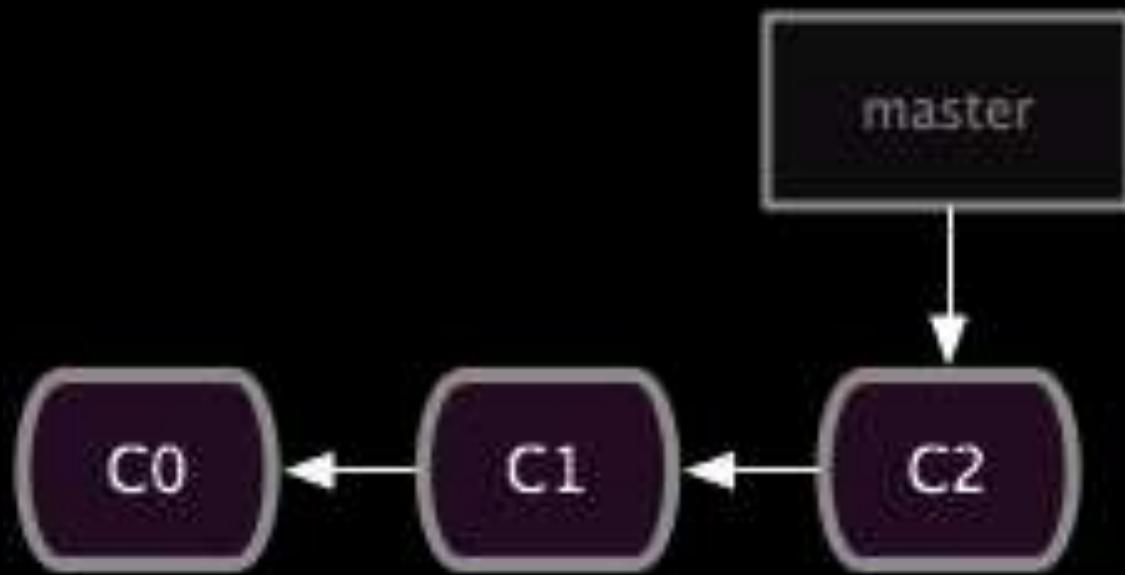




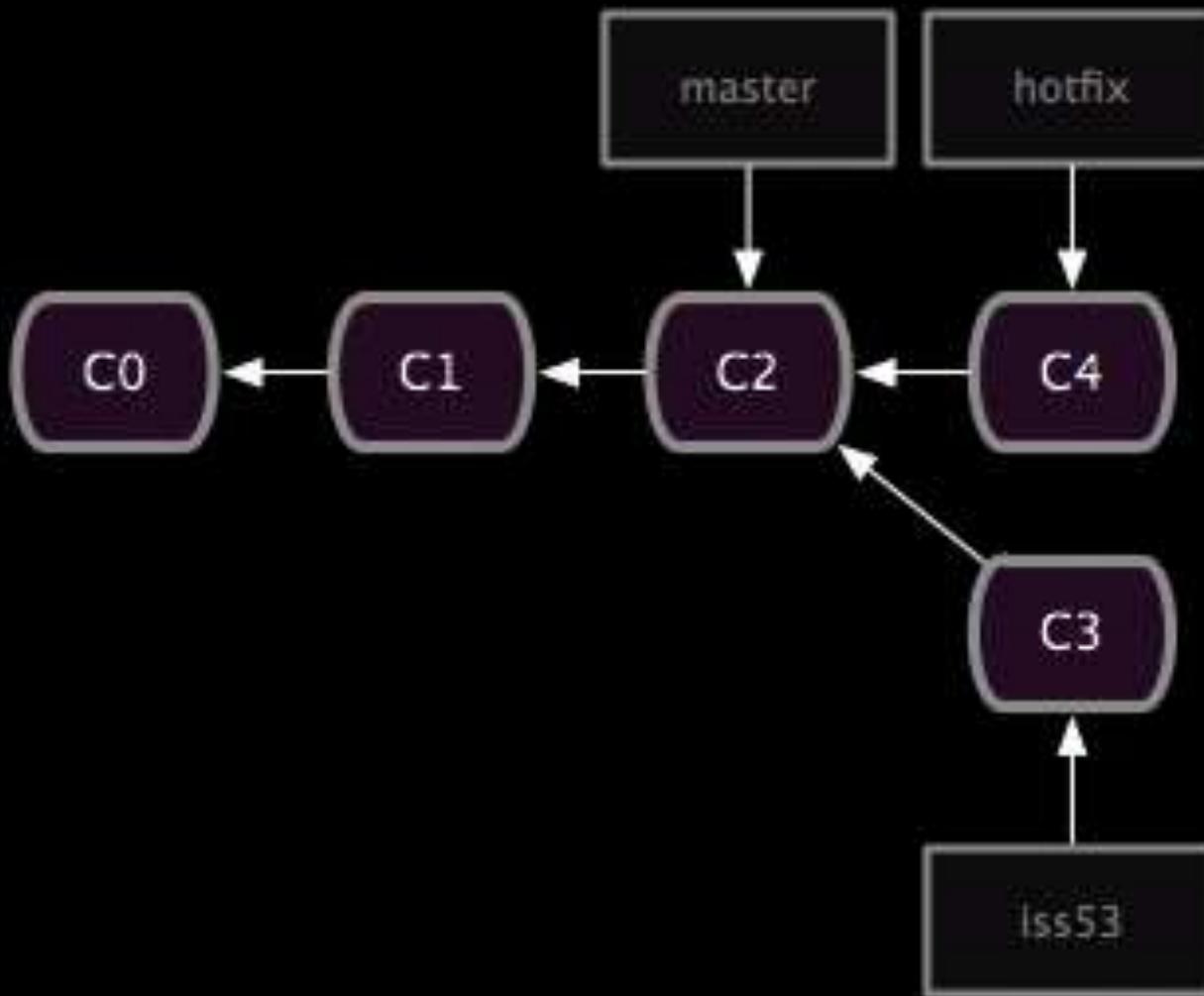
Version Control with Git

Why track/manage revisions?

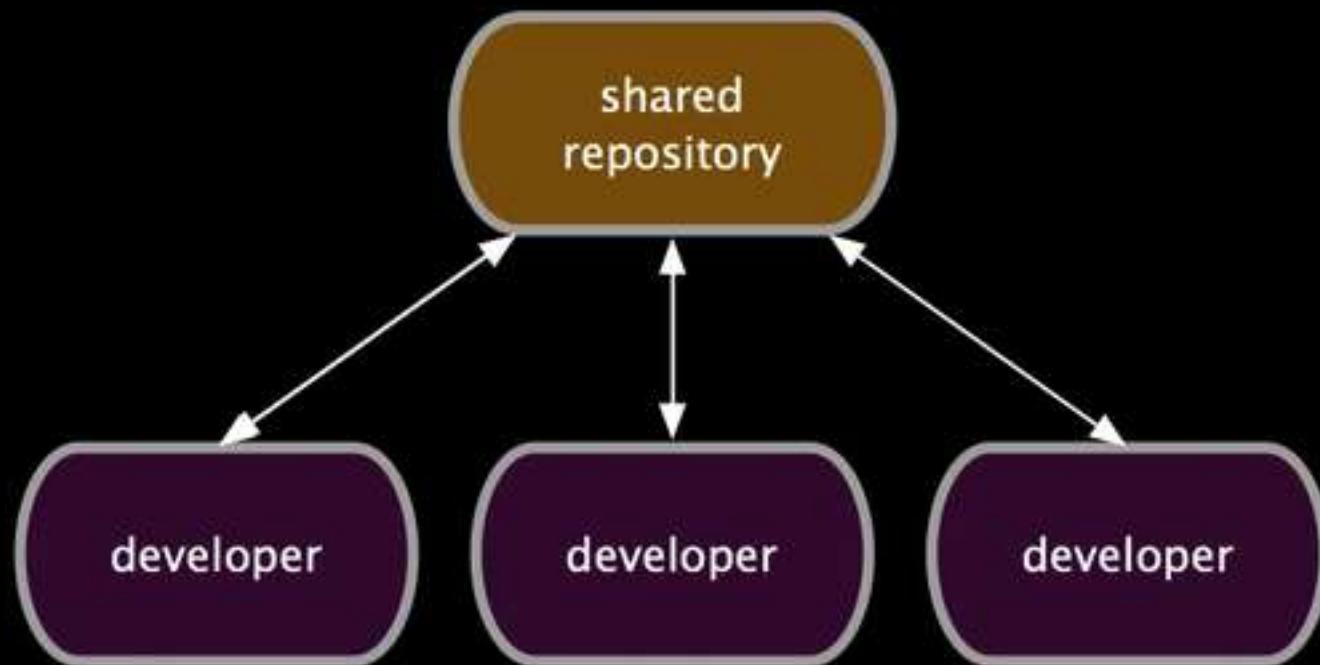
Backup: Undo or refer to old stuff



Branch: Maintain old release while working on new



Collaborate: Work in parallel with teammates



Version Control Systems (VCSs)

- Help you track/manage/distribute revisions
- Standard in modern development
- Examples:

older



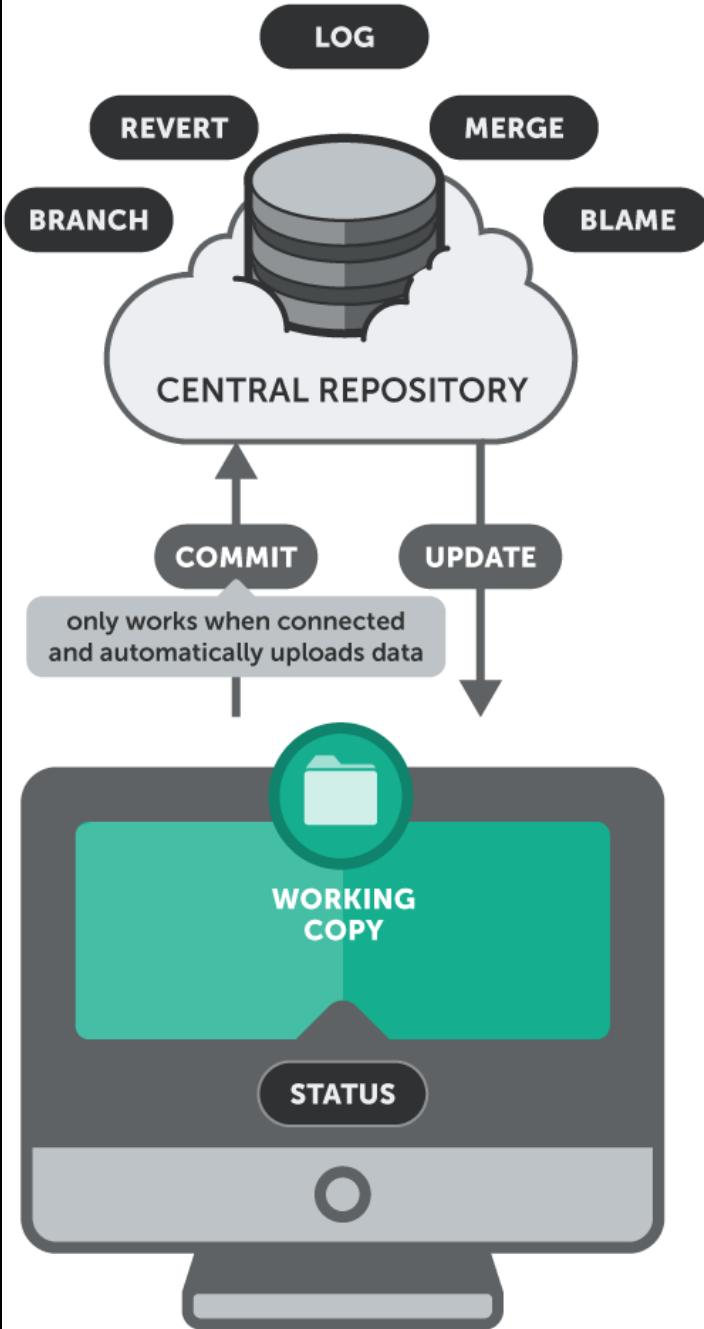
- Revision Control System (RCS)
- Concurrent Versions System (CVS)
- Subversion (SVN)

newer

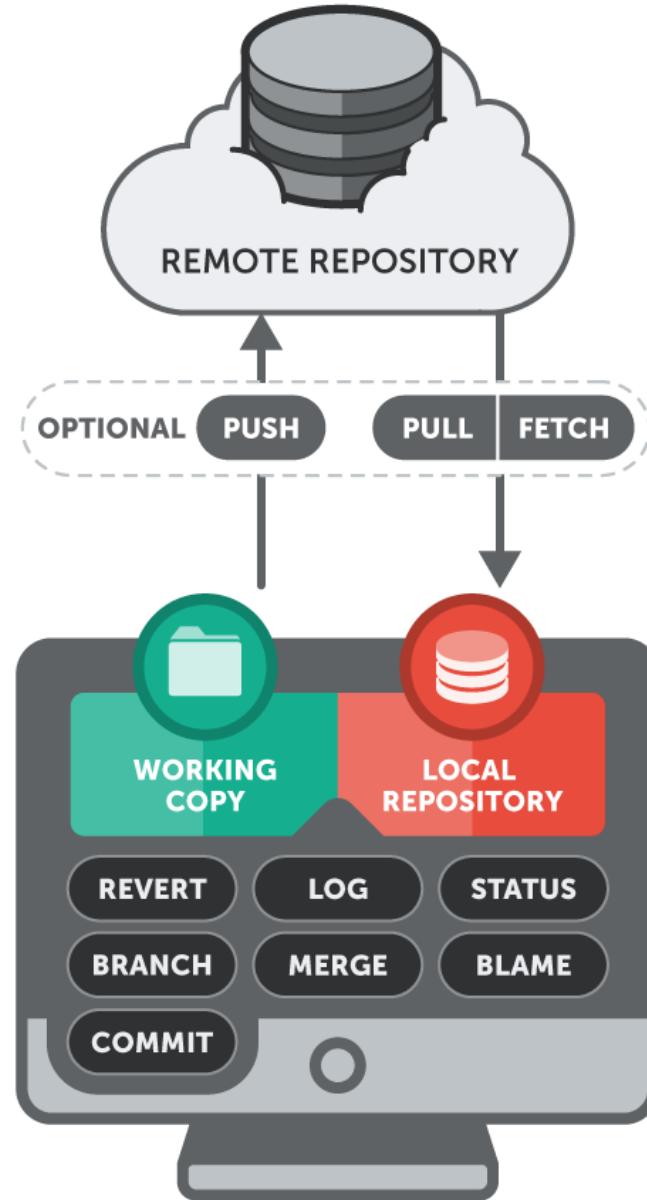
- **Git**

Our focus

SUBVERSION



GIT



Secure | https://git-scm.com/downloads

The screenshot shows the 'Downloads' section of the git-scm website. At the top right is a search bar. Below it, a large monitor icon displays the latest source release information: 'Latest source Release 2.14.1' with a link to 'Release Notes (2017-08-04)' and a 'Downloads for Windows' button. To the left of the monitor, there's a grey box containing download links for Mac OS X, Windows, Linux, and Solaris. Below this box, a note says 'Older releases are available and the Git source repository is on GitHub.' On the far left, a sidebar lists 'About', 'Documentation', 'Blog', 'Downloads' (which is currently selected), and 'Community'. A callout box in the sidebar promotes the 'Pro Git book'.

git --distributed-even-if-your-workflow-isnt

Search entire site...

About

Documentation

Blog

Downloads

GUI Clients

Logos

Community

The entire [Pro Git book](#) written by Scott Chacon and Ben Straub is available to [read online for free](#). Dead tree versions are available on [Amazon.com](#).

Downloads

Mac OS X Windows
 Linux Solaris

Older releases are available and the Git source repository is on GitHub.

GUI Clients

Git comes with built-in GUI tools (`git-gui`, `gitk`), but there are several third-party tools for users looking for a platform-specific experience.

[View GUI Clients →](#)

Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

[View Logos →](#)

<https://github.com>



Why GitHub? ▾ Team Enterprise Explore ▾ Marketplace Pricing ▾

Search GitHub



Sign in

Sign up

Where the world builds software

Millions of developers and companies build, ship, and maintain their software on GitHub—the largest and most advanced development platform in the world.

