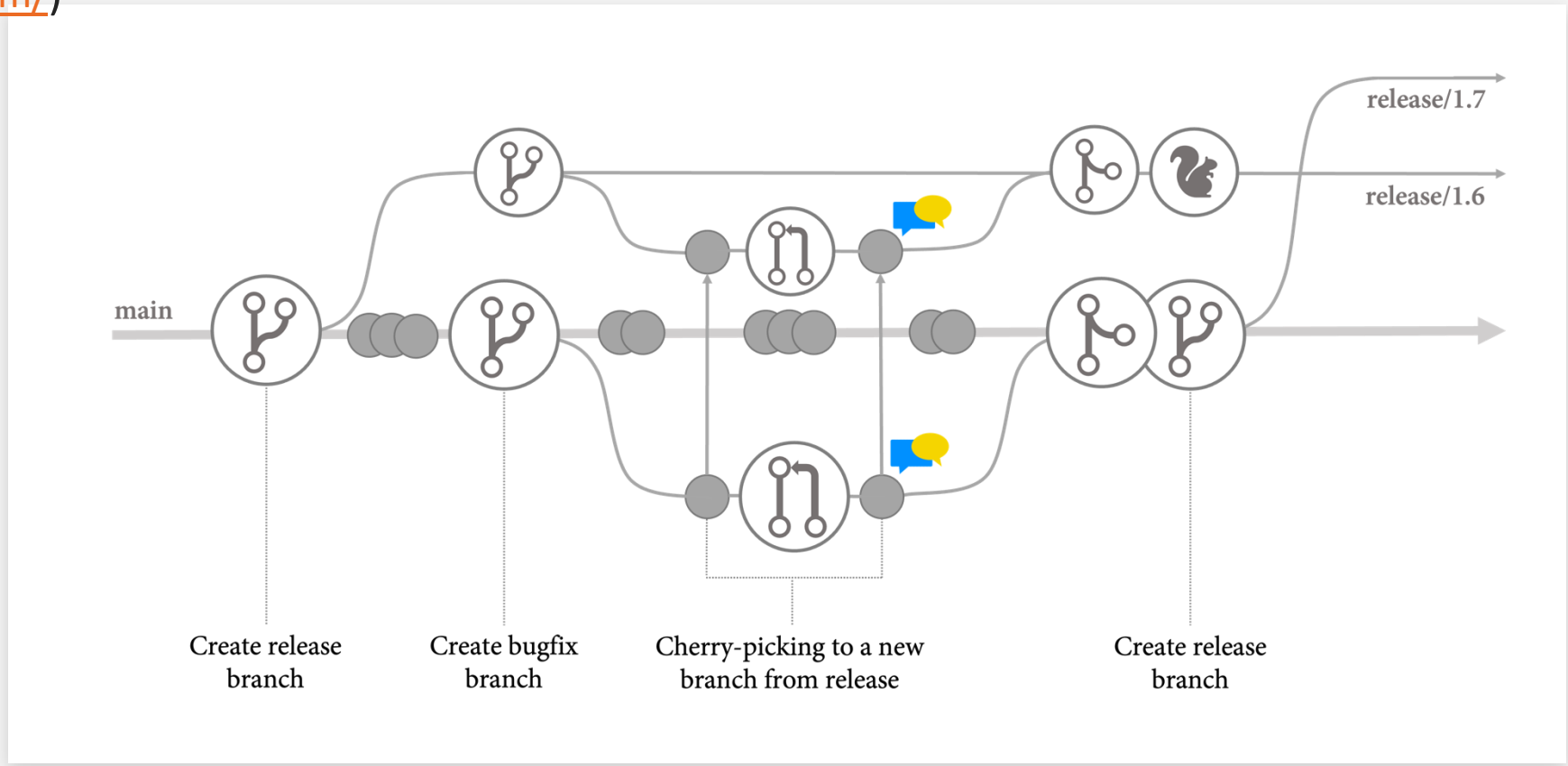
Branching workflows

- **GitHub flow** (trunk-based workflow, https://docs.github.com/en/get-started/quickstart/github-flow)



