Git

Sławomir Piotrowski

History

				BitKeeper	Mercurial
SCCS	RCS	CVS	SVN	Git	
1972	1982	1986	2000		2005

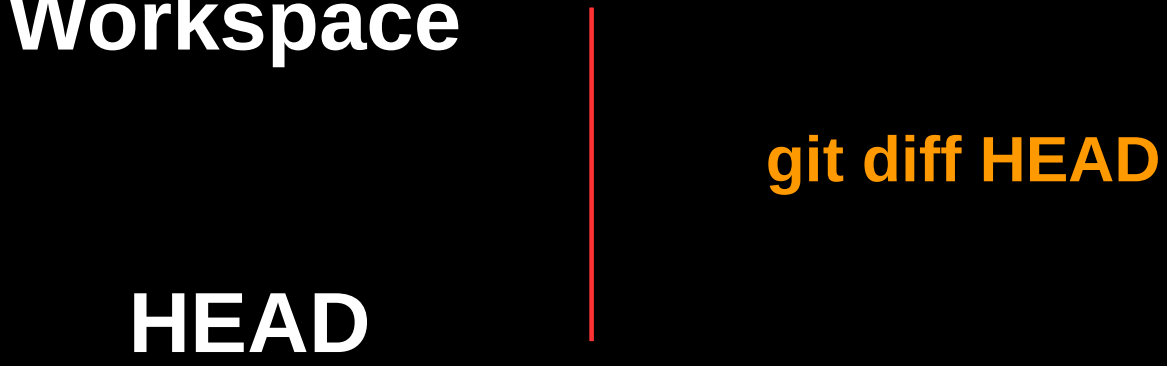
Linus Torvalds

Linux Kernel

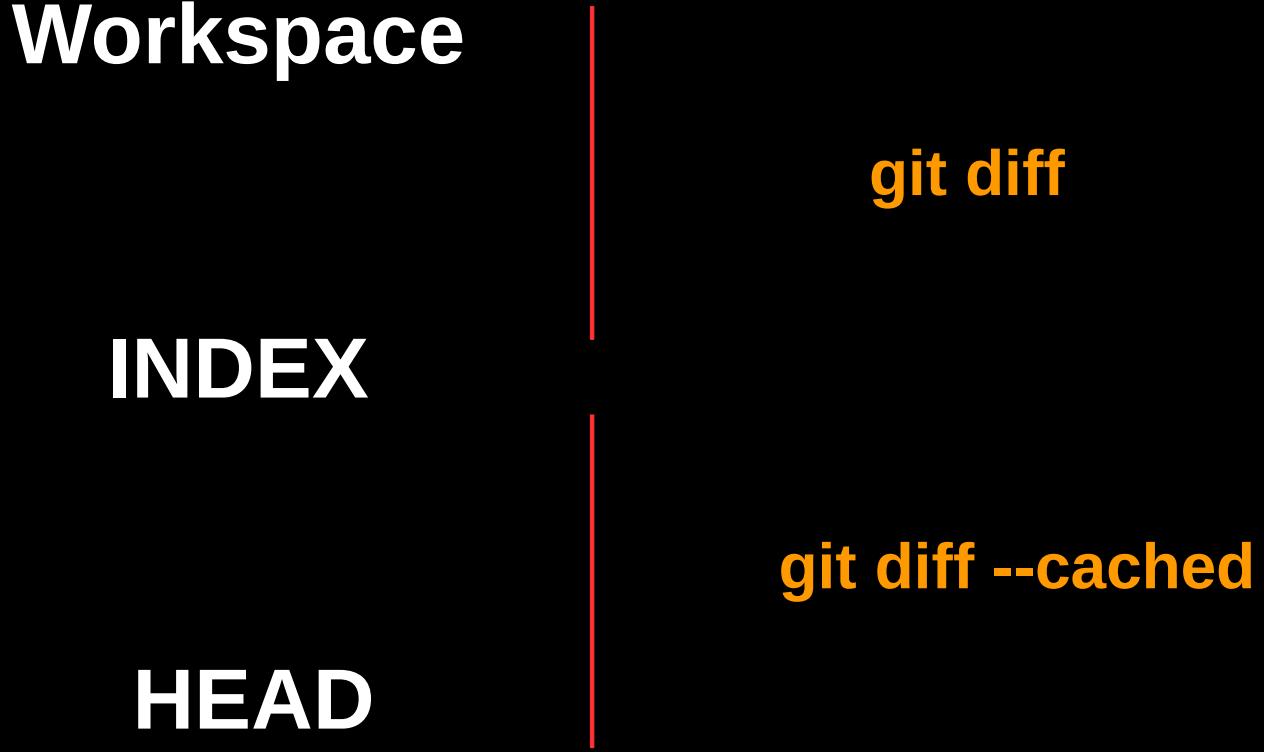
Tarballs and patches

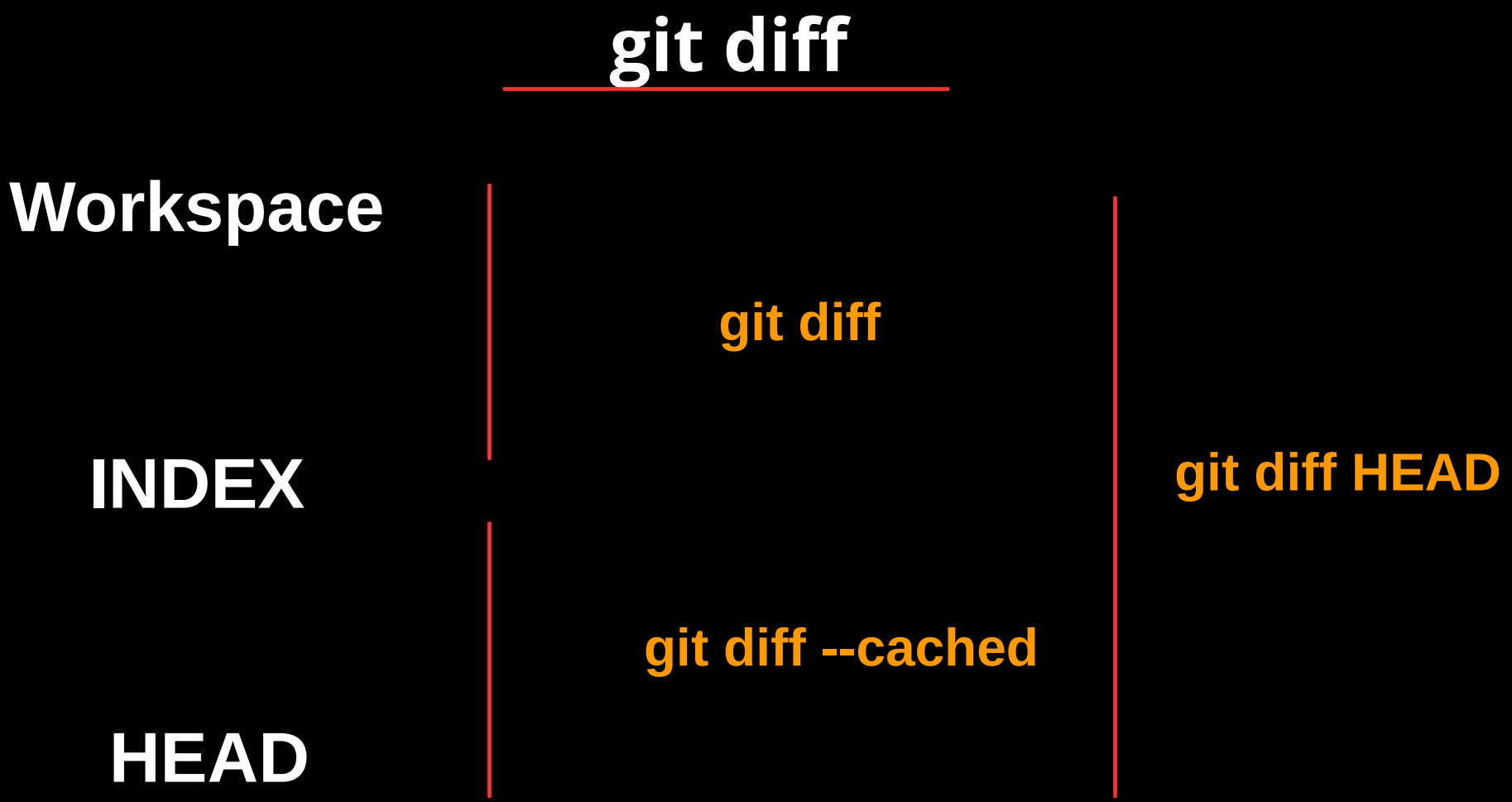
Workspace, INDEX, HEAD
SHA1
Reset

git diff



git diff