Email address

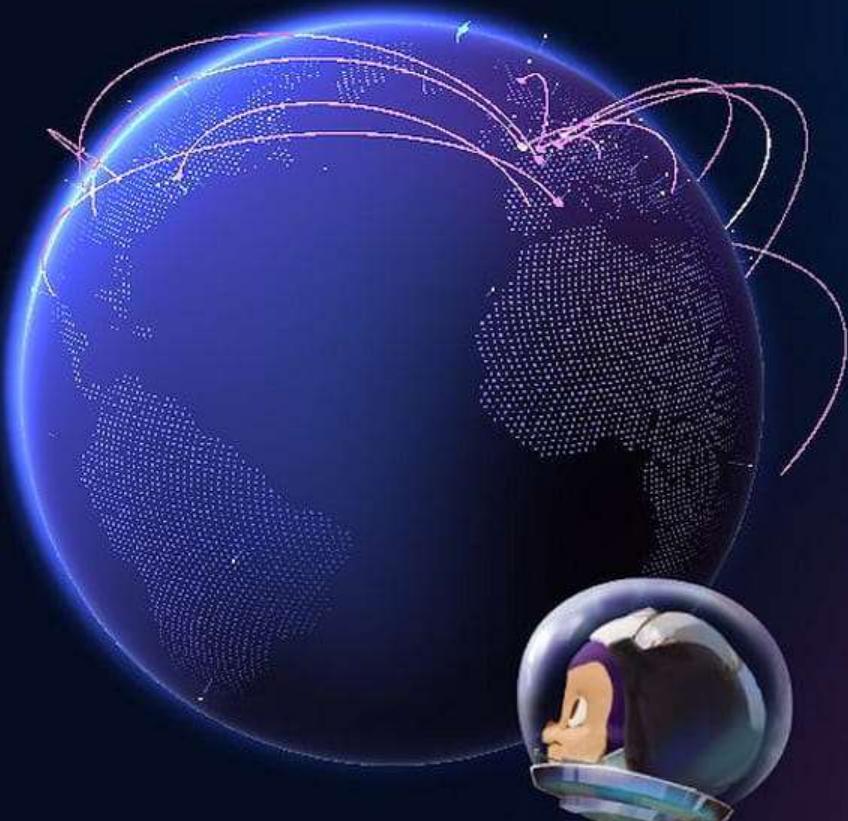
Sign up for GitHub

56+ million
Developers

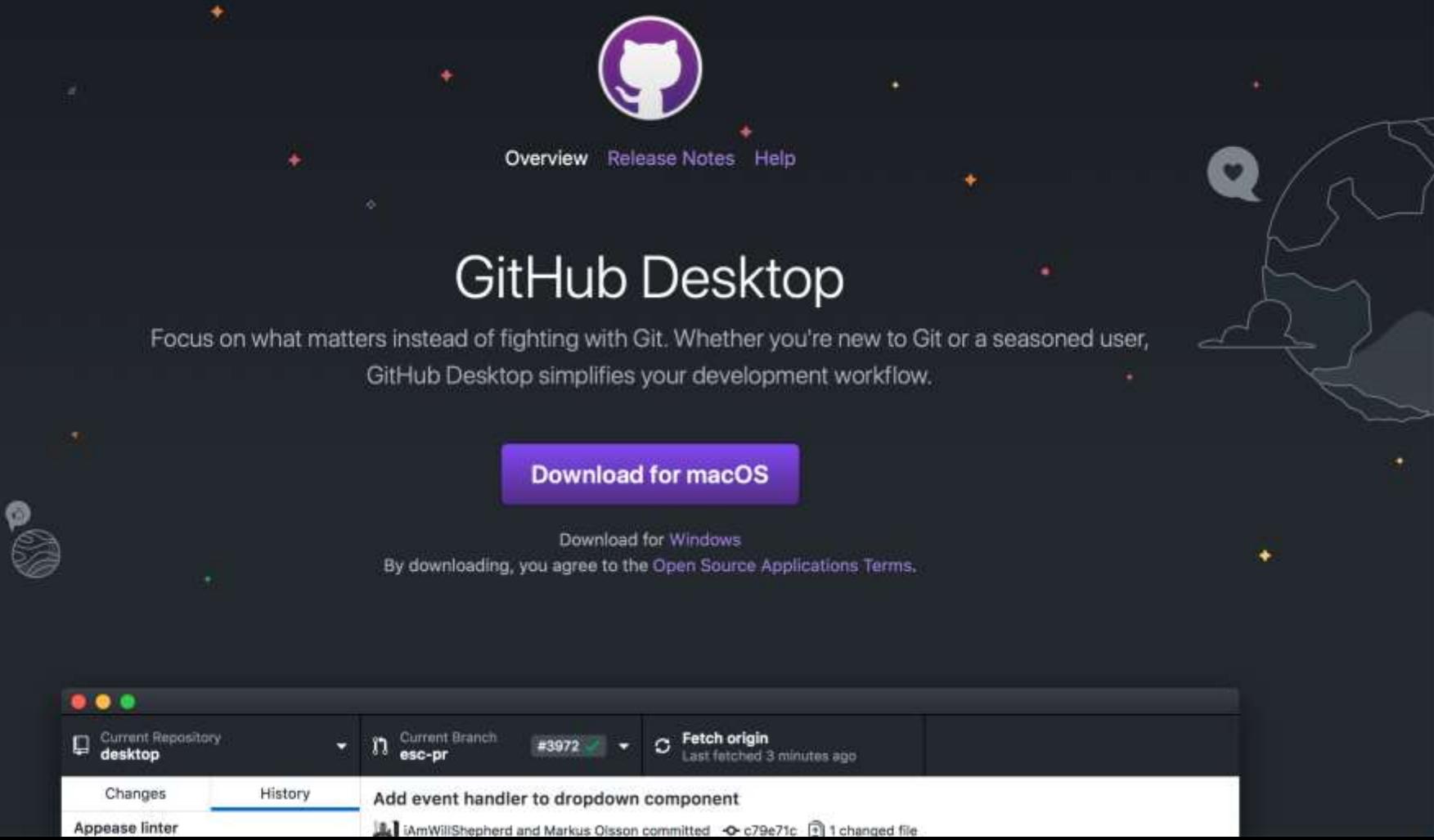
3+ million
Organizations

100+ million
Repositories

72%
Fortune 50



<https://desktop.github.com>



The image shows the GitHub Desktop landing page and a screenshot of the application interface.

Landing Page:

- Header:** GitHub logo, Overview, Release Notes, Help.
- Title:** GitHub Desktop
- Text:** Focus on what matters instead of fighting with Git. Whether you're new to Git or a seasoned user, GitHub Desktop simplifies your development workflow.
- Buttons:** Download for macOS, Download for Windows.
- Note:** By downloading, you agree to the Open Source Applications Terms.

Application Screenshot:

- Toolbar:** Current Repository: desktop, Current Branch: esc-pr #3972, Fetch origin: Last fetched 3 minutes ago.
- Navigation:** Changes (selected), History, Appear Linter.
- Details:** Add event handler to dropdown component, AmWillShepherd and Markus Olsson committed c79e71c 1 changed file.

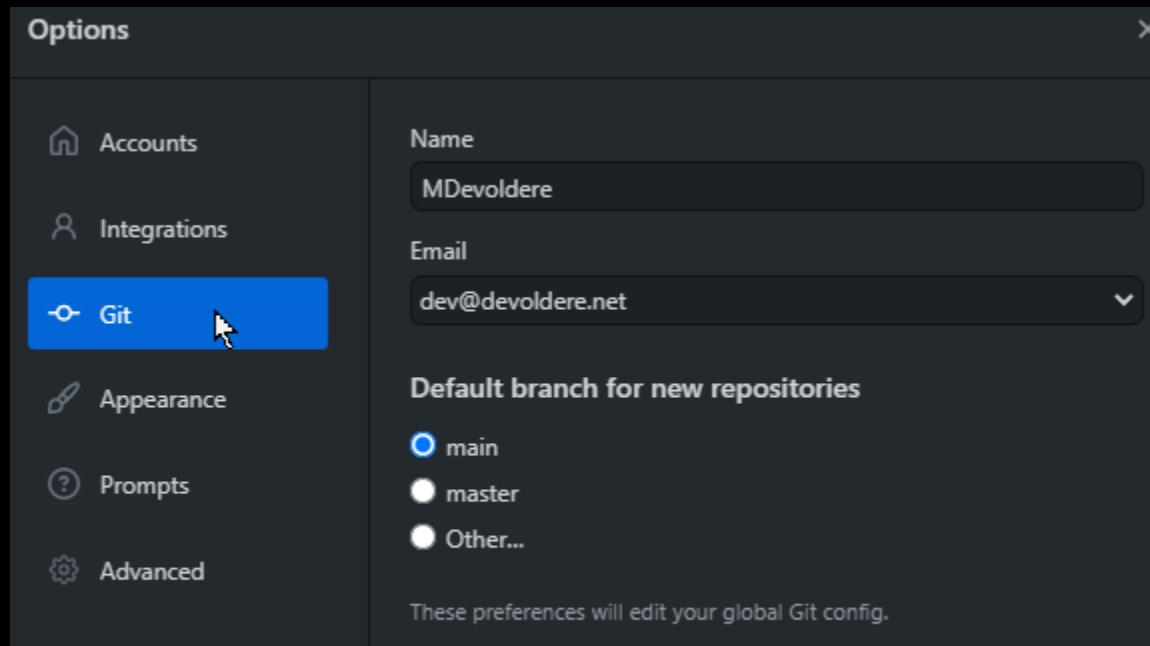
Configure your Git client

- Check config info:

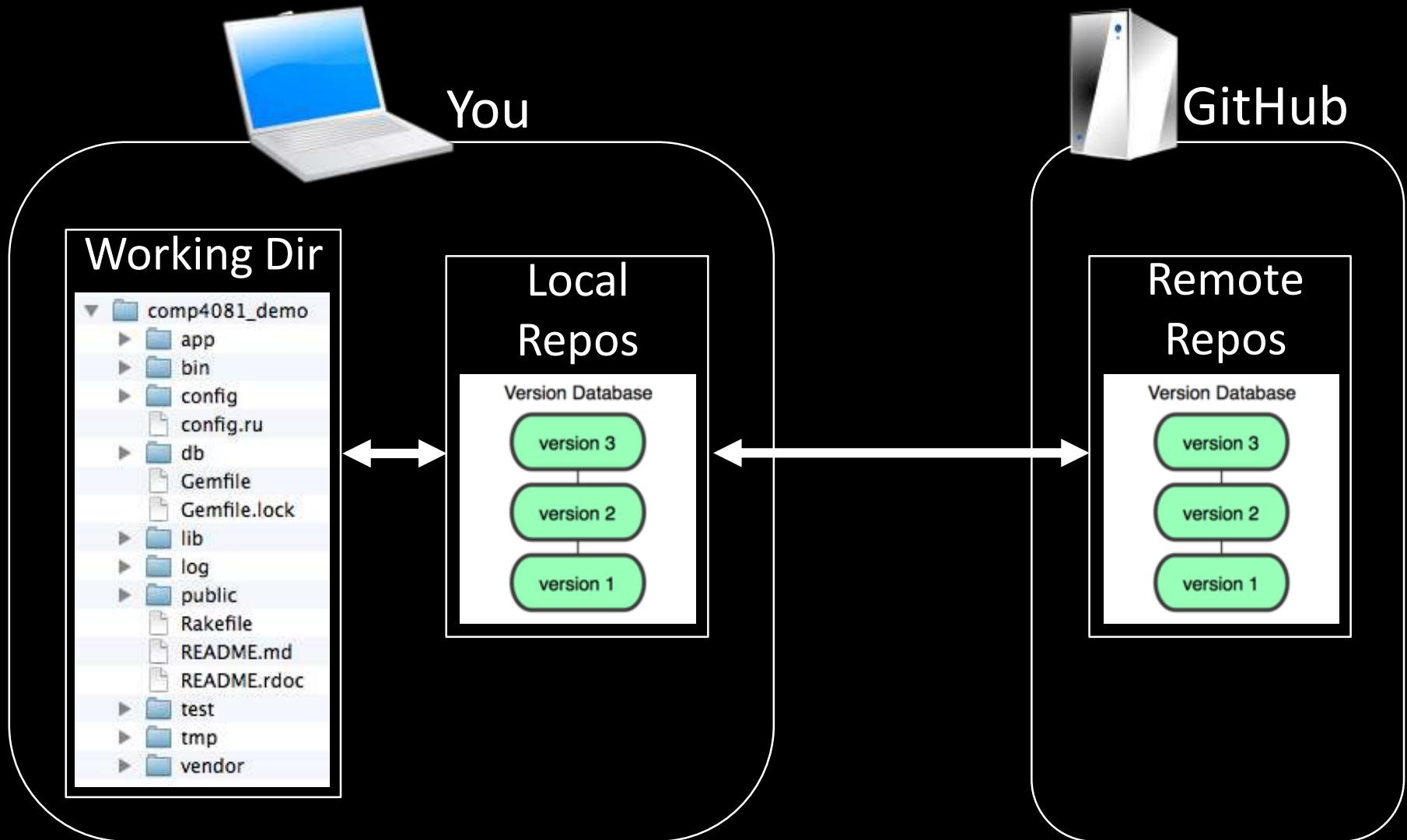
```
$ git config --list  
user.name=MDevoldere  
user.email=mdevoldere@arfpa.asso.fr
```

- Fix if necessary:

```
$ git config --global user.name "John Doe"  
$ git config --global user.email jdoe@example.com
```



GitHub-User Perspective



Let's begin with an example...



You



GitHub

Log into GitHub and create a repos (with add README option)



You



GitHub

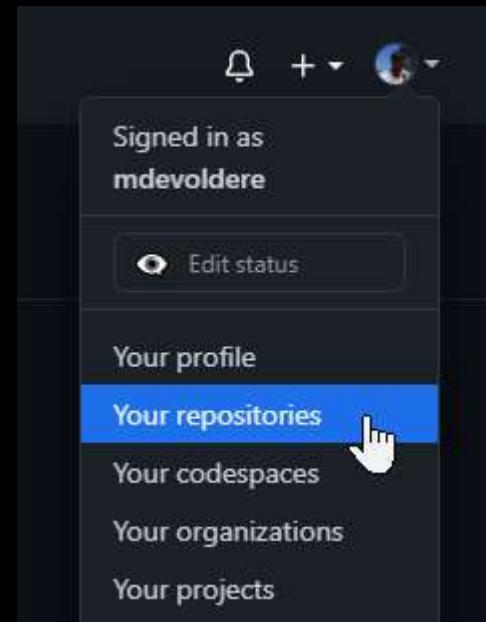
Remote
Repos

Version Database

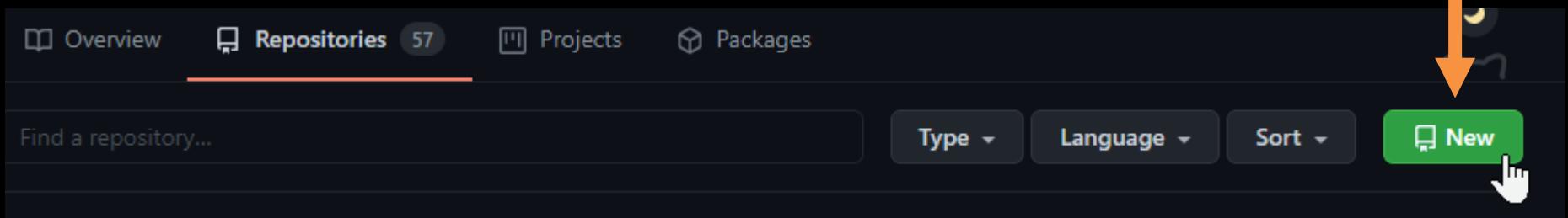
version 1

<https://github.com>

1. Go to your Repos page



2. On Repos page, click « New »



Repository template

Start your repository with a template repository's contents.

No template ▾

Owner *



mdevoldere ▾

Repository name *

my-repo



Great repository names are short and memorable. Need inspiration? How about [special-journey](#)?

Description (optional)

 **Public**

Anyone on the internet can see this repository. You choose who can commit.

 **Private**

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

Add a README file

This is where you can write a long description for your project. [Learn more](#).

Add .gitignore

Choose which files not to track from a list of templates. [Learn more](#).

.gitignore template: [VisualStudio](#) ▾

Choose a license

A license tells others what they can and can't do with your code. [Learn more](#).

This will set  [main](#) as the default branch. Change the default name in your [settings](#).

Create repository

Signed in as
mdevoldere

Edit status

Your profile

Your repositories (selected)

Your codespaces

Your organizations

Your projects

infradev_2004

edu-dataset

Find a repository... Type - Language - Sort - New

Star Star

mdevoldere / **edu-dataset**

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 3 branches 0 tags Go to file Add file Code

 mdevoldere Initial commit 41960d5 6 days ago 1 commit

.gitignore Initial commit 6 days ago

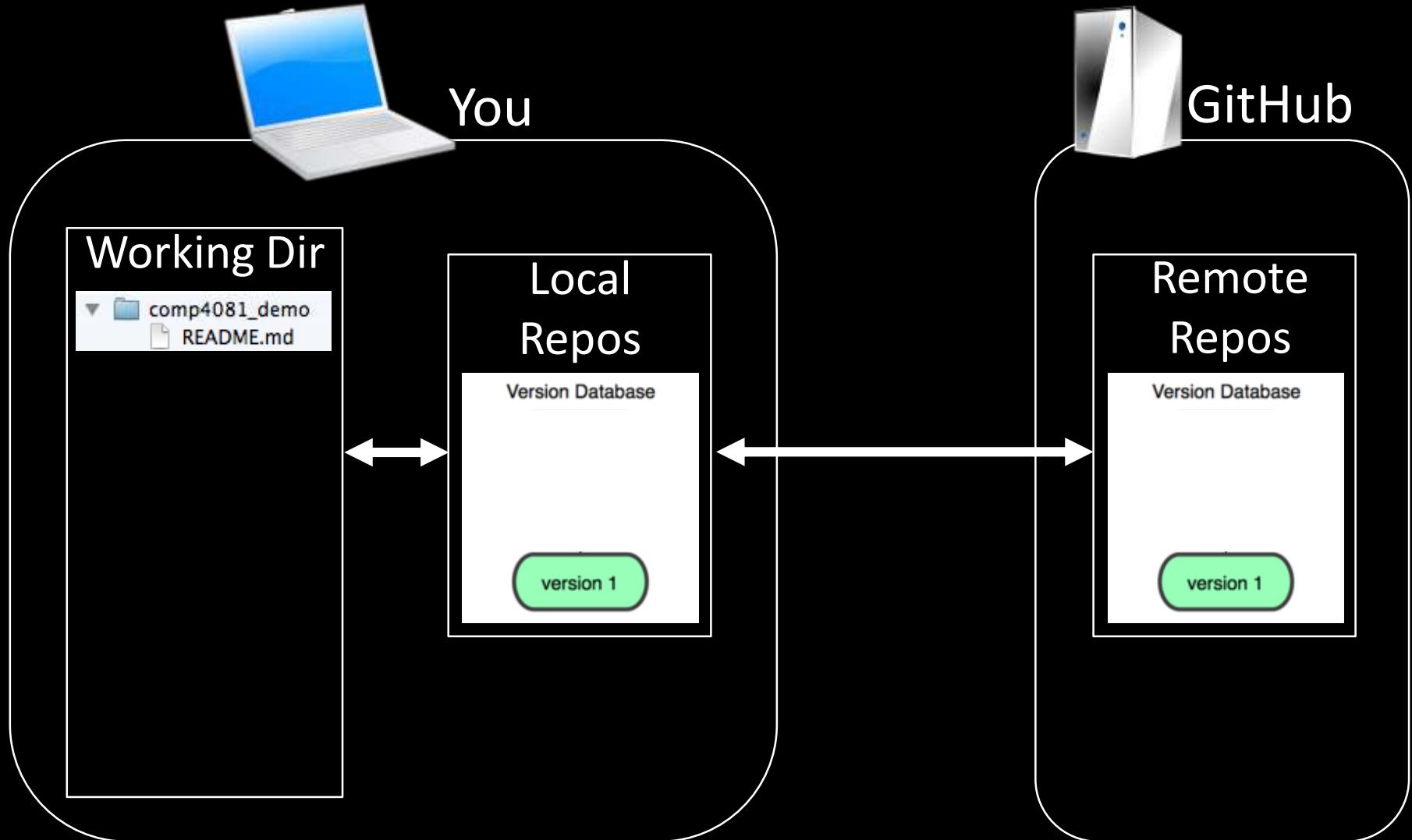
LICENSE Initial commit 6 days ago

README.md Initial commit 6 days ago

README.md

edu-dataset

```
$ git clone https://github.com/arfp/comp4081_demo.git
```



 mdevoldere / edu-dataset

Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

main ▾

3 branches

0 tags

Go to file

Add file ▾

Code



mdevoldere Initial commit

.gitignore

Initial commit

LICENSE

Initial commit

README.md

Initial commit

README.md

edu-dataset

Clone

HTTPS SSH GitHub CLI

<https://github.com/mdevoldere/edu-dataset>

Use Git or checkout with SVN using the web URL.