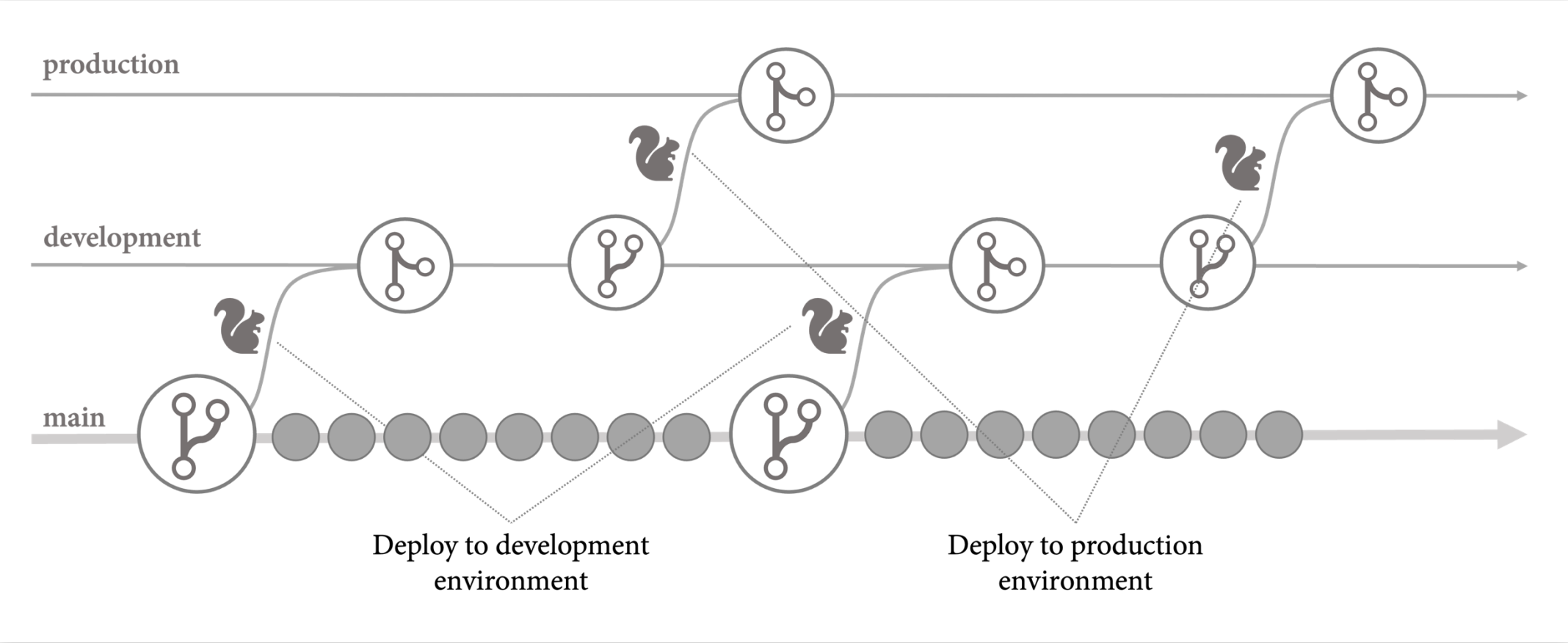
Branching workflows

- **Release flow** (https://devblogs.microsoft.com/devops/release-flow-how-we-do-branching-on-the-vsts-team/)



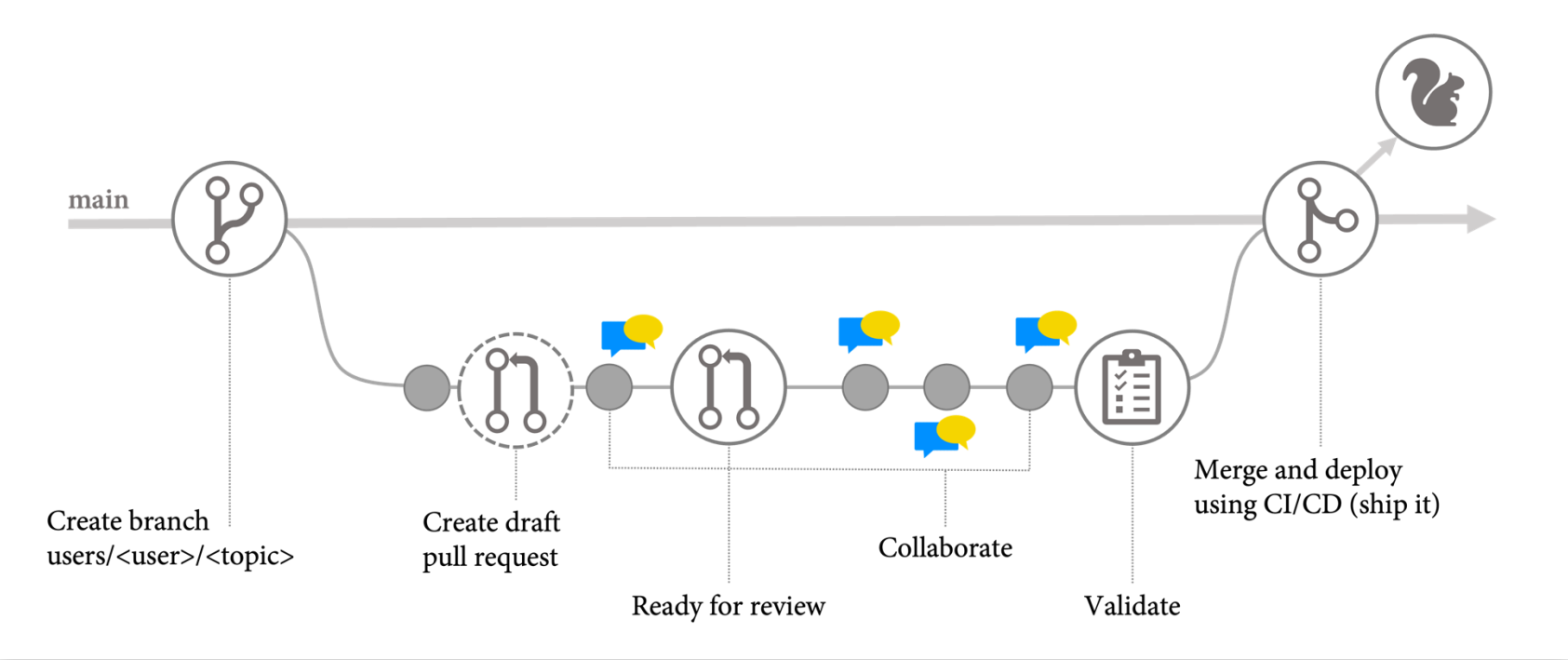
Branching workflows

- **GitLab flow** (https://docs.gitlab.com/ee/topics/gitlab_flow.html)



