



# git

Software Engineering II  
WS 2019/20

Enterprise Platform and Integration Concepts

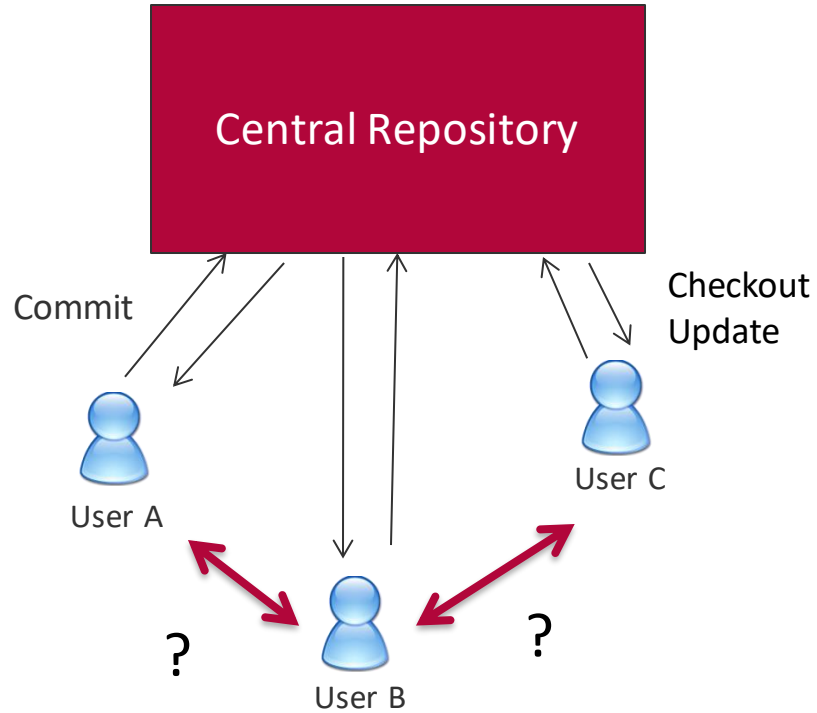
# Outline

1. Basics
2. Local
3. Collaboration

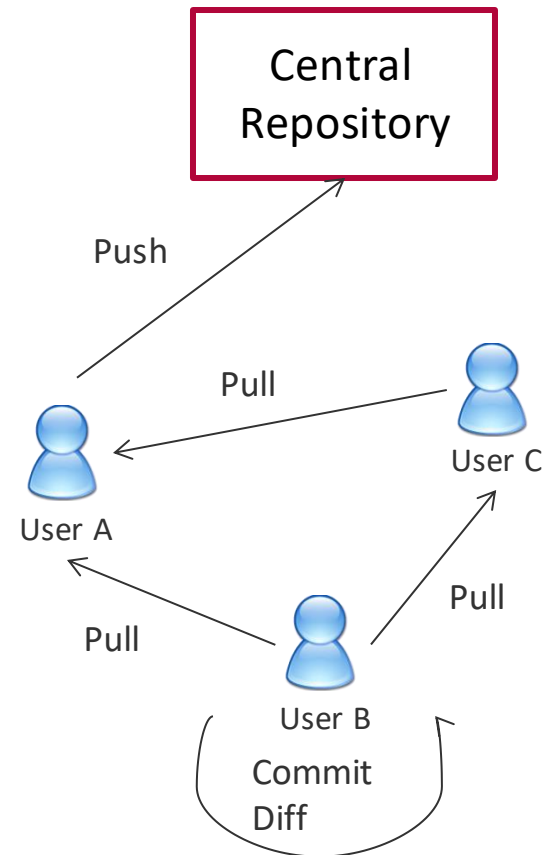


<https://xkcd.com/1597/>

# Centralized vs Distributed VCS



VS.



# Centralized vs Distributed VCS



- Distributed VCS are mostly used like centralized VCS
  - Same features (branches, tags, merging)
- Local commits are a blessing and a curse
  - Commits can be made while offline
  - Higher chances of code diverging
- Pull-Requests are better than patch files