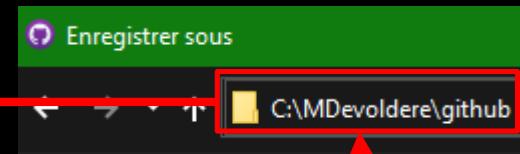
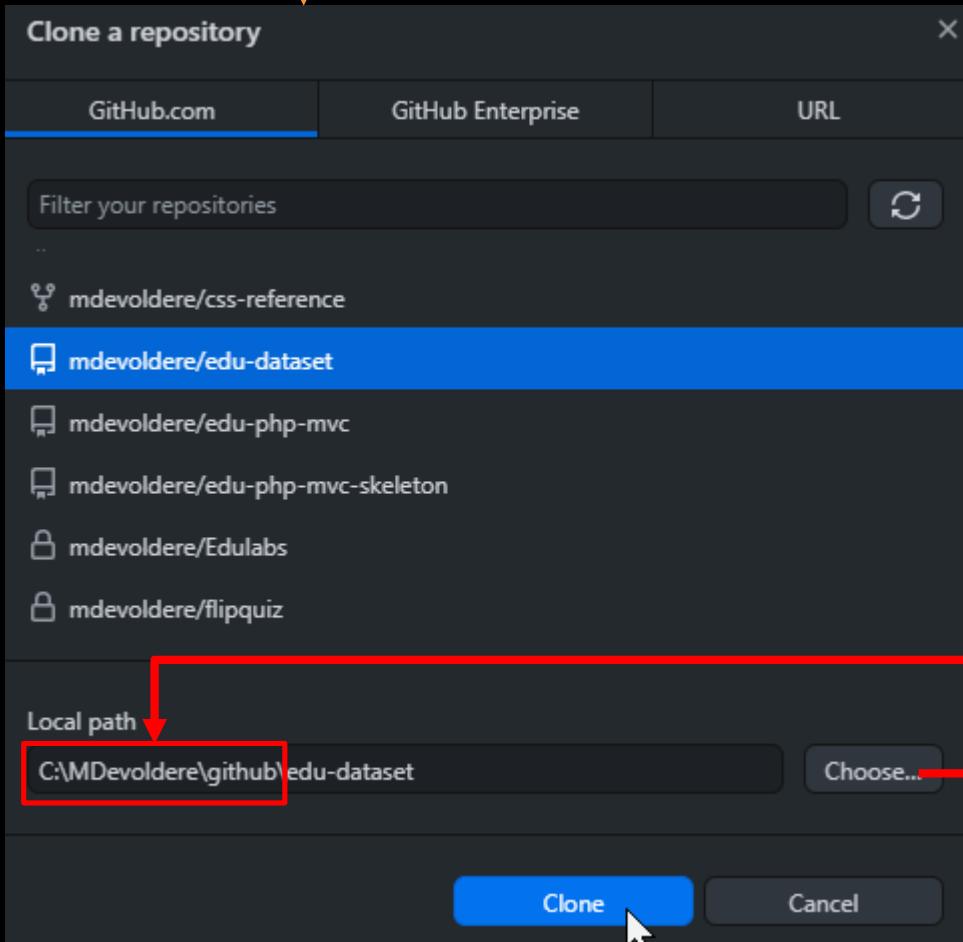
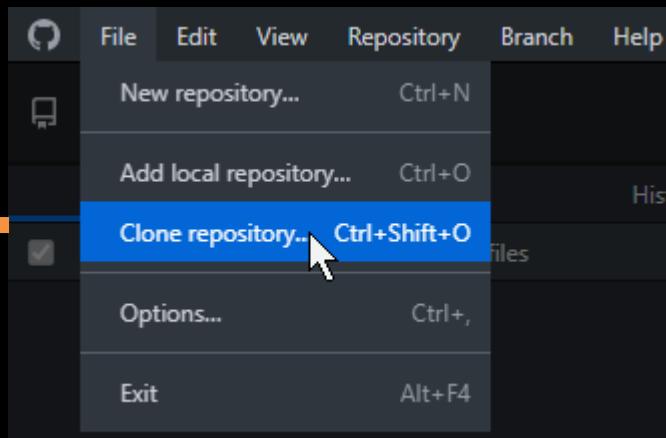
Open with GitHub Desktop

Open with Visual Studio

Download ZIP

```
git clone https://github.com/mdevoldere/edu-dataset.git
```

GitHub Desktop



Clone

Cancel

⬇️ Cloning edu-dataset

Receiving objects: 100% (5/5), 15.13 KiB | 1.38 MiB/s, done.



Disque local (C:) > MDevoldere > github > edu-dataset >

Nom	Modifié le	Type	Taille
.git	07/05/2021 11:17	Dossier de fichiers	
.gitignore	07/05/2021 11:17	Fichier source Git l...	7 Ko
LICENSE	07/05/2021 11:17	Fichier	35 Ko
README.md	07/05/2021 11:17	Markdown File	1 Ko

Local Repository

Disque local (C:) > MDevoldere > github > edu-dataset	
Nom	Modifié le
.git	07/05/2021 11:17
.gitignore	07/05/2021 11:17
LICENSE	07/05/2021 11:17
README.md	07/05/2021 11:17

Working Directory (the files you are working on)

edu-dataset

Fichier Accueil Partage Affichage

Volet de navigation Volet de visualisation

Volets

Disposition

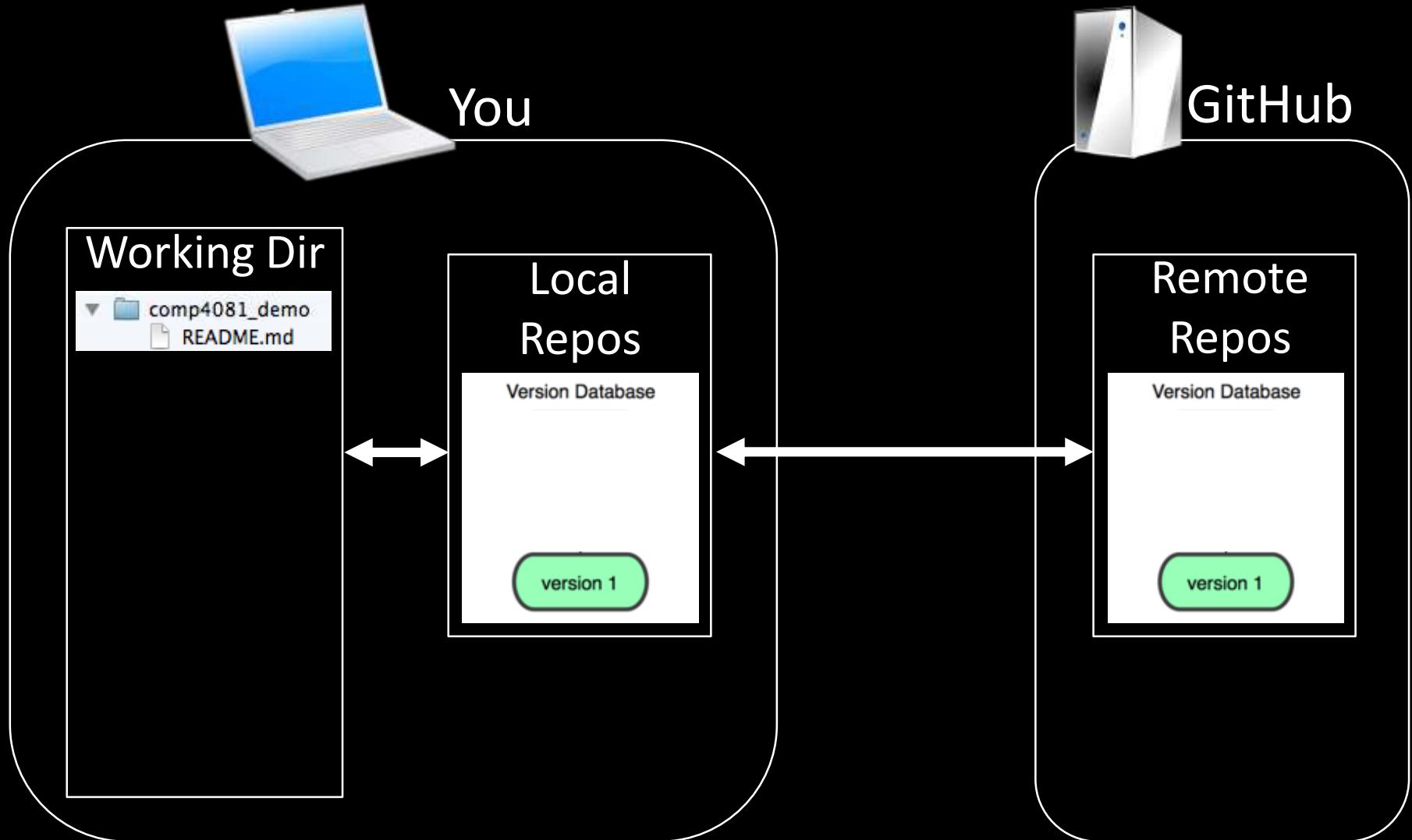
Affichage actuel

Éléments masqués

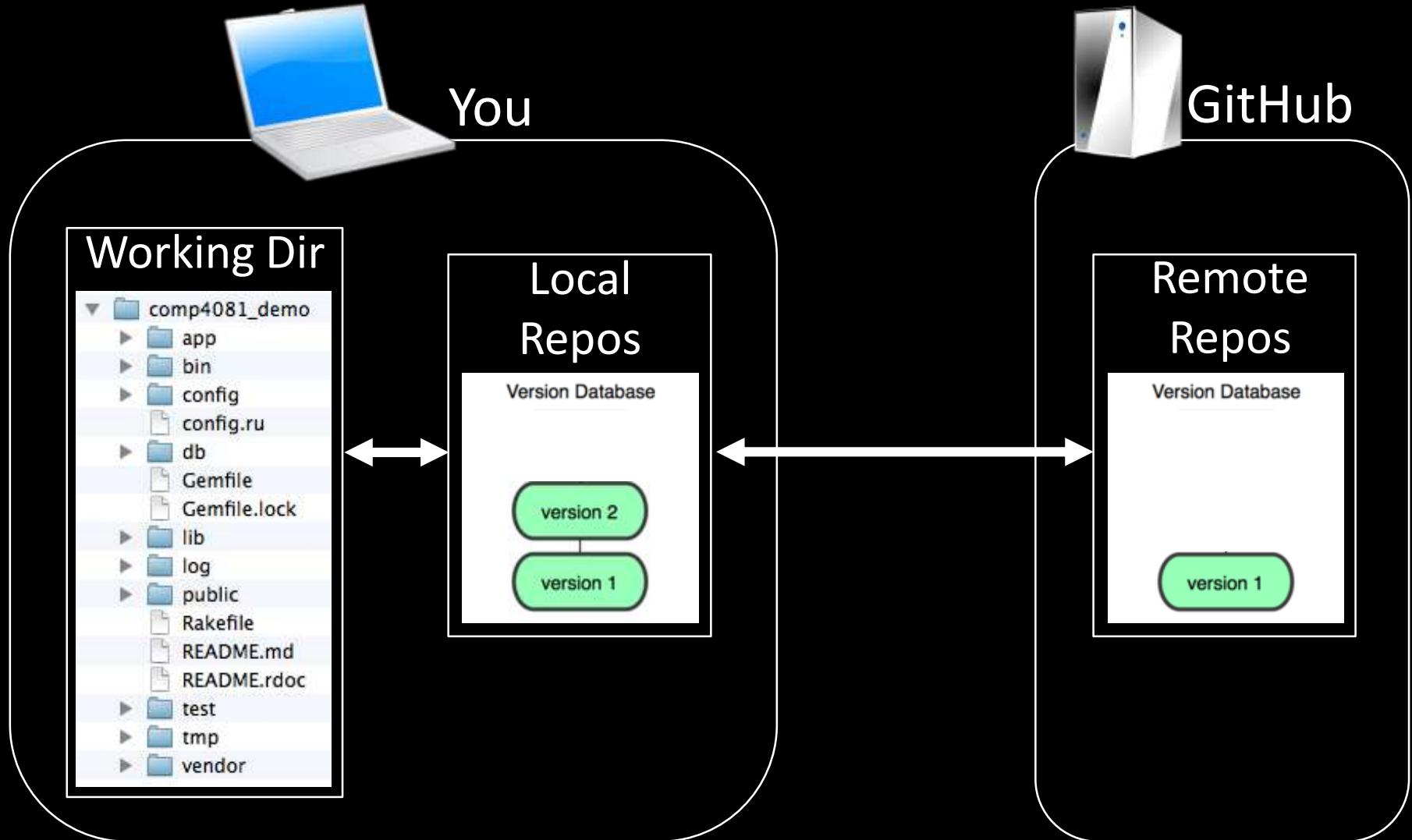
Accès rapide Bureau Téléchargements Documents

Nom	Modifié le	Type	Taille
.git	07/05/2021 11:23	Dossier de fichiers	
.gitignore	07/05/2021 11:17	Fichier source Git l...	7 Ko
LICENSE	07/05/2021 11:17	Fichier	35 Ko
README.md	07/05/2021 11:17	Markdown File	1 Ko

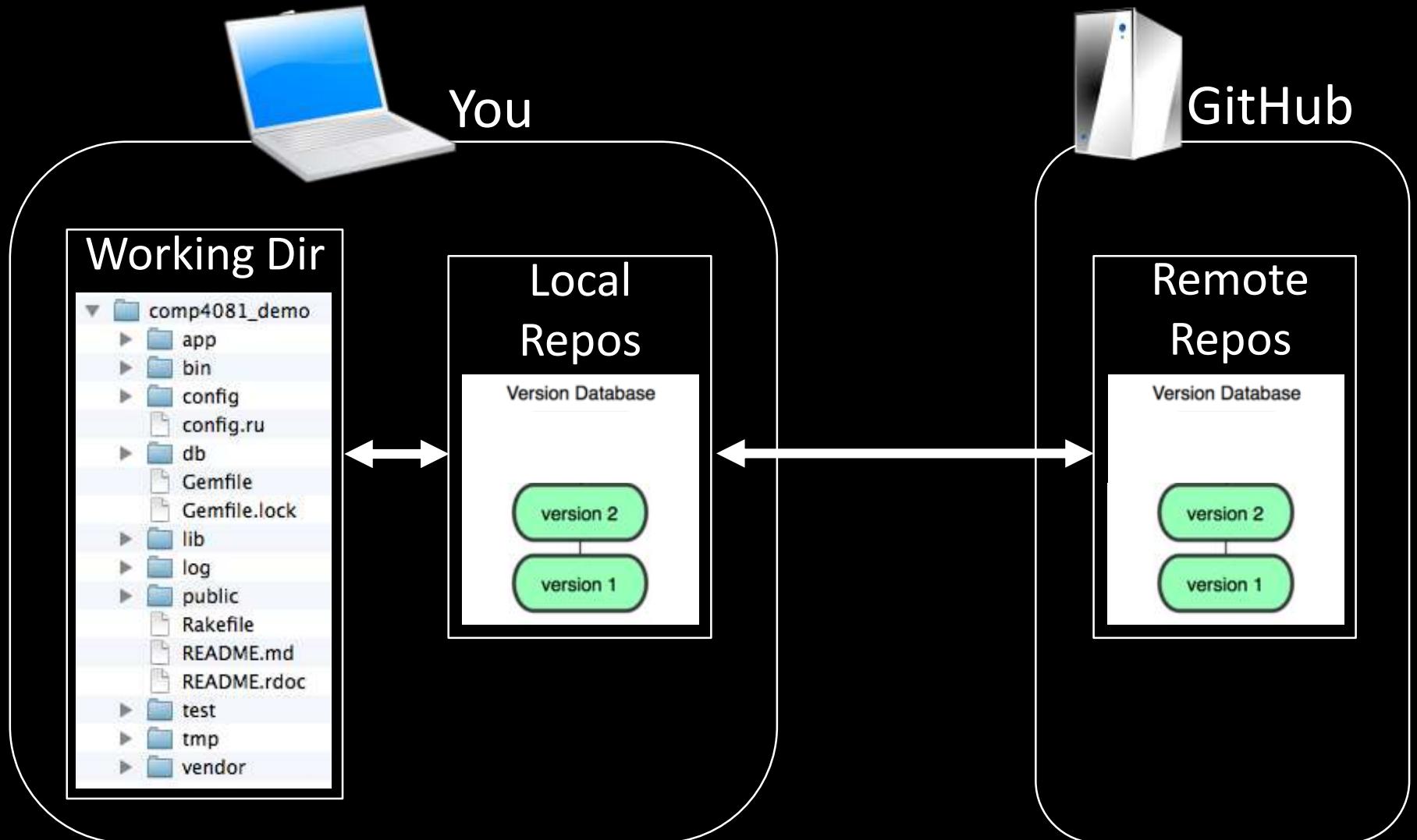
```
$ git clone https://github.com/arfp/comp4081_demo.git
```



