

Using Git Effectively

Use case

- 4 people are working on a software project
- Everyone is responsible for a different feature
- Some files need to be edited by everyone
- This creates conflicts when trying to develop all features simultaneously

Why use git?

- More than just a way to save your work
- Can allow for complex workflows to manage team development
- A distributed system – most operations can be performed locally and propagated to remotes

object database



blob

tree

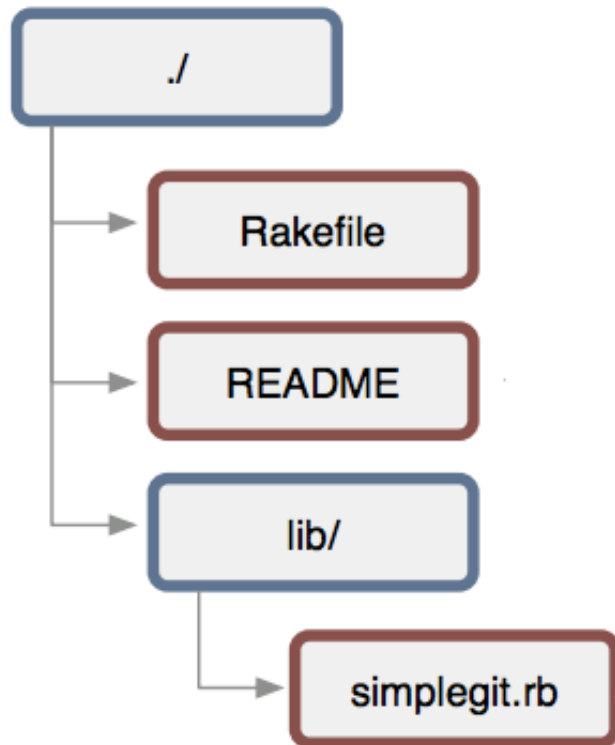
commit

tag

object database

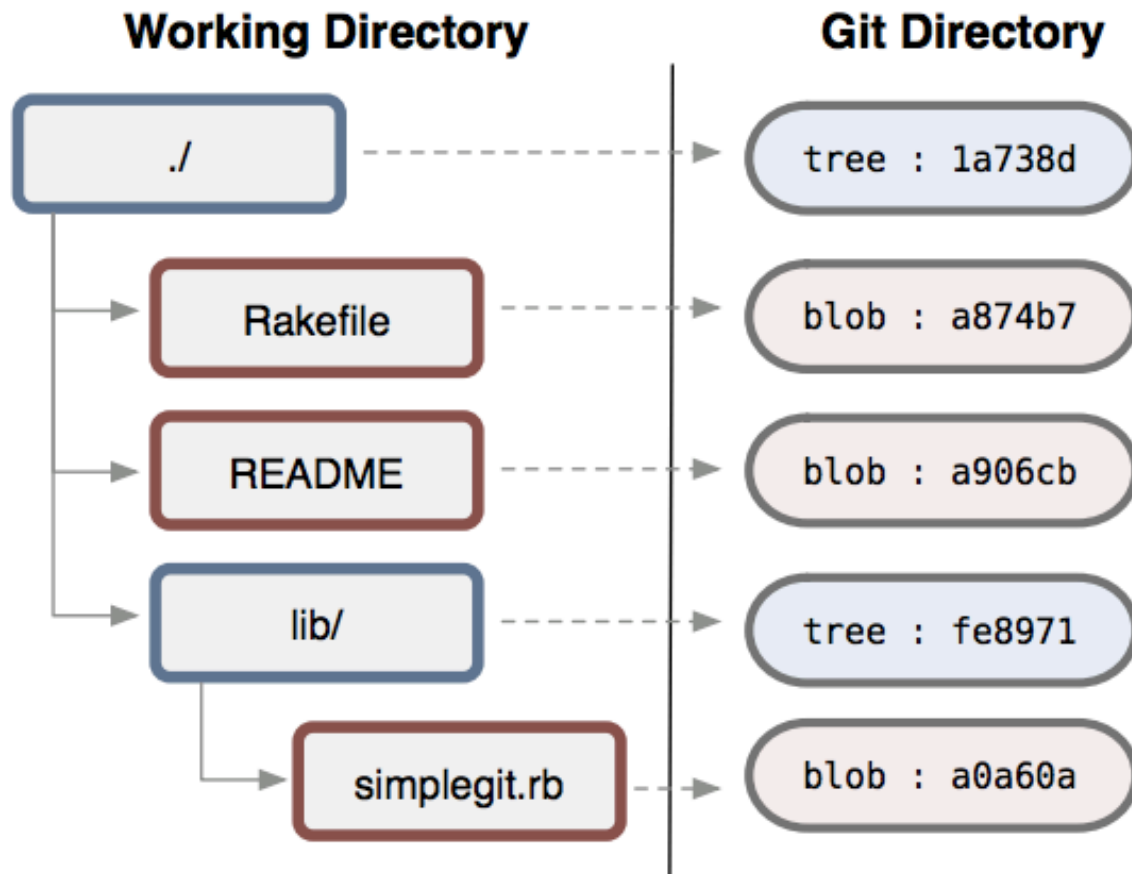
blob

Working Directory



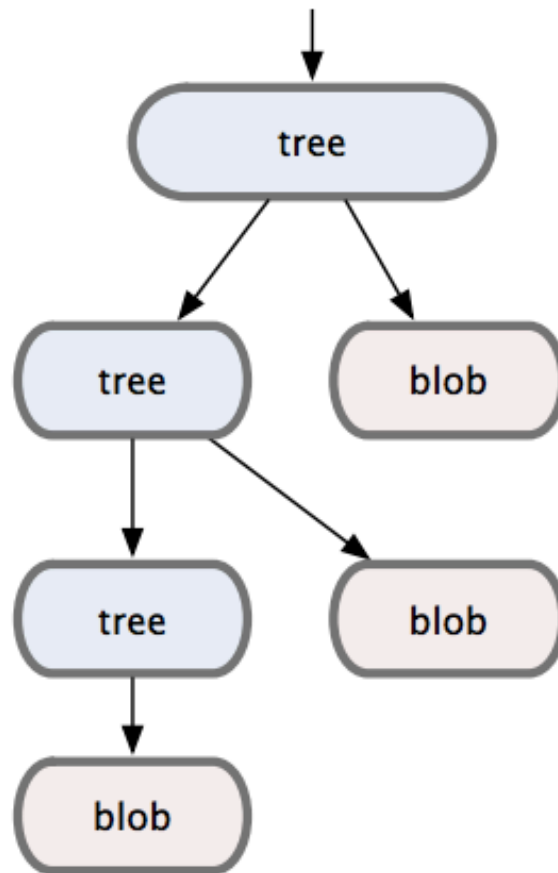
object database

tree



object database

commit



object database

commit : e1b3ec

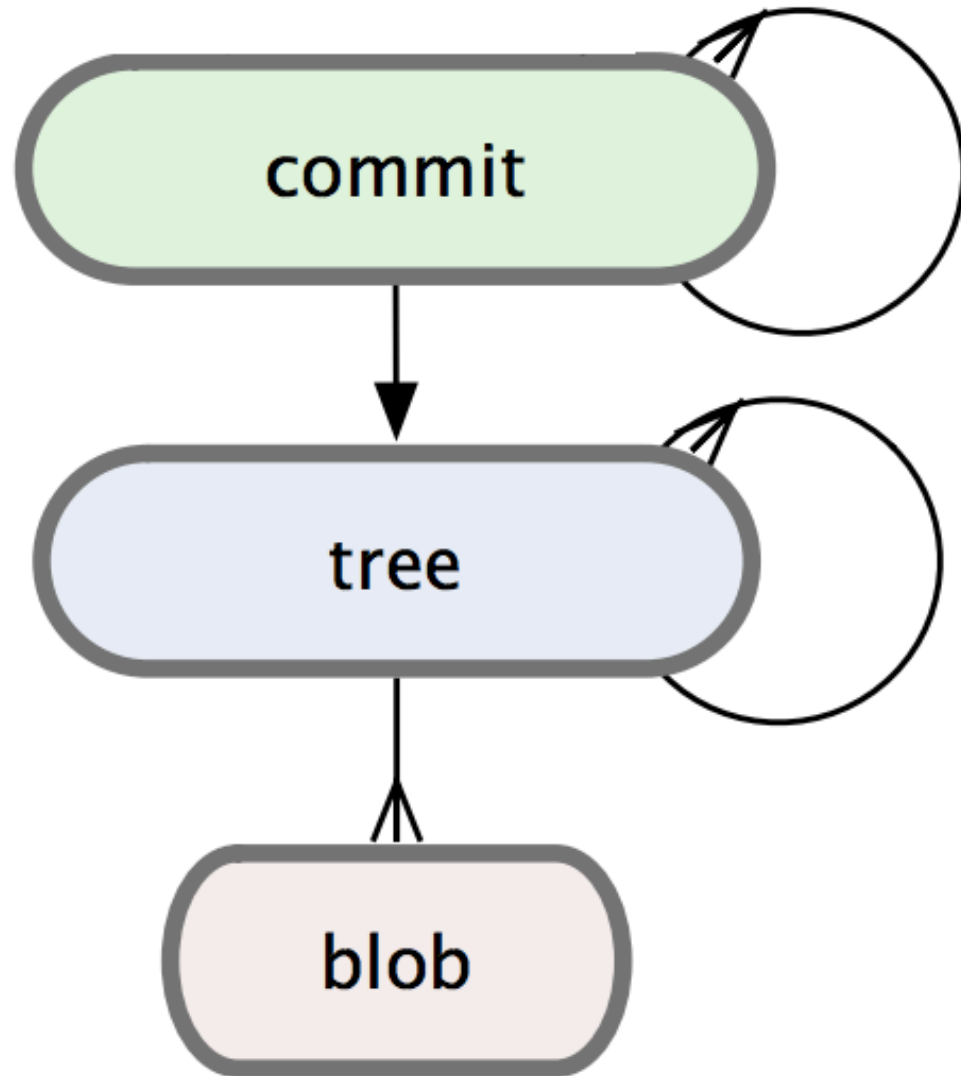
zlib::deflate

```
commit [content size]\0
```