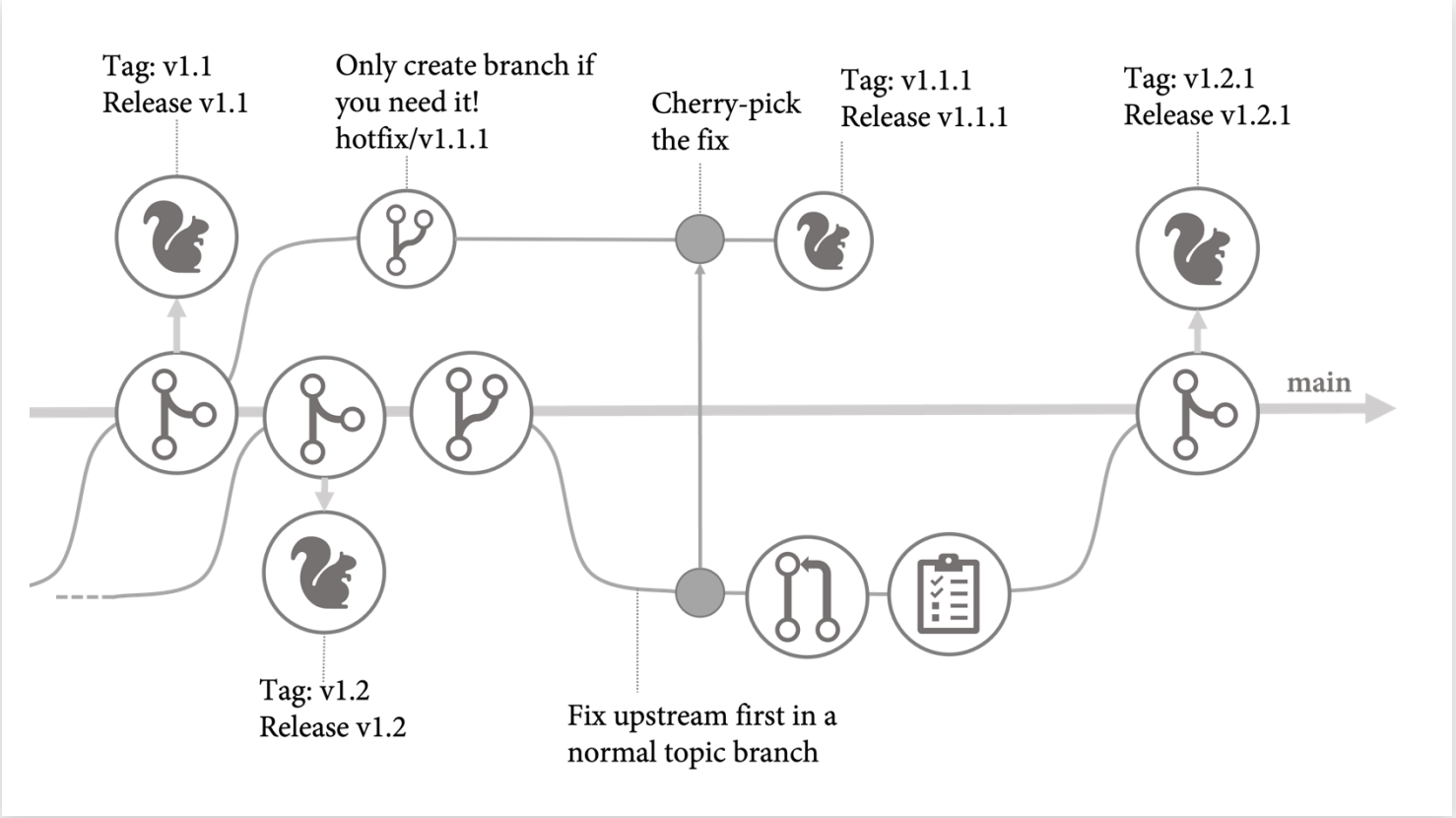
Branching workflows

- **MyFlow** (https://wulfland.github.io/MyFlow/)
  - Trunk-based (main, branch protection, CODEOWNERS)
  - Private topic branches (`users/<username>/<id>_<topic>`)
  - (Draft) pull requests, auto merge, `git push -f` / `git push origin +<branch-name>`



Branching workflows

- **MyFlow** (https://wulfland.github.io/MyFlow/)
  - Releases / Tags / cherry-pick