git diff
Workspace
INDEX
HEAD
git diff
git diff --cached
git diff HEAD

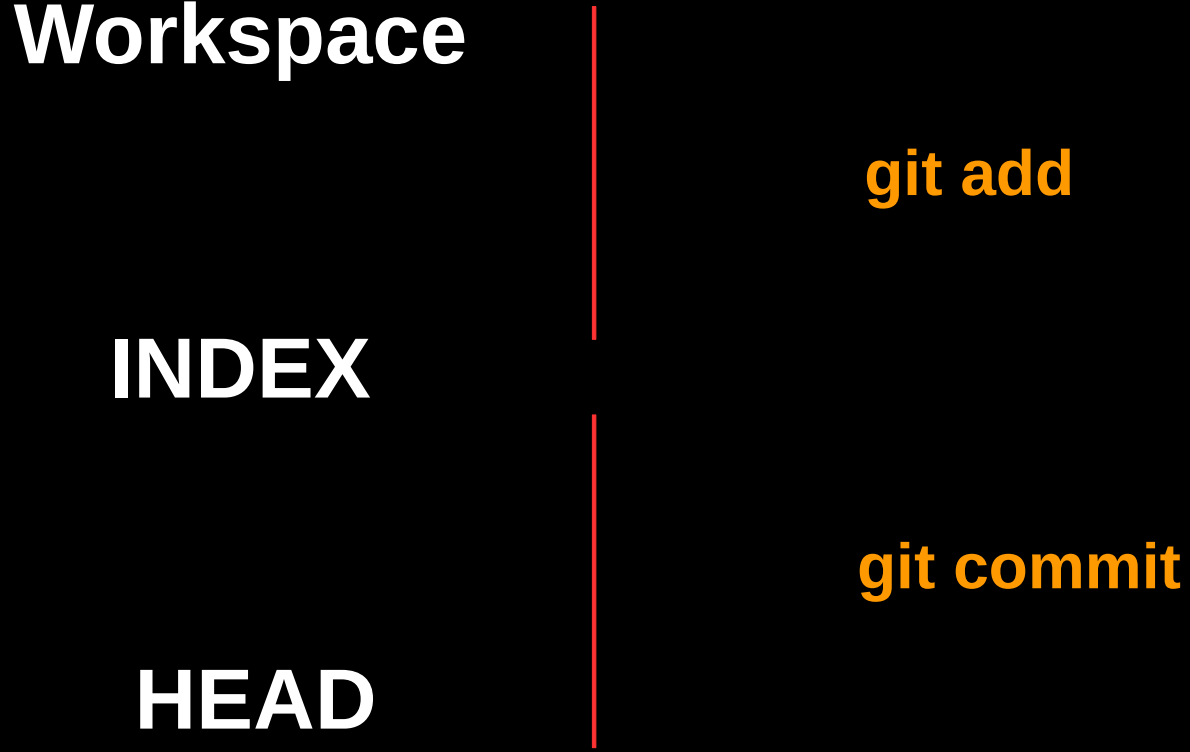
New repo

```
mkdir hello_world
cd hello_world
git init

git config user.email "sentinel@atteo.com"
git config user.name "Sławek Piotrowski"
```

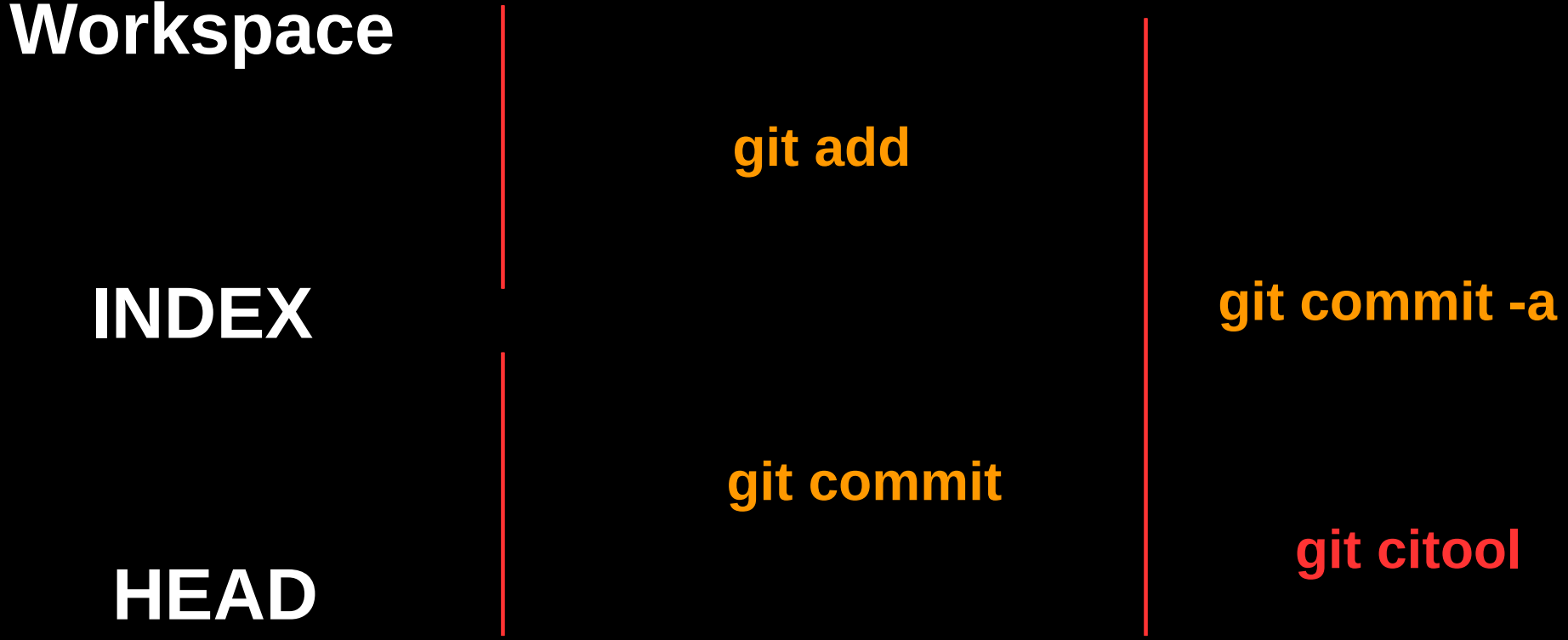
First commit

```
git add <file>
git commit
```