# Git Objects



## Blob

- Content of a file
- Nothing else

## Tree

- File structure
- References Blobs

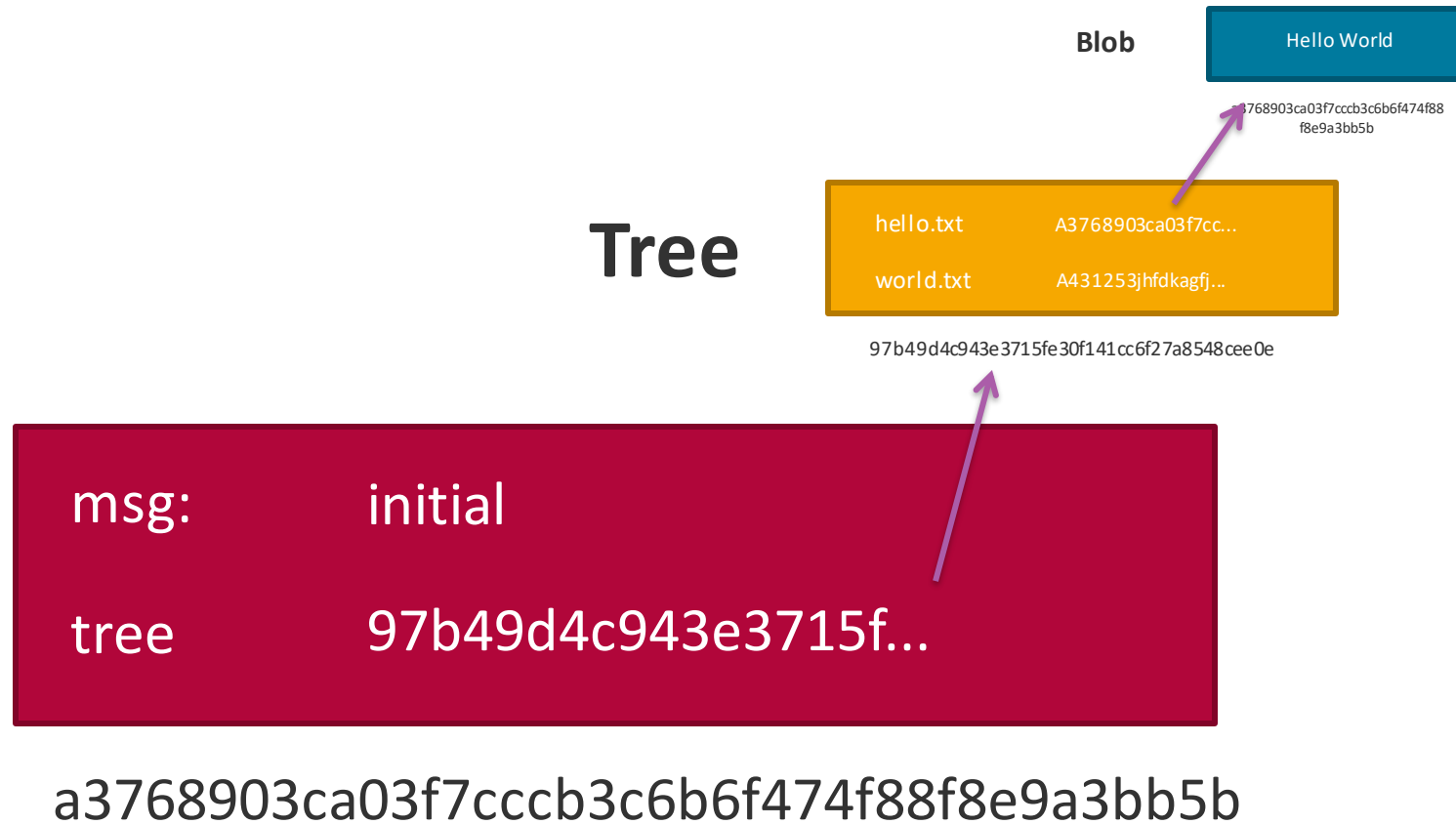
## Commit

- References Tree object
- Metadata
- 0..\* parent commits

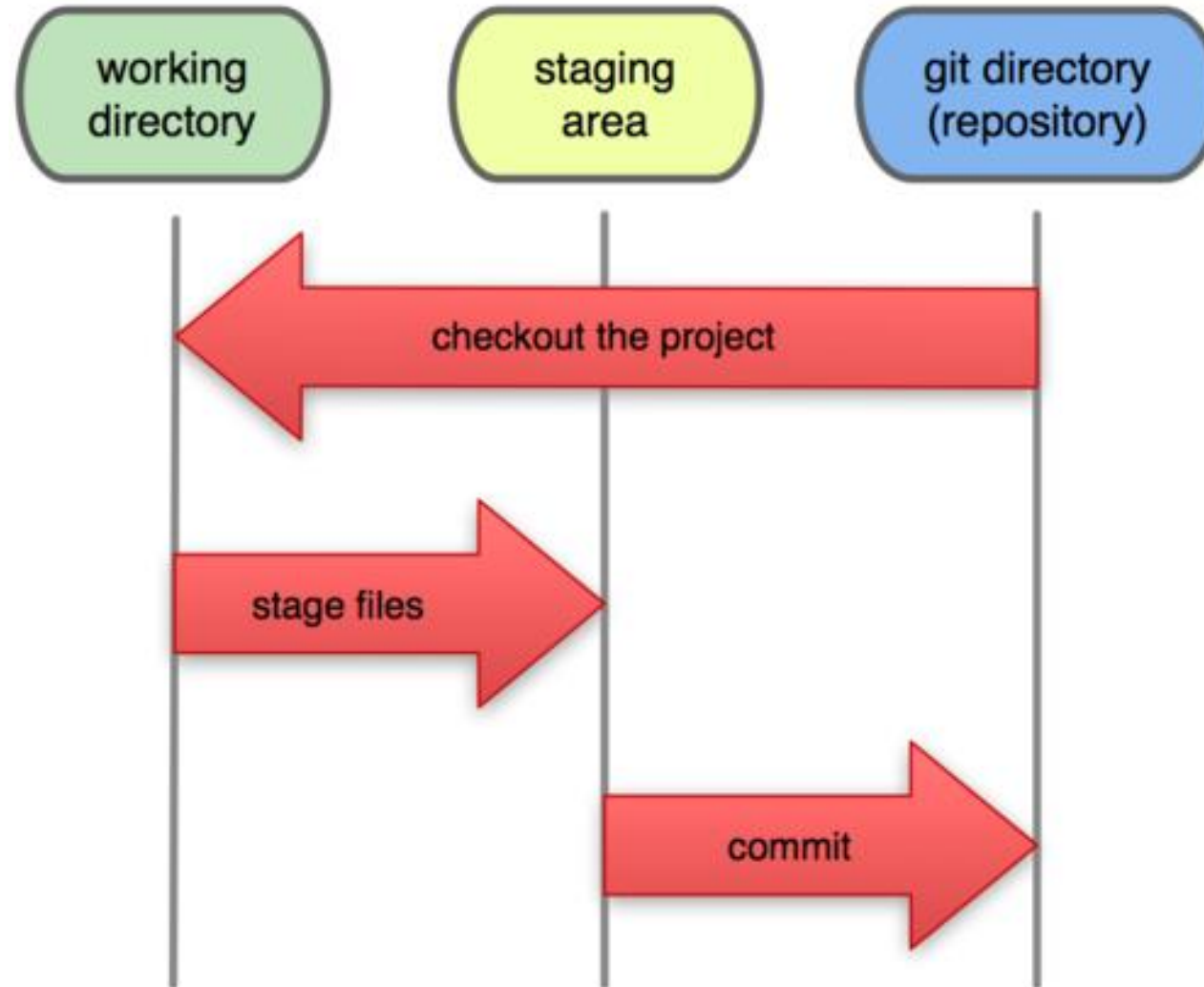
## Tag

- Reference to other object