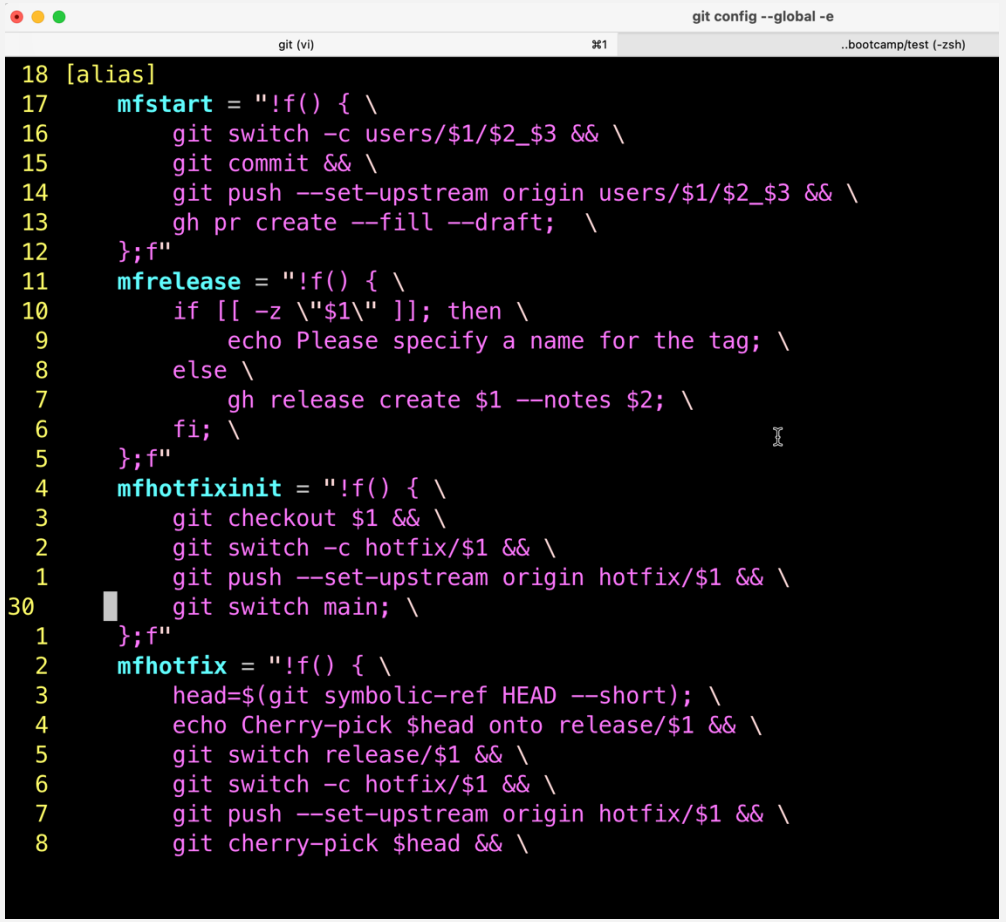


Git aliases

Git aliases

```
$ git config --global alias.last "log -1"
$ git config --global alias.lol "log --oneline --graph --decorate --all"
$ git config --global -e
```

- Functions ”!f(){ … };f "
- $1, $2, $3, … $@
- Line endings: \
- && or ; to separate commands