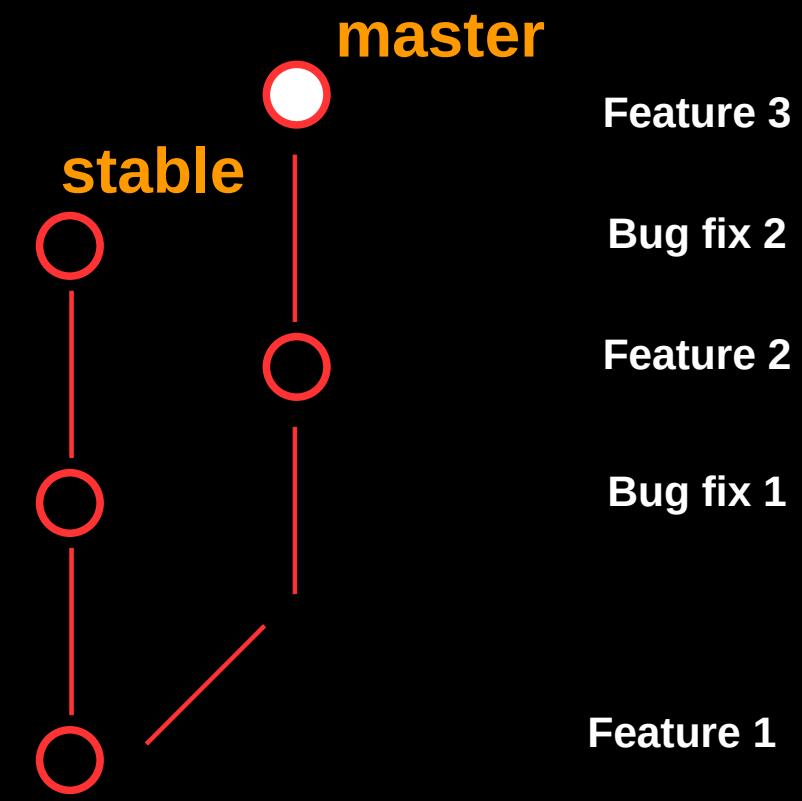
commit



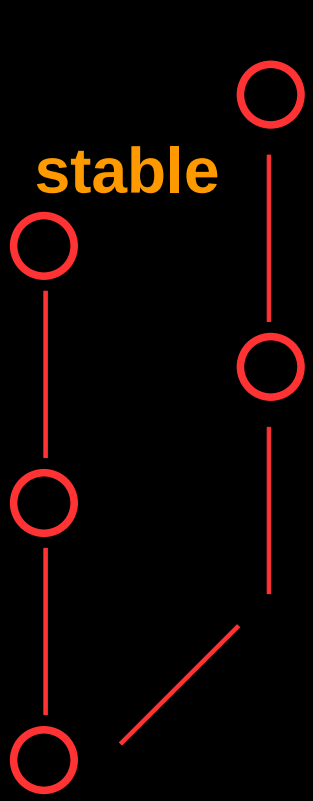
commit


Workspace
INDEX
HEAD
git add
git commit
git commit -a
git citool

gitk --all

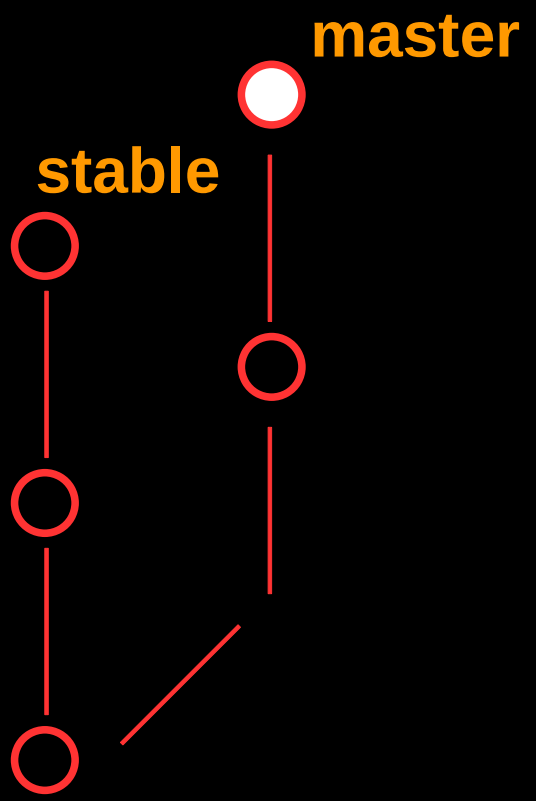

master
stable
Feature 3
Bug fix 2
Feature 2
Bug fix 1
Feature 1

Branches

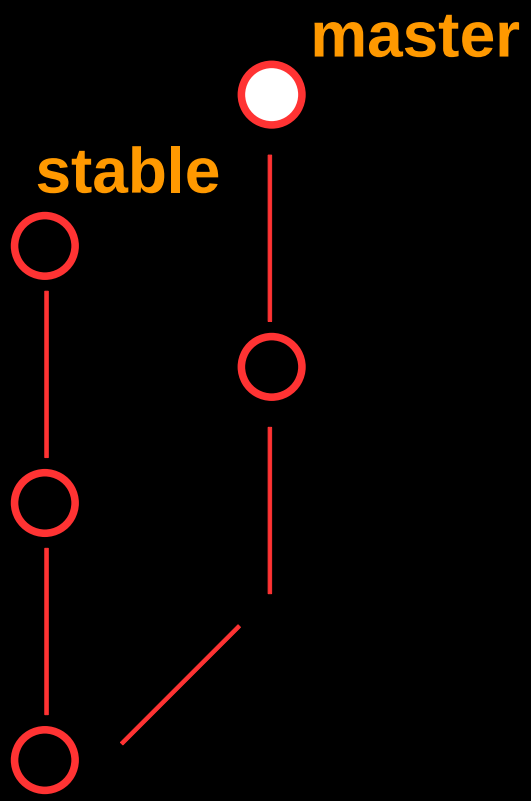



```
git branch

git branch <new-branch>
git branch <new-branch> <sha1>
git checkout -b <new-branch> <sha1>

git branch -d <new-branch>
git branch -D <new-branch>
```

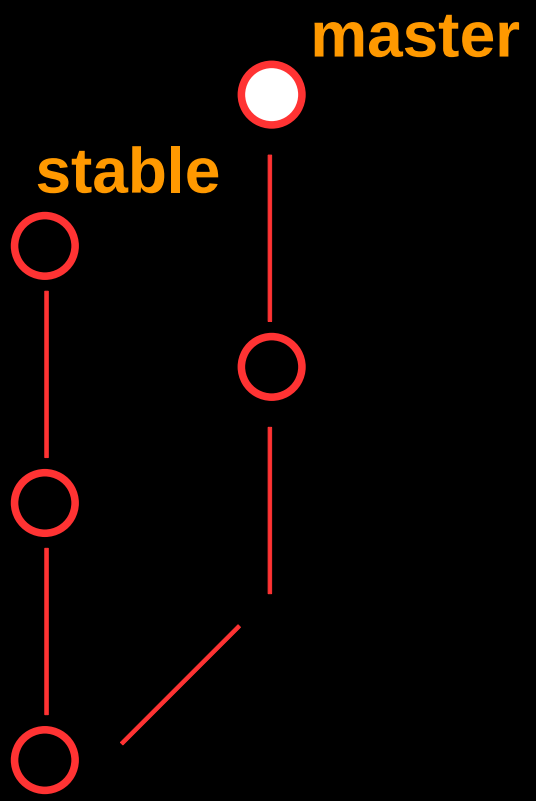
Create graph



Remove master branch

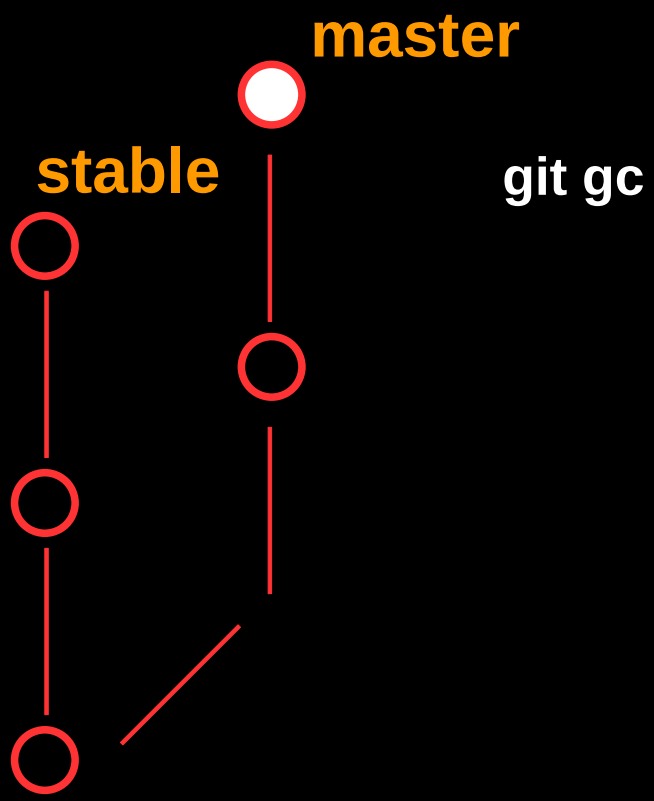


Bring back master branch




```
git fsck --no-reflogs --dangling  | grep commit | awk '{print $3}' | xargs gitk --all
```

Garbage collector




```
git -c gc.reflogExpire=0 -c gc.reflogExpireUnreachable=0 -c gc.rerereresolved=0 \
    -c gc.rerereunresolved=0 -c gc.pruneExpire=now gc
```