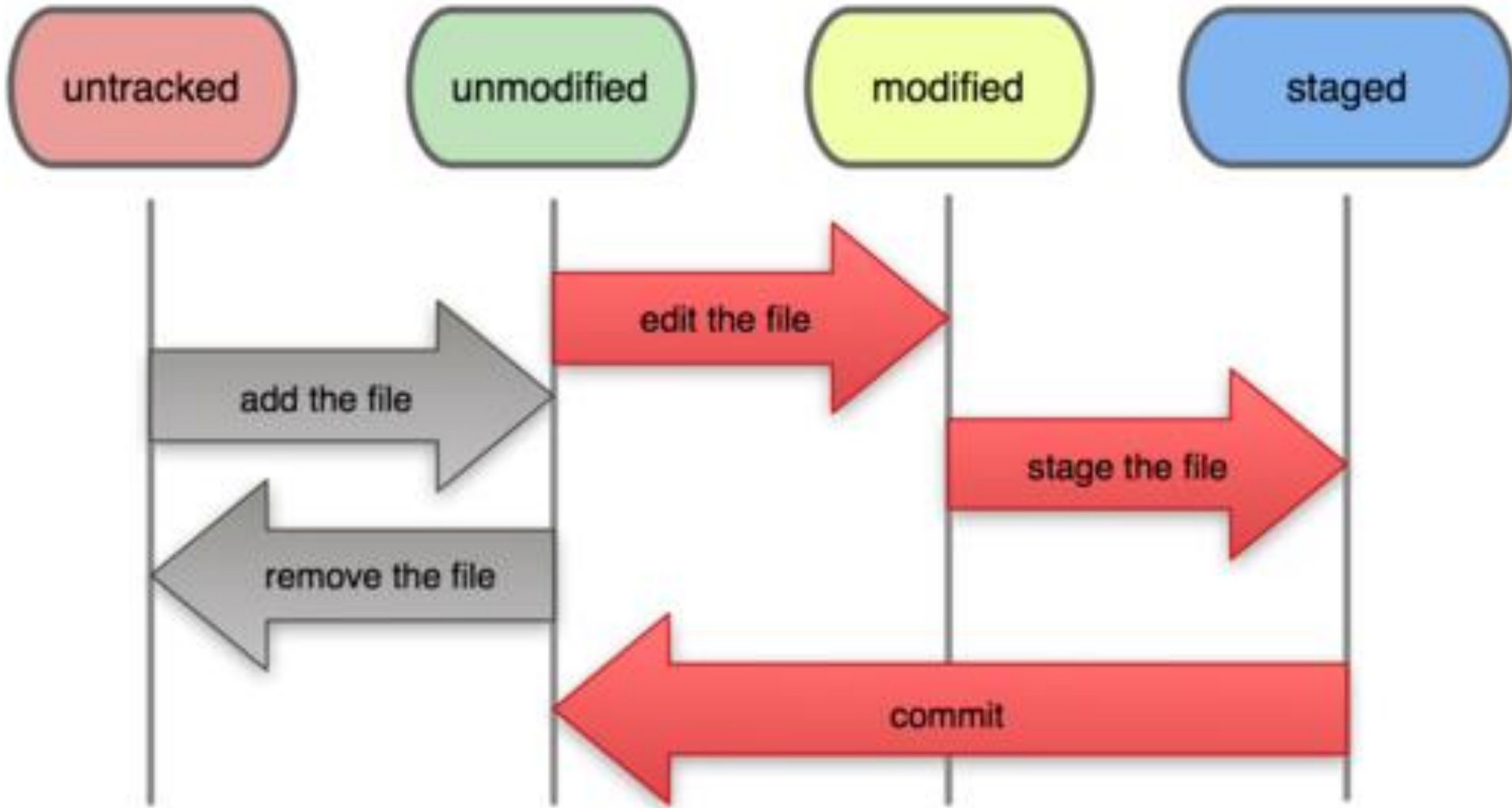
# Commit



# Local Operations

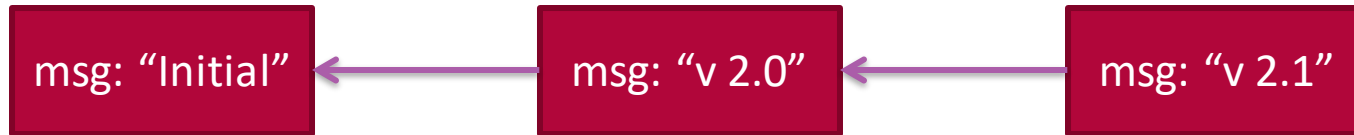


# File Status Lifecycle



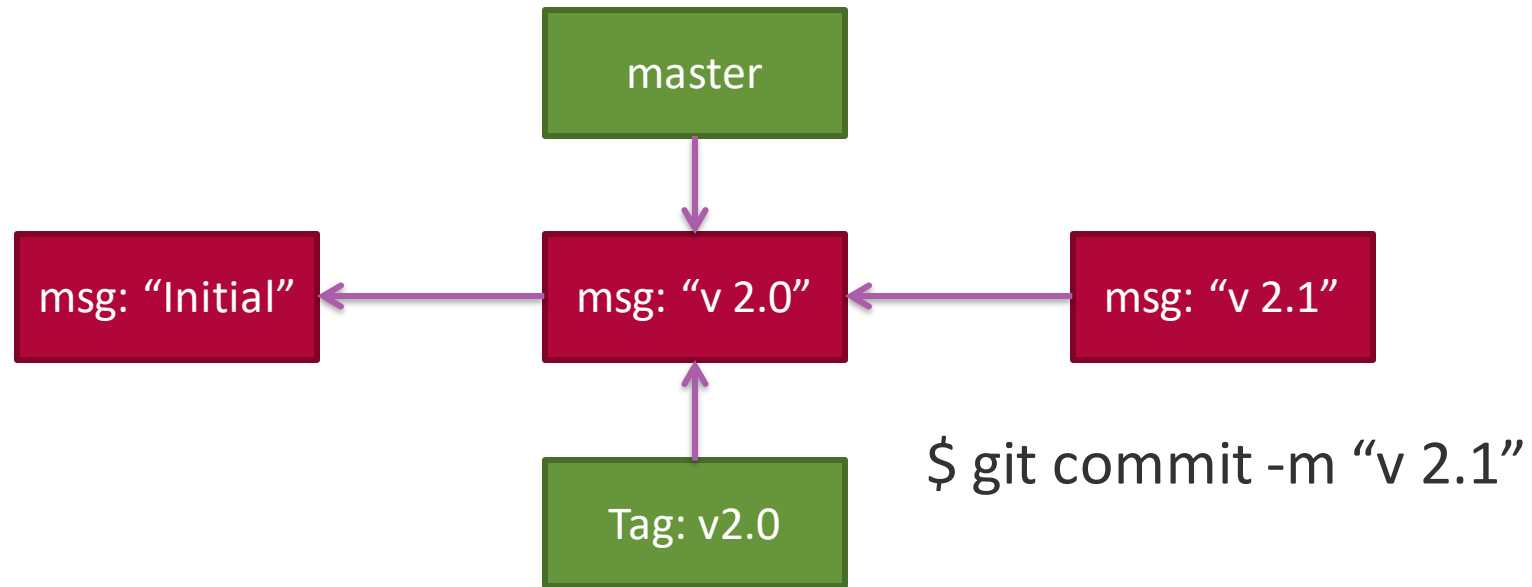


# Commit Parent

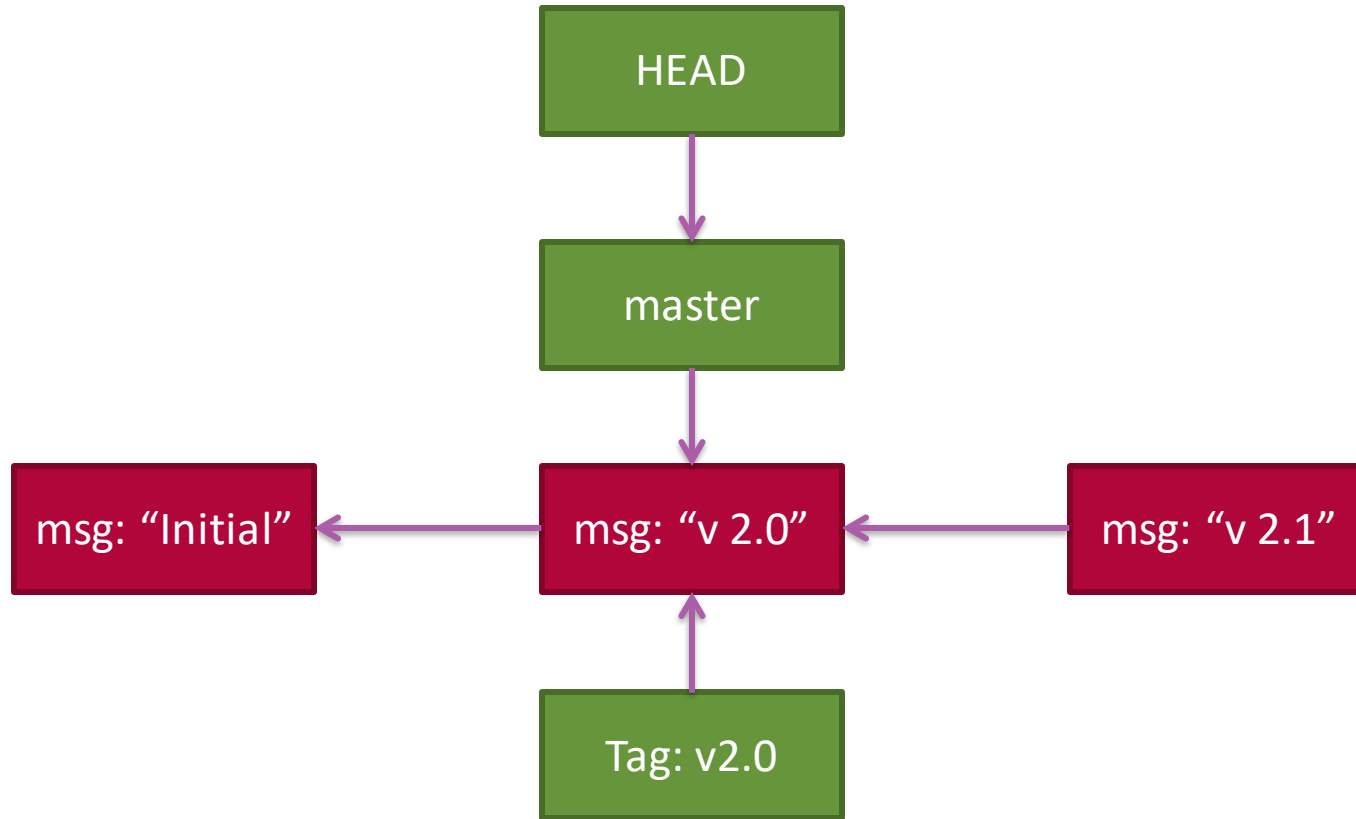


`$ git commit -m "v 2.1"`

