Intro to Git & GitHub

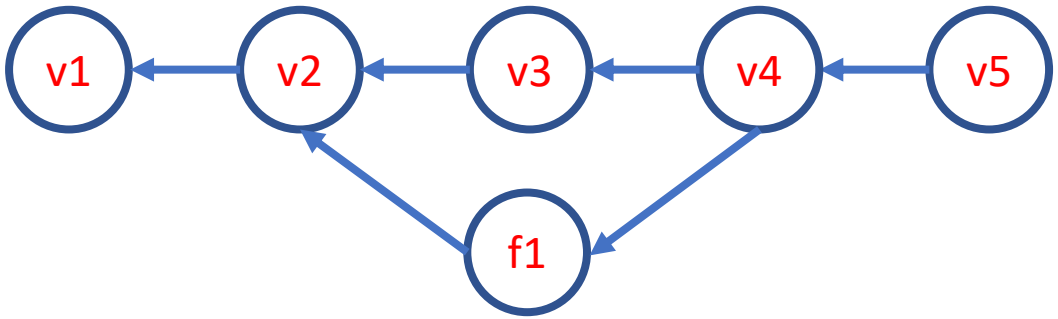
Jay / TDMDAL

Site for this workshop: https://tdmdal.github.io/git-workshop

What's Git

- A version control system
  - manage the evolution of a set of files (repository / repo)
  - usually for source code or text files

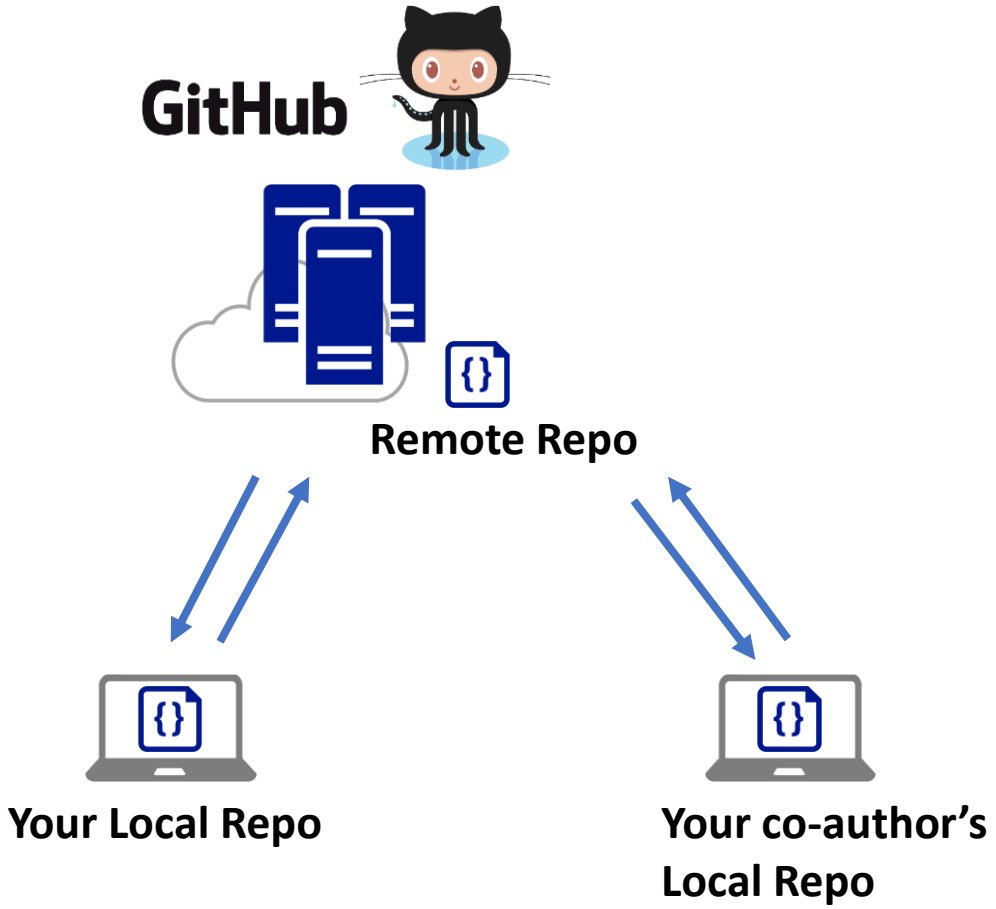
- Version control?
  - keep track of changes: version 1, version 2, etc.
  - like "Track Changes" or "undo" in MS Word, but much more powerful



What's GitHub

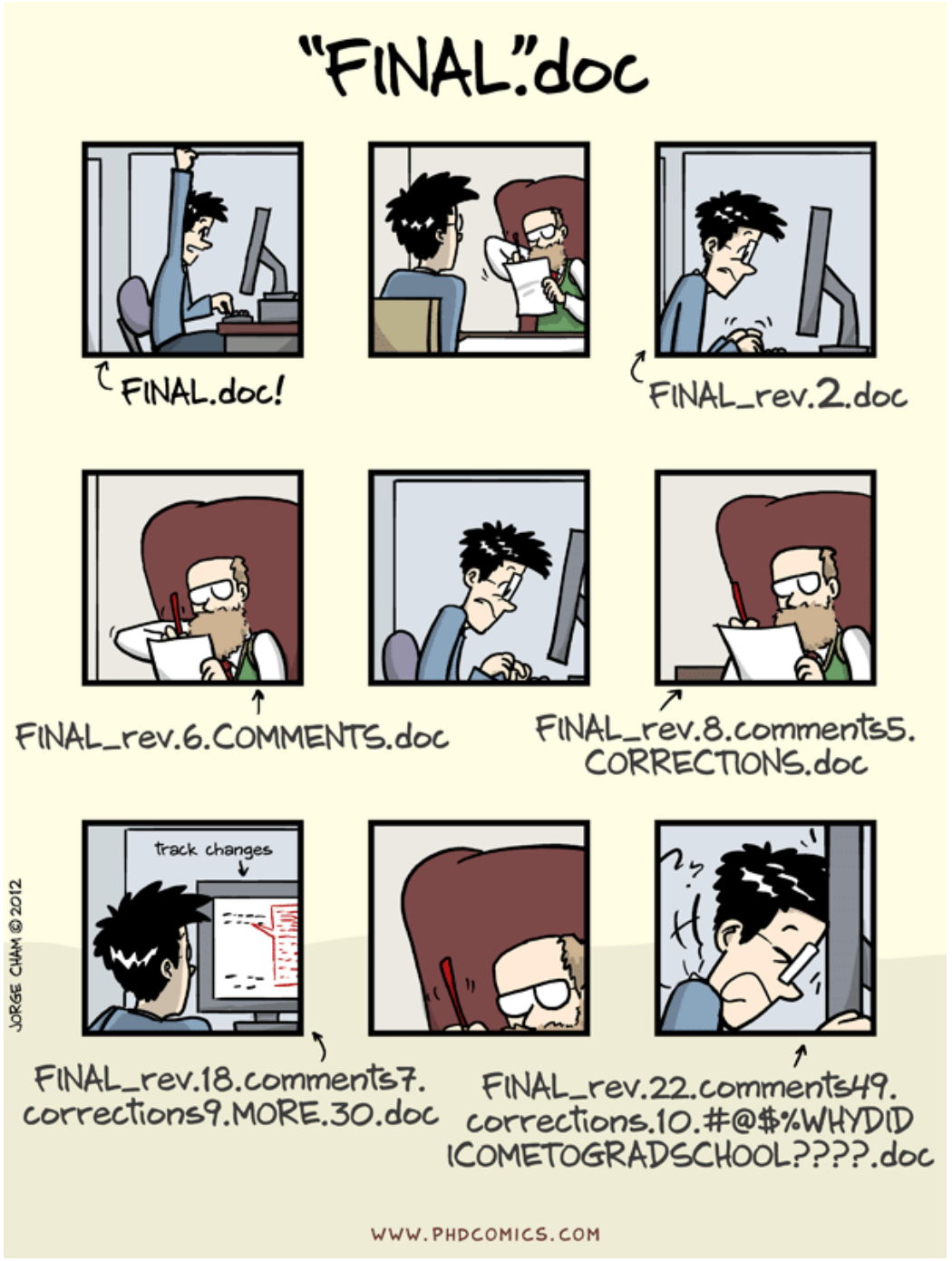
- A git-aware online repo host
- Enable repo sharing and collaboration
  - raise issues, pull request, etc.
- Free public and private repo (*)
- Other repo hosts exist
  - e.g. gutbucket, gitlab, etc.