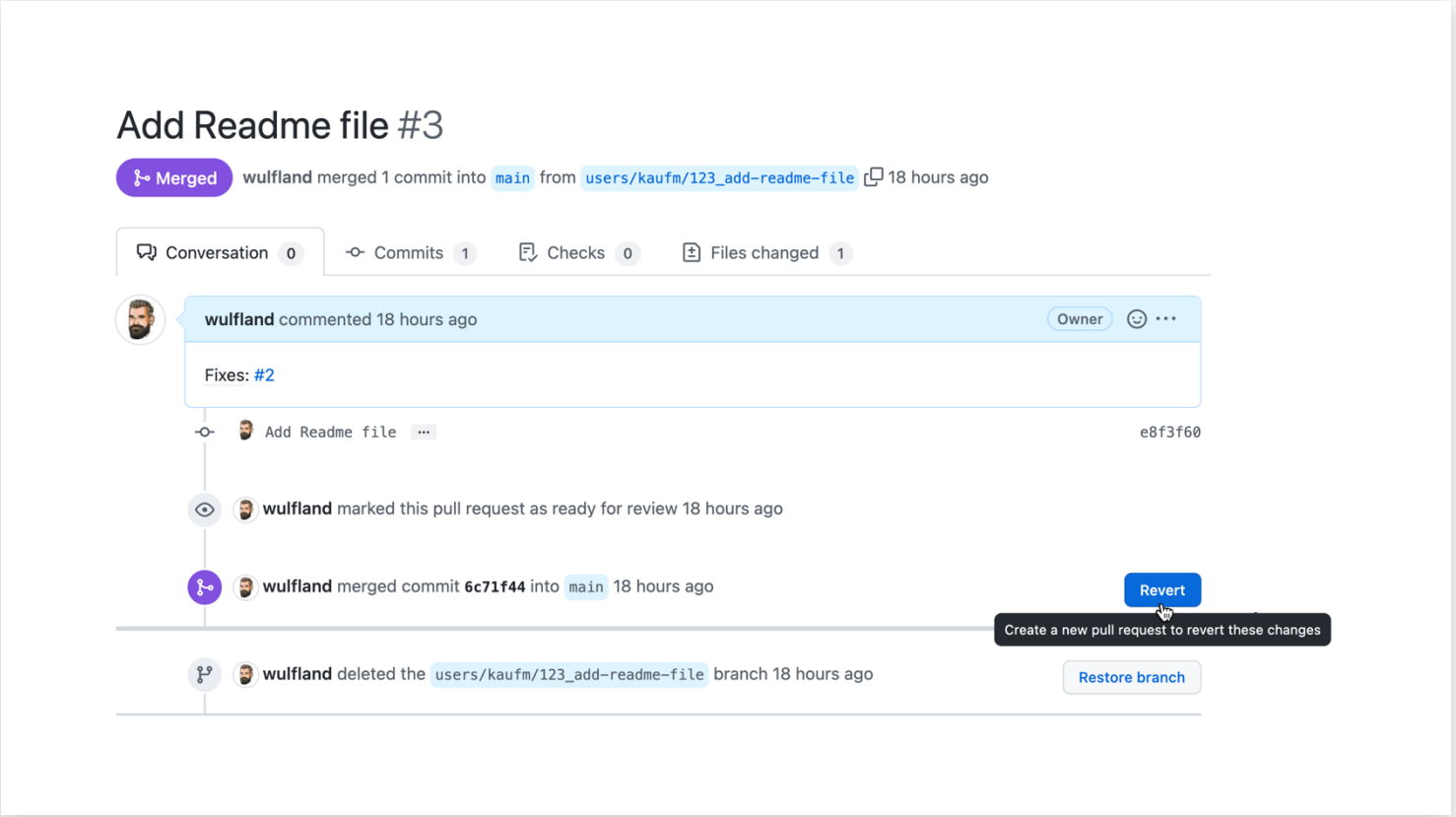


Reverting changes

Reverting changes

Undo changes in a safe way

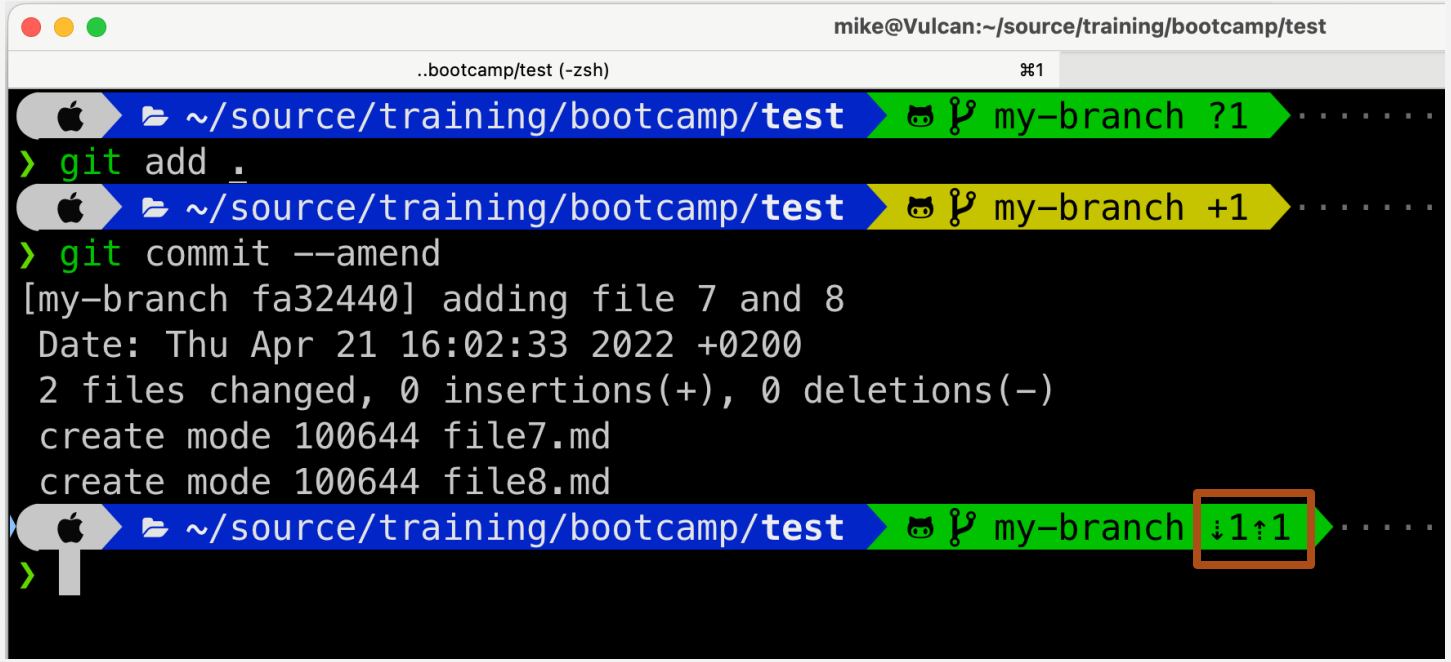
```
$ git revert <REF>
```



Ammending changes

Ammending changes

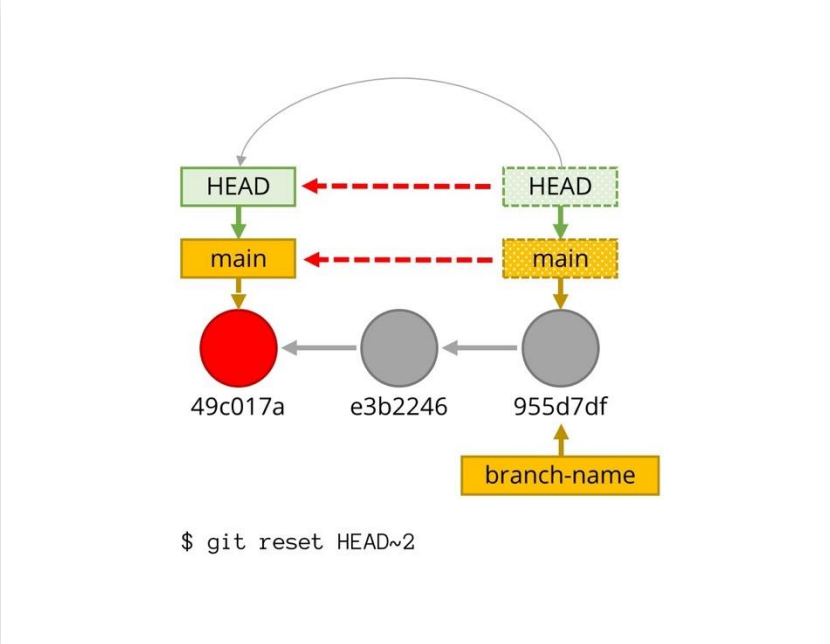
```
$ git commit --amend
$ git push --force
```