# Branches & Tags



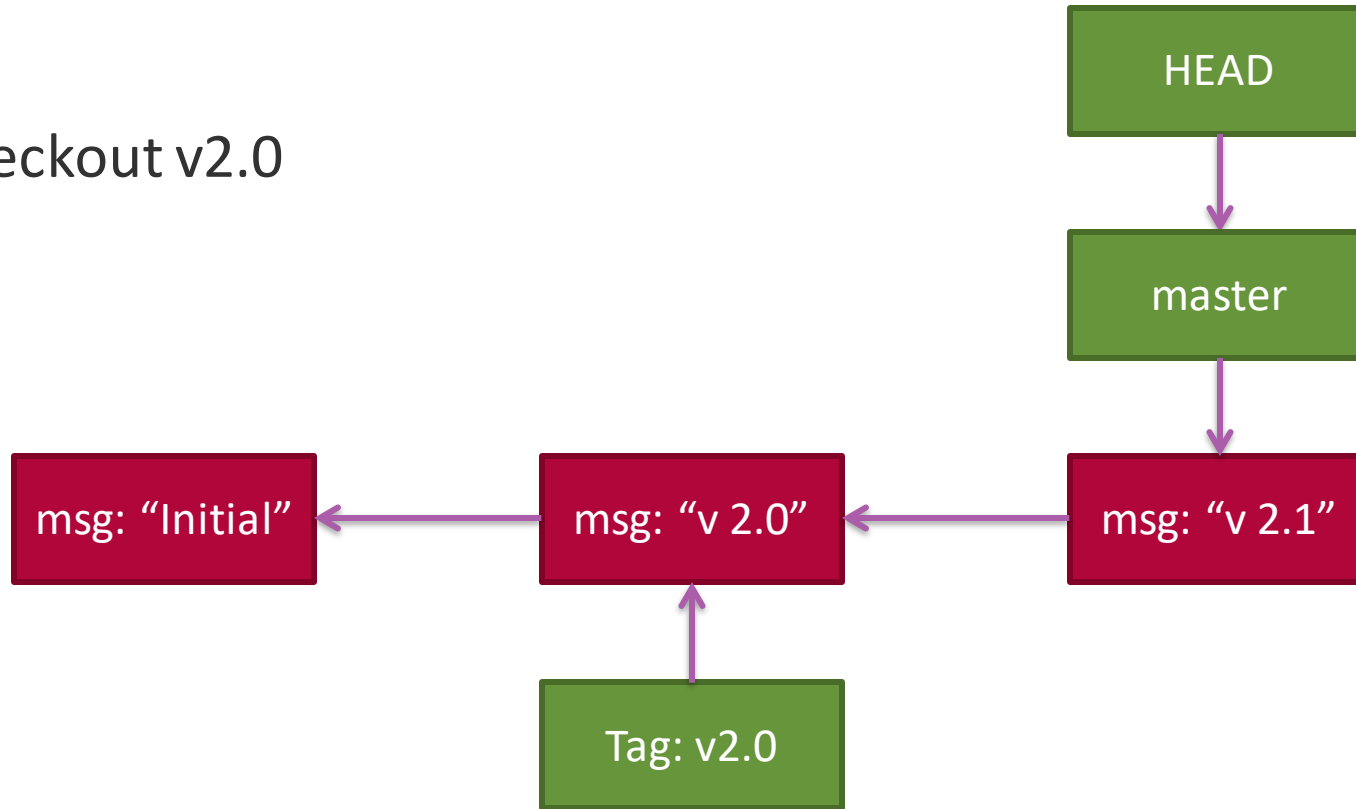
# Head



# Detached Head



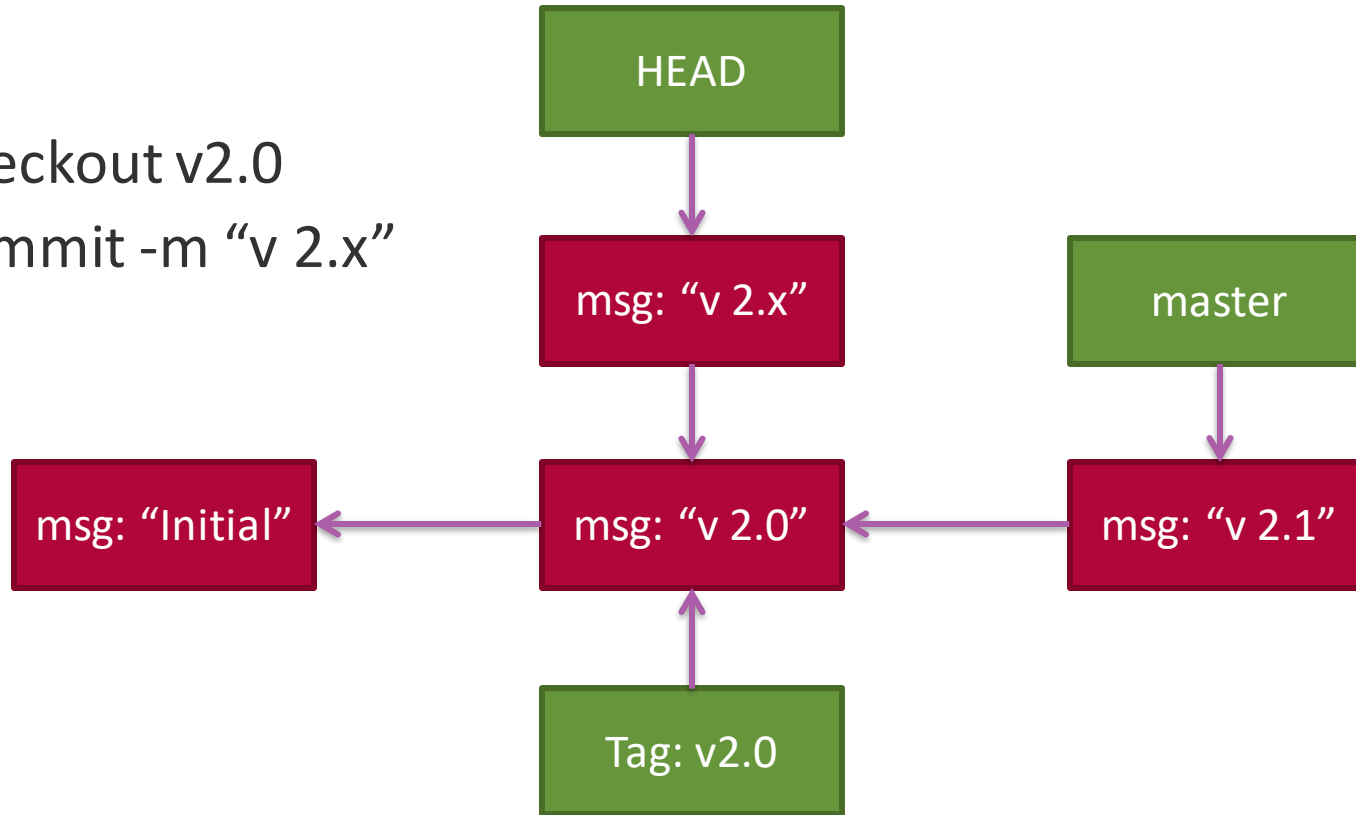
\$ git checkout v2.0



# Detached Head



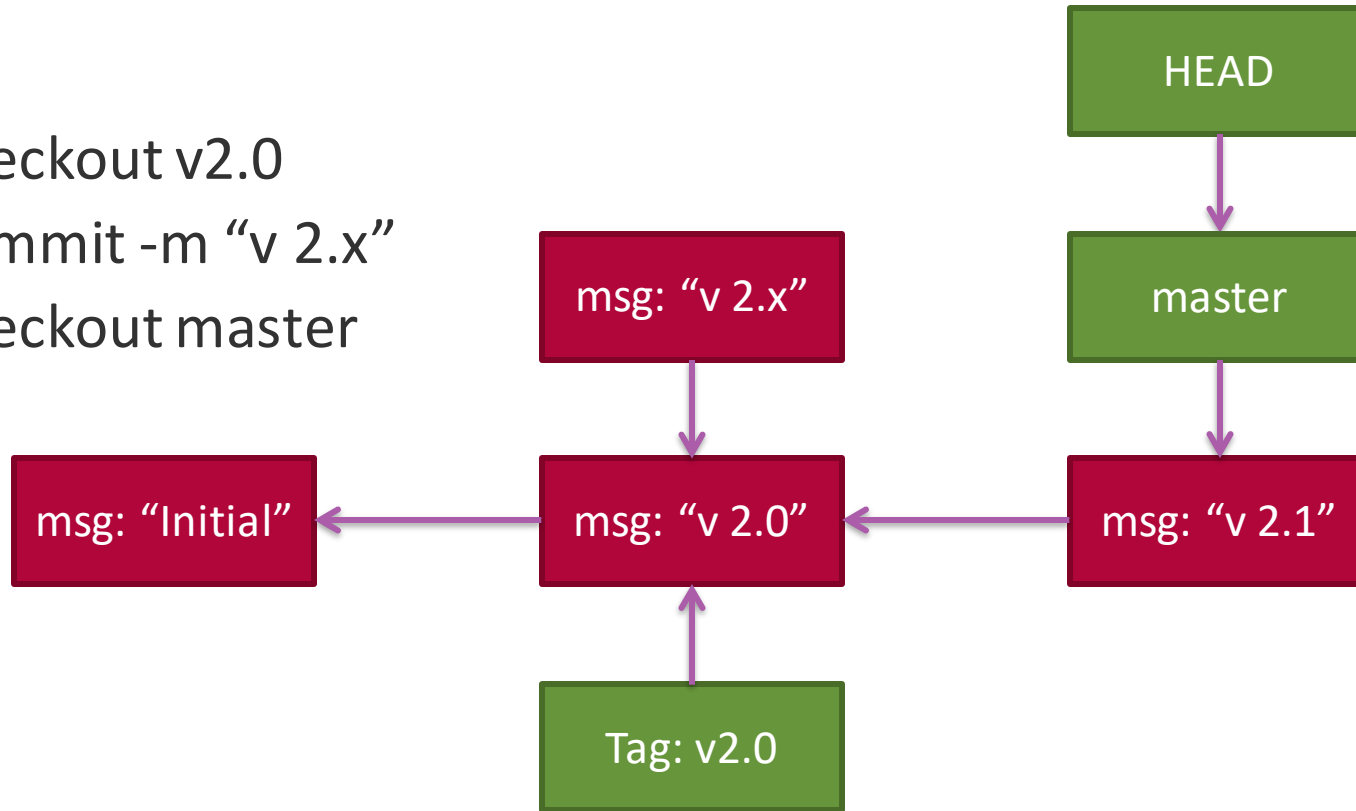
```
$ git checkout v2.0  
$ git commit -m "v 2.x"
```