Ref: https://github.com/pricing



Why Git & GitHub

- Organize (record keeping; traceability)
  - Track, compare and undo changes
  - Manage multiple versions/ideas at the same time efficiently
  - Backup your work
- Share
- Collaborate
  - co-authors
  - open source community



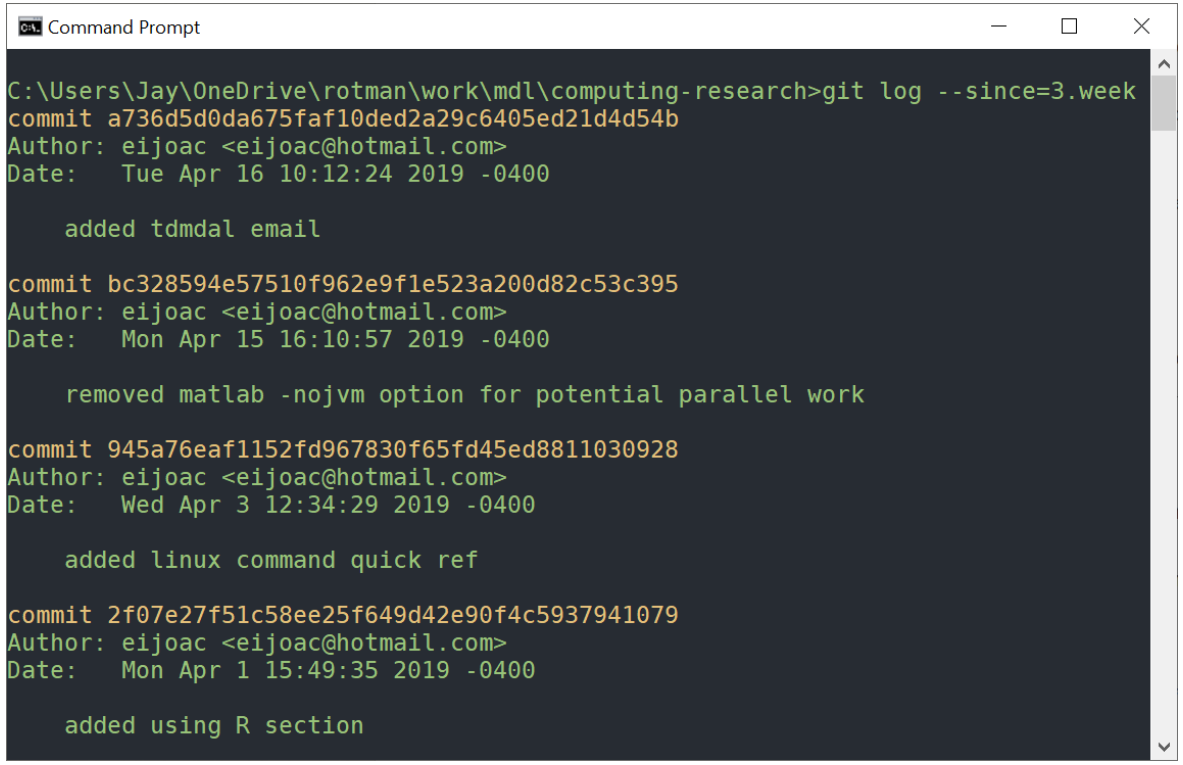
Other Benefits

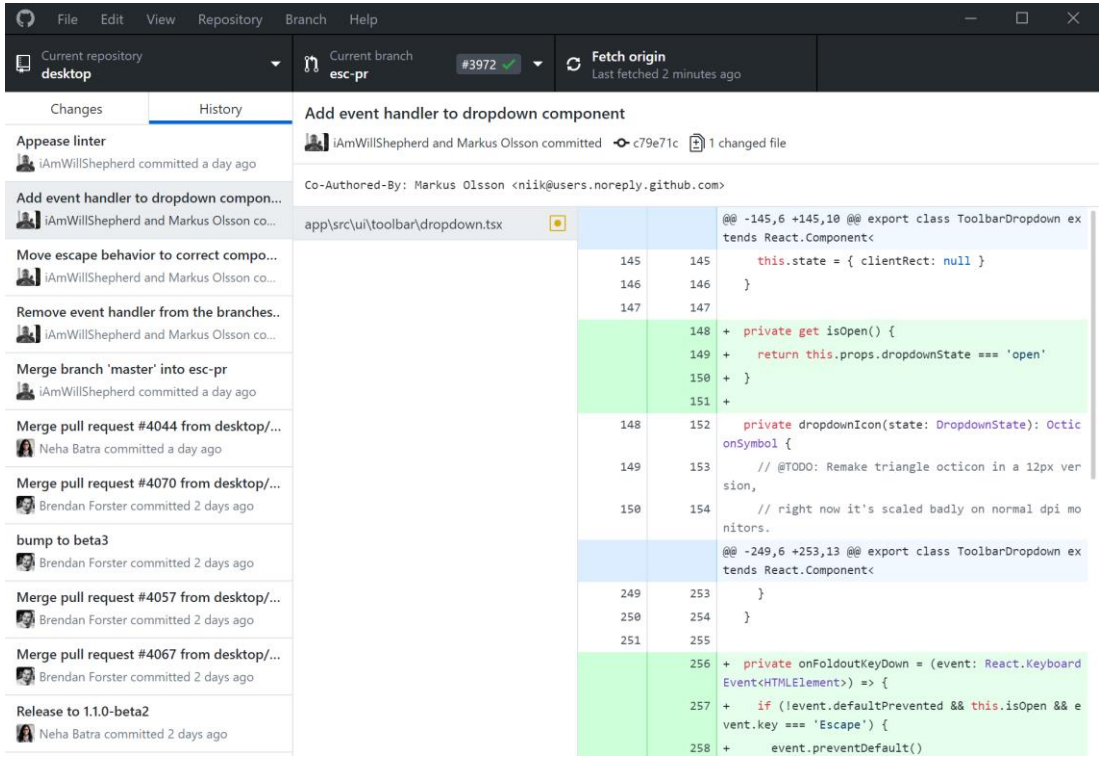
- GitHub can host your (static) website, i.e., a free web host
  - GitHub Pages: https://pages.github.com/
  - ex. https://tdmdal.github.io

- Digital presence for your research or classroom work
  - ex. https://github.com/jesusfv/
  - ex. https://github.com/Computational-Content-Analysis-2018

Using Git: Command Line vs GUI Clients

- Installation: https://git-scm.com/downloads
- Command Line vs GUI clients (many of them)





The simplest git workflow (demo)

1. make changes to your files
2. add/stage the changes (to the staging area): `git add`
3. commit the changes (i.e., record version history): `git commit`
4. repeat (back to 1)…

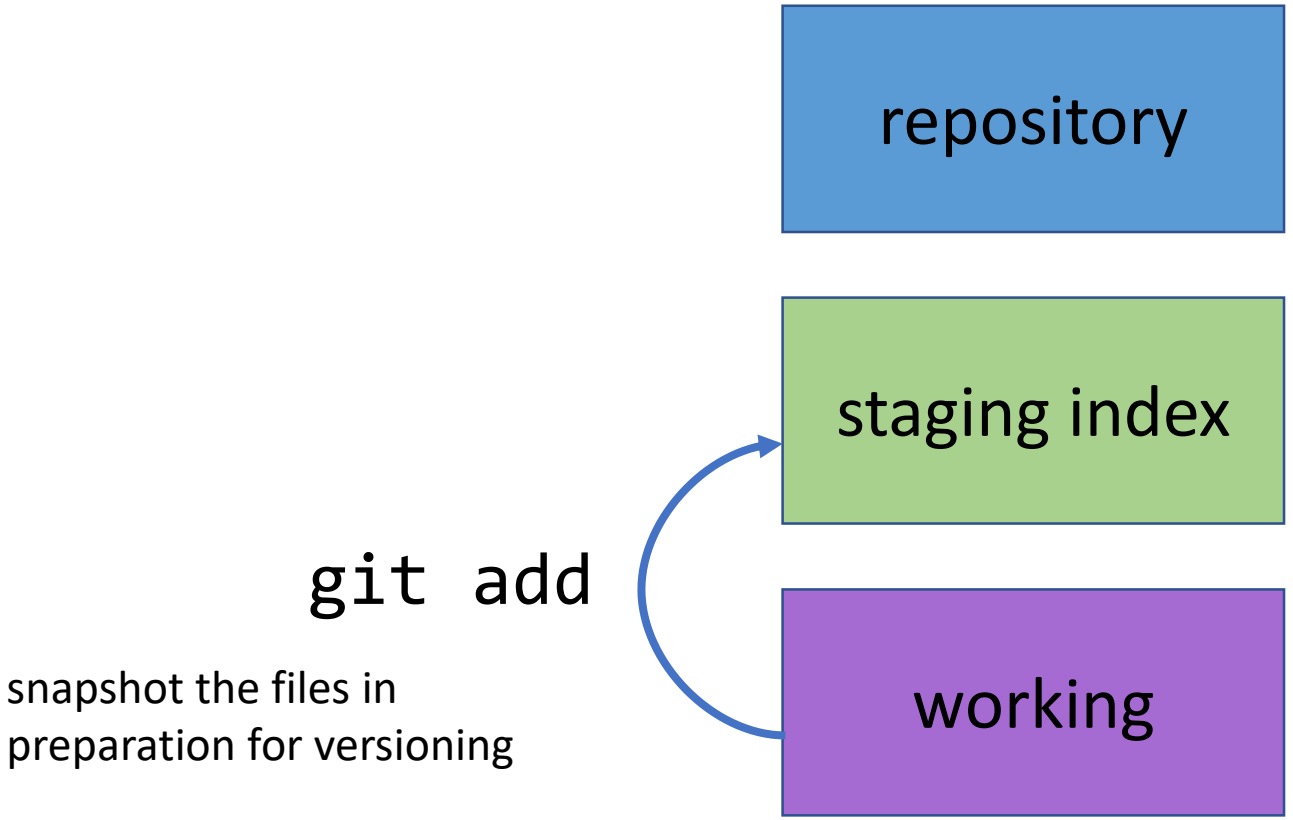
Configure git for first-time use: `git config`
Create a new local repo: `git init`
check commit history: `git log`; `git show`
compare difference between changes: `git diff`