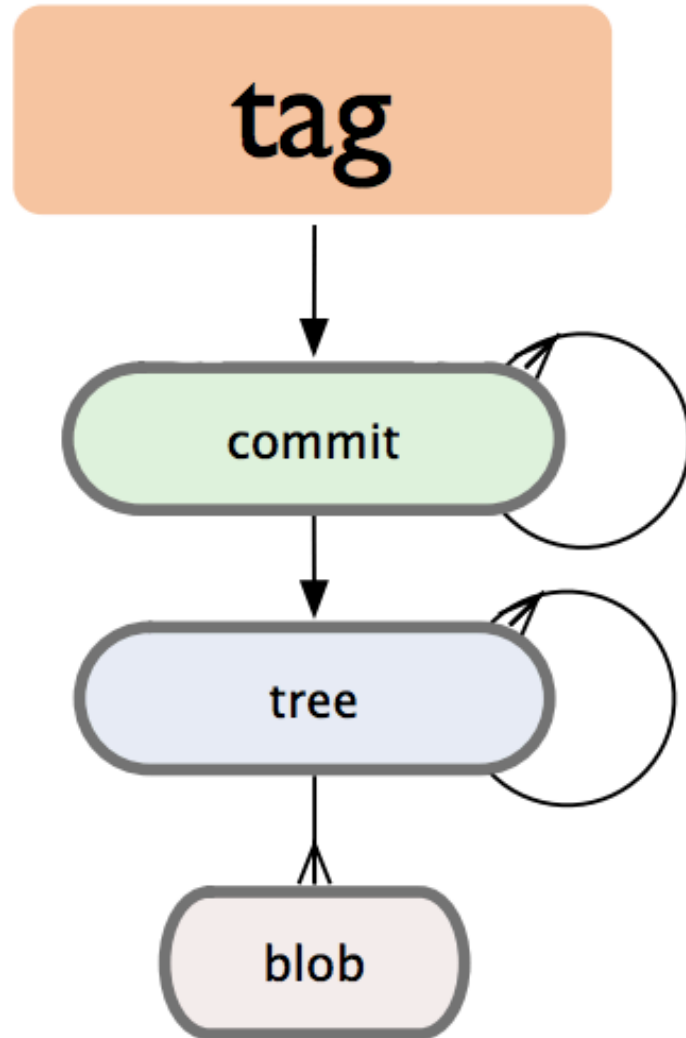
```
tree e1b3ec
parent allbef
author Scott Chacon
    <schacon@gmail.com> 1205624433
committer Scott Chacon
    <schacon@gmail.com> 1205624433

my second commit, which is better than
the first
```