Switch branch

Workspace

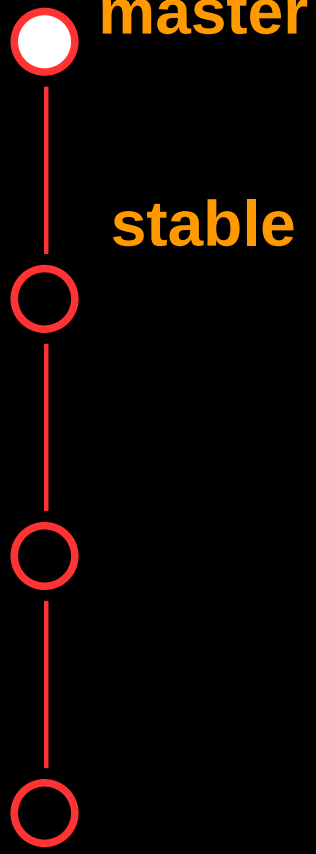
=

INDEX

=

HEAD

git checkout <branch>



Switch branch

Workspace

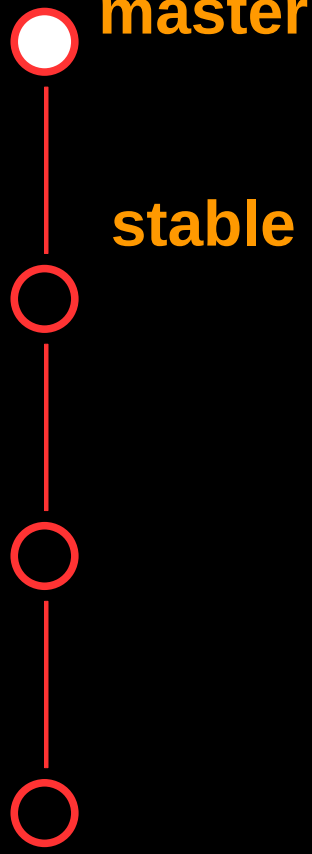
≠

INDEX

=

HEAD

git stash
git checkout <branch>
git stash pop



Switch branch

Workspace

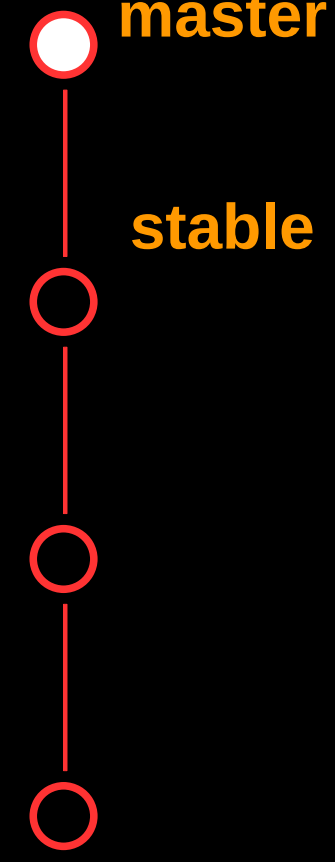
≠

INDEX

≠

HEAD

git reset --hard
git checkout <branch>



reset

commit

Workspace

≠

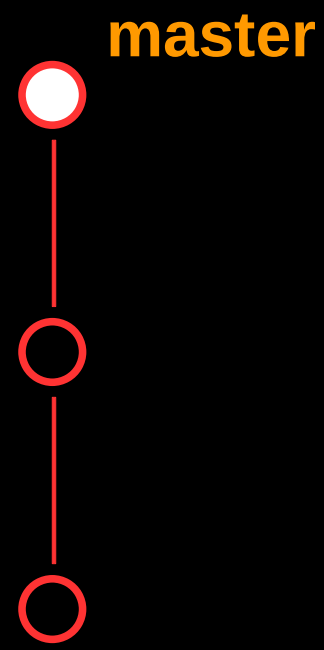
INDEX

≠

HEAD

git add

git commit



commit

Workspace