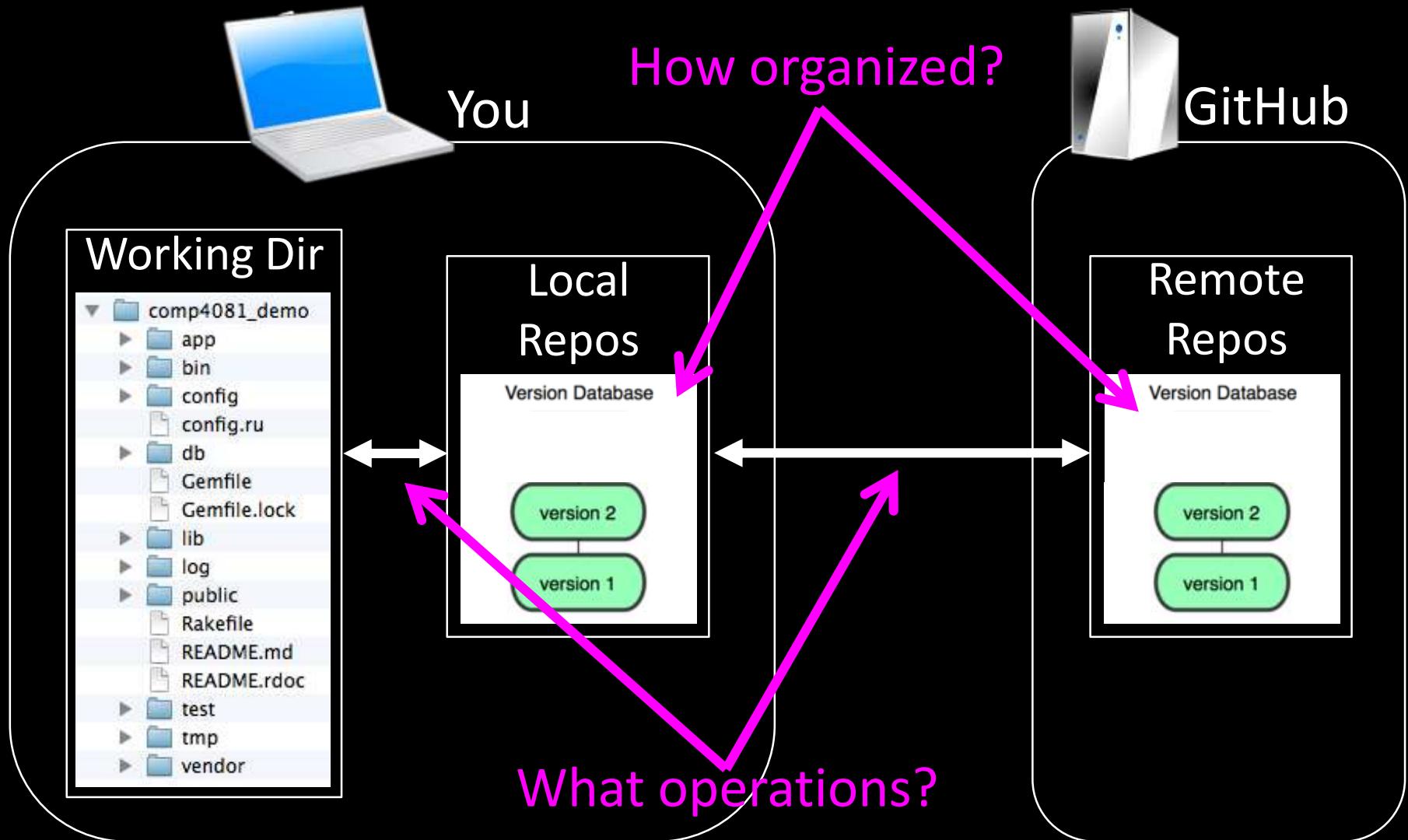
```
$ cd comp4081_demo  
// Add/edit files  
$ git add -A  
$ git commit -m "Created project skeleton"
```



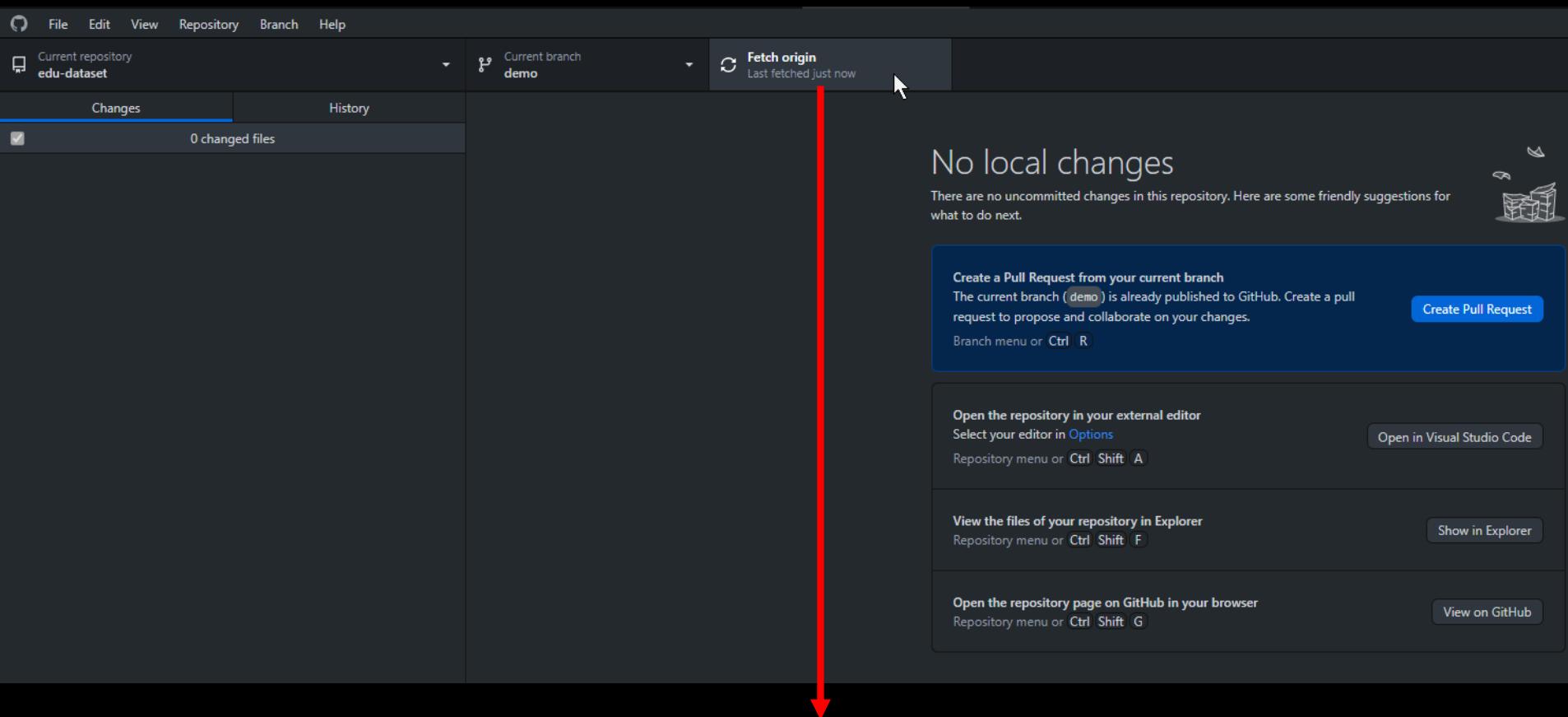
```
$ git push
```



Questions to answer



Important: before starting to work



Update your local repository
to make sure you're in sync
with the remote repository

git fetch origin

Add / Edit files

The screenshot shows a code editor interface with a dark theme. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The current file is Program.cs, indicated by the tab on the right. The Explorer sidebar on the left shows a project structure under 'EDU-DATASET': my-project (containing obj, my-project.csproj, and Program.cs), .gitignore, LICENSE, and README.md. The Program.cs file is selected and highlighted with a red box. The main editor area displays the following C# code:

```
1  using System;
2
3  namespace my_project
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              Console.WriteLine("Hello World!");
10         }
11     }
12 }
```

File Edit View Repository Branch Help

Current repository
edu-dataset

Current branch
demo

Fetch origin
Last fetched 6 minutes ago

Changes 2 History my-project\Program.cs

2 changed files

my-project\my-project.csproj

my-project\Program.cs

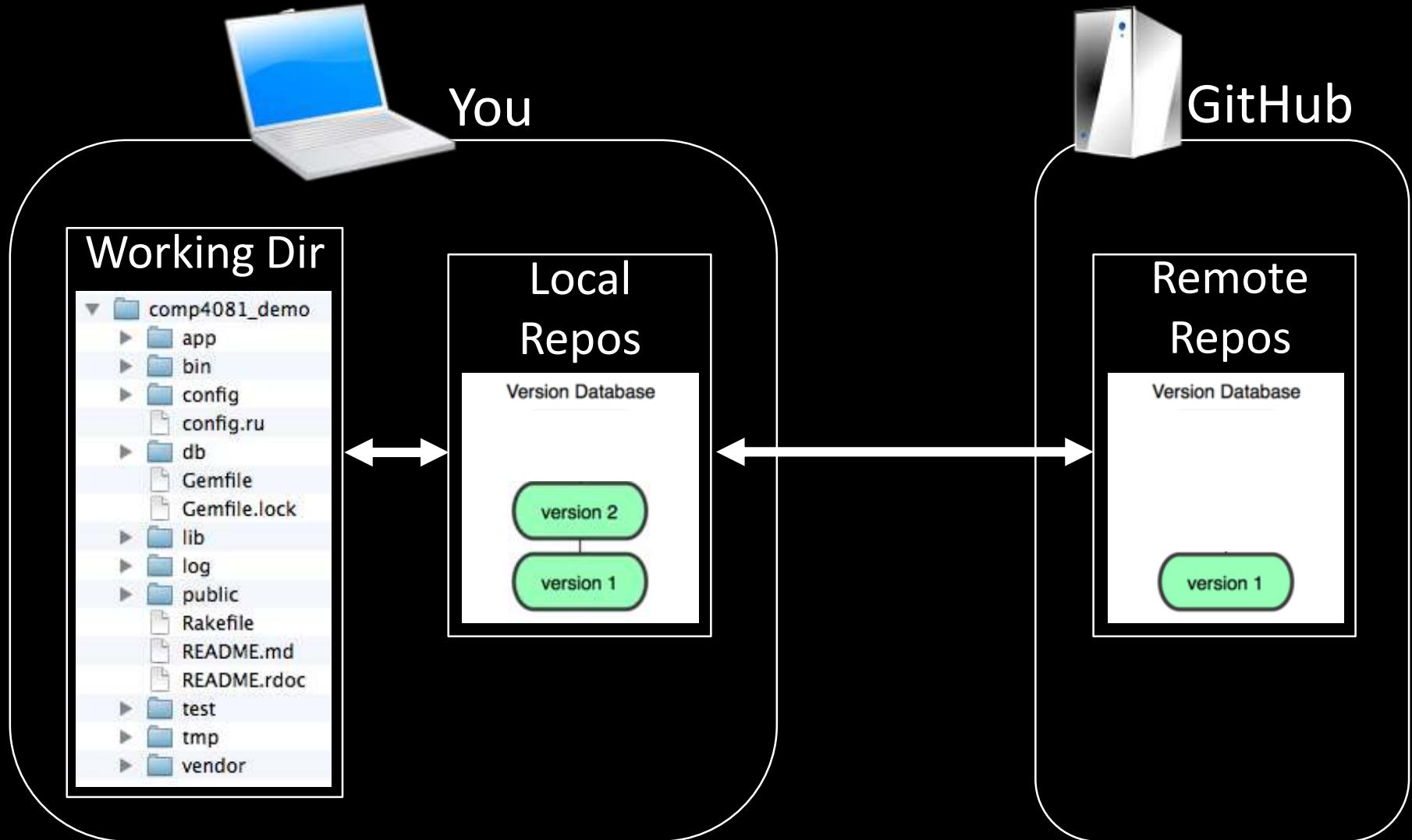
```
@@ -0,0 +1,12 @@
1 +•using System;
2 +
3 +namespace my_project
4 +{
5 +    class Program
6 +    {
7 +        static void Main(string[] args)
8 +        {
9 +            Console.WriteLine("Hello World!");
10 +
11 +
12 +    }
13 +}
```

create my-project

A fantastic Hello world App !

Commit to demo

```
$ cd comp4081_demo  
// Add/edit files  
$ git add -A  
$ git commit -m "Created project skeleton"
```



The screenshot shows a GitHub desktop application interface. At the top, there's a navigation bar with File, Edit, View, Repository, Branch, and Help. Below the navigation bar, the current repository is set to "edu-dataset" and the branch is "demo". A status bar indicates a push to "origin" was made 11 minutes ago.

The main area displays a commit history. The first commit is highlighted in blue and titled "create my-project" by user "MDevoldere" 4 minutes ago. The commit message is "A fantastic Hello world App !". It shows changes to two files: "my-project\Program.cs" and "my-project\my-project.csproj".

The "History" tab is selected, and a tooltip "Select branch to compare..." is visible over the "Changes" tab.

The code diff for "my-project\Program.cs" is shown on the right:

```
@@ -0,0 +1,12 @@
1 +using System;
2 +
3 +namespace my_project
4 +{
5 +    class Program
6 +    {
7 +        static void Main(string[] args)
8 +        {
9 +            Console.WriteLine("Hello World!");
10 +        }
11 +    }
12 +}
```

File Edit View Repository Branch Help

Current repository
edu-dataset

Changes History 

Select branch to compare...

create my-project MDevoldere • 4m

Initial commit MDevoldere • 6d

Current branch demo

Push origin Last fetched 11 minutes ago 1 ↑

create my-project MDevoldere -O- 6e33b37 ± 2 changed files ⚙ New

A fantastic Hello world App !

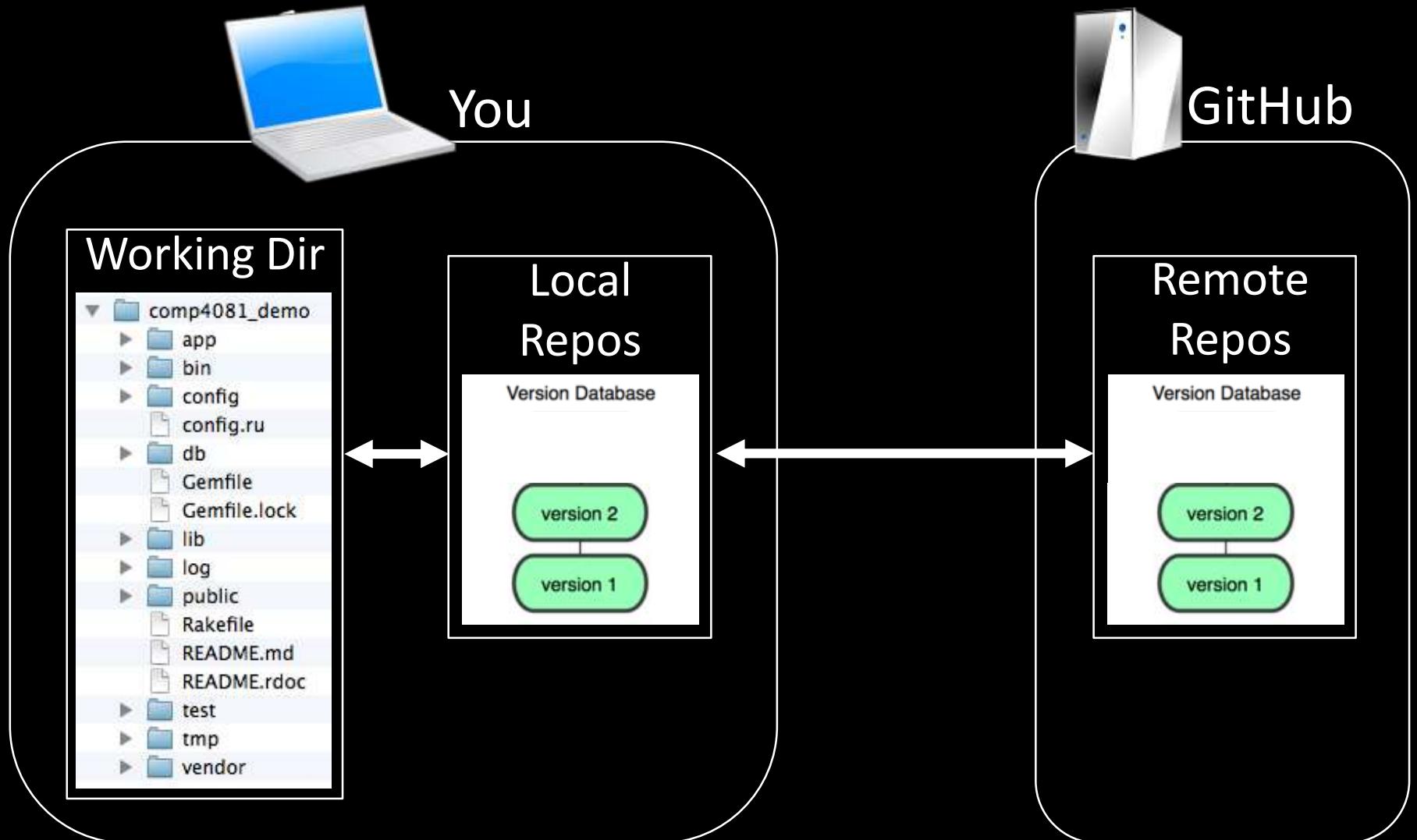
my-project\Program.cs

```
@@ -0,0 +1,12 @@
1 +using System;
2 +
3 +namespace my_project
4 +{
5 +    class Program
6 +    {
7 +        static void Main(string[] args)
8 +        {
9 +            Console.WriteLine("Hello World!");
10 +        }
11 +    }
12 +}
```

my-project\my-project.csproj

 Pushing to origin
Hang on...

```
$ git push
```



File Edit View Repository Branch Help

Current repository: edu-dataset

Changes

Push Ctrl+P

Pull Ctrl+Shift+P

Remove... Ctrl+Backspace

View on GitHub Ctrl+Shift+G

Open in Command Prompt Ctrl+`

Show in Explorer Ctrl+Shift+F

Open in Visual Studio C... Ctrl+Shift+A

Create issue on GitHub Ctrl+I

Repository settings...