object database



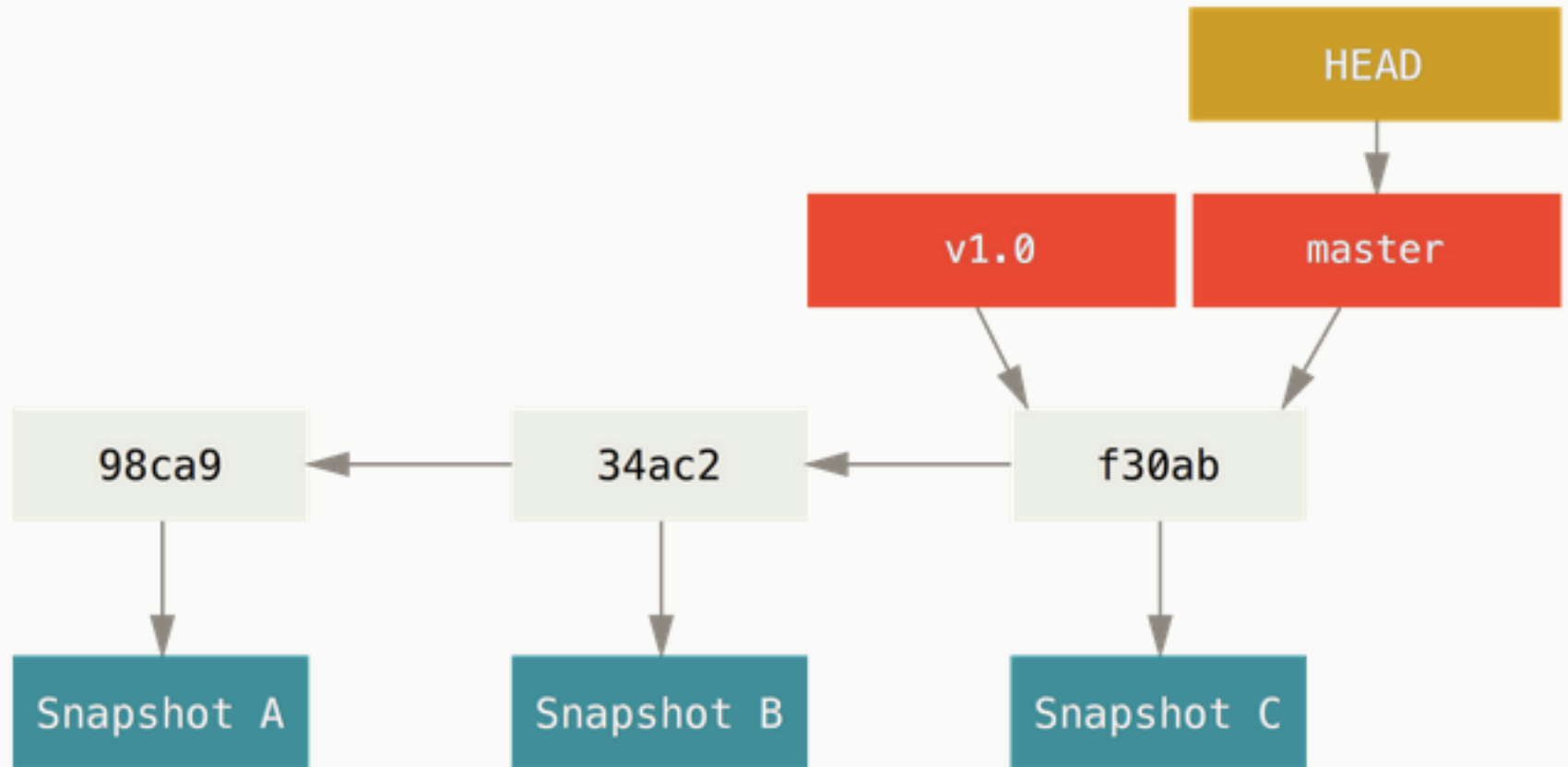
object database

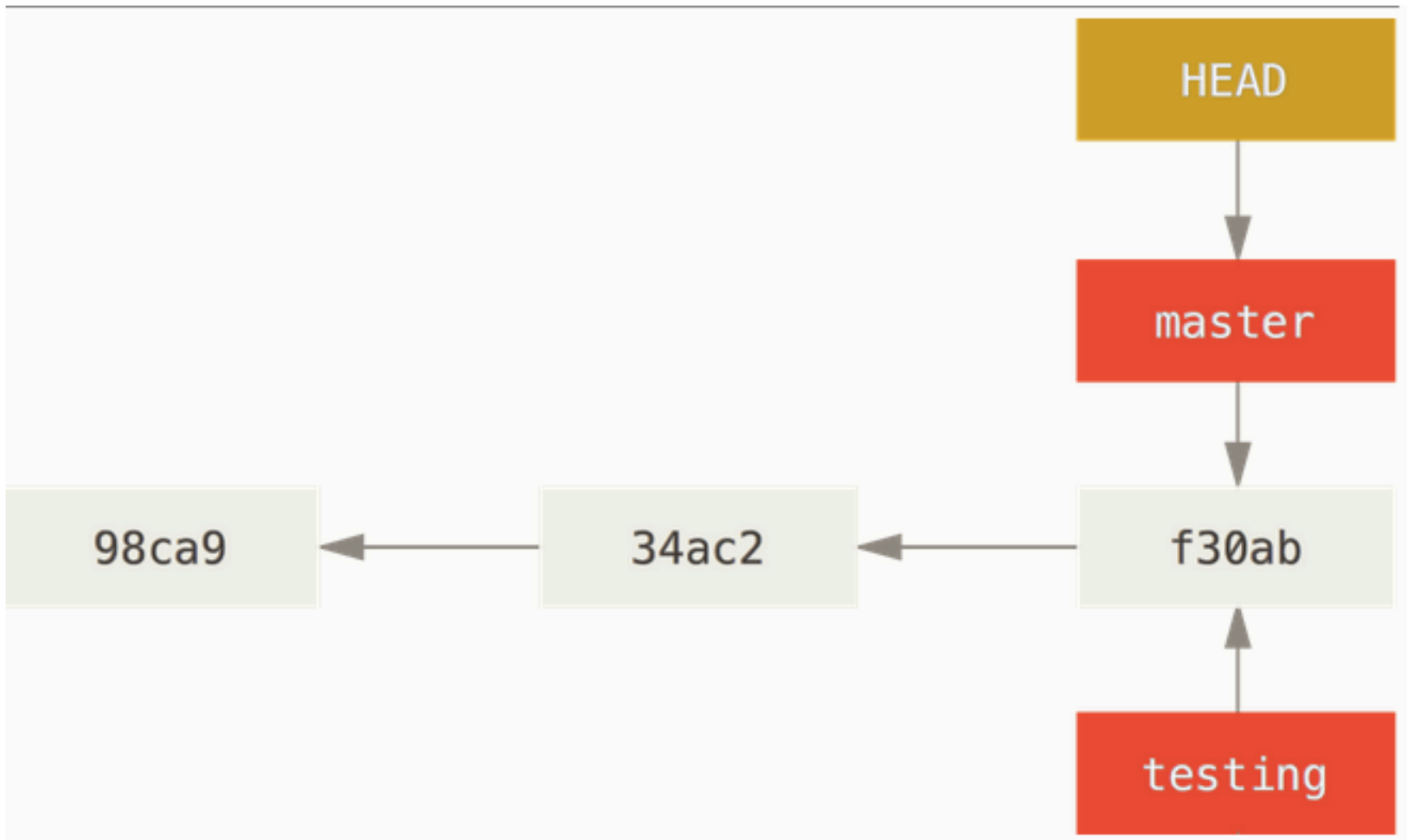