# Detached Head



```
$ git checkout v2.0  
$ git commit -m "v 2.x"  
$ git checkout master
```

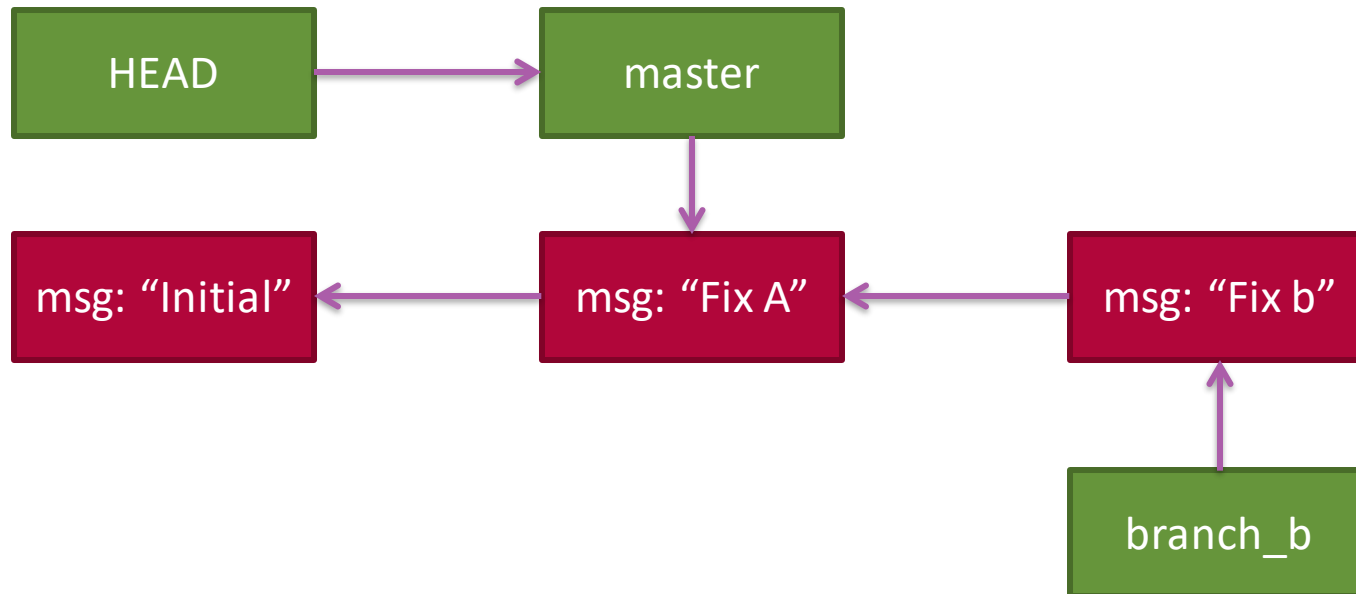


**Tip:**  
List all commits:  
git reflog

# Fast-forward



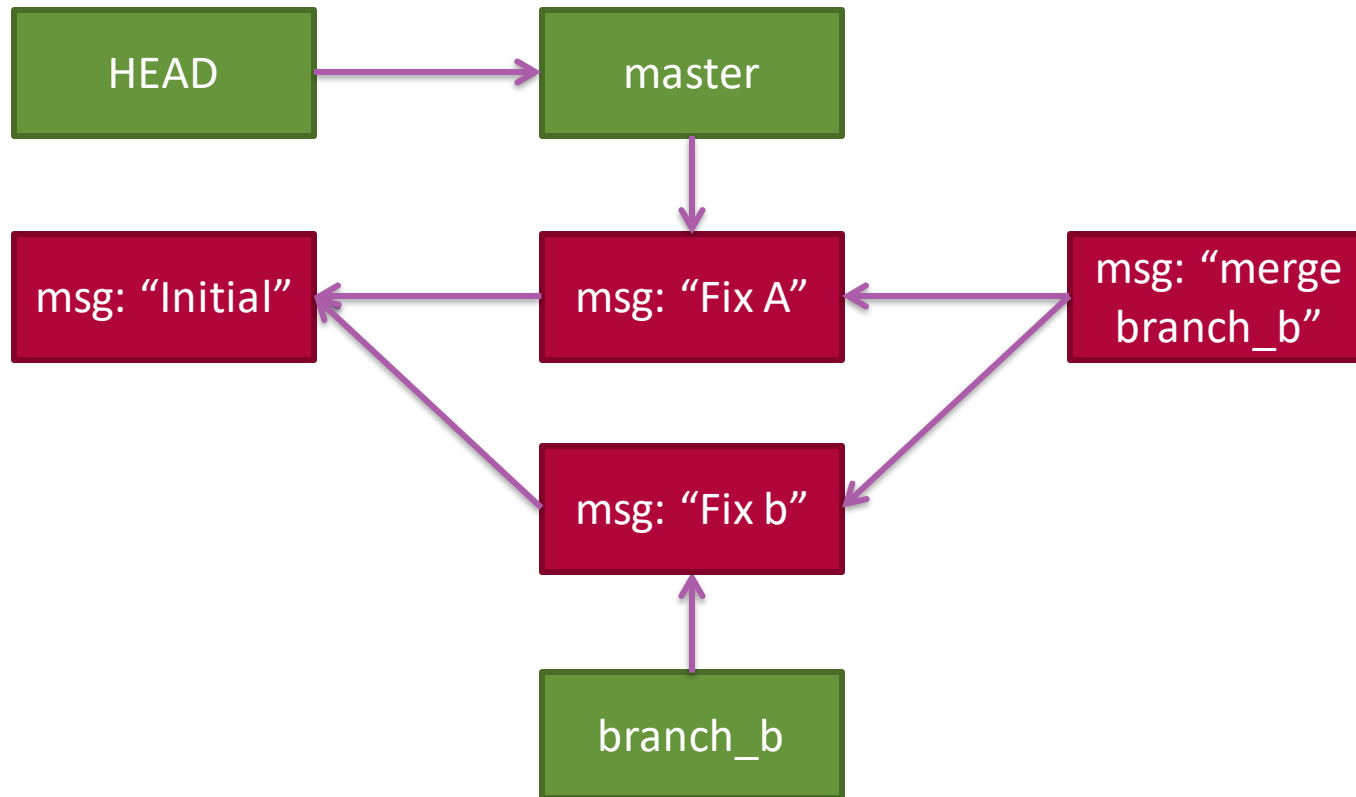
\$ git merge branch\_b



# Merge

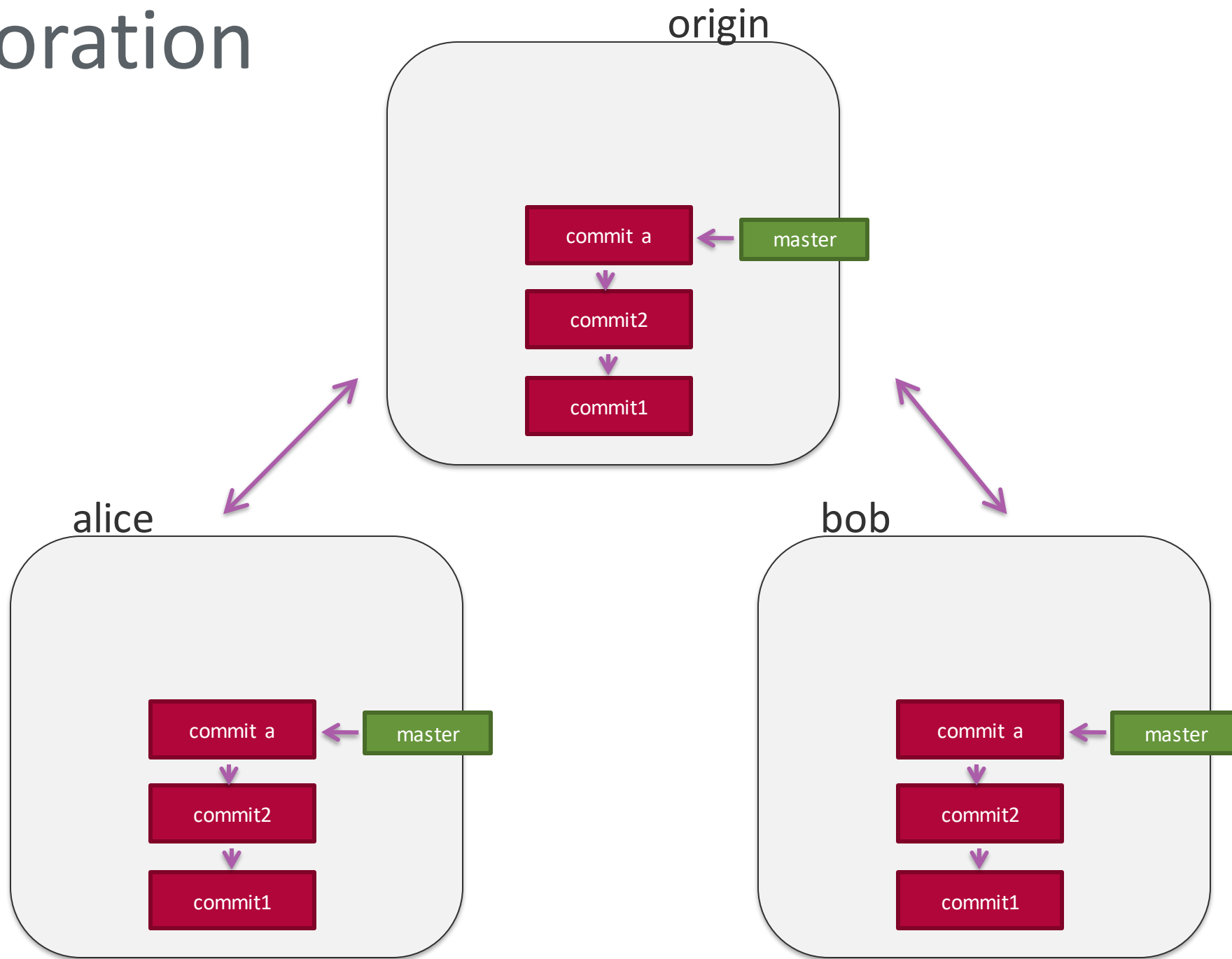


\$ git merge branch\_b





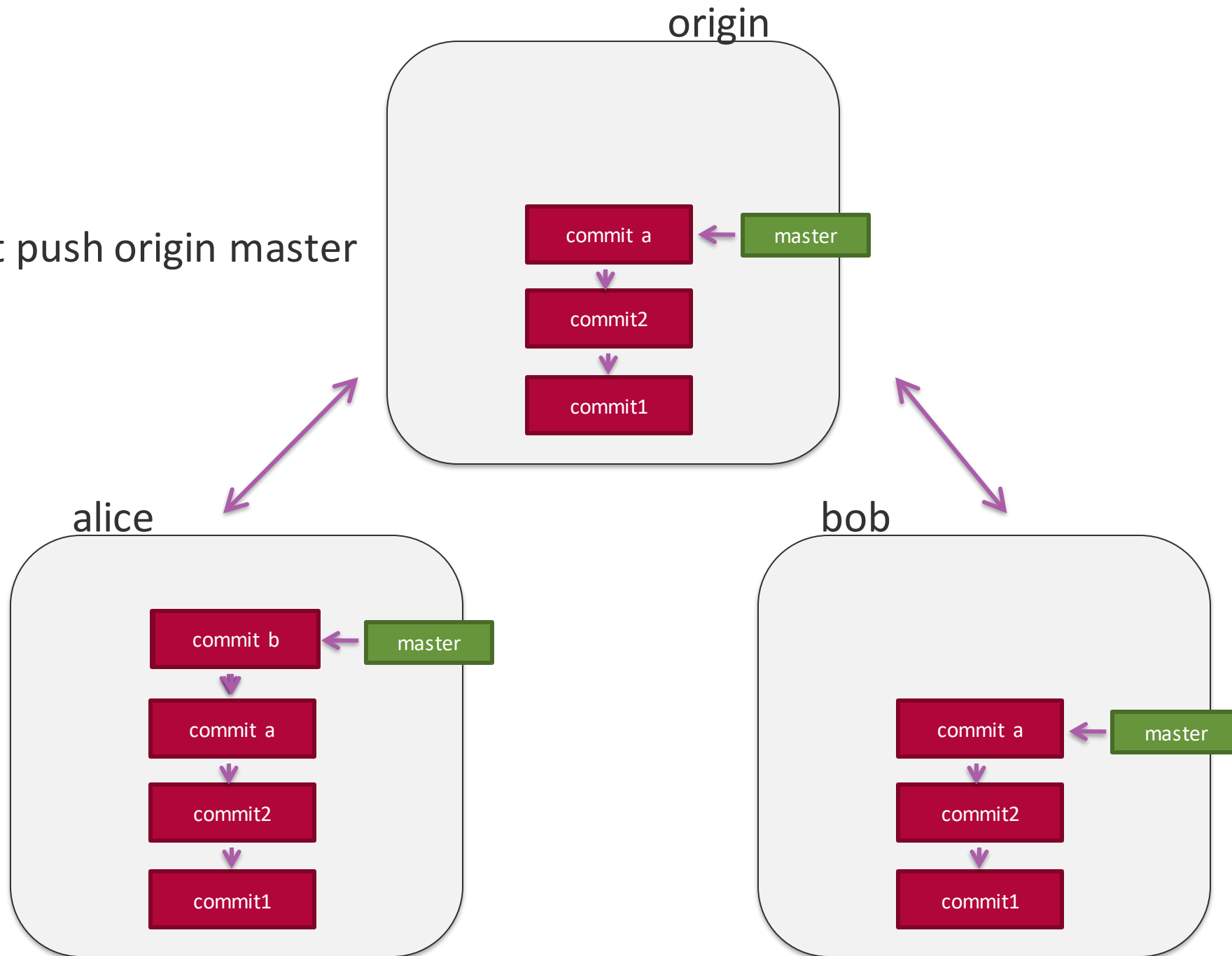