Reseting your repository

Resetting your repository

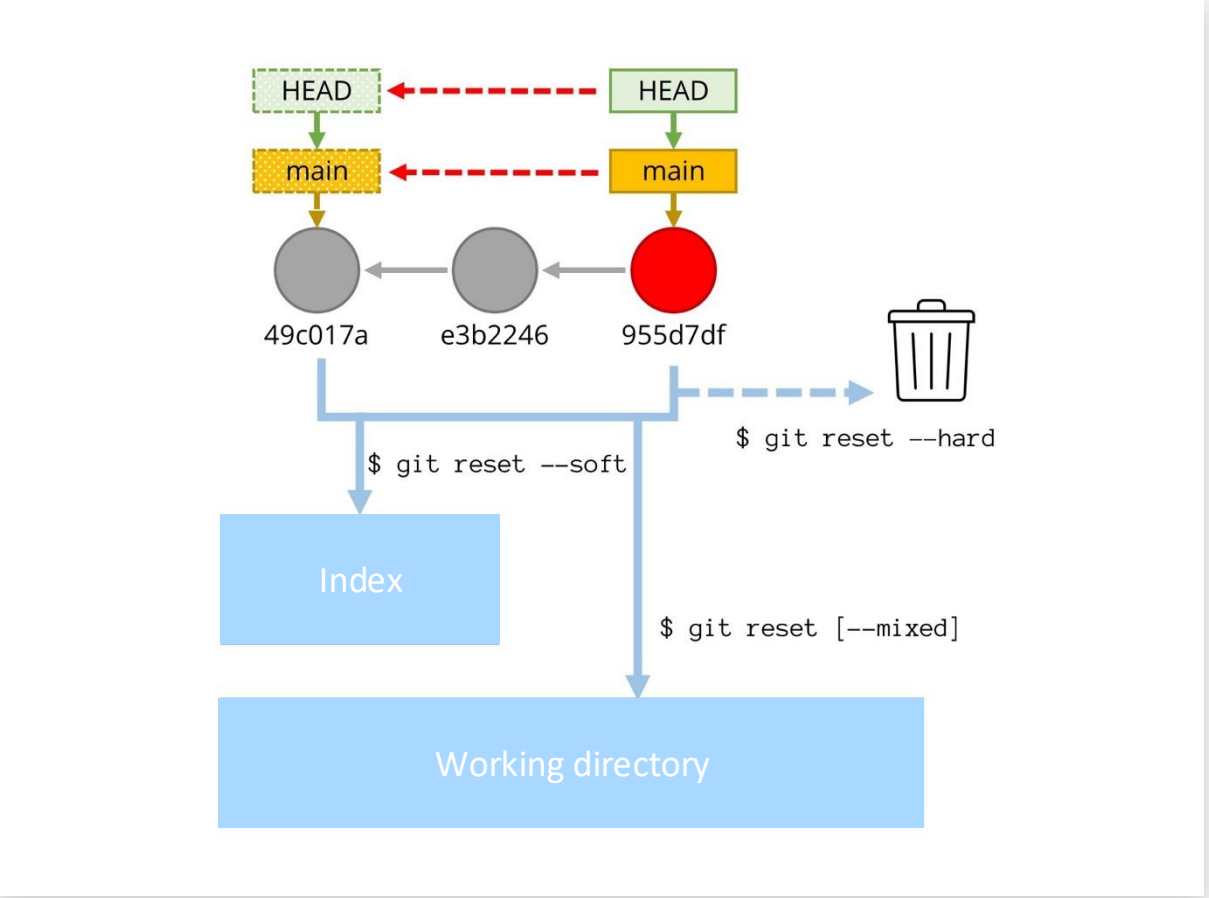
- Moving the current branch to another position



Resetting your repository

- What happens with the changes?

```
$ git reset [--mixed]
$ git reset --hard

$ git reset --soft
$ git reset --merge | --keep
```