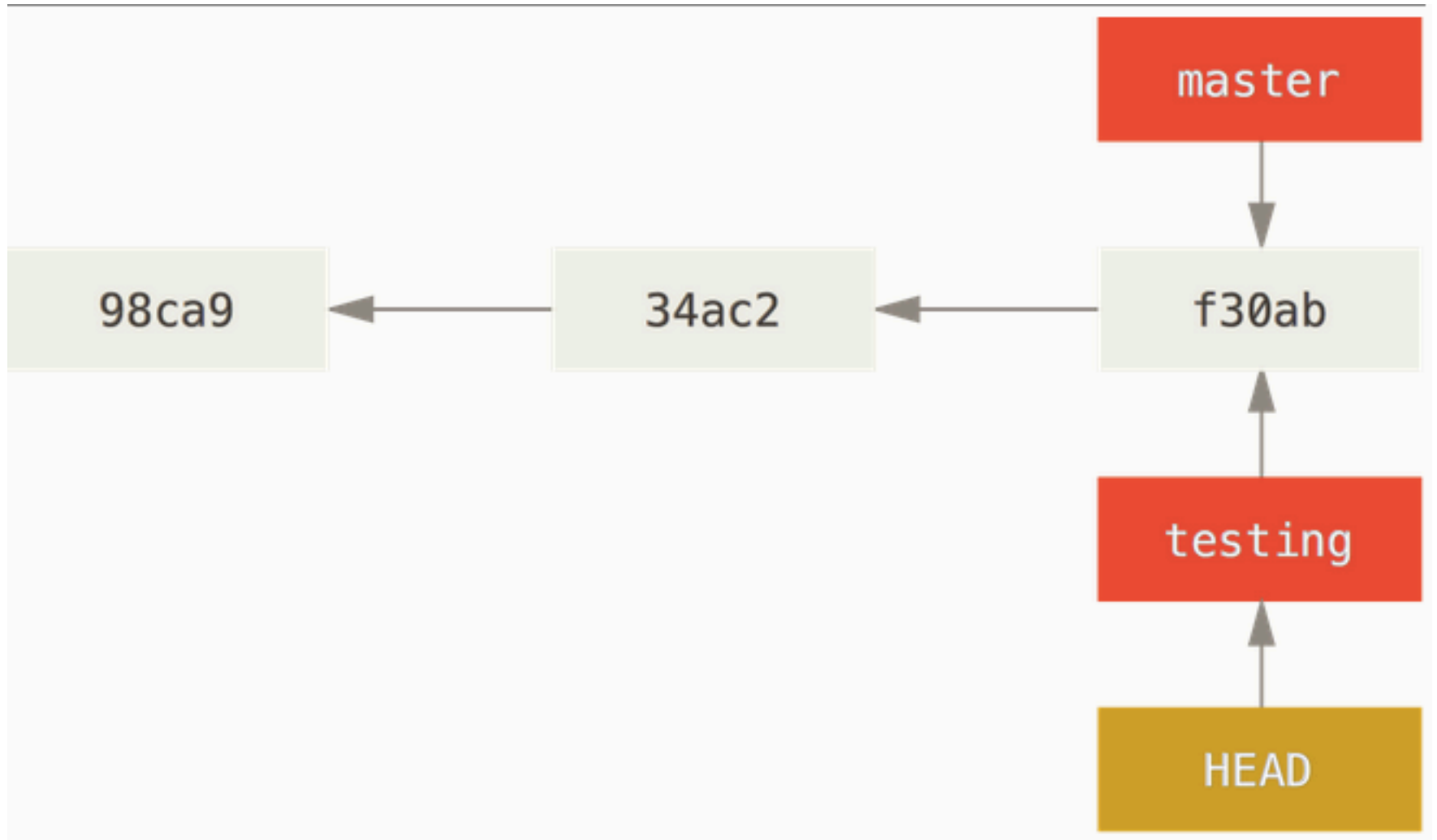
Branching

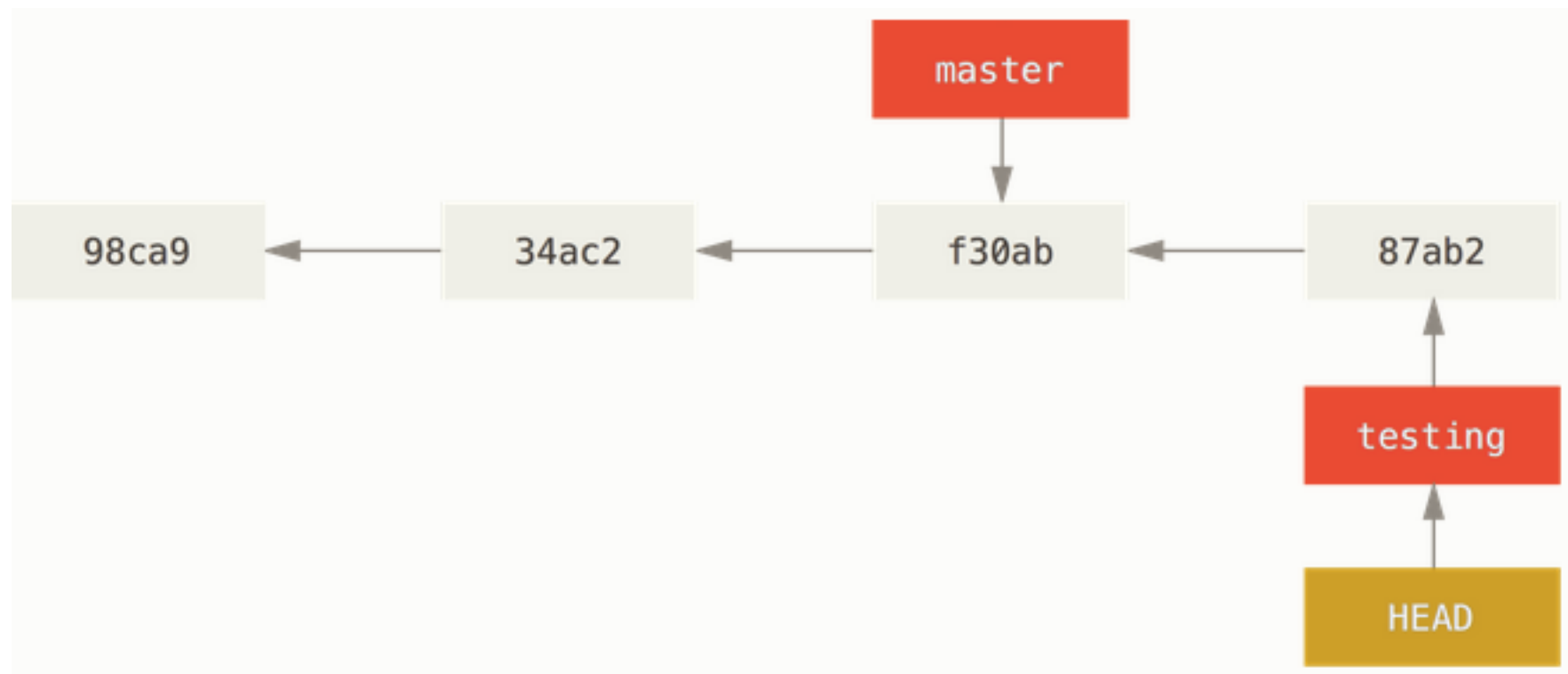
- Branches are simply pointers to a given commit
- HEAD points to what branch you're currently on