≠

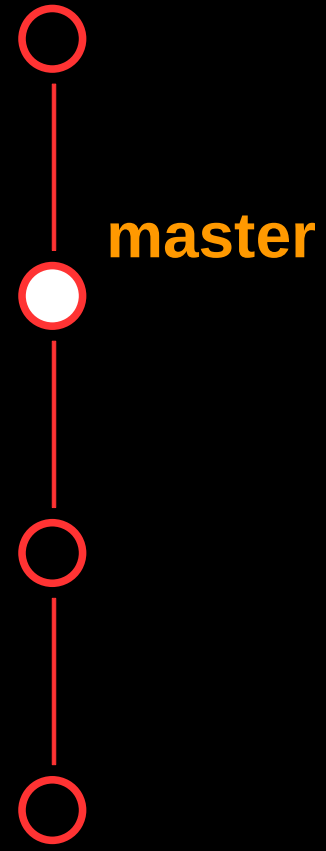
INDEX

≠

HEAD

git add

git commit



commit

Workspace

≠

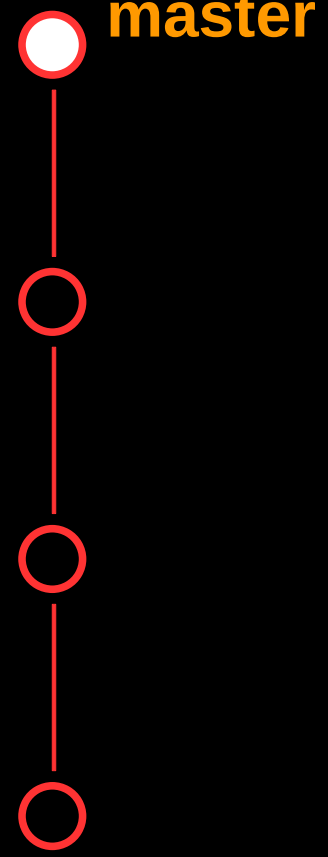
INDEX

=

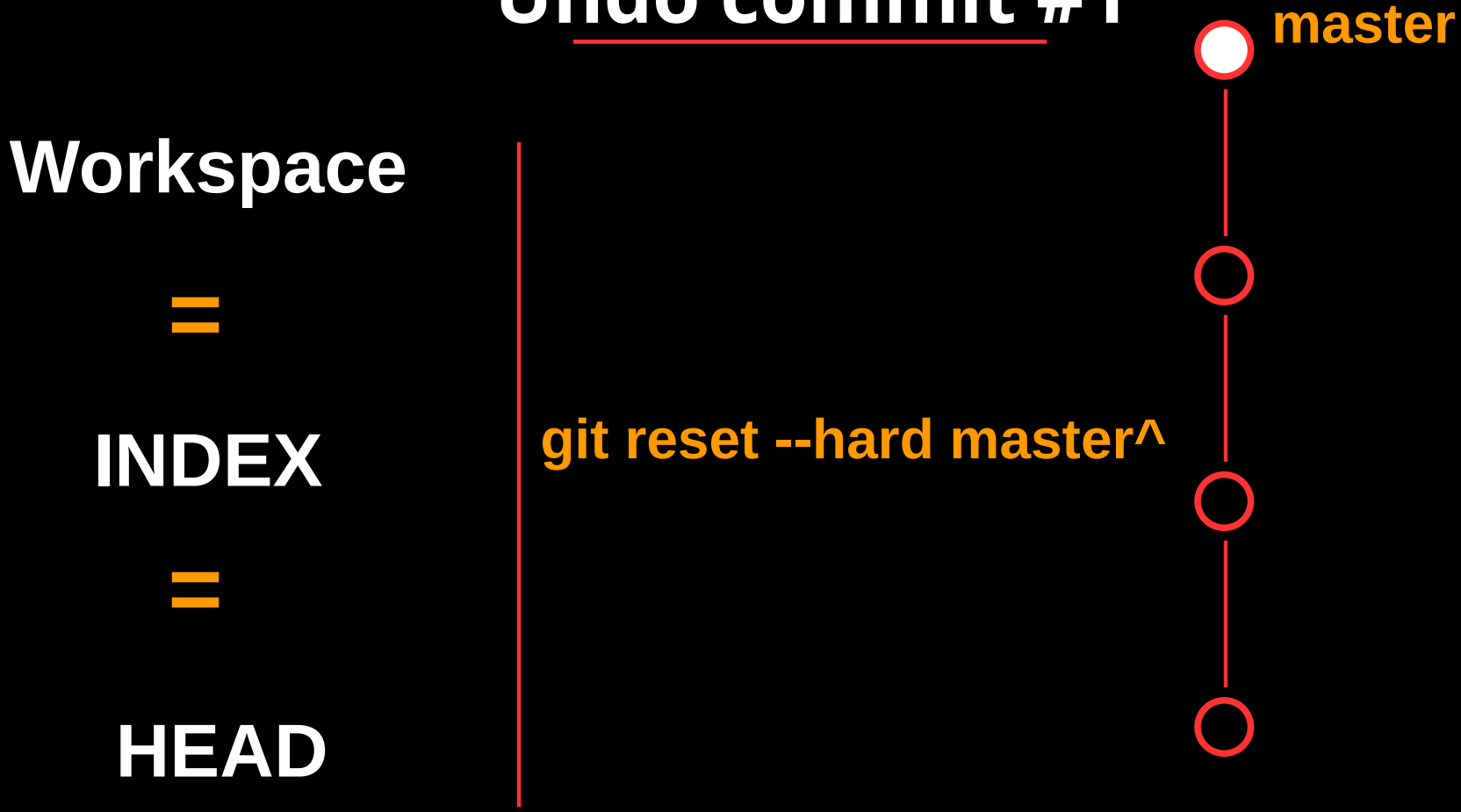
HEAD

git add

git commit



Undo commit #1



Undo commit #2

Workspace

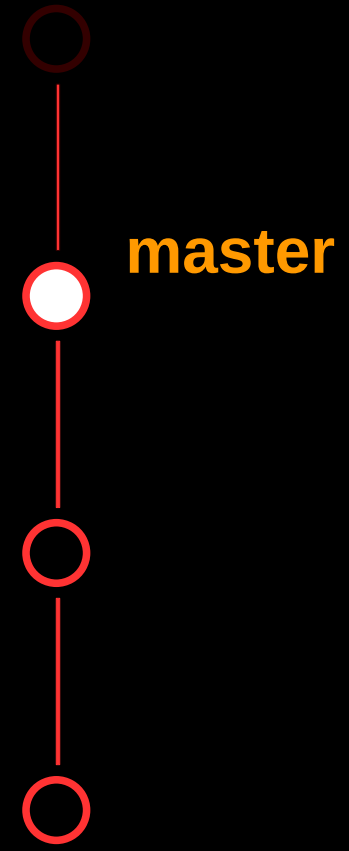
=

INDEX

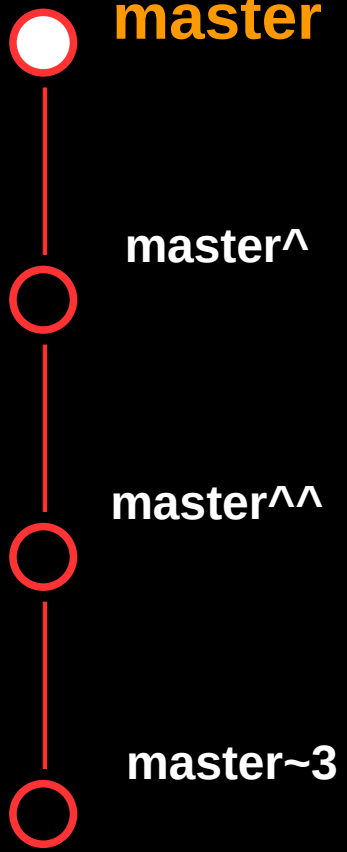
=

HEAD

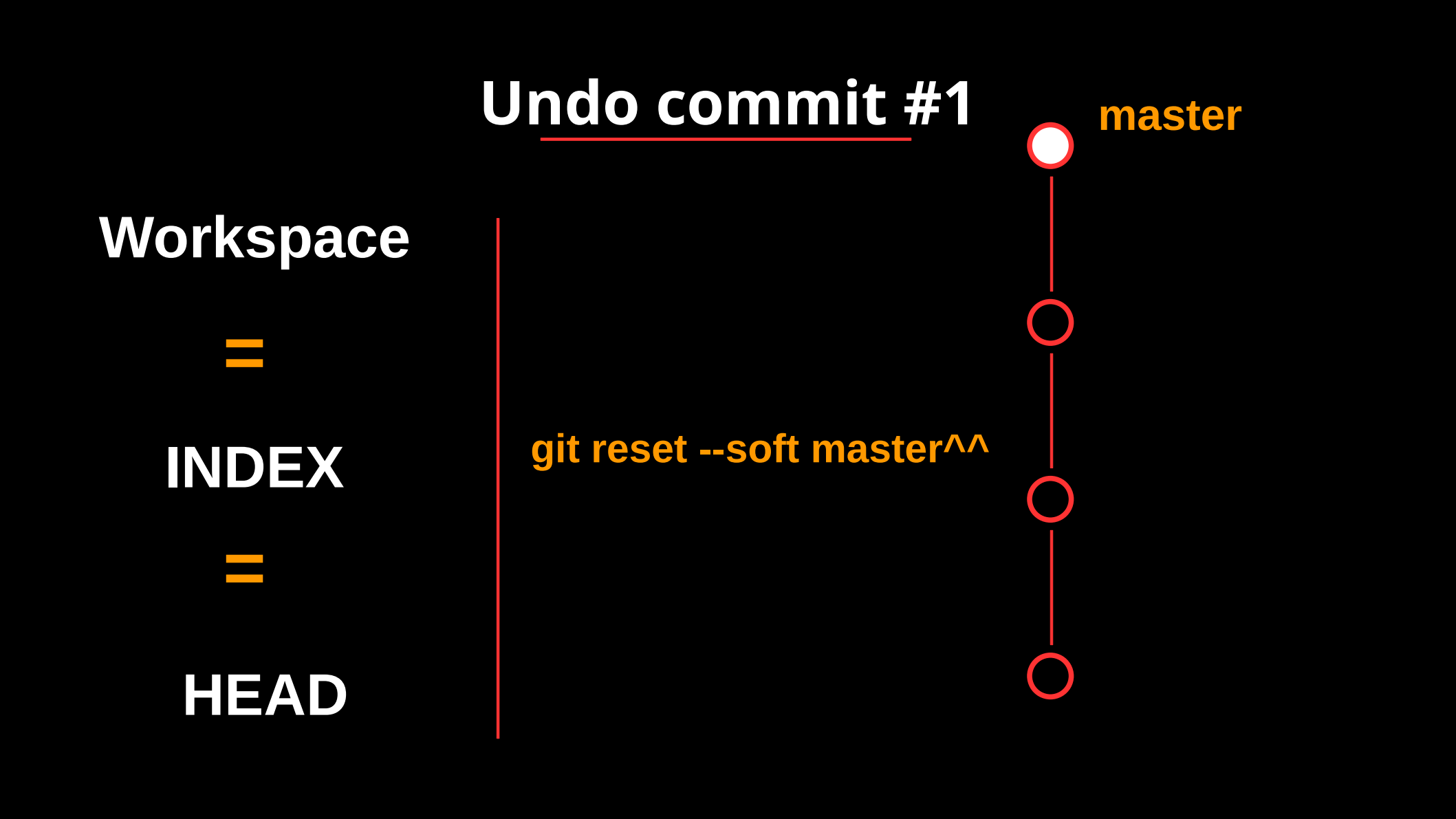
git reset --hard master^



Commit naming



Undo commit #1



Undo commit #2