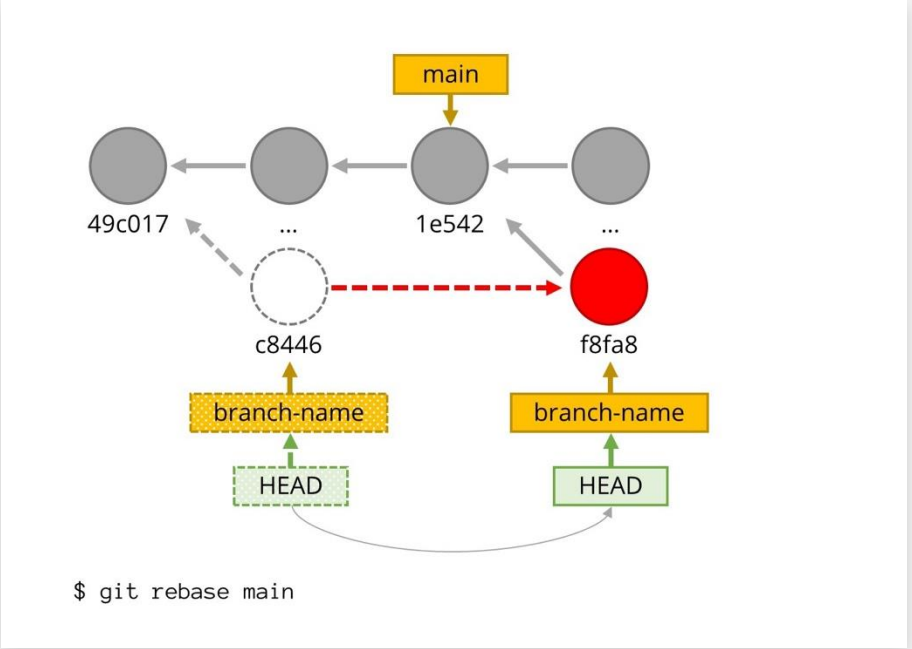


Rebase

Rebase

- Rebase

```
$ git rebase <REF>
$ git rebase –I
$ git commit --amend
$ git rebase --continue
$ git rebase --abort
```