Current branch: demo

Last fetched 6 minutes ago

Fetch origin

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Create a Pull Request from your current branch
The current branch (`demo`) is already published to GitHub. Create a pull request to propose and collaborate on your changes.
Branch menu or `Ctrl + R`

Create Pull Request

Open the repository in your external editor
Select your editor in [Options](#)
Repository menu or `Ctrl + Shift + A`

Open in Visual Studio Code

View the files of your repository in Explorer
Repository menu or `Ctrl + Shift + F`

Show in Explorer

Open the repository page on GitHub in your browser
Repository menu or `Ctrl + Shift + G`

View on GitHub

<https://github.com/mdevoldere/edu-dataset>

A screenshot of a GitHub repository page for the user mdevoldere. The repository name is "my-project". The commit history shows the following initial commits:

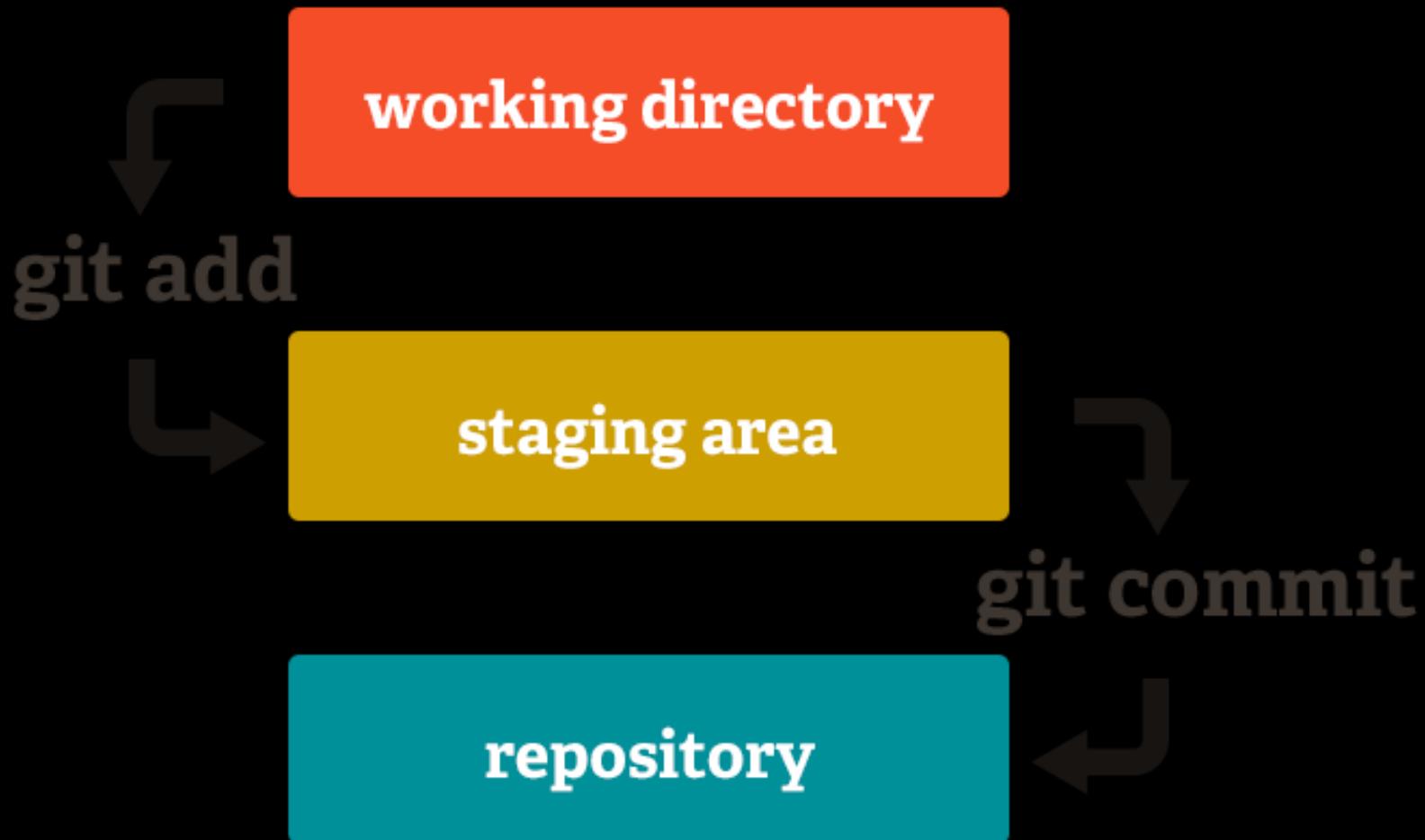
- my-project (create my-project) by mdevoldere 9 minutes ago
- .gitignore (Initial commit) by mdevoldere 6 days ago
- LICENSE (Initial commit) by mdevoldere 6 days ago
- README.md (Initial commit) by mdevoldere 6 days ago

Below the commit history, there are two sections for commit logs:

- Commits on May 7, 2021**
 - create my-project by mdevoldere 10 minutes ago
- Commits on May 1, 2021**
 - Initial commit by mdevoldere 6 days ago

A large red arrow points downwards from the top right towards the commit history area.

How the repos is organized



File Edit View Repository Branch Help

Current repository
edu-dataset

Changes 2 History

2 changed files

my-project\my-project.csproj

my-project\Program.cs

Current branch
demo

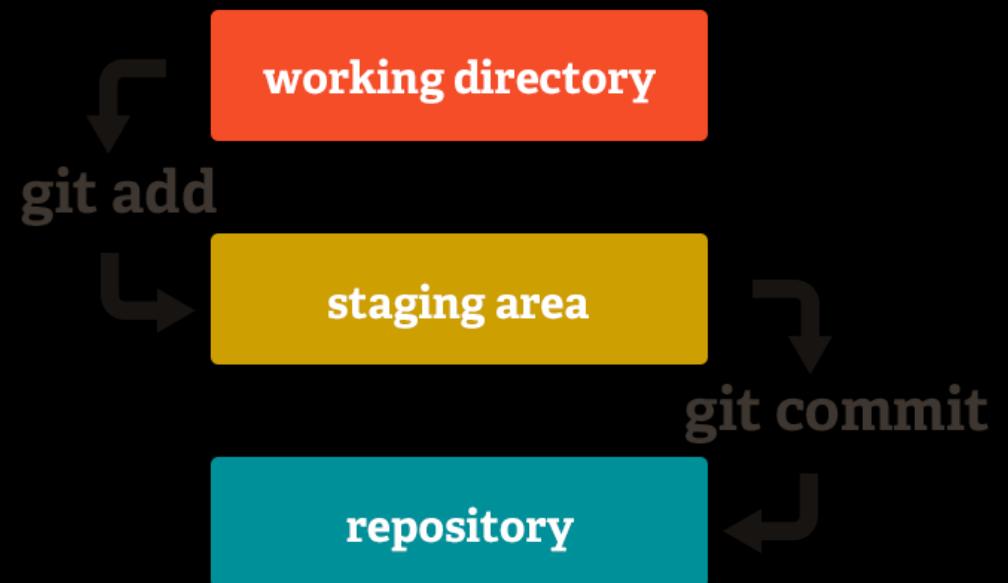
Fetch origin
Last fetched 6 minutes ago

my-project\Program.cs

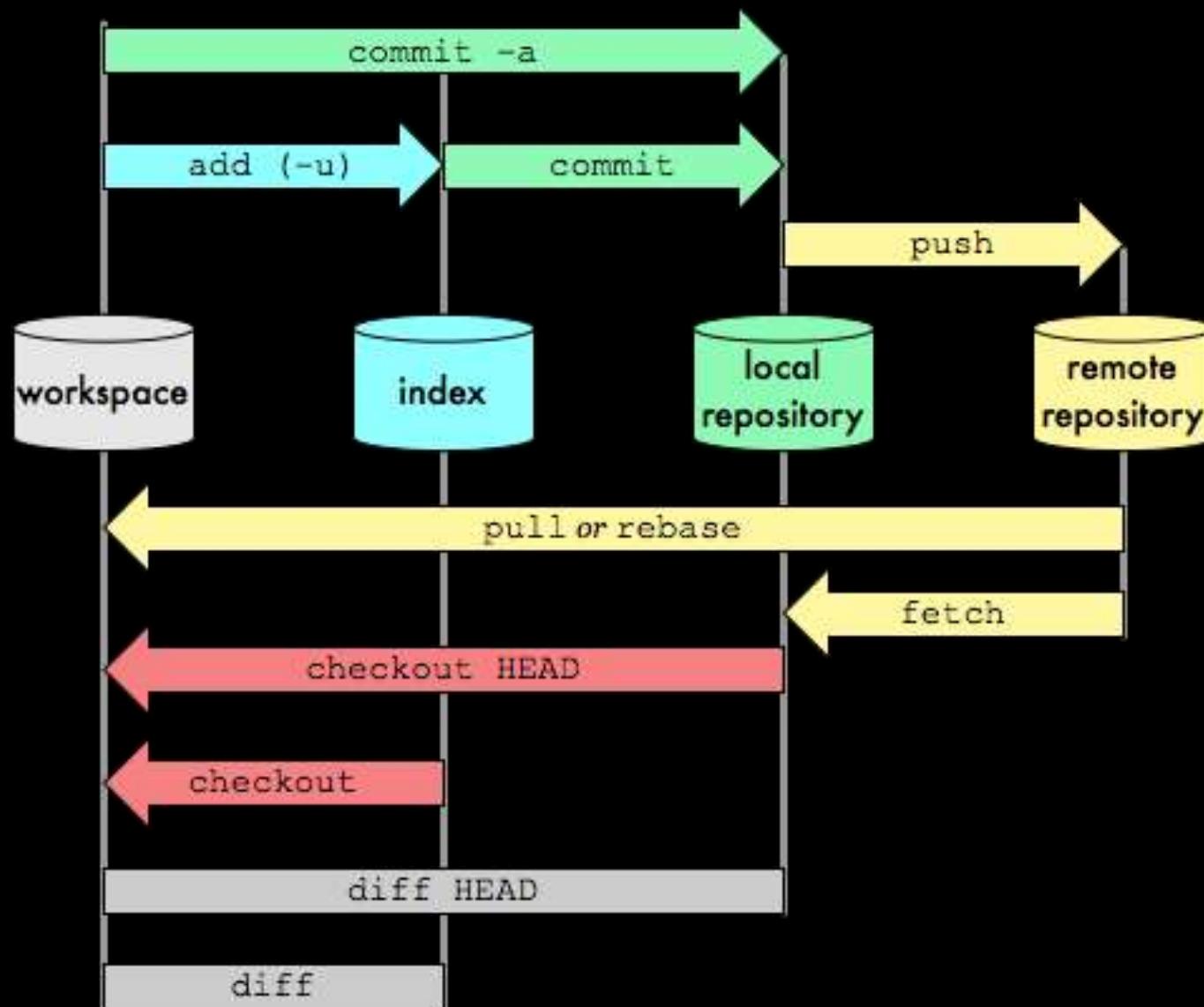
```
@@ -0,0 +1,12 @@
1 +•using System;
2 +
3 +namespace my_project
4 +{
5 +    class Program
6 +    {
7 +        static void Main(string[] args)
8 +        {
9 +            Console.WriteLine("Hello World!");
10 +
11 +
12 +    }
13 +}
```

create my-project

A fantastic Hello world App !



How the repos is organized



A screenshot of the Github Desktop application interface. At the top, there's a navigation bar with 'File', 'Edit', 'View', 'Repository', 'Branch', and 'Help'. Below the navigation bar, the current repository is set to 'edu-dataset' and the current branch is 'demo'. A status message indicates 'Push origin' was last fetched 11 minutes ago, with one update available. The main area is divided into two sections: 'Changes' and 'History'. The 'History' tab is selected, showing a commit titled 'create my-project' by 'MDevoldere' 4 minutes ago. The commit message is 'A fantastic Hello world App !'. The commit details show two files: 'my-project\Program.cs' and 'my-project\my-project.csproj'. The code editor on the right displays the content of 'Program.cs':

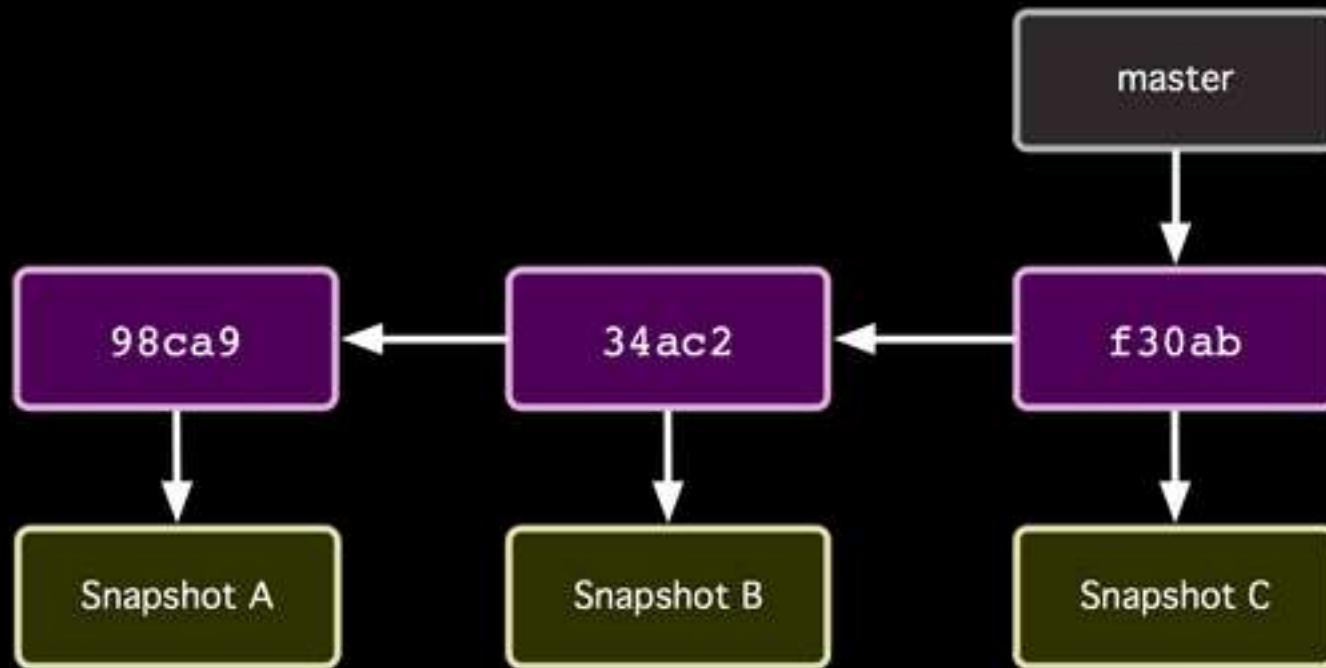
```
@@ -0,0 +1,12 @@
1 +using System;
2 +
3 +namespace my_project
4 +{
5 +    class Program
6 +    {
7 +        static void Main(string[] args)
8 +        {
9 +            Console.WriteLine("Hello World!");
10 +        }
11 +    }
12 +}
```

Github Desktop

How are commits organized?

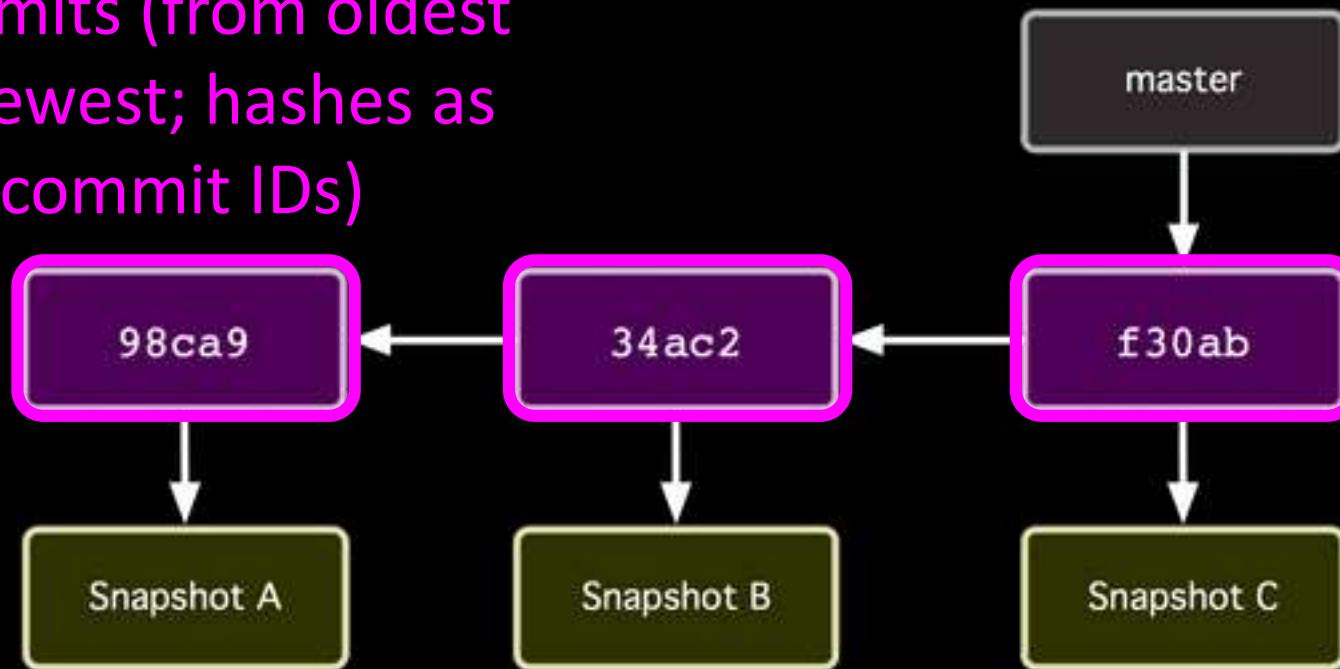
A screenshot of the Github.com website showing commit history. The top navigation bar includes 'Commits on May 7, 2021' and 'Commits on May 1, 2021'. The first commit listed is 'create my-project' by 'mdevoldere' committed 10 minutes ago. The second commit listed is 'Initial commit' by 'mdevoldere' committed 6 days ago. On the right side, there's a large 'Github.com' logo.