Workspace

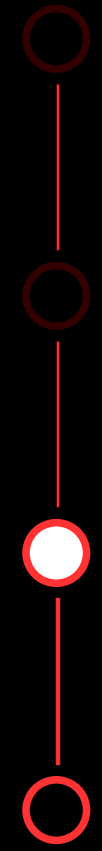
=

INDEX

≠

HEAD

git reset --soft master^^



master

Join commits #3

Workspace

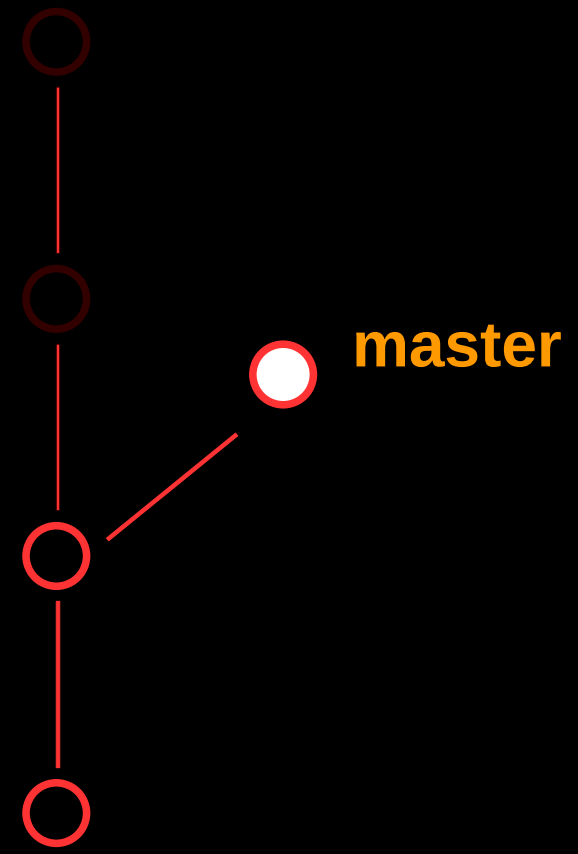
=

INDEX

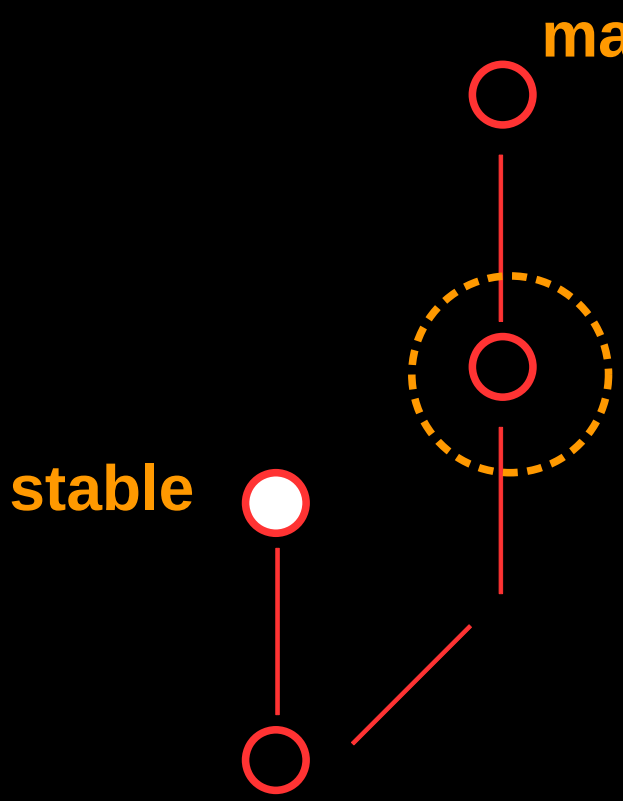
=

HEAD

git reset --soft master^^
git commit -a

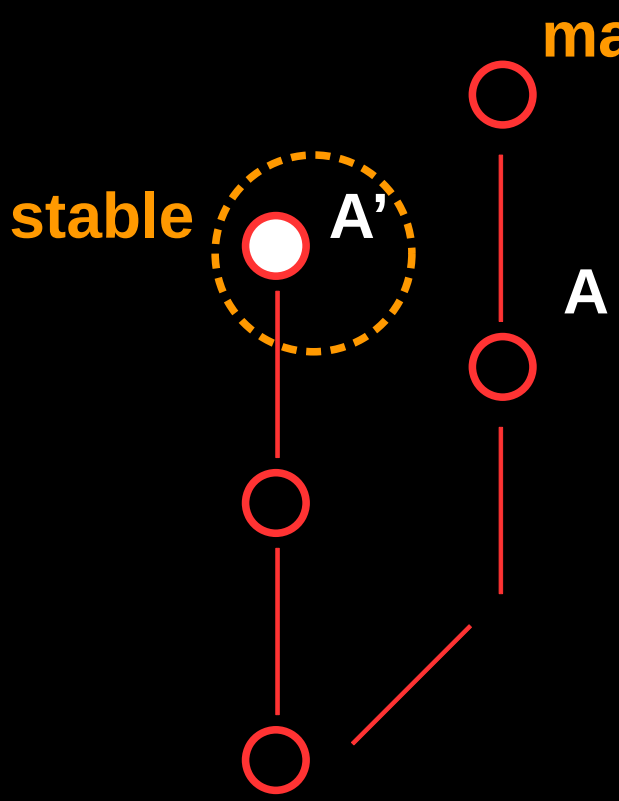


cherry-pick #1



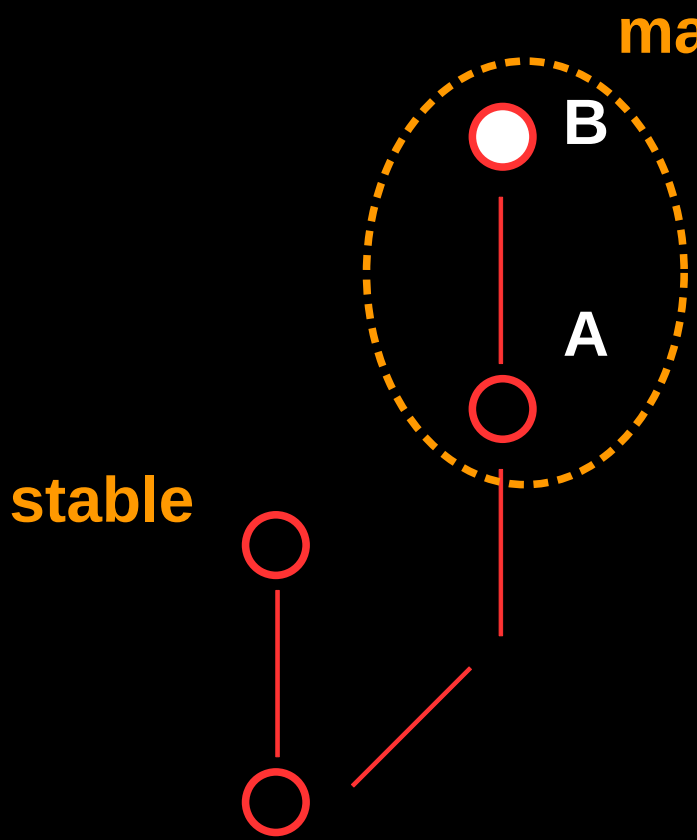
git cherry-pick master^

cherry-pick #2



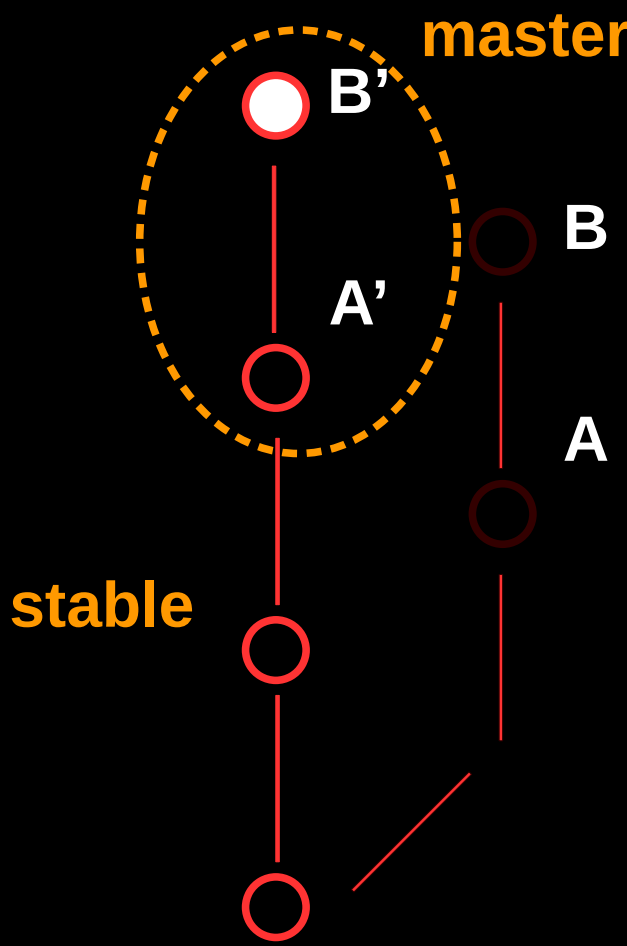
git cherry-pick master^

rebase #1