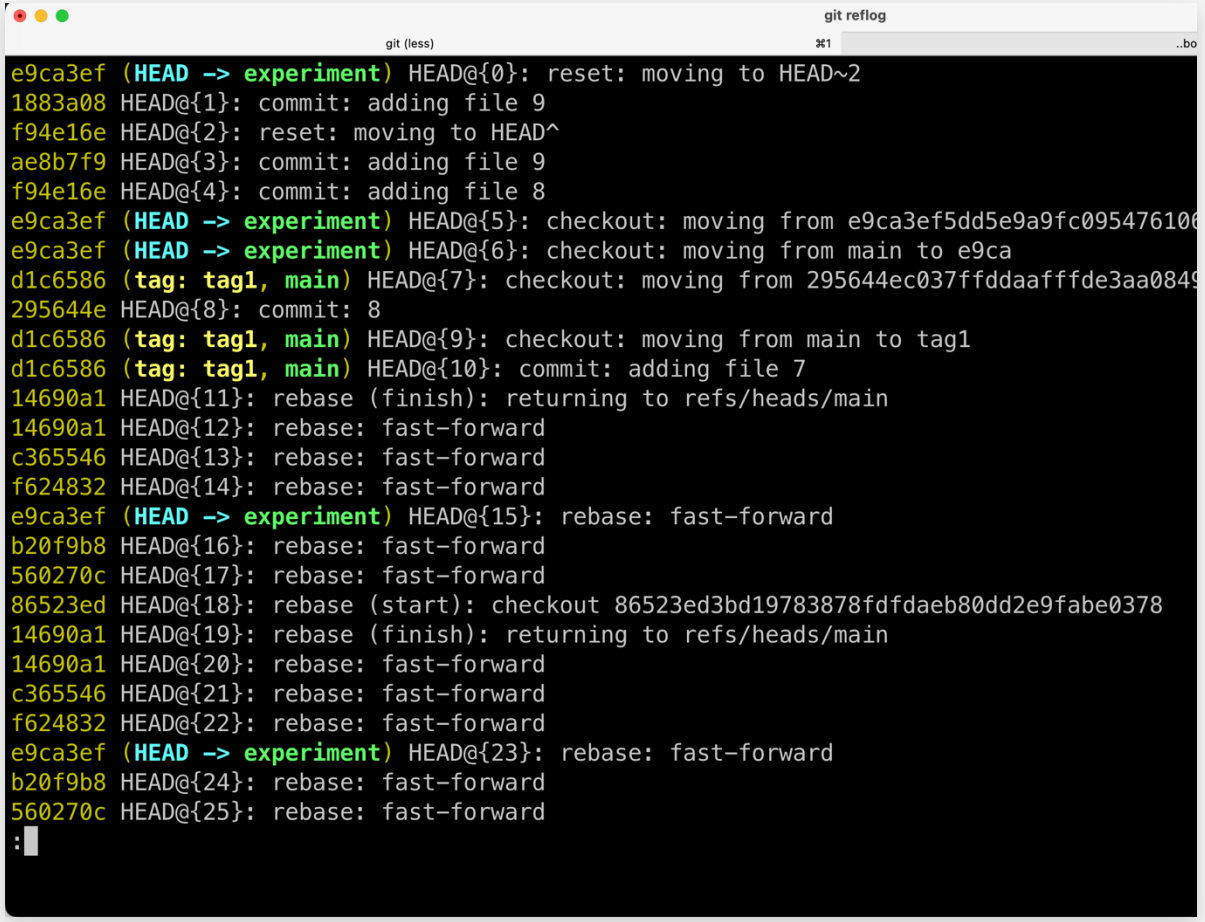


Cherry-picking and the reflog

Git RefLog

- The RefLog is **local**!
- Expires:
  - gc.reflogExpireUnreachable (default 30 days)
  - gc.reflogExpire (default 90 days)

```
$ git reflog [show]

$ git cherry-pick <REF>
$ git cherry-pick --edit <REF>
$ git cherry-pick –no-commit <REF>
```

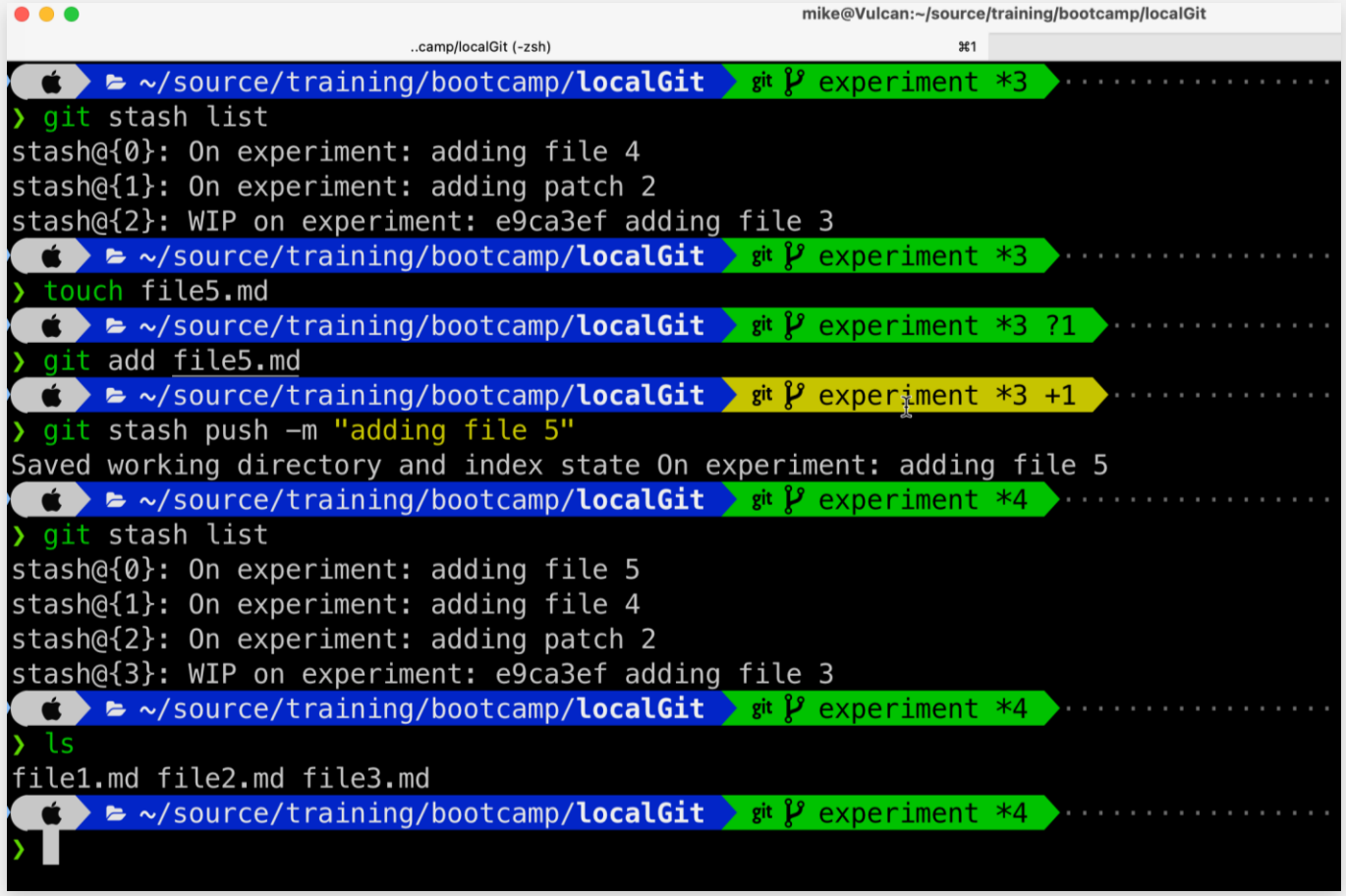


The stash

The stash

- LIFO: Last In – First Out

```
$ git stash [push]
$ git stash push –m "<message>"
$ git stash list
$ git stash pop [--index]
$ git stash branch <name>
```



Let‘s connect

Michael Kaufmann

Managing Director Xebia Germany

Microsoft Regional Director, MVP

Linkedin.com/in/mikaufmann

mkaufmann@xpirit.com

@mike_kaufmann

@wulfland

https://writeabout.net

Let's connect!