https://tdmdal.github.io/git-workshop/basic-git-workflow.html

Git Concepts – three trees/areas

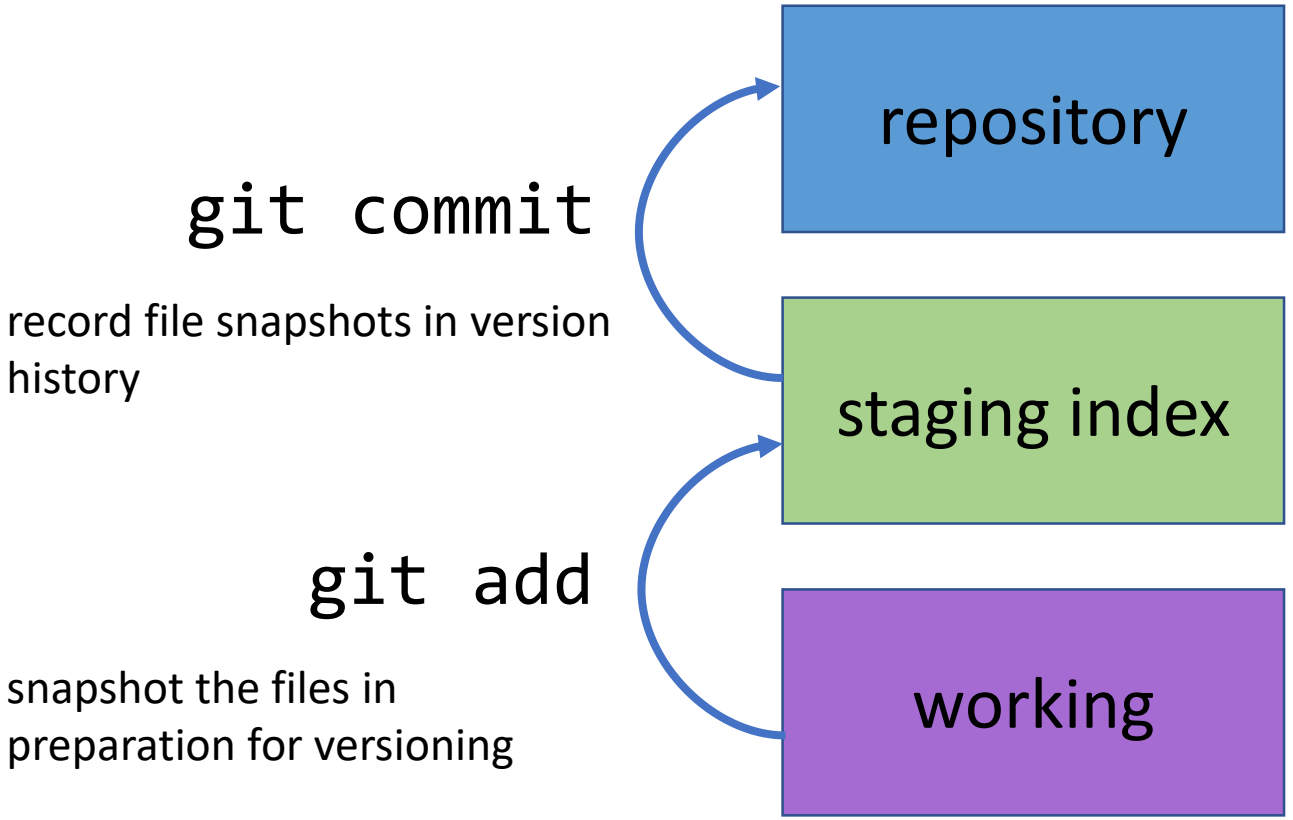
Source: Git Essential Training by Kevin Skoglund on lynda.com

Git Concepts – three trees/areas



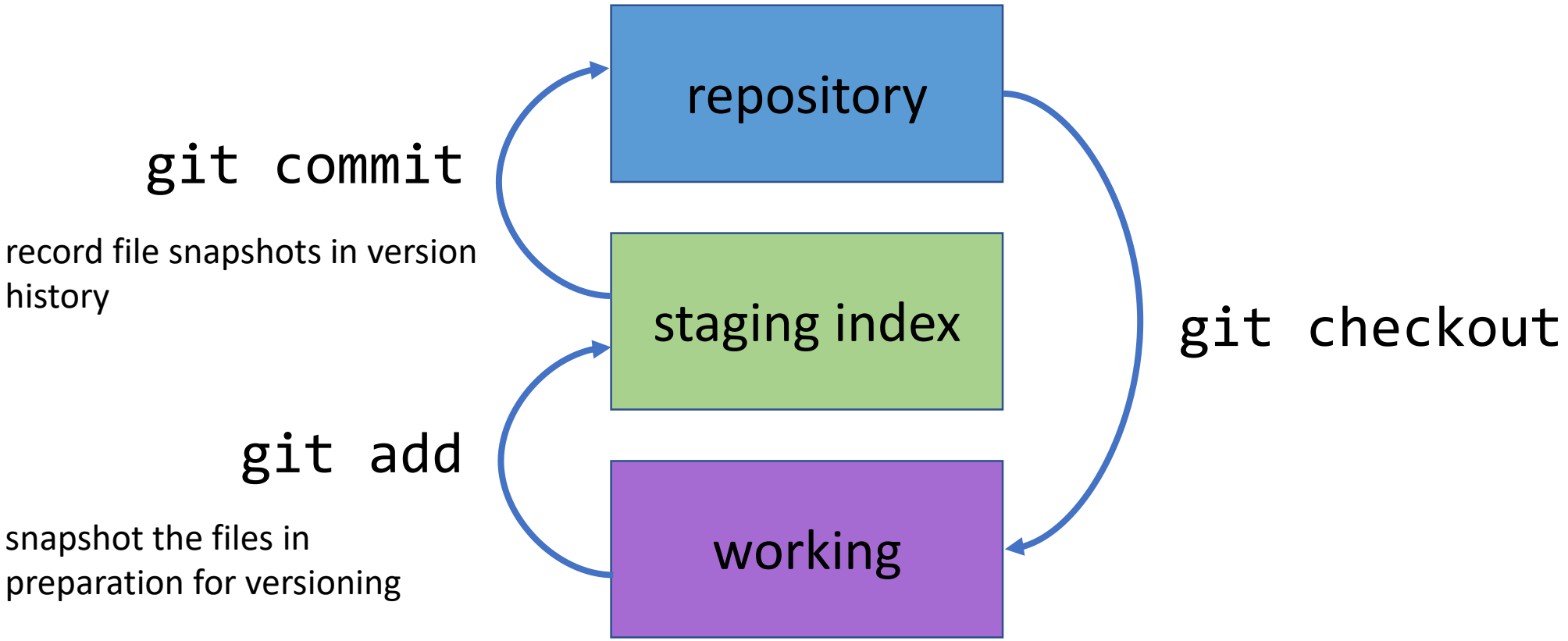
Source: Git Essential Training by Kevin Skoglund on lynda.com

Git Concepts – three trees/areas



Source: Git Essential Training by Kevin Skoglund on lynda.com

Git Concepts – three trees/areas



Source: Git Essential Training by Kevin Skoglund on lynda.com

Hands-on?

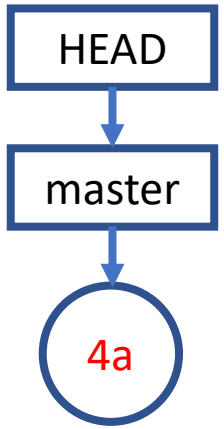
Option 1: Install Git: https://git-scm.com/downloads

Option2: Use this in-browser Linux emulator for Git practice.

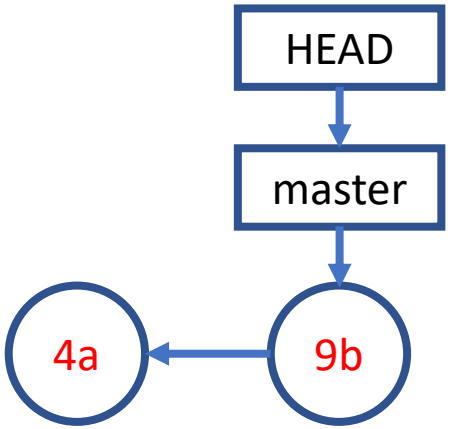
- may have problem accessing internet (i.e. when you use github)

https://tdmdal.github.io/git-workshop/basic-git-workflow.html

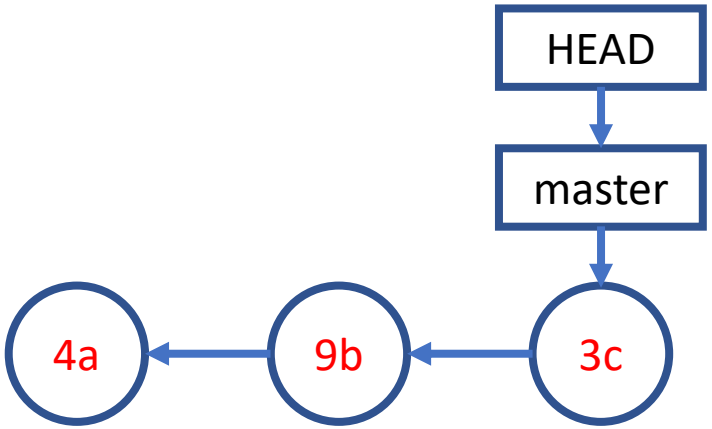
Git Concepts – First commit



Git Concepts – Second commit



Git Concepts – Third commit and so on…



Remove and Rename Files (FYI)