# Collaboration



# Push



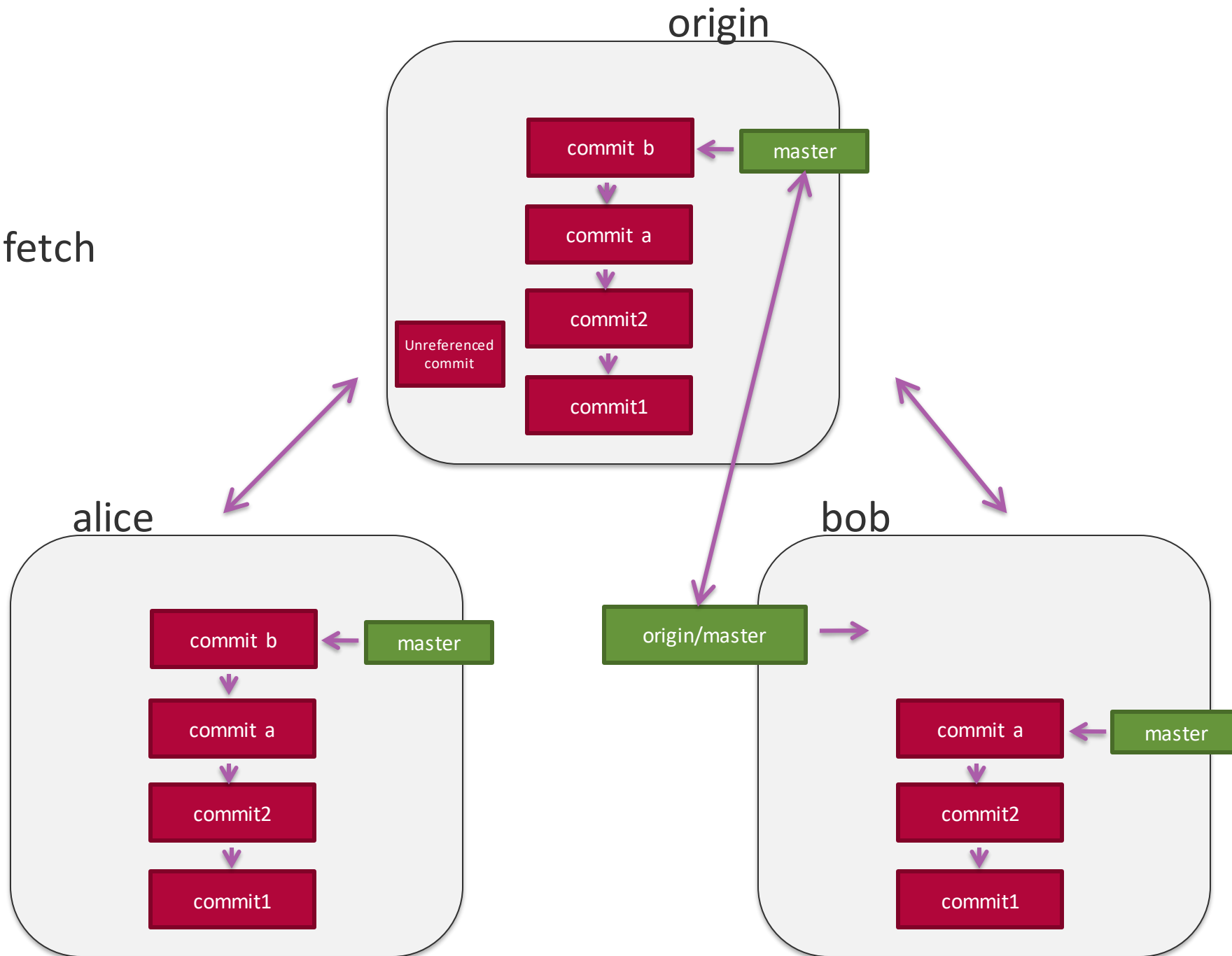
alice\$ git push origin master



# Fetch



bob\$ git fetch

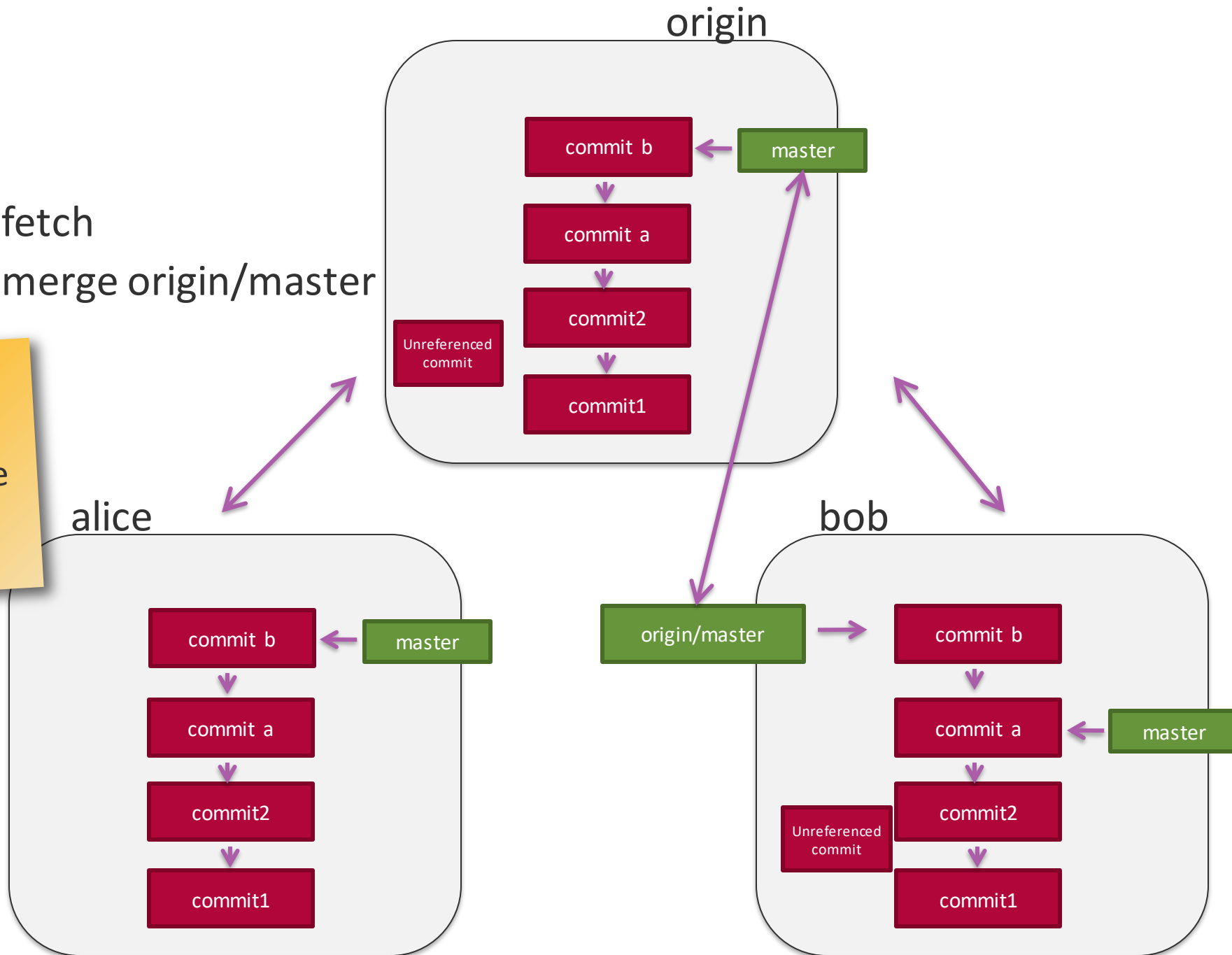


# Pull



```
bob$ git fetch
bob$ git merge origin/master
```

**Tip:**  
fetch + merge  
= pull

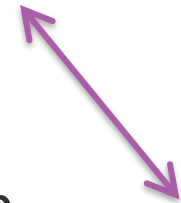
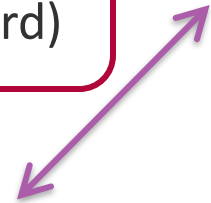