How the repos is organized

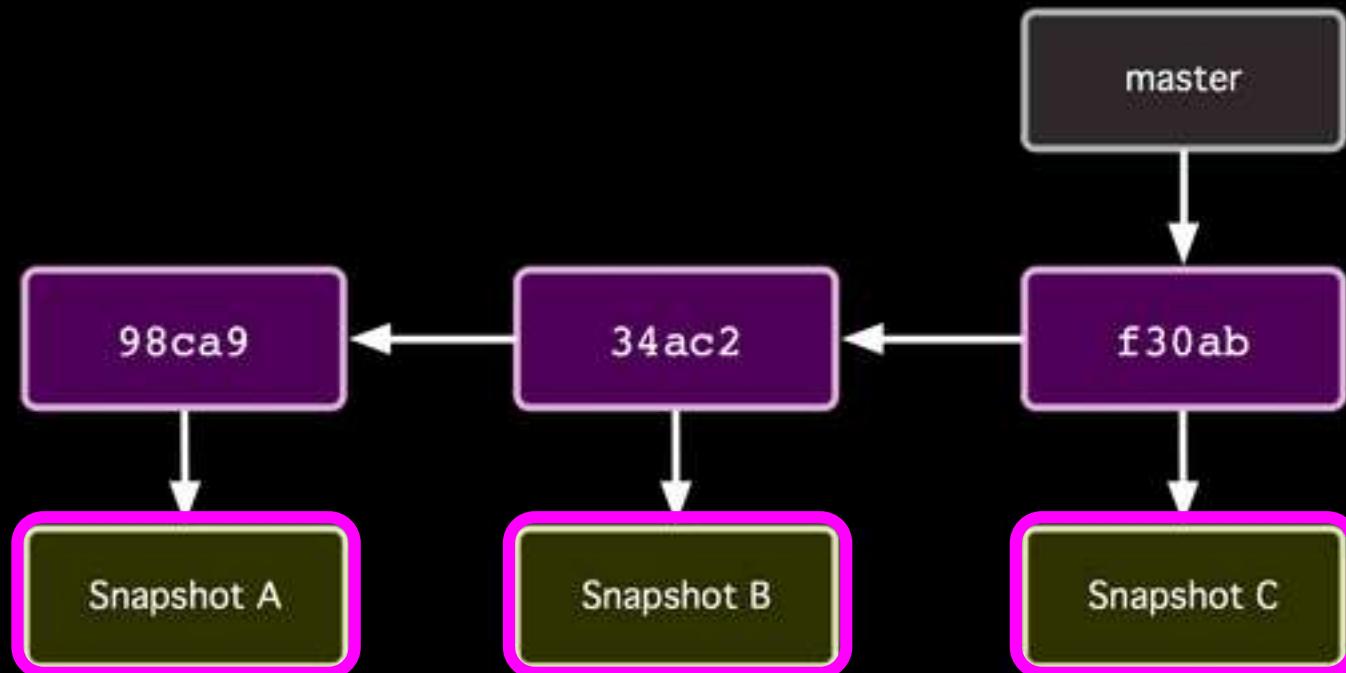


How the repos is organized

Commits (from oldest
to newest; hashes as
commit IDs)



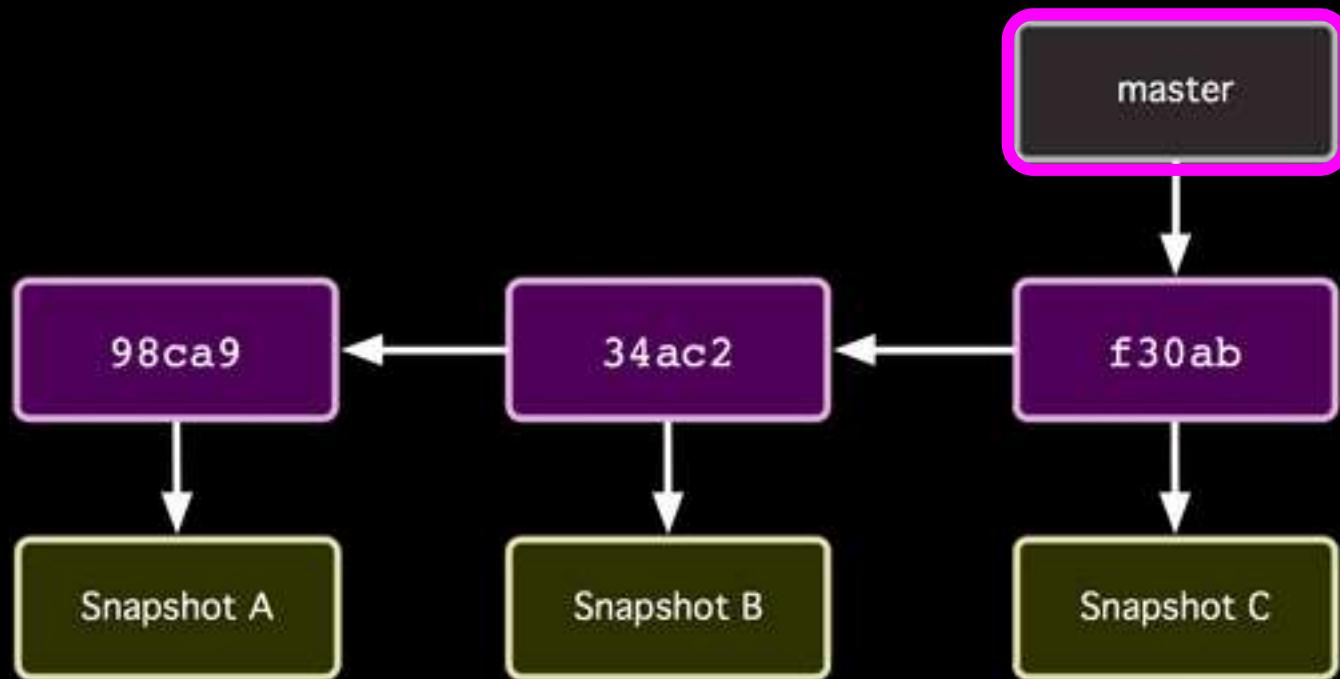
How the repos is organized



Snapshot of all files
at each commit

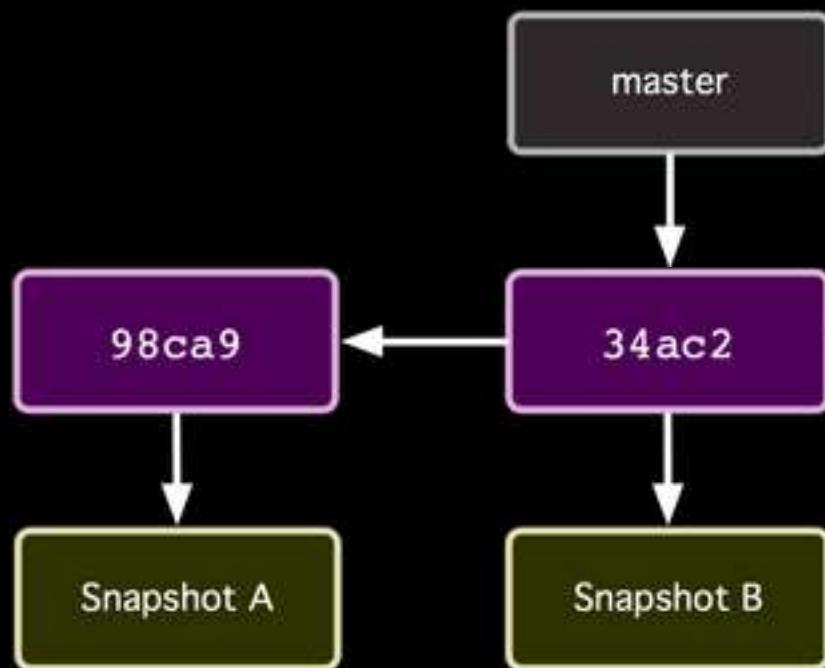
How the repos is organized

Branch (last commit)

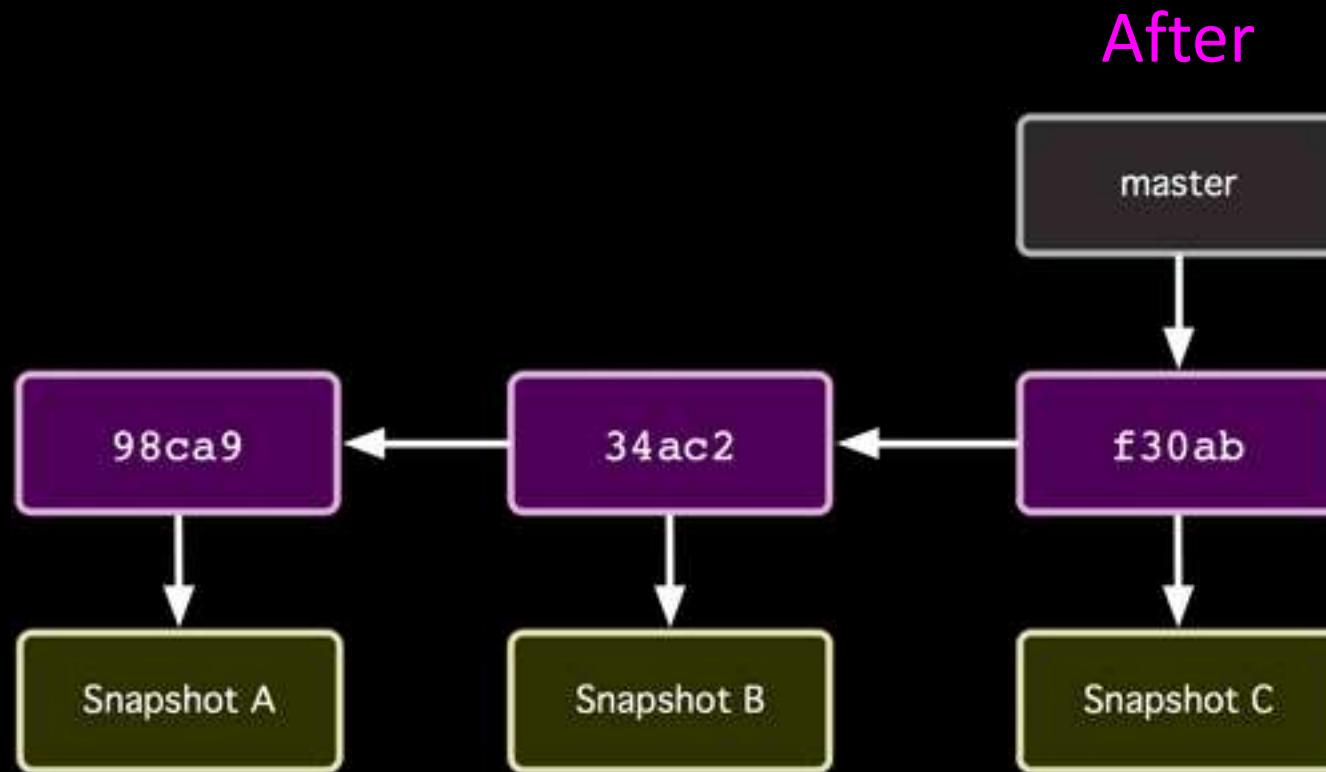


How commit works

Before



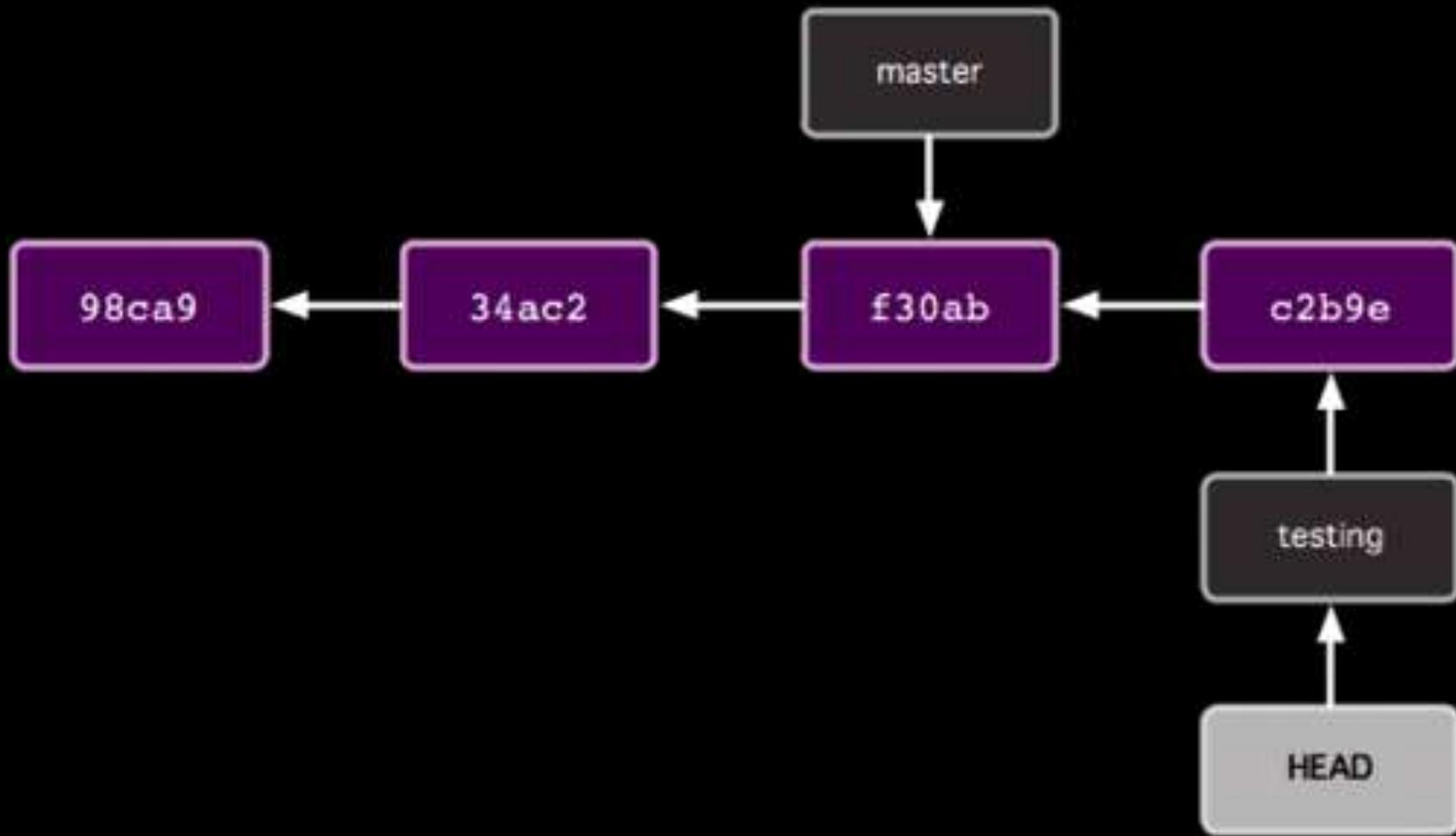
How commit works



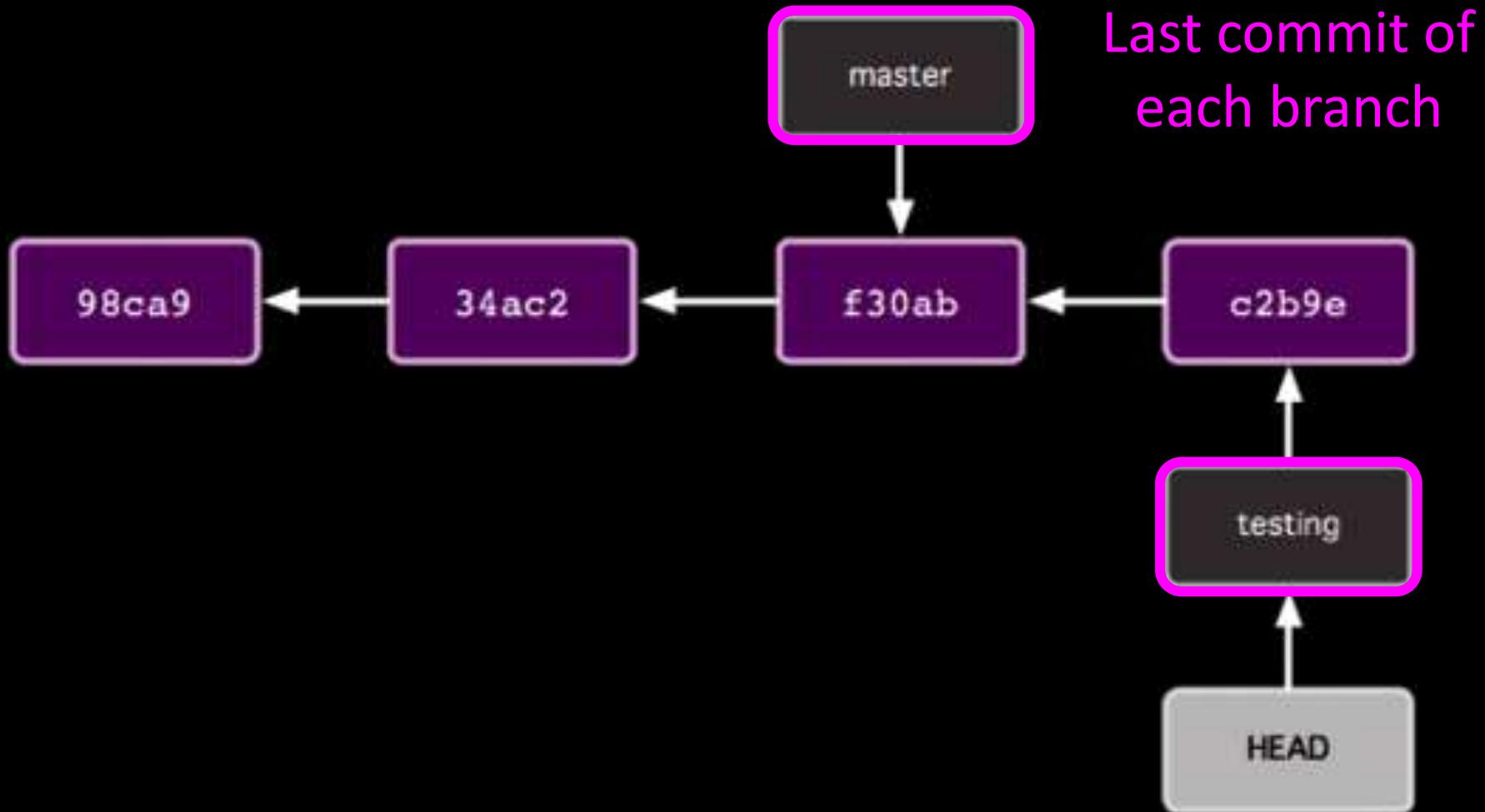
Common Workflow

1. Create temp local branch
 2. Checkout temp branch
 3. Edit/Add/Commit on temp branch
 4. Checkout master branch
 5. Pull to update master branch
 6. Merge temp branch with updated master
 7. Delete temp branch
 8. Push to update server repos
-
- Make changes in local branch
- Merge with GitHub repos