- Remove files (demo)

```
git rm <file>
```

- Rename files

```
git mv <file_old> <file_new>
```

- After removing or rename files

```
git commit –m "<remove or rename msg>"
```

Undo (1 / FYI)

- Undo working directory changes

```
git checkout -- <file>
```

- Retrieve old version of a file (to staging index & working dir) (demo)

```
git checkout <commit-id> -- <file>
```

- Unstaging files

```
git reset HEAD <file>
```

Undo (2 / FYI)

- Amending last commit

```
git commit –amend –m “commit message”
```

- Reverting a commit (by adding a new commit to undo last commit)

```
git revert <commit-id>
```

- Undo multiple commits

```
git reset [--soft|--mixed|--hard] <commit-id>
```

Suppress Tracking: .gitignore file

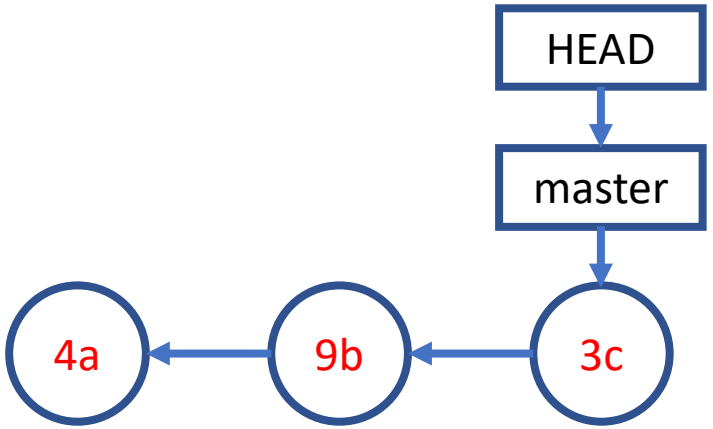
- Don’t track certain files for a project
  - e.g. my_project/.gitignore

my_project/.gitignore

```
*.log
log/
data/
!data/sample.csv
```

A collection of useful .gitignore templates.

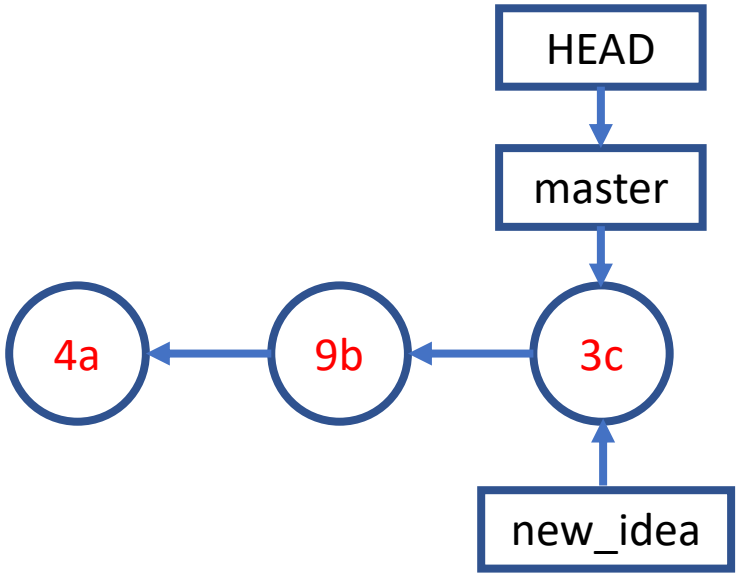
Branching (another common git workflow)



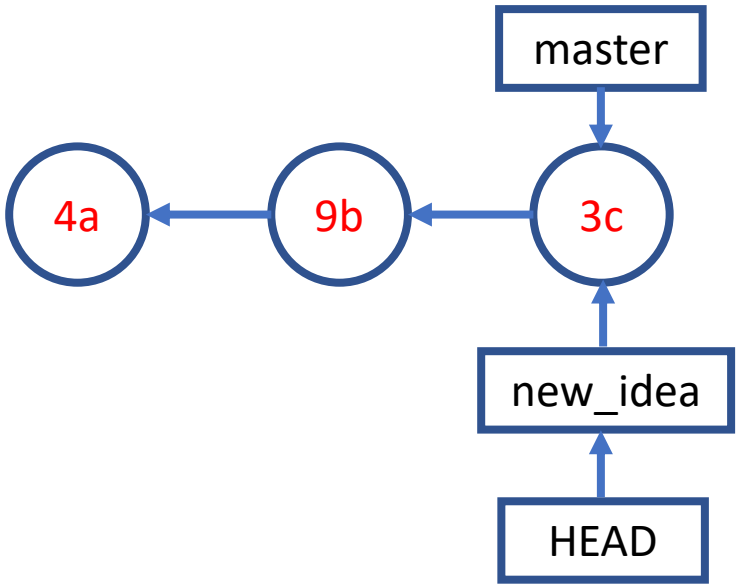
Inspiration source: Git Essential Training by Kevin Skoglund on lynda.com