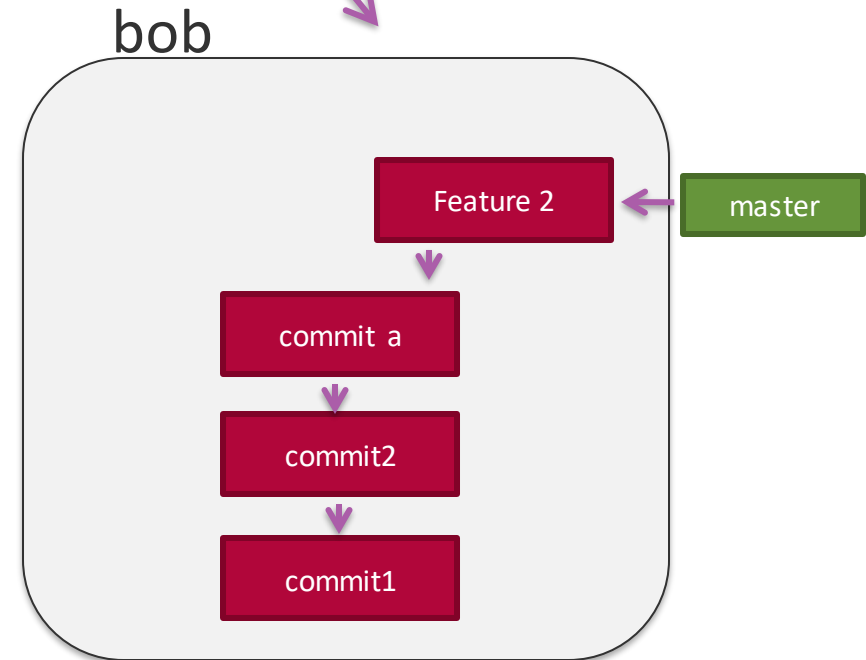
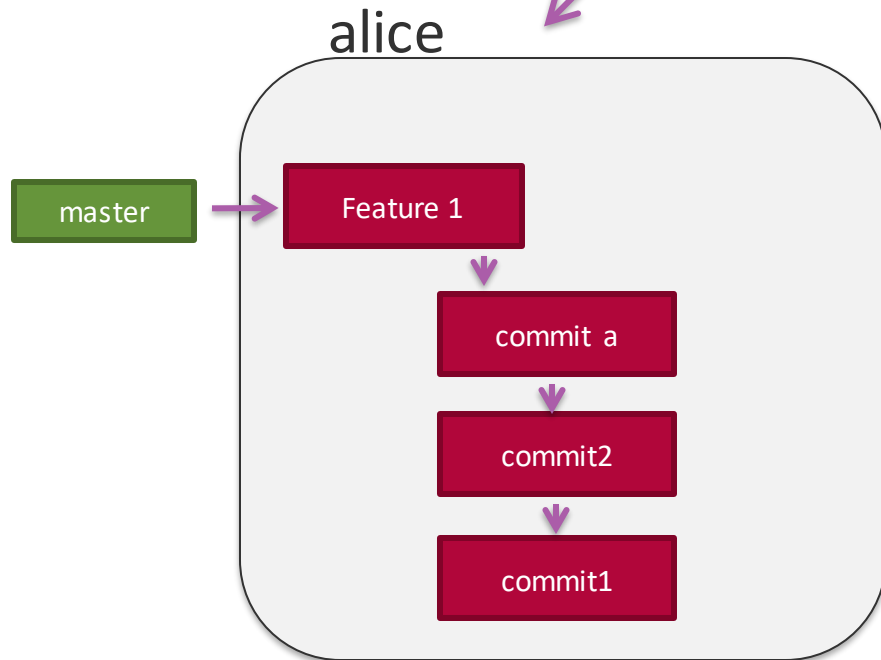
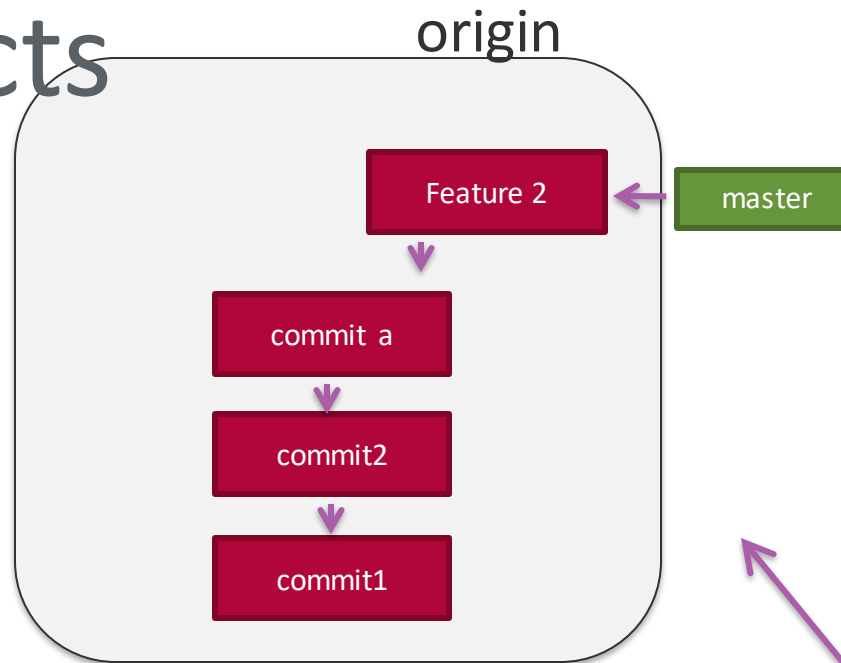


# Push with Conflicts



alice\$ git push origin master

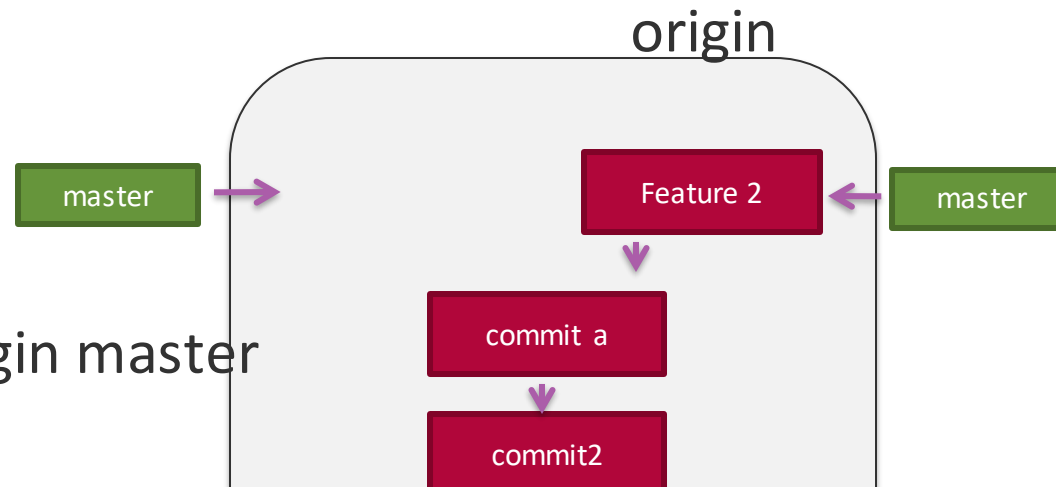
! [rejected] master -> master (non-fast-forward)



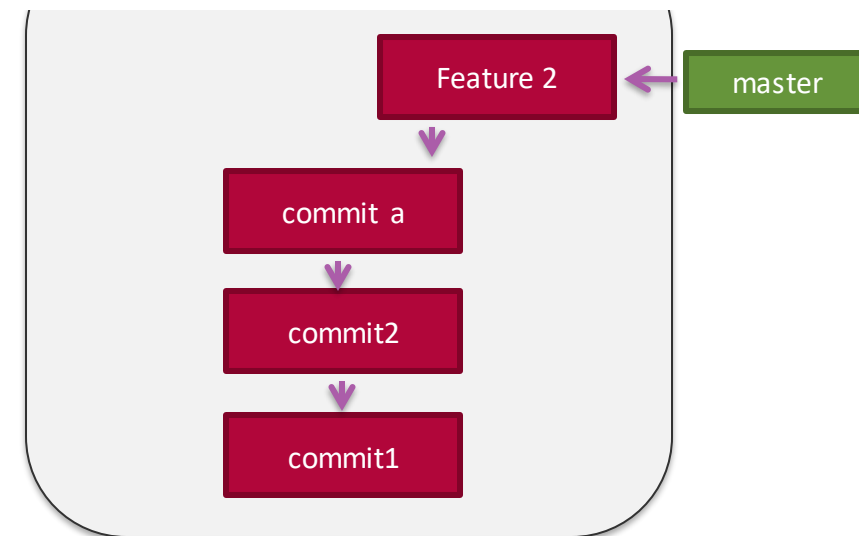
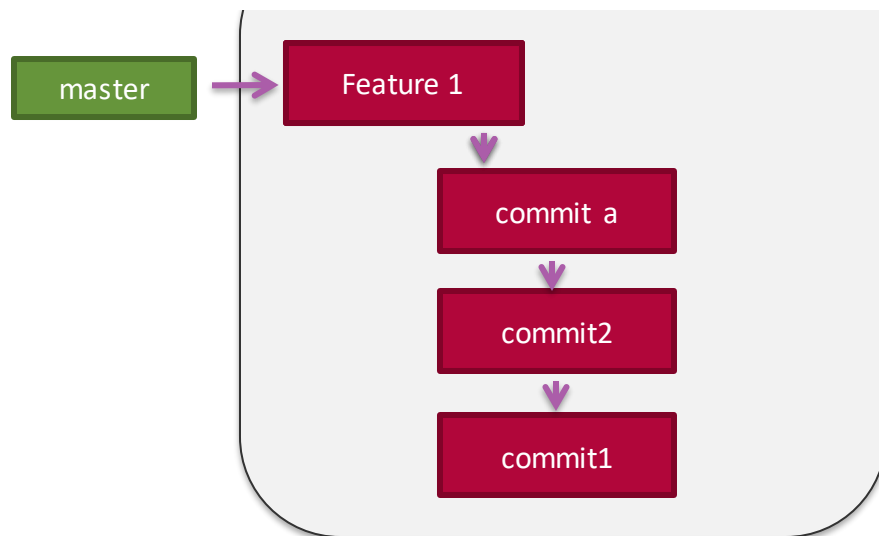
# Push --force