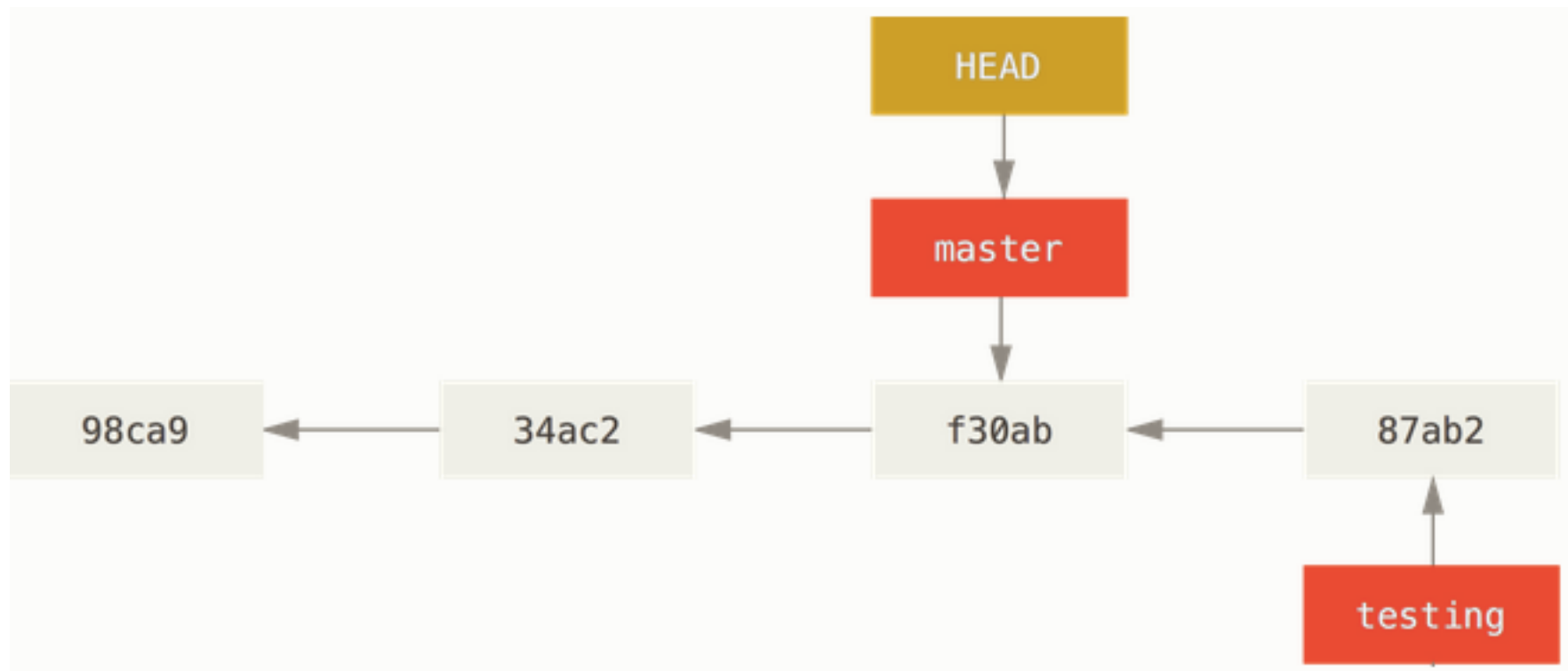
Branching Example

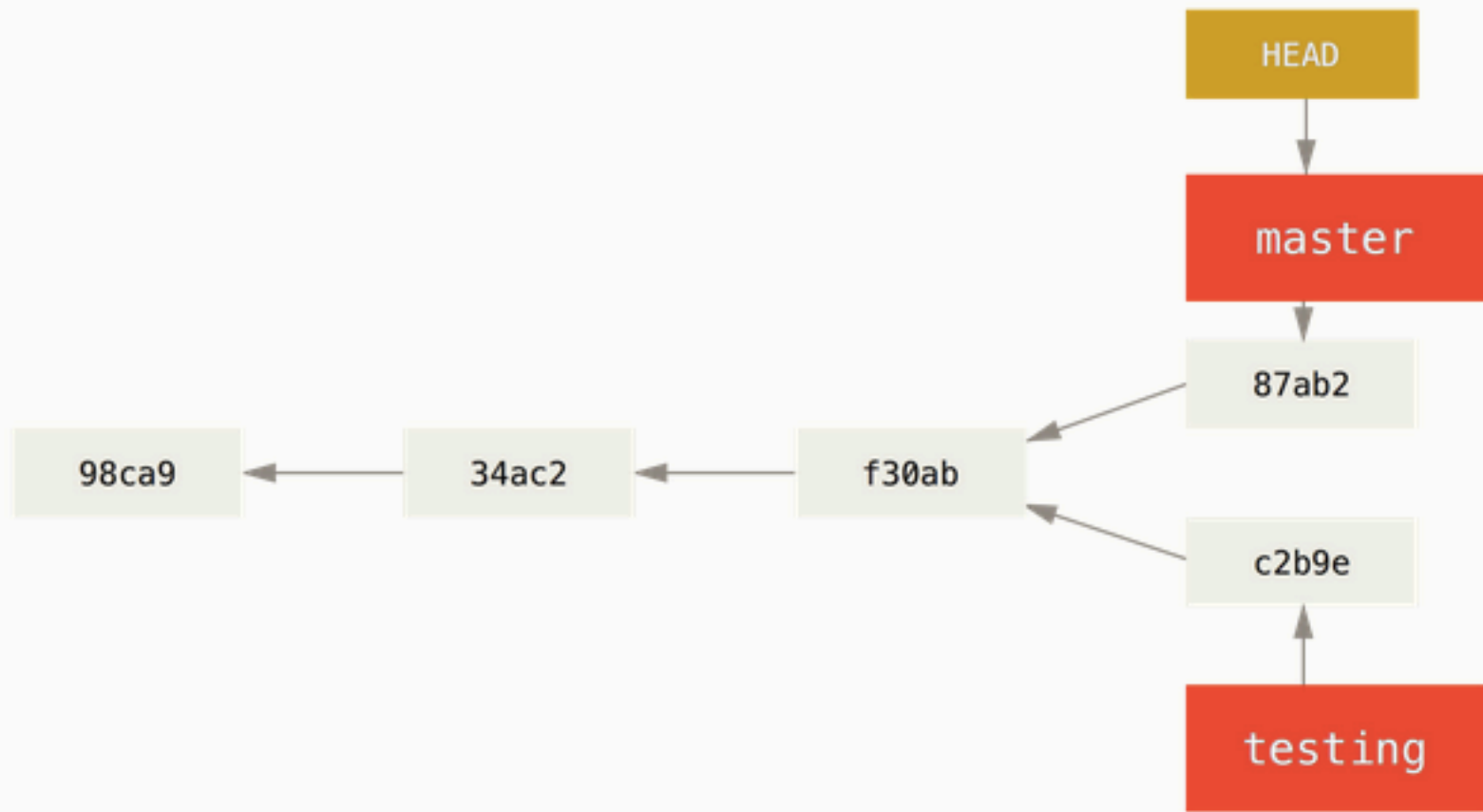












Resources

- <http://nvie.com/posts/a-successful-git-branching-model/>
- <http://git-scm.com/>
- <https://try.github.io/levels/1/challenges/1>
- <http://pcottle.github.io/learnGitBranching/>

Image Credits

- Scott Chacon – [Getting Git presentation](#)
- Git handbook – [Branches in a Nutshell](#)