Branch

```
git branch new_idea
```

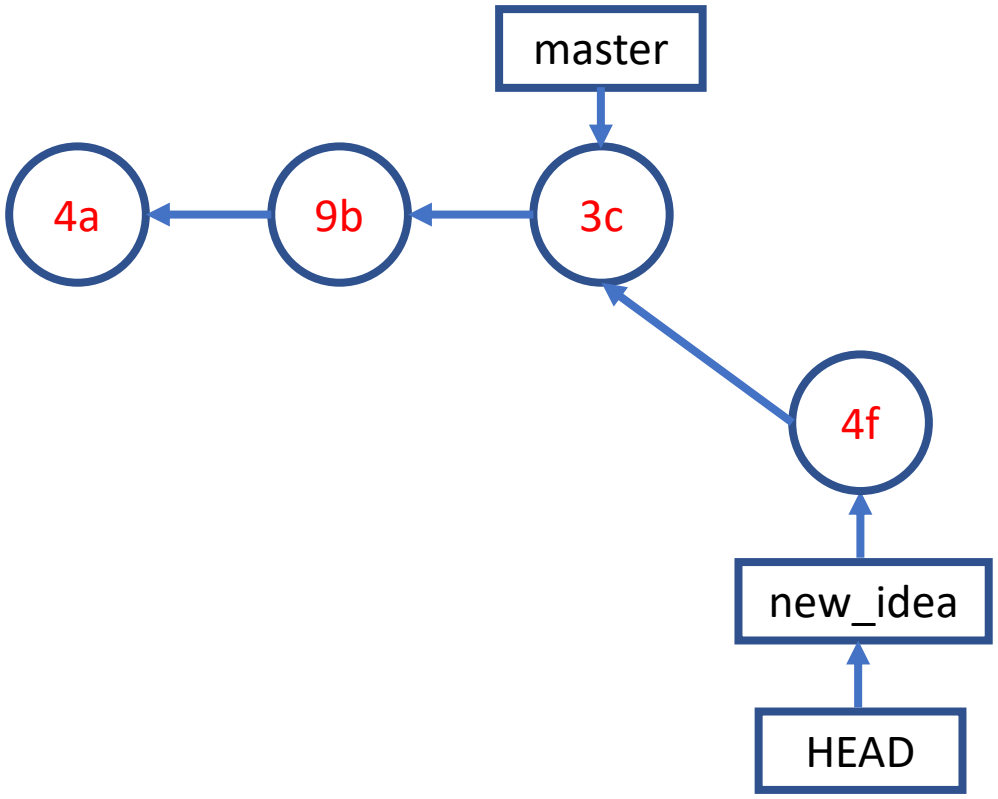


Branch `git checkout new_idea`



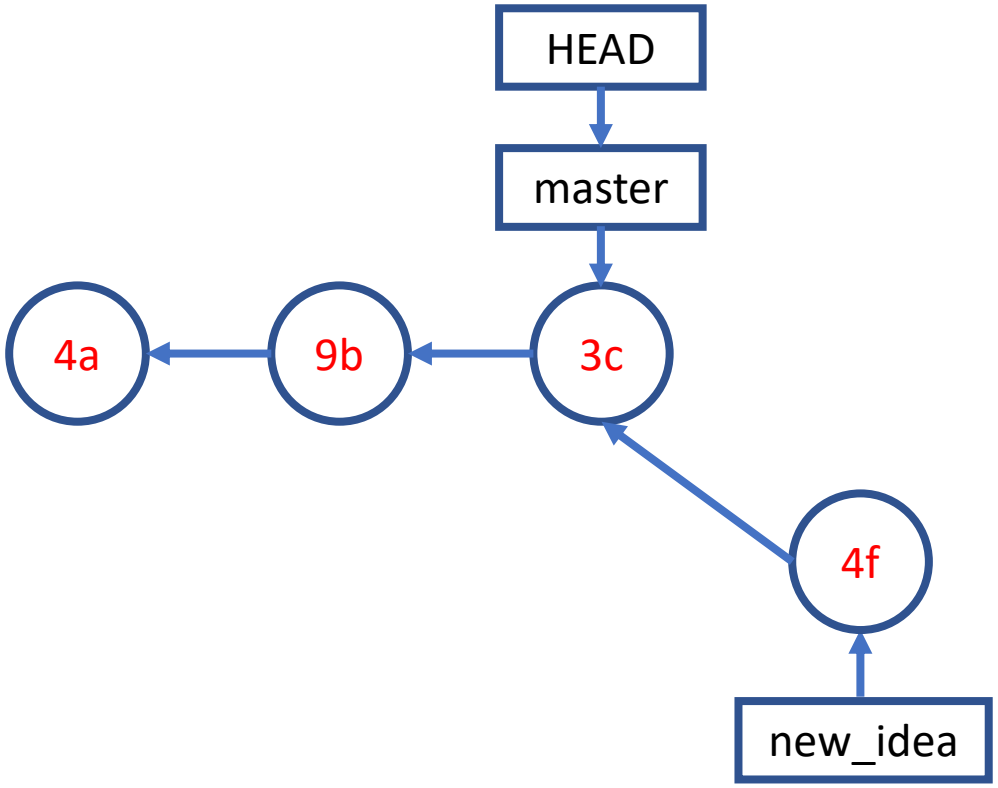
Branch

```
git add; git commit;
```



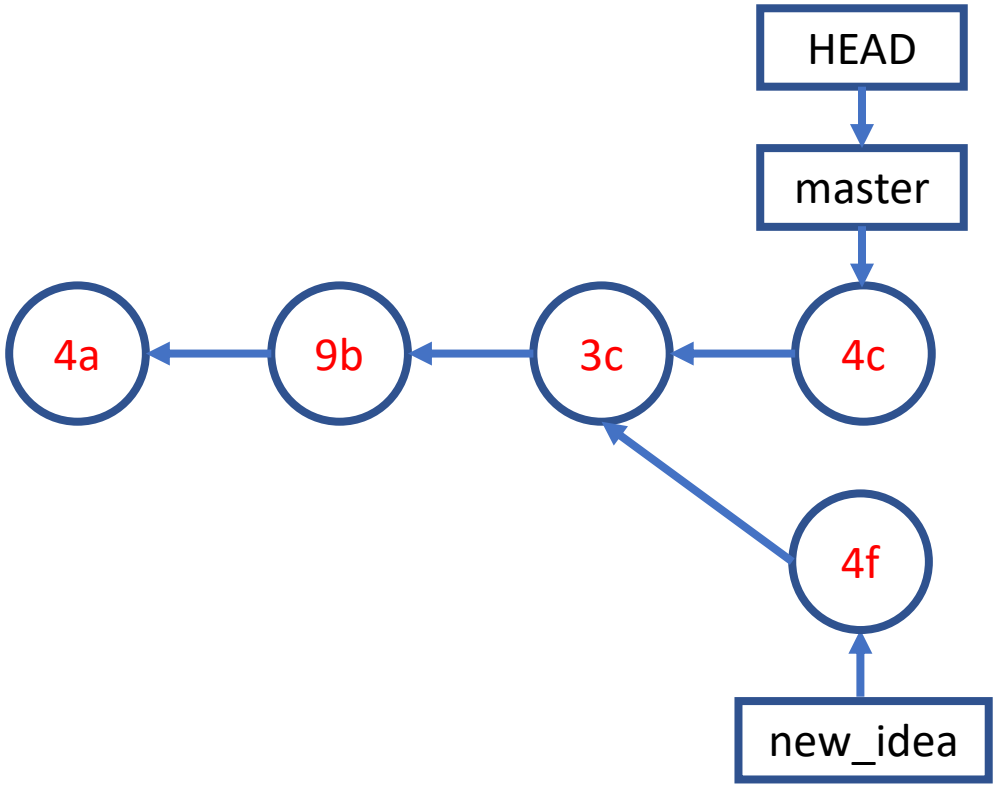
Branch

```
git checkout master
```



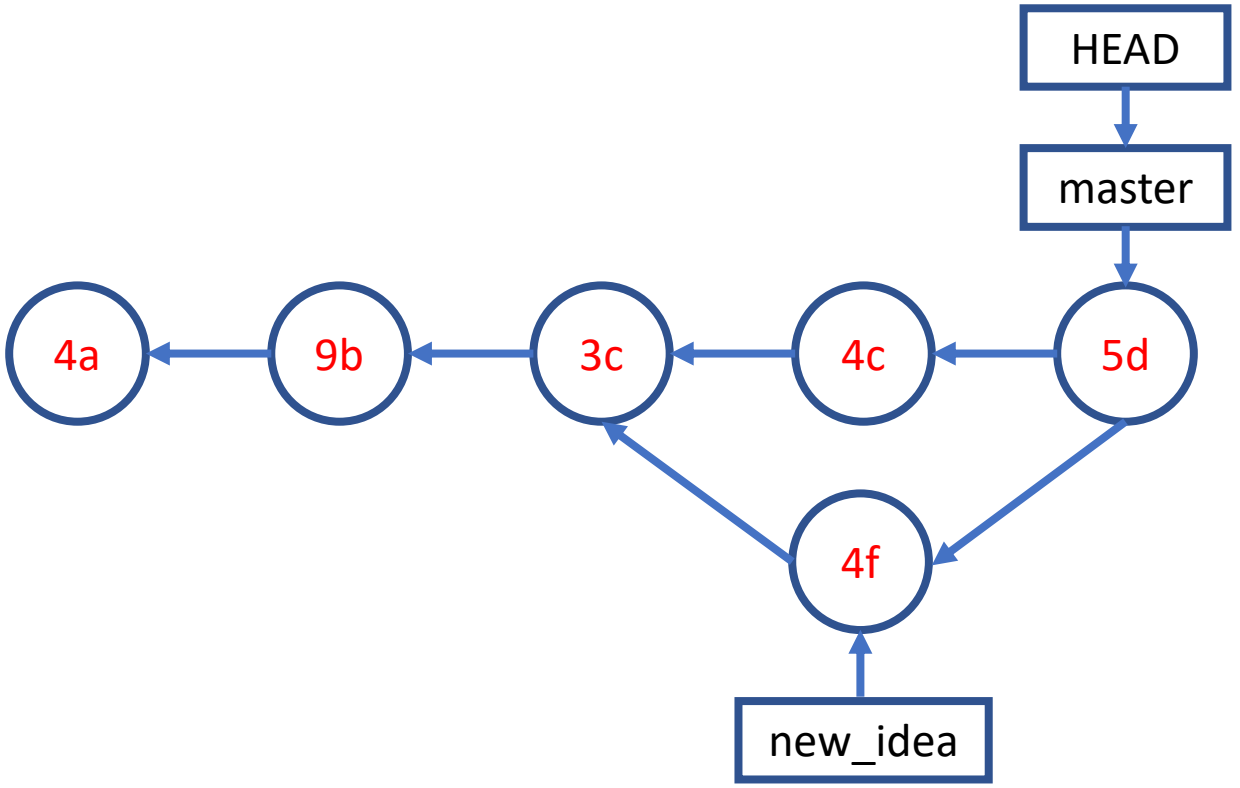
Branch

```
git add; git commit;
```



Merge

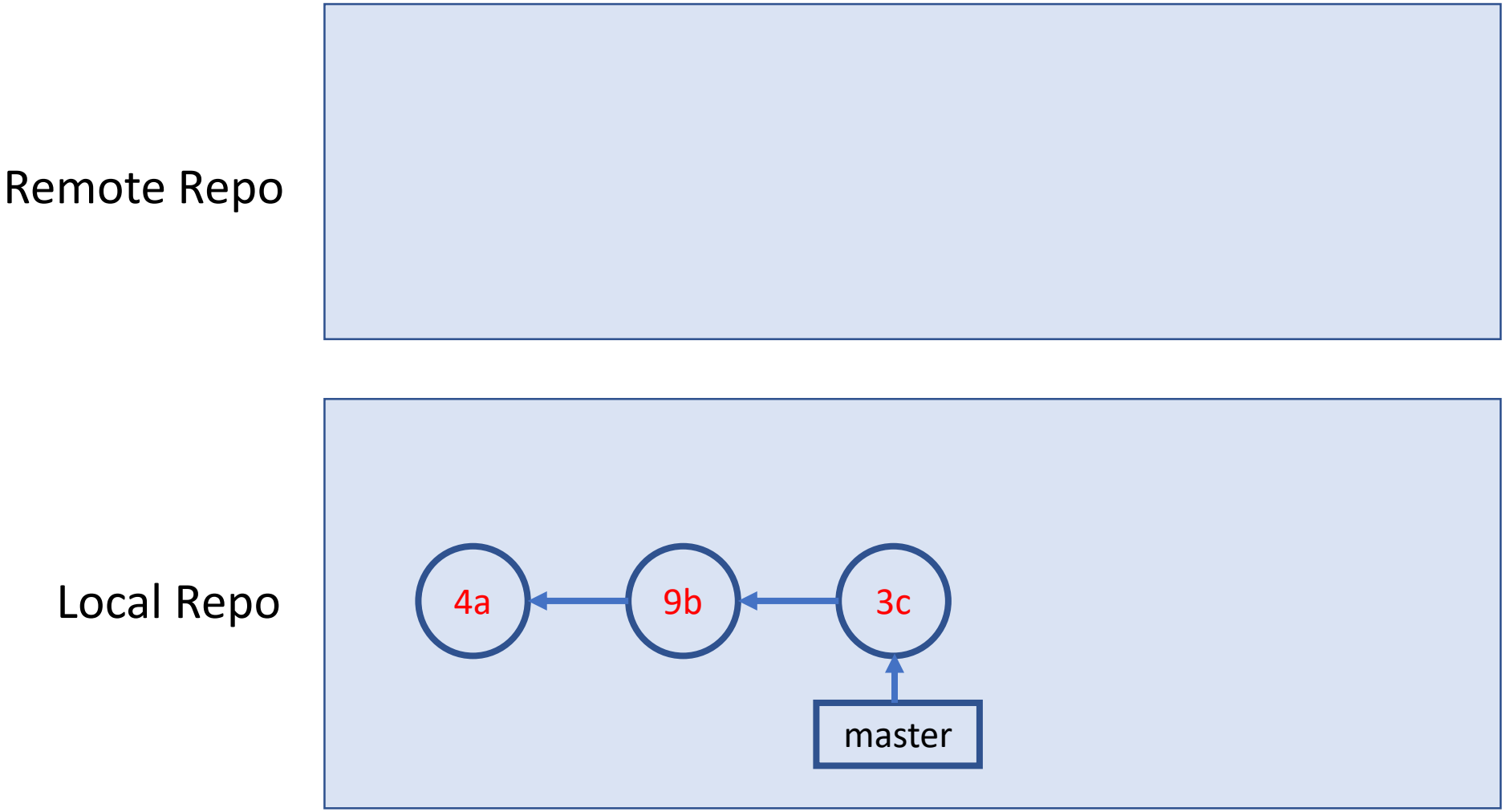
```
git merge new_idea
```



Work with GitHub (demo)