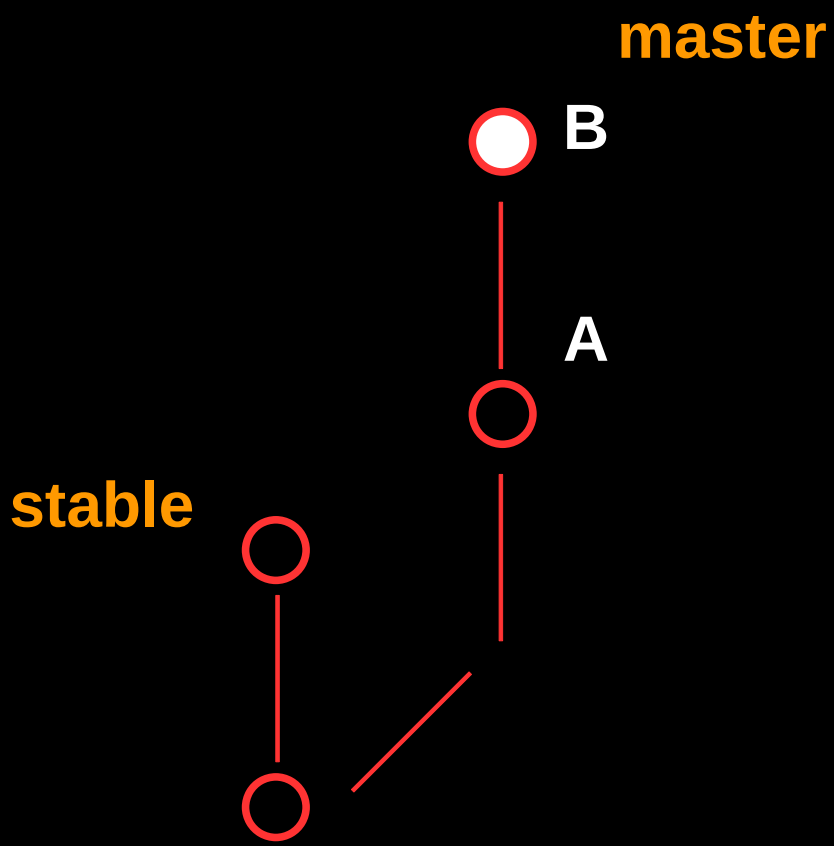
git rebase stable

rebase #2



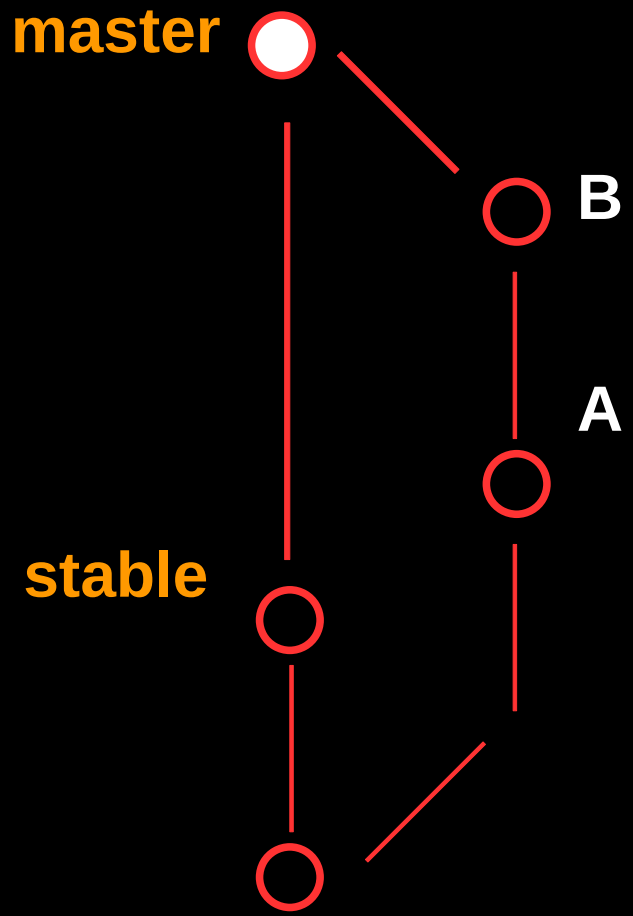
git rebase stable

merge #1



git merge stable

merge #2



git merge stable


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.

Limbo state?

```
git rebase --abort
git cherry-pick --abort
git reset --hard
gitk --all
```

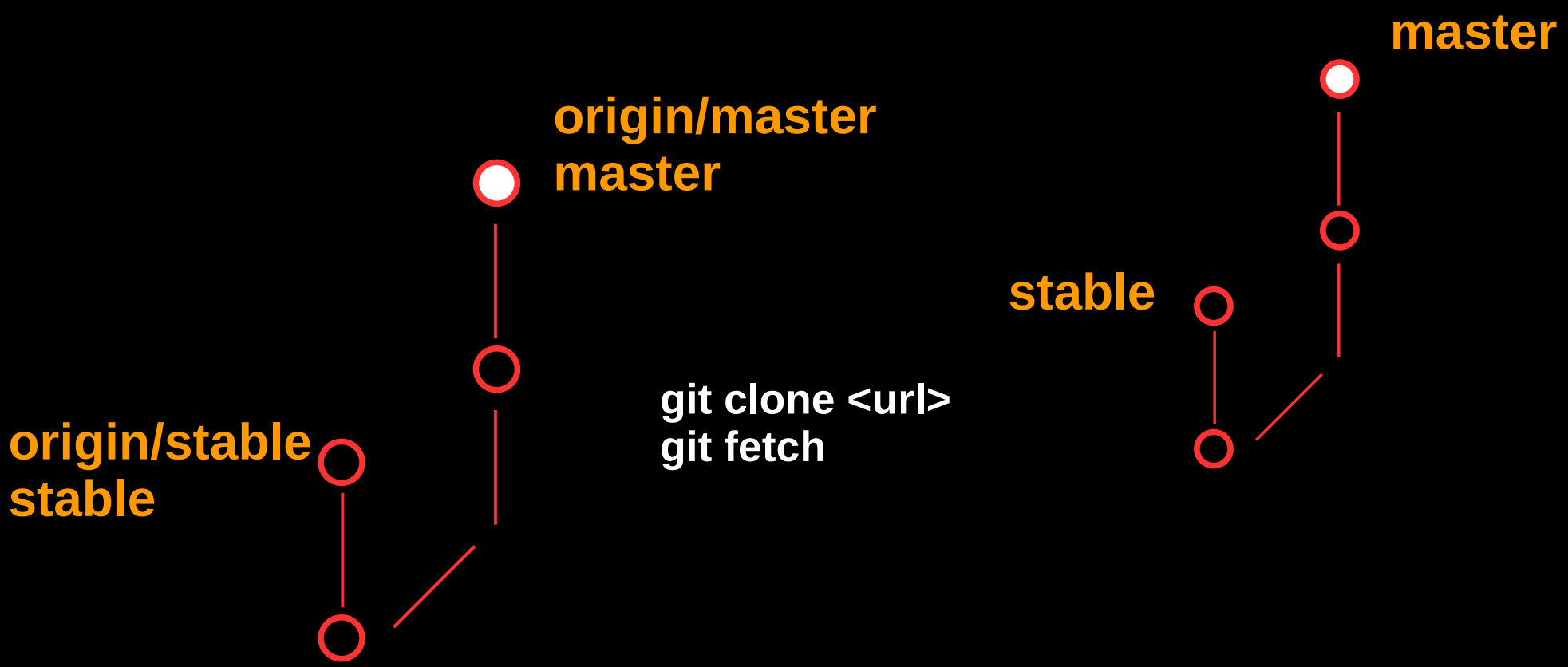
Limbo state?

Creating GitHub account

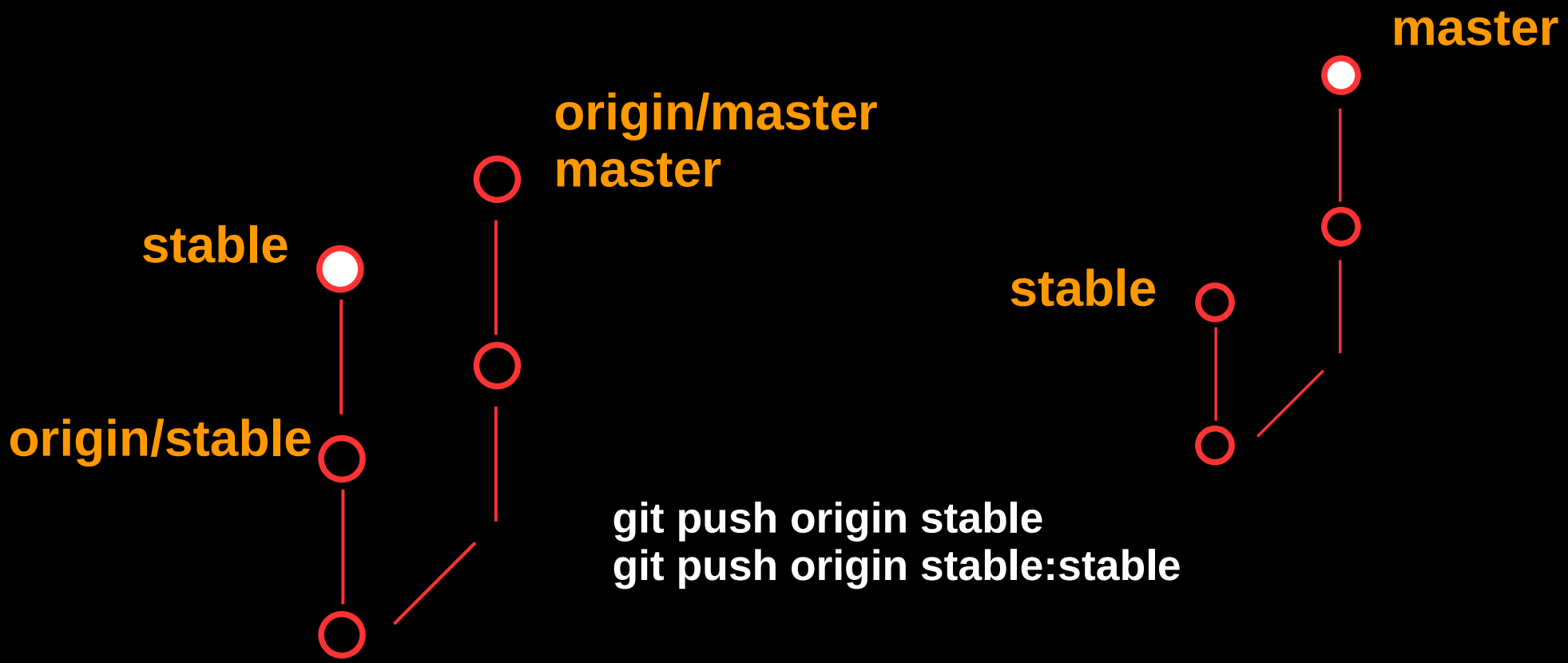
~/.netrc (on Windows _netrc)