alice\$ git push --f origin master



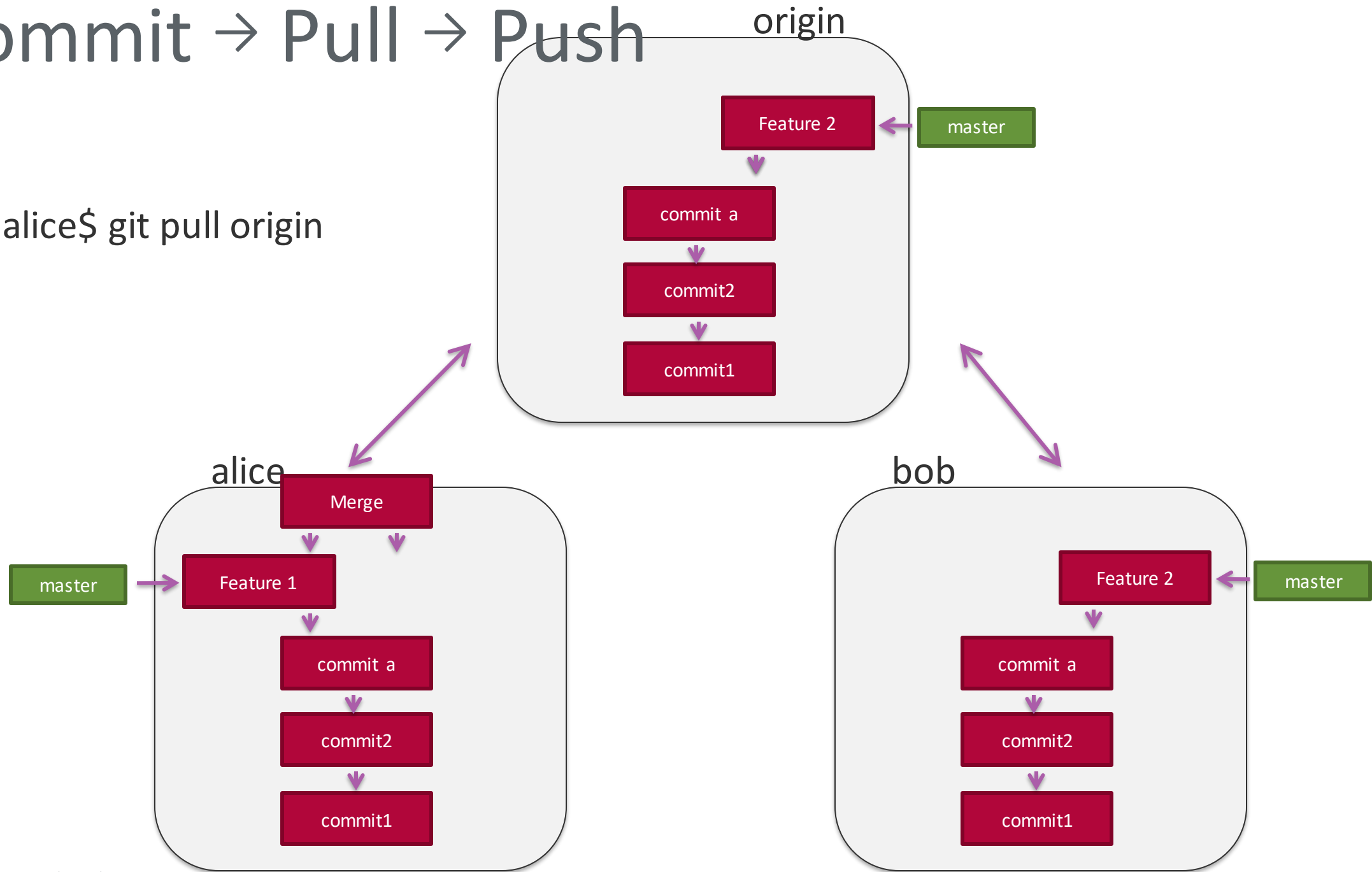
Never **EVER** push --force



# Commit → Pull → Push



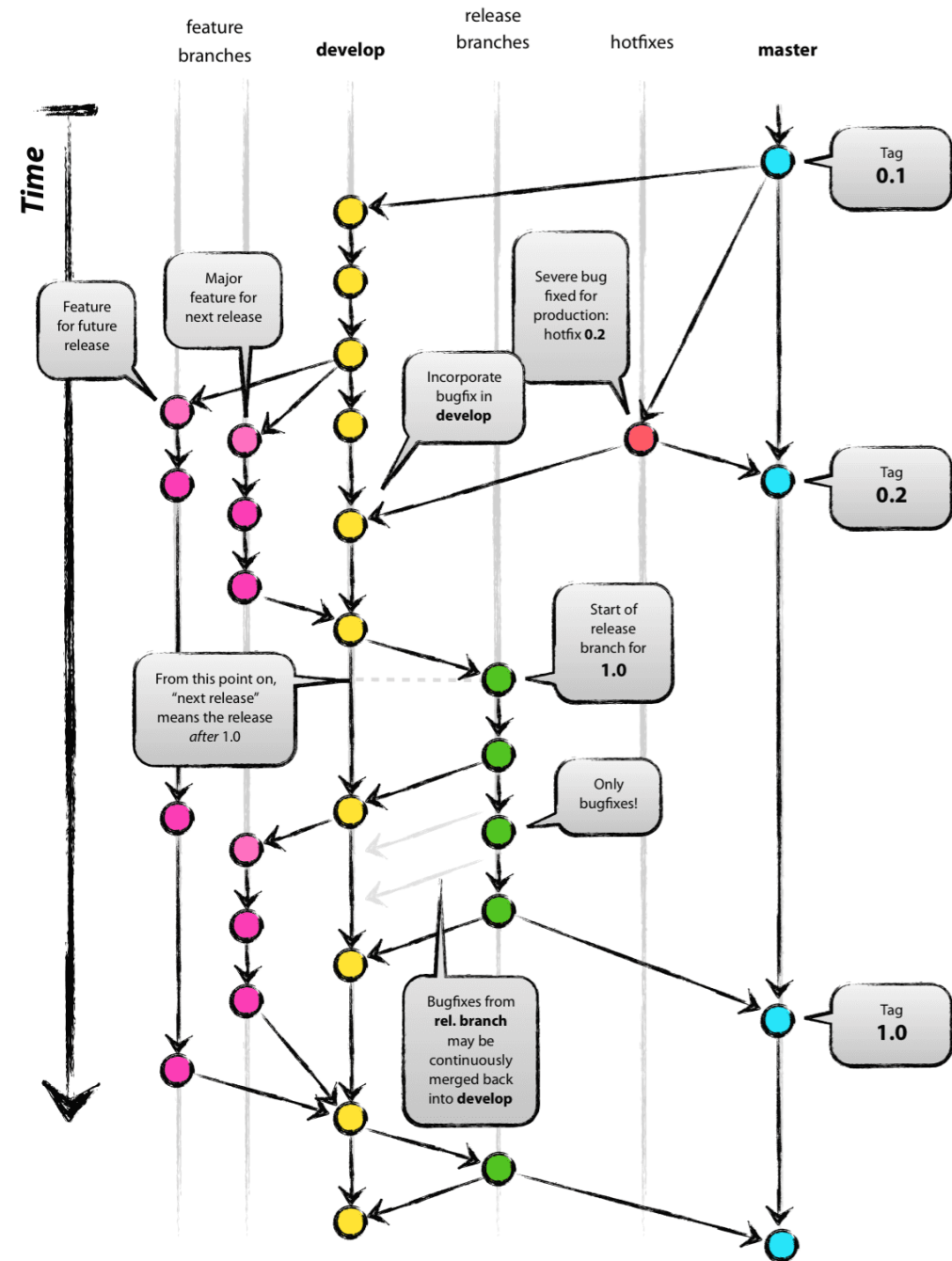
alice\$ git pull origin



# Branching



- Many ways to structure branches
- Some helpful tips:
  - Never merge in master or release branches
  - Never break build in shared branches





# What happened?



- git log
- git diff
- git blame

# Summary



## 1. Basics

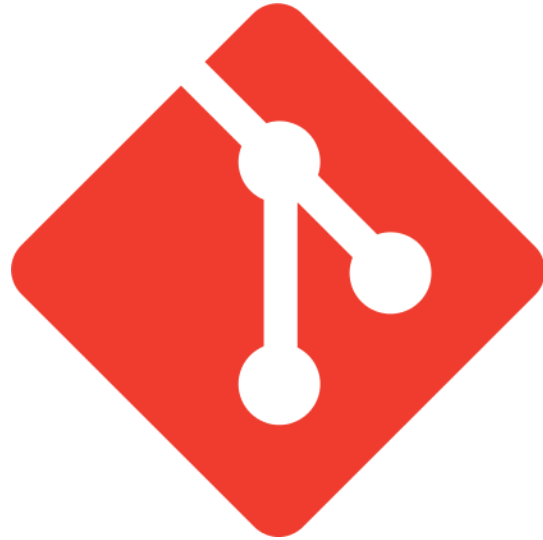
- Objects

## 2. Local

- Checkout
- Add
- Commit

## 3. Collaboration

- Pull
- Push



# git