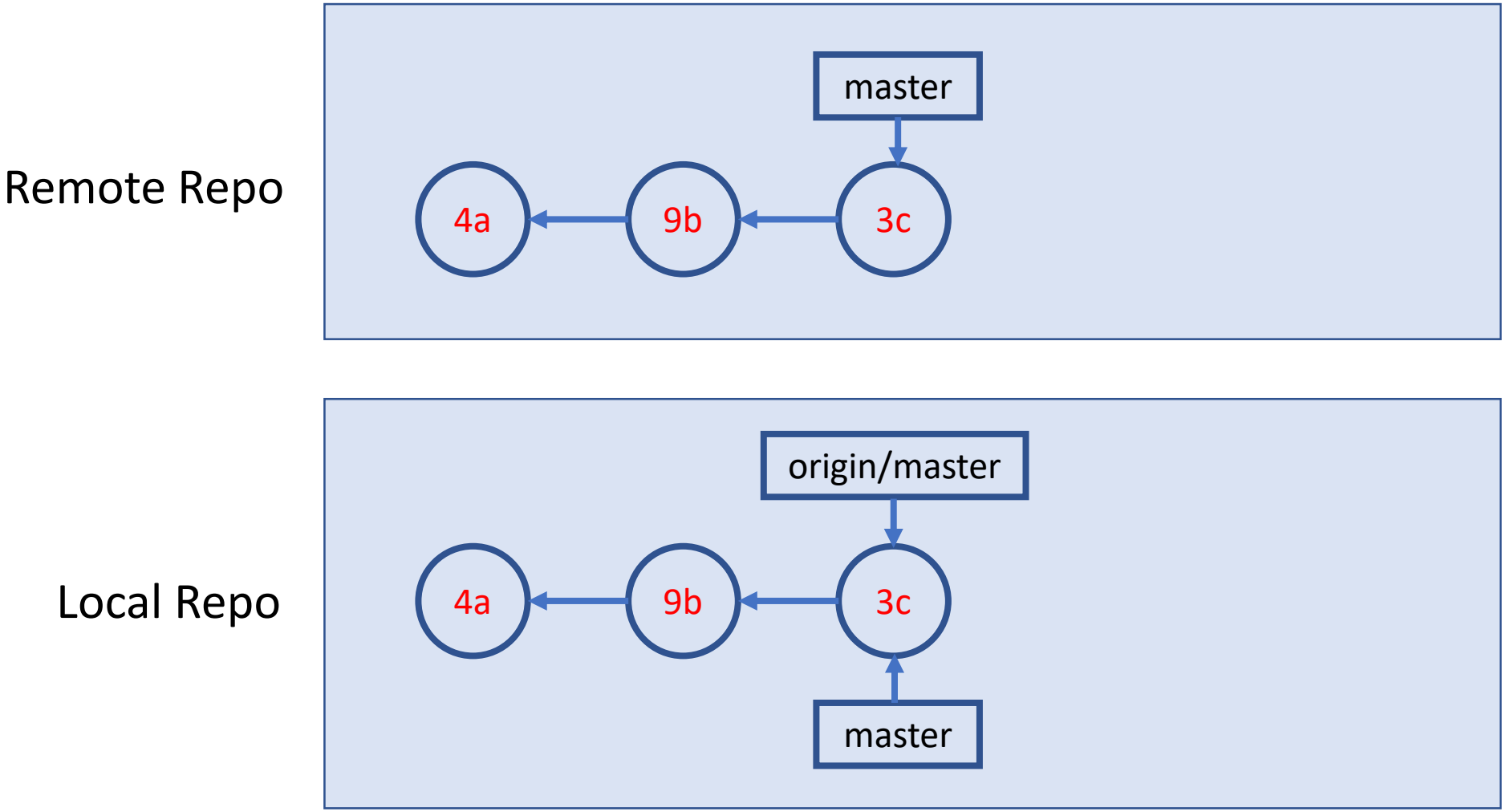
- GitHub Account
- Create a GitHub project repo & push your code there `git remote add` `git push`
  - backup
  - collaborate with your co-authors
  - collaborate with open source community
- Use a public repo as your project starting point `fork & git clone`

A Simple Remote Repo Workflow



Source: Git Essential Training by Kevin Skoglund on lynda.com

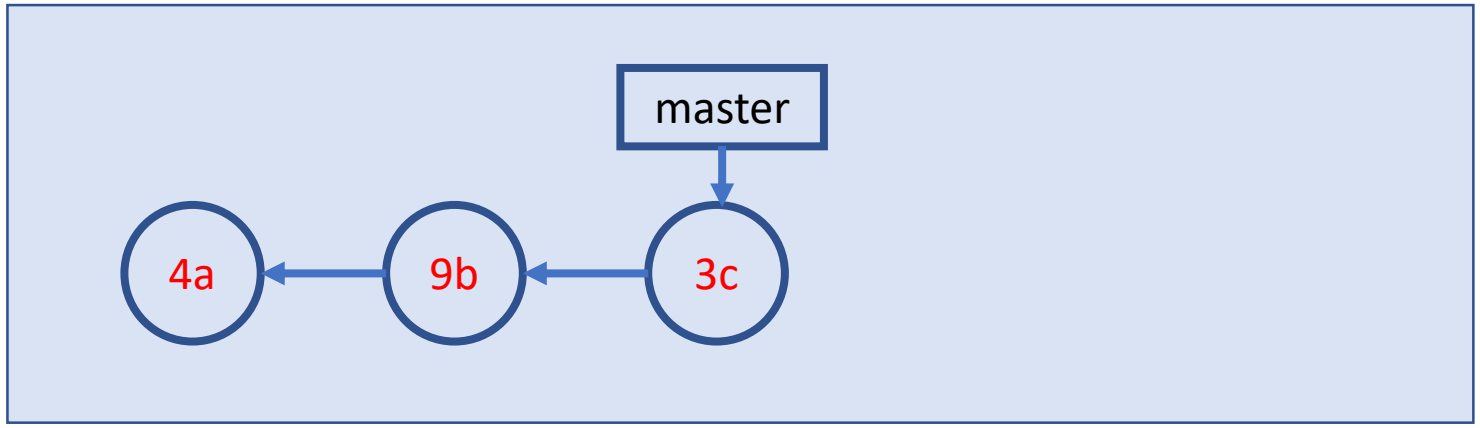
A Simple Remote Repo Workflow `git push`



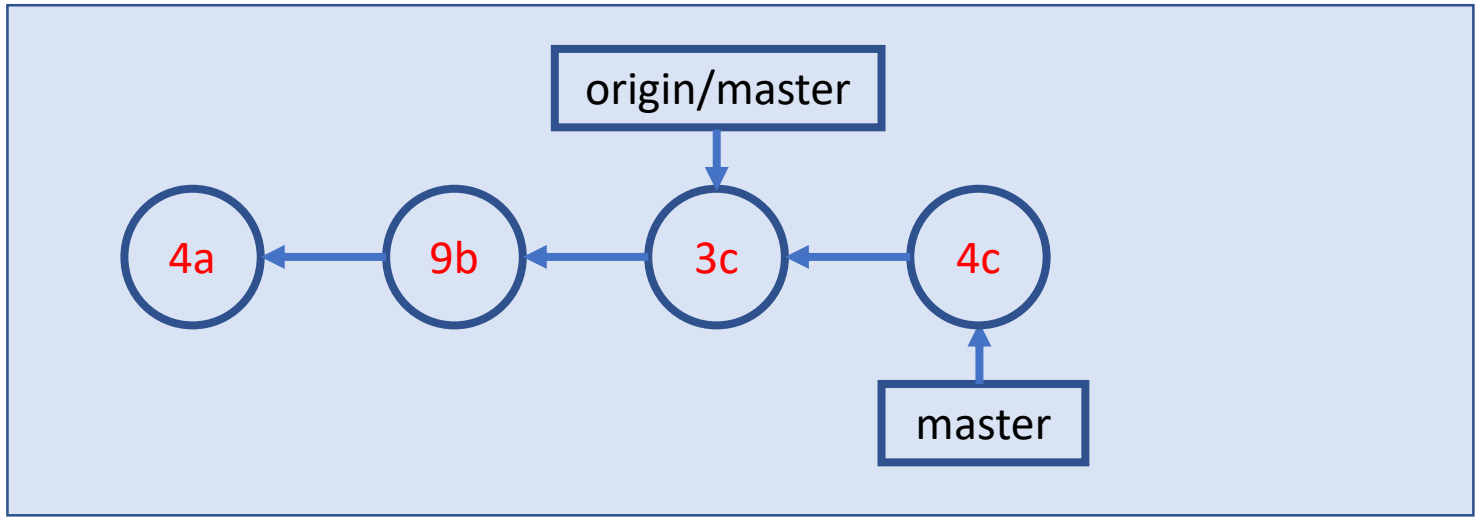
Source: Git Essential Training by Kevin Skoglund on lynda.com