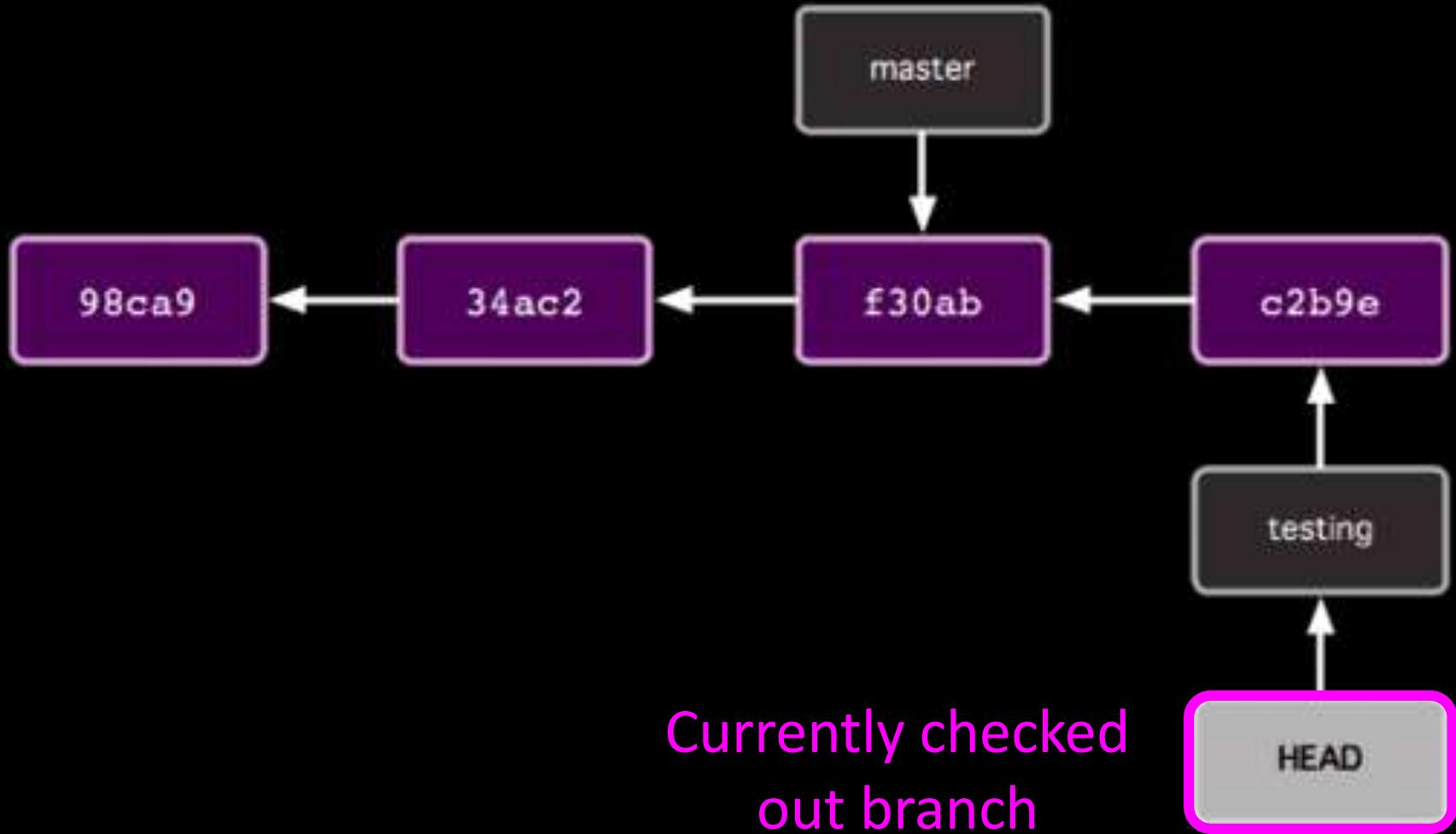
Organization with two branches



Organization with two branches

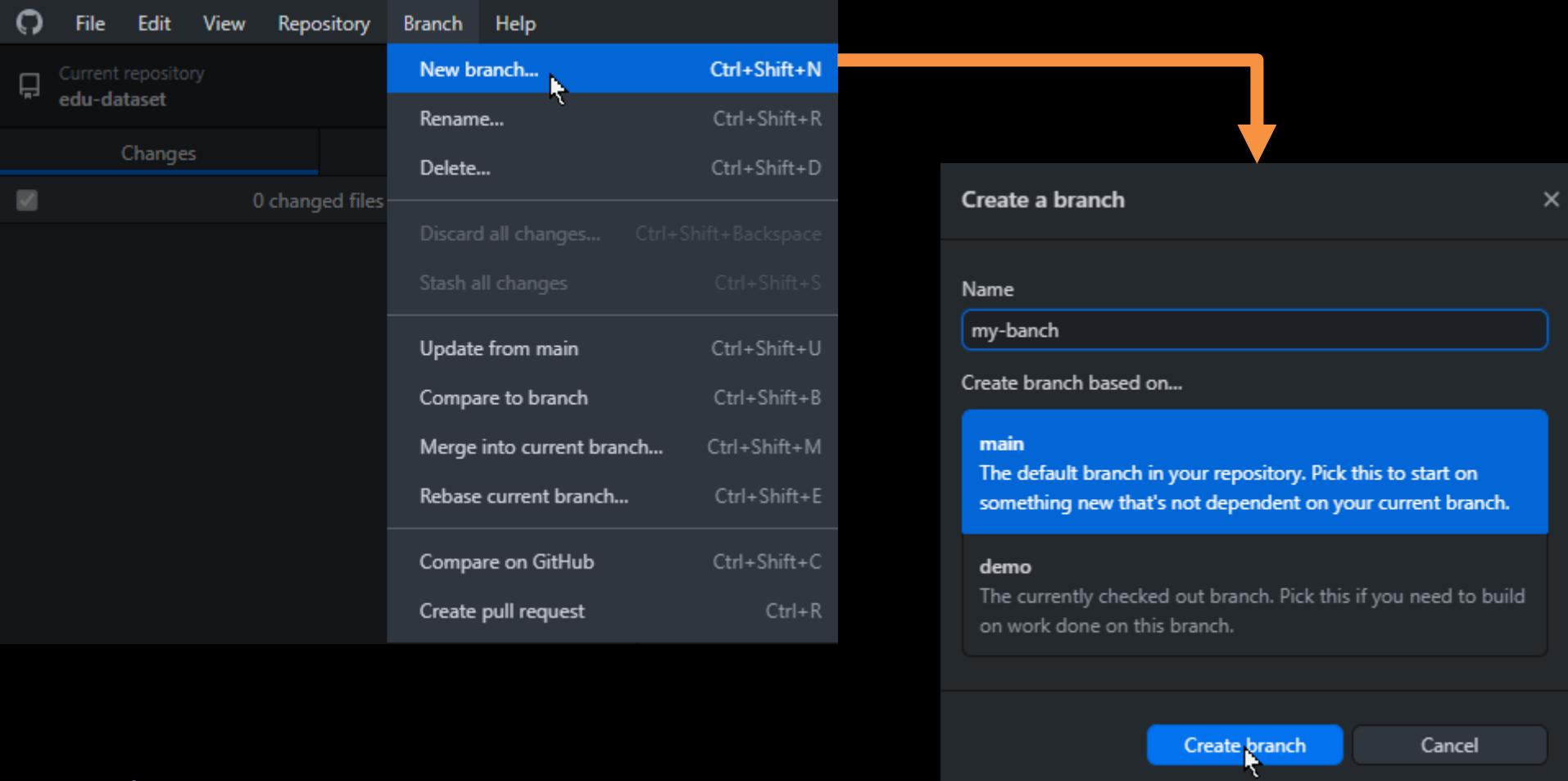


Organization with two branches



Common Workflow

1. Create temp local branch
2. Checkout temp branch
3. Edit/Add/Commit on temp branch
4. Checkout master branch
5. Pull to update master branch
6. Merge temp branch with updated master
7. Delete temp branch
8. Push to update server repos



```
$ git branch my-branch  
$ git checkout my-branch  
ou  
$ git checkout -b my-banch
```

File Edit View Repository Branch Help

Current repository
edu-dataset

Current branch
demo

Fetch origin
Last fetched 5 minutes ago

Changes History Branches Pull requests

0 changed files

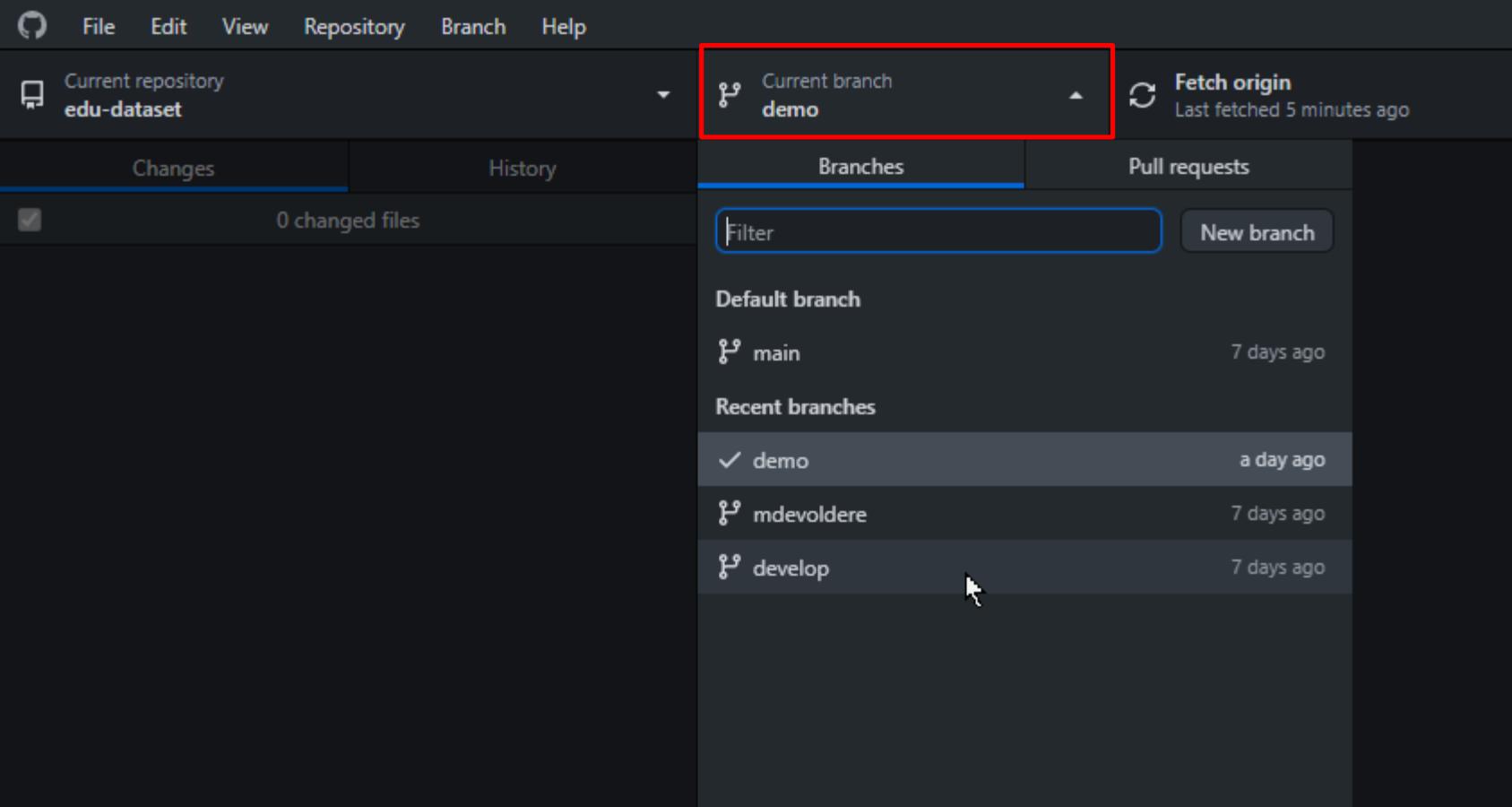
Filter New branch

Default branch

main 7 days ago

Recent branches

✓ demo	a day ago
└ mdevoldere	7 days ago
└ develop	7 days ago



File Edit View Repository Branch Help

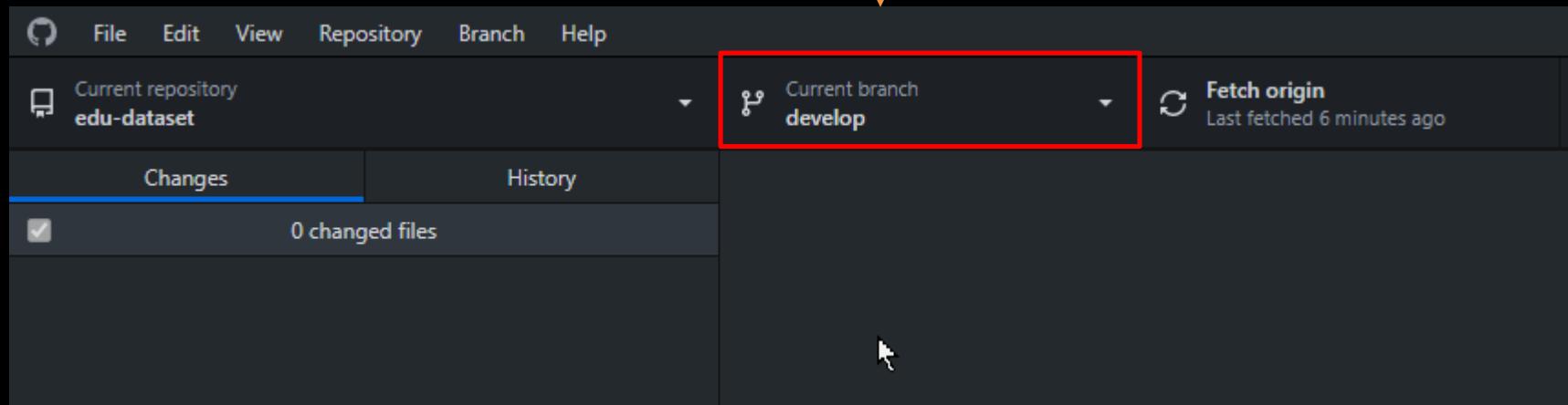
Current repository
edu-dataset

Current branch
develop

Fetch origin
Last fetched 6 minutes ago

Changes History

0 changed files





EXPLORER

EDU-DATASET

.gitignore

LICENSE

README.md

Select a ref to checkout

- + Create new branch...
- + Create new branch from...
- 🔗 Checkout detached...
- demo 6e33b37b
- mdevoldere 41960d57
- main 41960d57 ▼
- develop 41960d57
- origin/demo Remote branch at 6e33b37b
- origin/mdevoldere Remote branch at 41960d57
- origin/main Remote branch at 41960d57
- origin/develop Remote branch at 41960d57
- origin/HEAD Remote branch at 41960d57

Show All Commands Ctrl + Shift + PGo to File Ctrl + PFind in Files Ctrl + Shift + FStart Debugging F5Toggle Terminal Ctrl + `