machine github.com login sentinel@atteo.com password <password>

clone

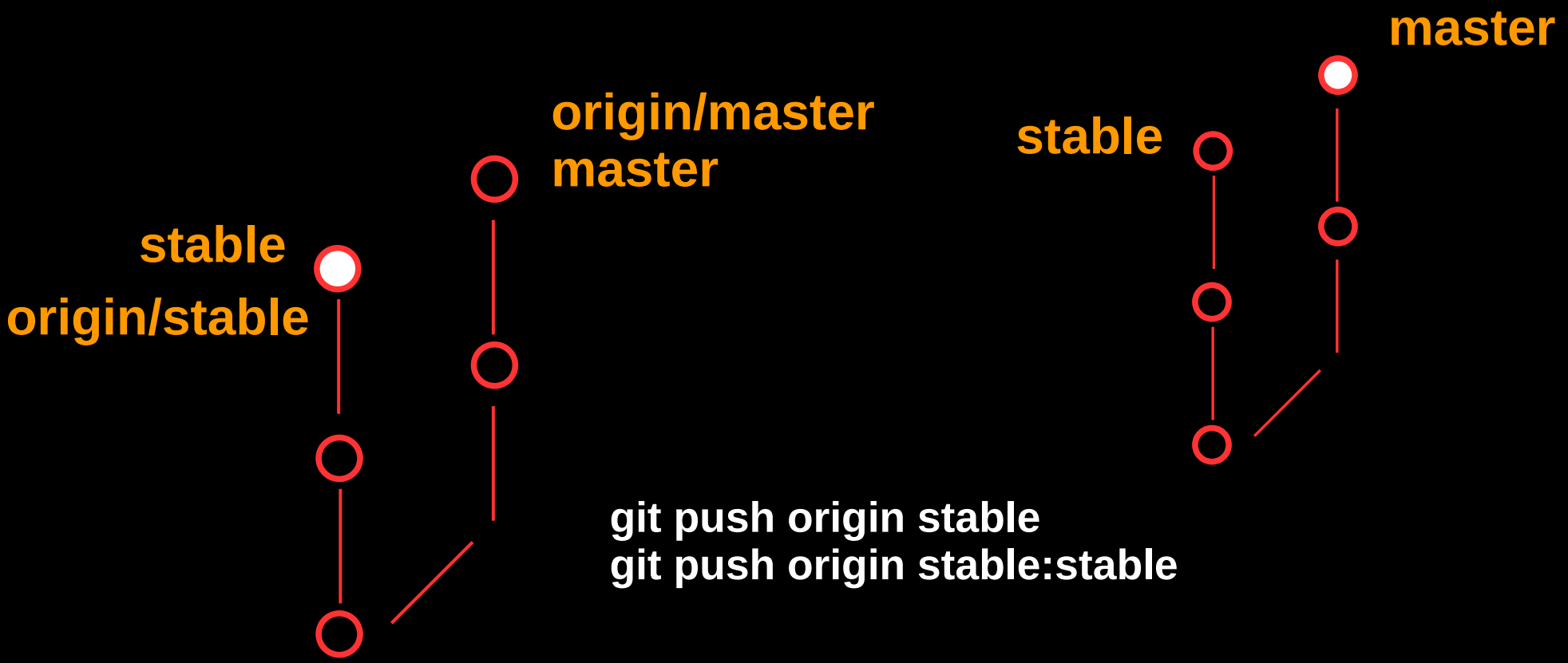

origin/master
master
origin/stable
stable
git clone <url>
git fetch
master
stable

push #1



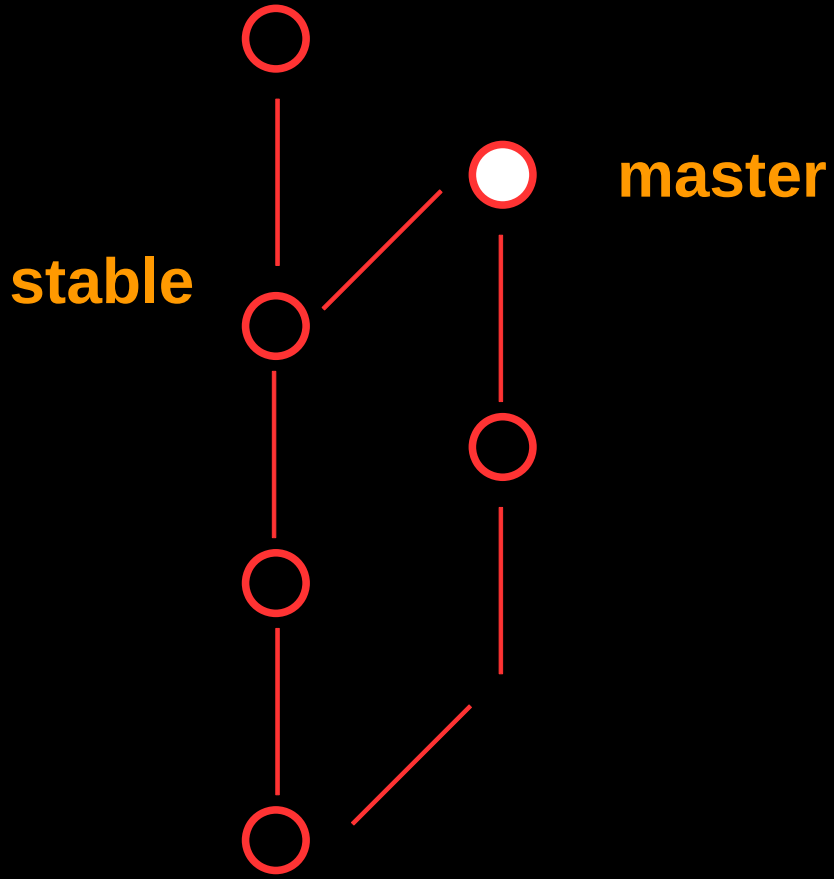
git push origin stable
git push origin stable:stable

push #2

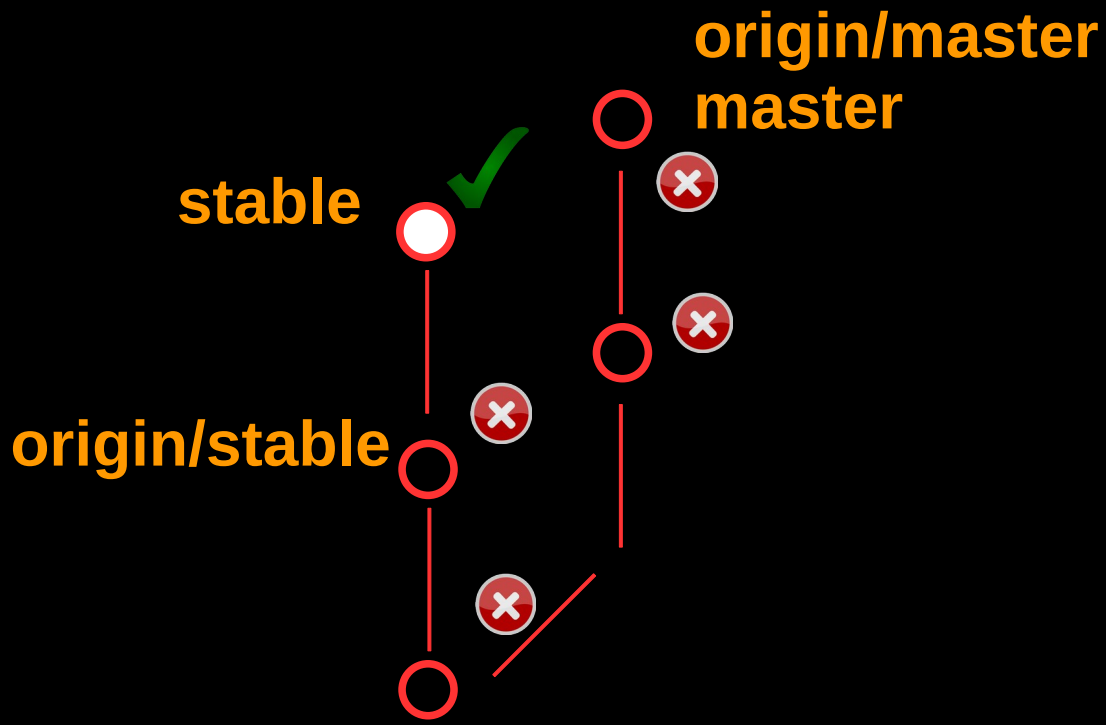


git push origin stable
git push origin stable:stable

Branching model



History rewriting



git rebase
git rebase -i
git cherry-pick
git push -f

Tools

git rebase -i

git bisect

git gui blame

git filter-branch

git grep

git worktree

git log --all --graph –decorate

git reflog

Questions/Discussions