A Simple Remote Repo Workflow

Remote Repo

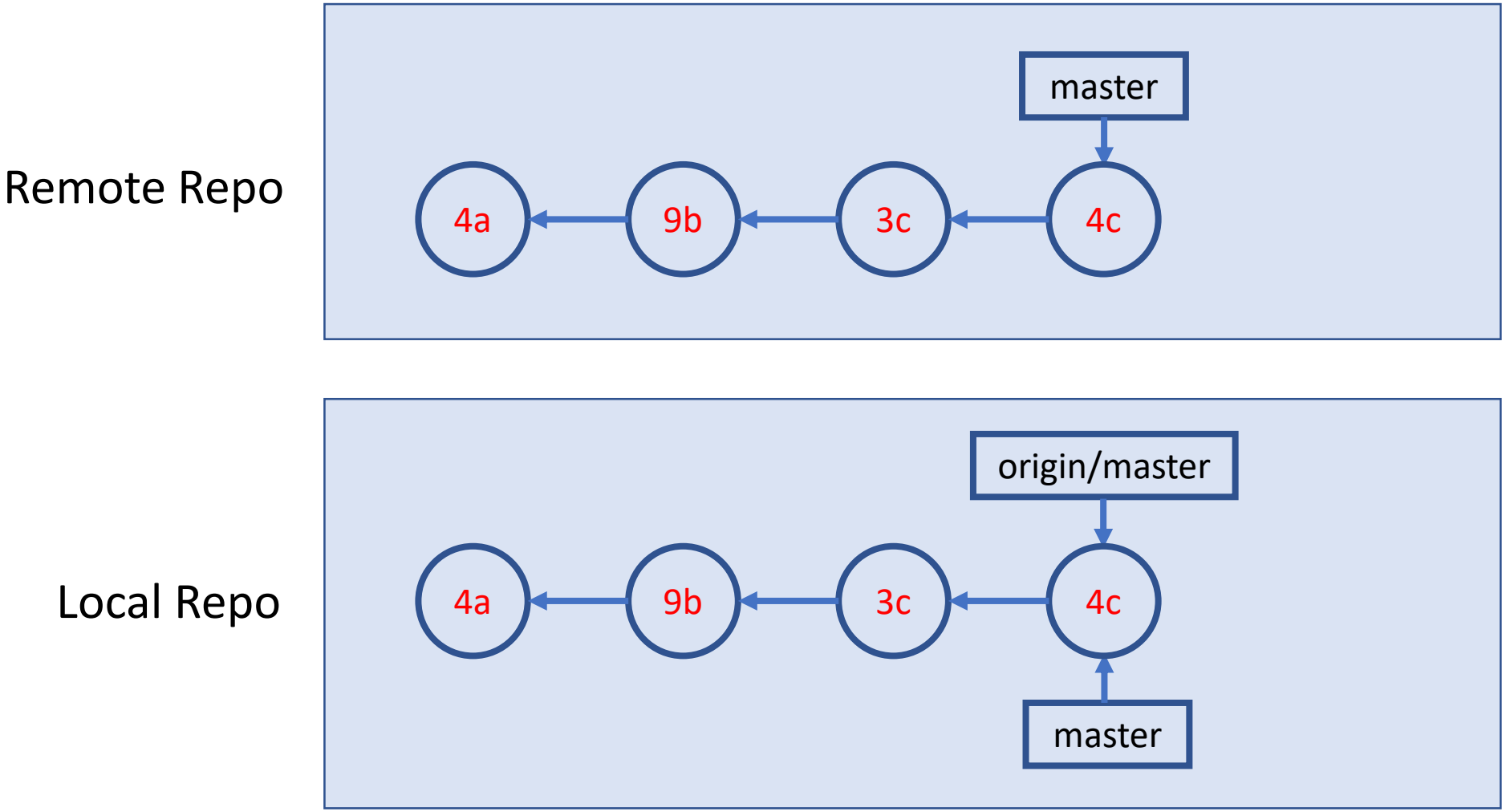


Local Repo



Source: Git Essential Training by Kevin Skoglund on lynda.com

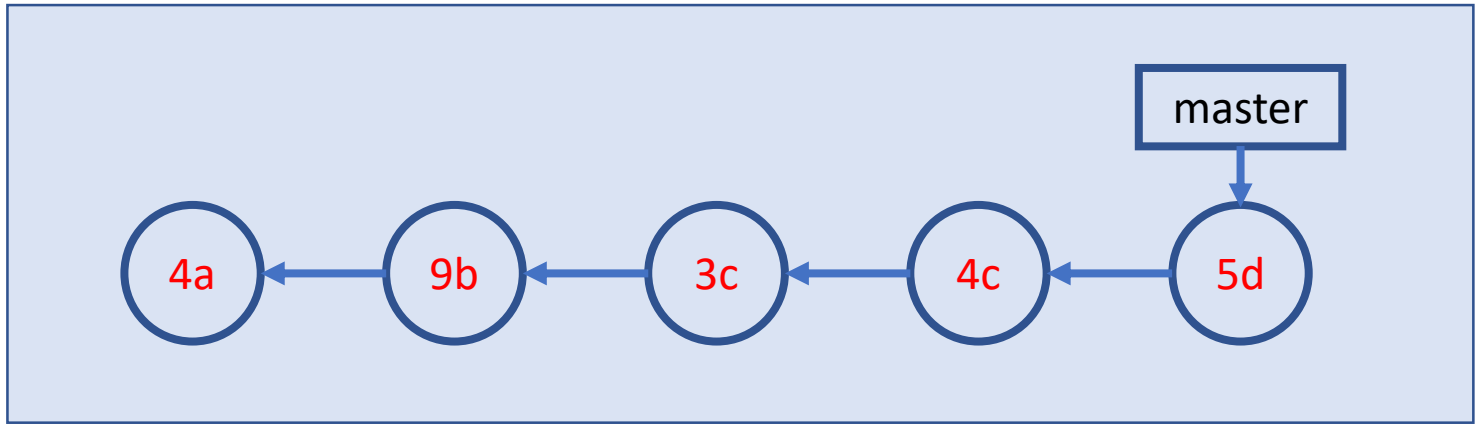
A Simple Remote Repo Workflow `git push`



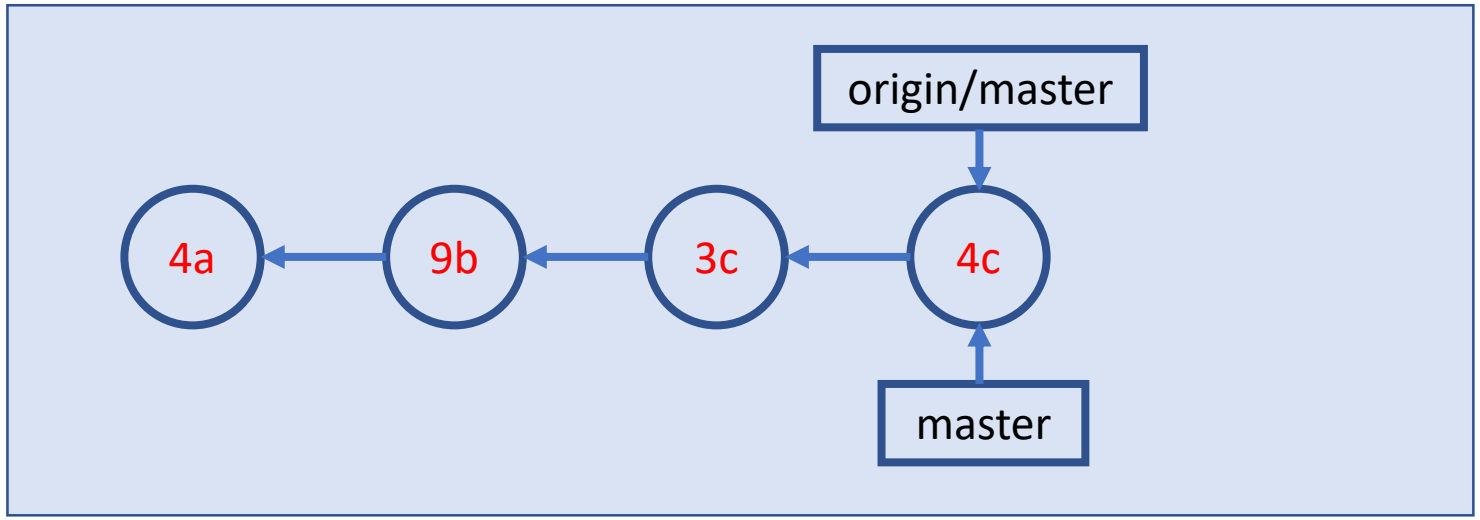
Source: Git Essential Training by Kevin Skoglund on lynda.com

A Simple Remote Repo Workflow

Remote Repo

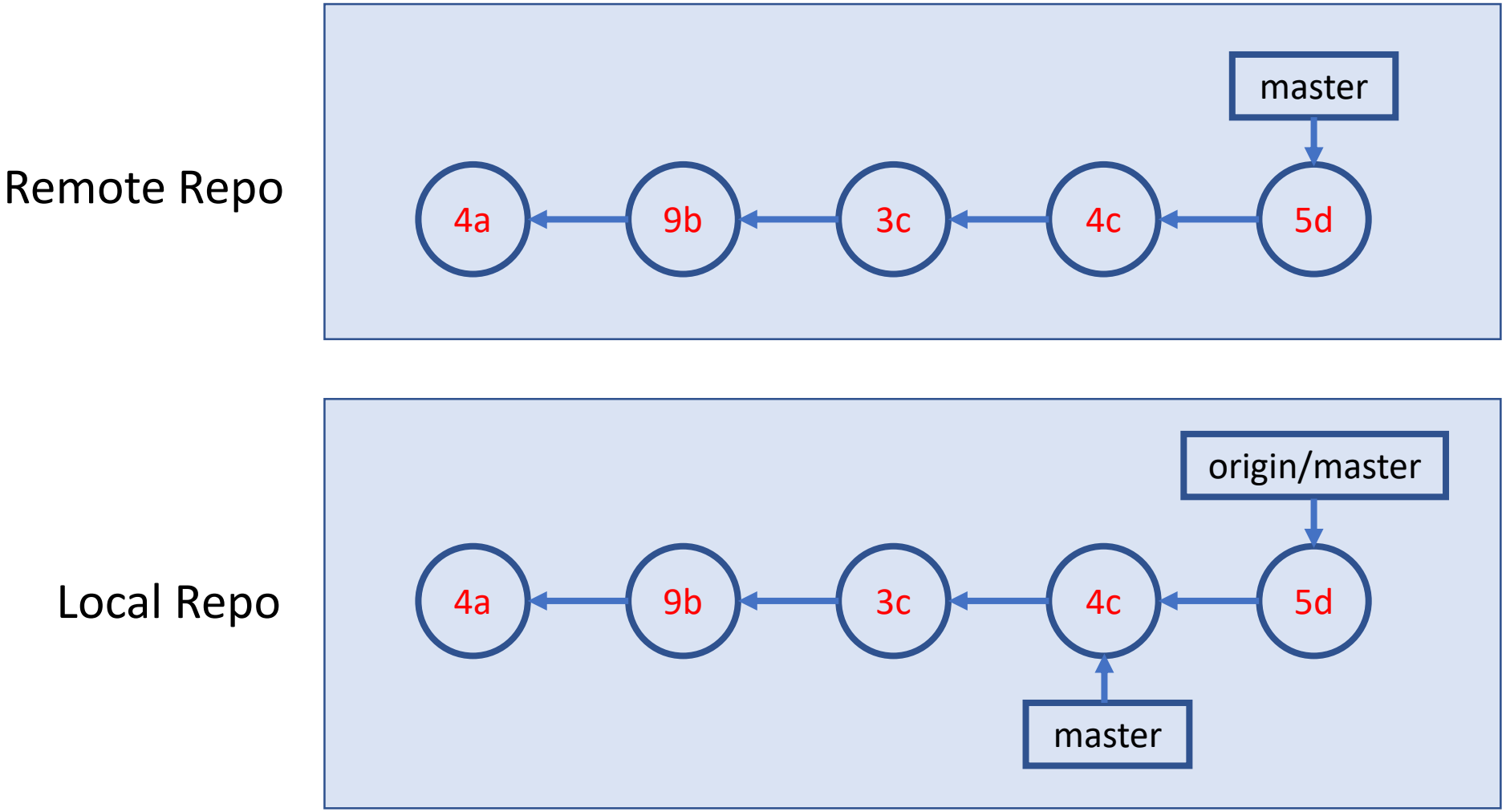


Local Repo



Source: Git Essential Training by Kevin Skoglund on lynda.com

A Simple Remote Repo Workflow `git fetch`

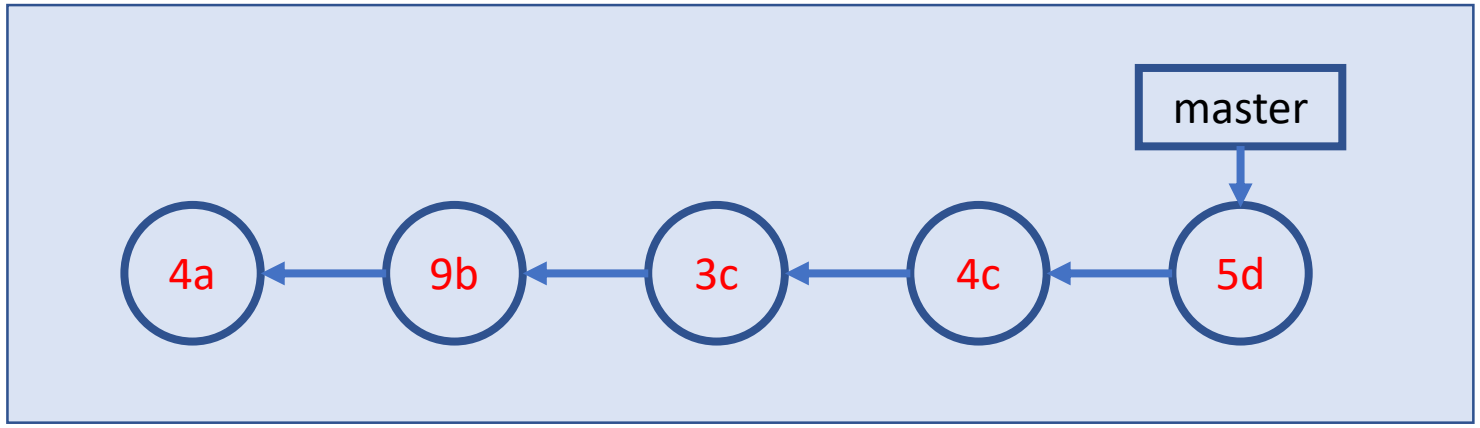


Source: Git Essential Training by Kevin Skoglund on lynda.com

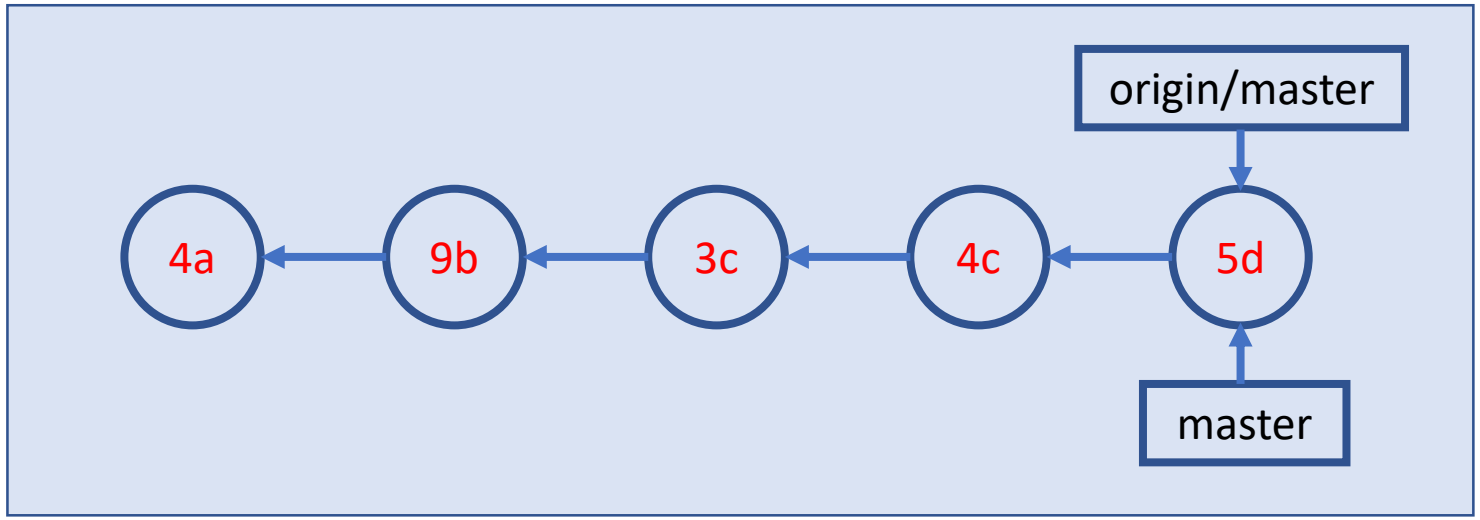
A Simple Remote Repo Workflow

`git merge`

Remote Repo



Local Repo



Source: Git Essential Training by Kevin Skoglund on lynda.com

Many more to explore on your own

- Git concept / command
  - merge conflict
  - remote branch
  - git reset
  - git stash, rebase, bisect
  - …
- Git best practice
  - workflows
  - commit size / message
  - …

Resources

- Git Ref Book: https://git-scm.com/book/en/v2

- Git Tutorials
  - Version Control with Git by Software Carpentry
  - Git Essential Training by Kevin Skoglund at lynda.com
    - login from here for UofT free access
  - Get Started Tutorials from Bitbucket Atlassian
  - GitHub Guides

- Git GUI (I recommend starting with command line)
  - dedicated GUI client: https://git-scm.com/downloads/guis
  - GUI integrated with IDE or code editor (e.g. RStudio, vscode, etc.)