> OUTLINE

develop

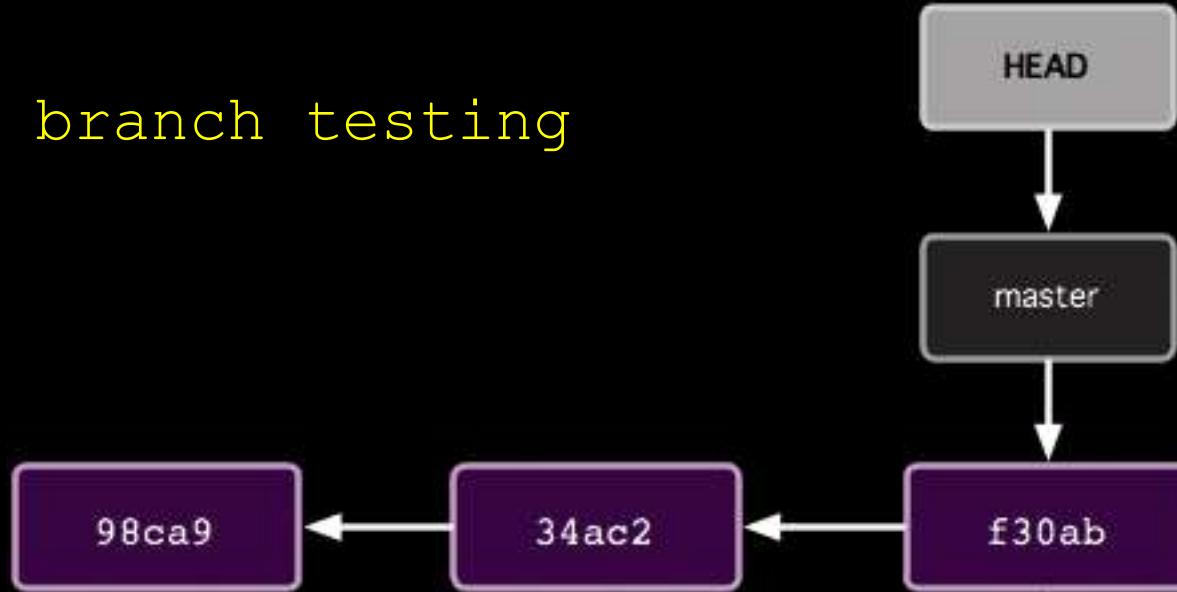


0 0 Δ 0



How git branch works

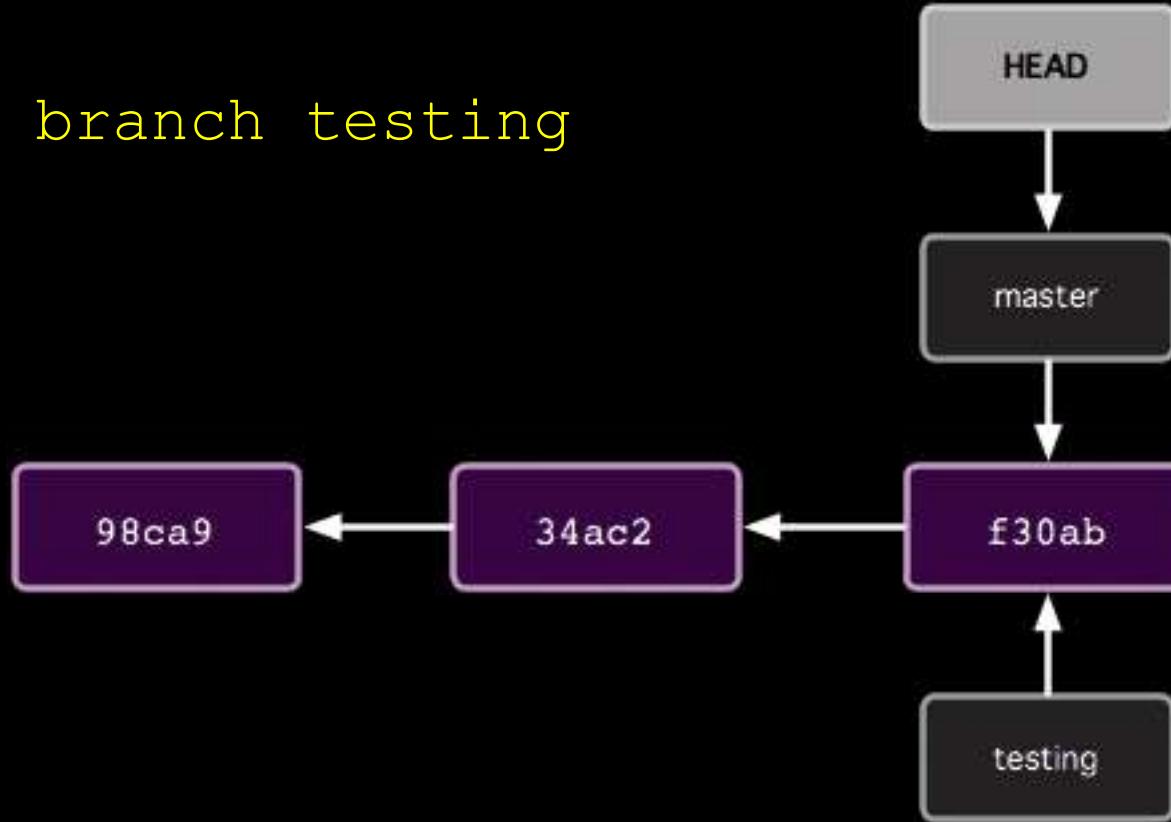
```
$ git branch testing
```



Before

How git branch works

```
$ git branch testing
```



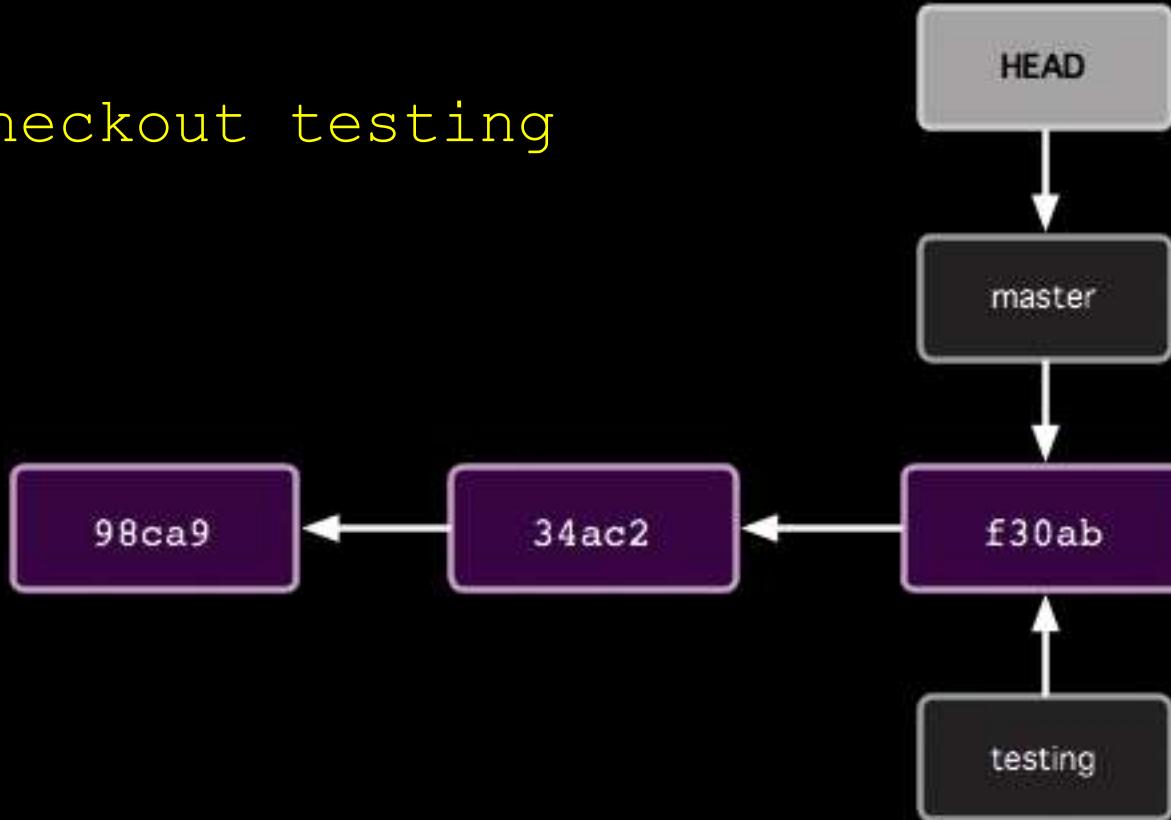
After

Common Workflow

1. Create temp local branch
2. Checkout temp branch
3. Edit/Add/Commit on temp branch
4. Checkout master branch
5. Pull to update master branch
6. Merge temp branch with updated master
7. Delete temp branch
8. Push to update server repos

How git checkout works

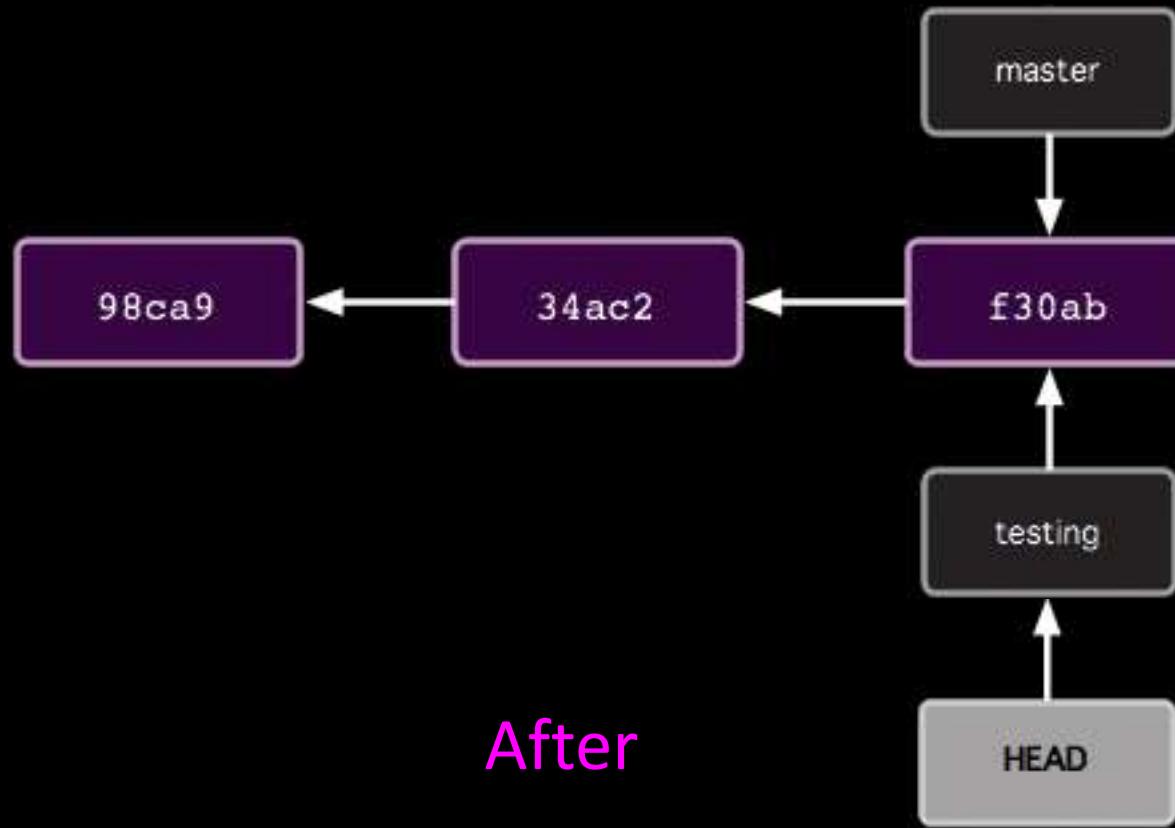
```
$ git checkout testing
```



Before

How git checkout works

```
$ git checkout testing
```



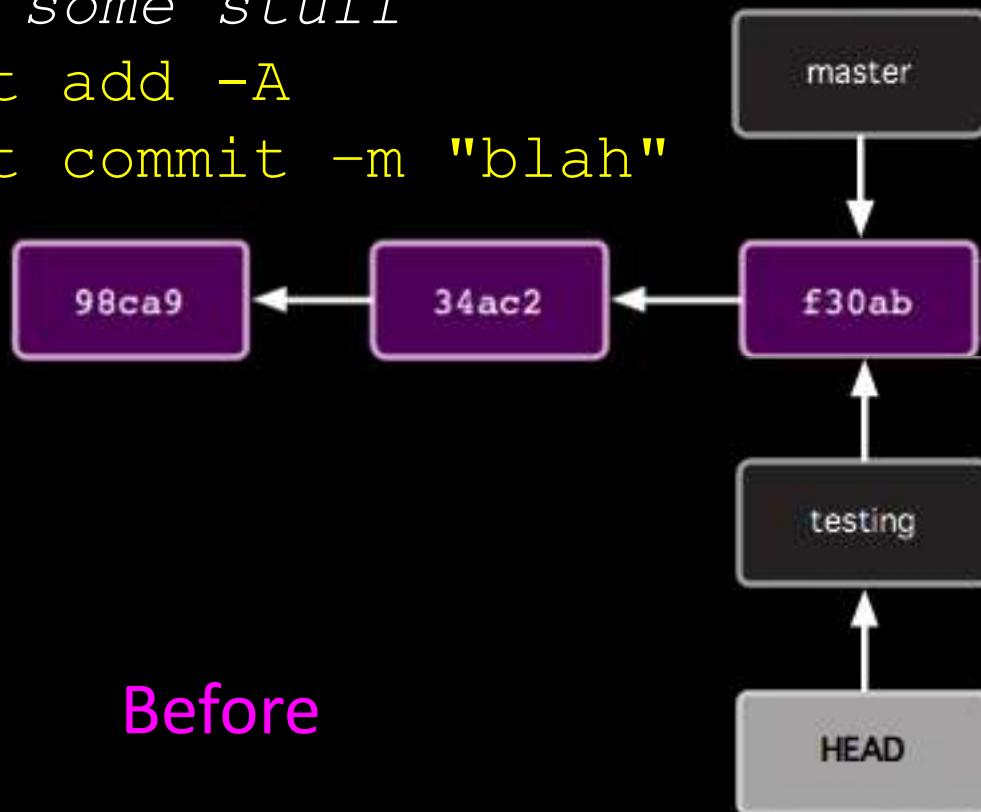
Common Workflow

1. Create temp local branch
2. Checkout temp branch
3. Edit/Add/Commit on temp branch
4. Checkout master branch
5. Pull to update master branch
6. Merge temp branch with updated master
7. Delete temp branch
8. Push to update server repos

How git commit works with multiple branches

Edit some stuff

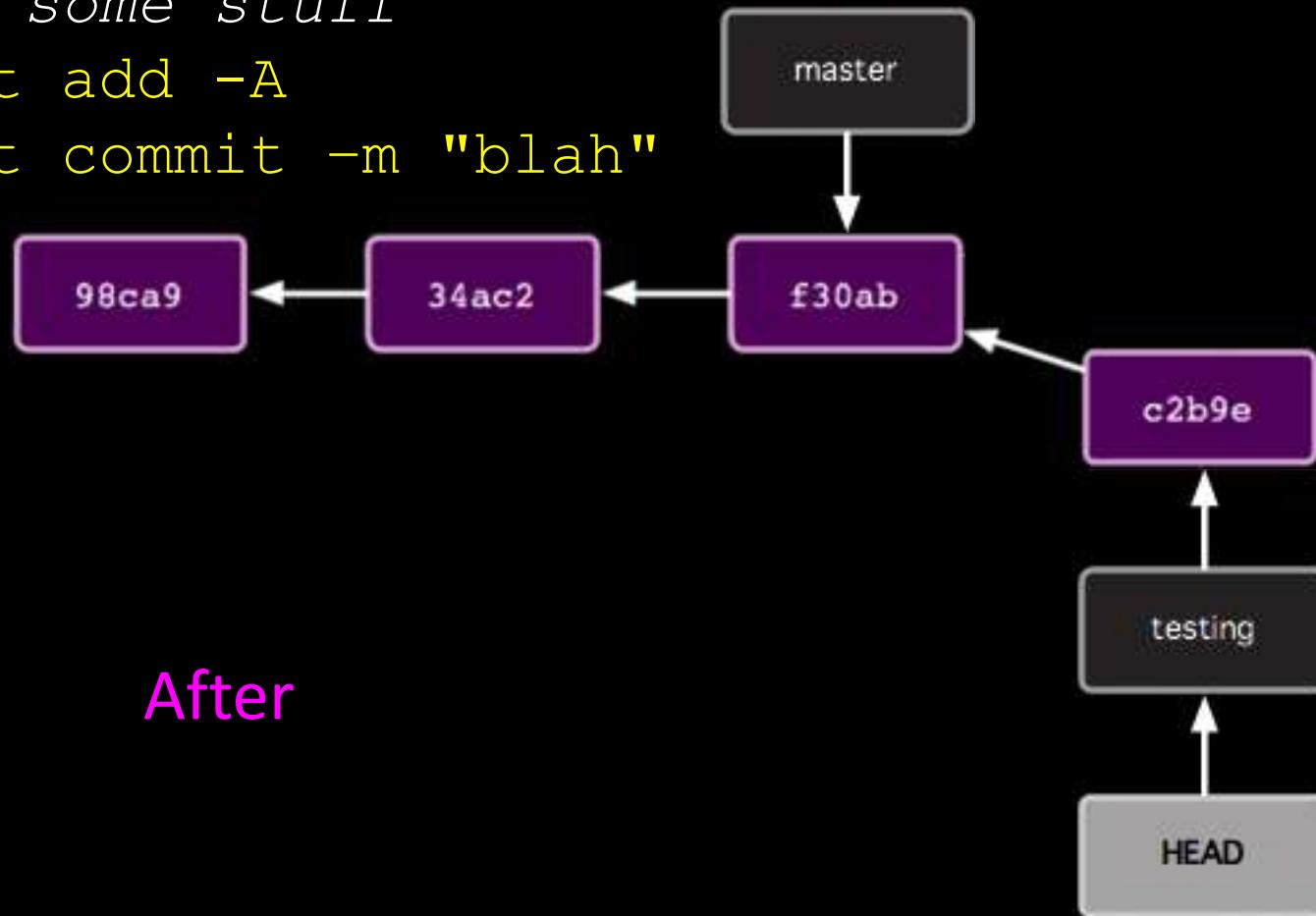
```
$ git add -A  
$ git commit -m "blah"
```



How git commit works with multiple branches

Edit some stuff

```
$ git add -A  
$ git commit -m "blah"
```



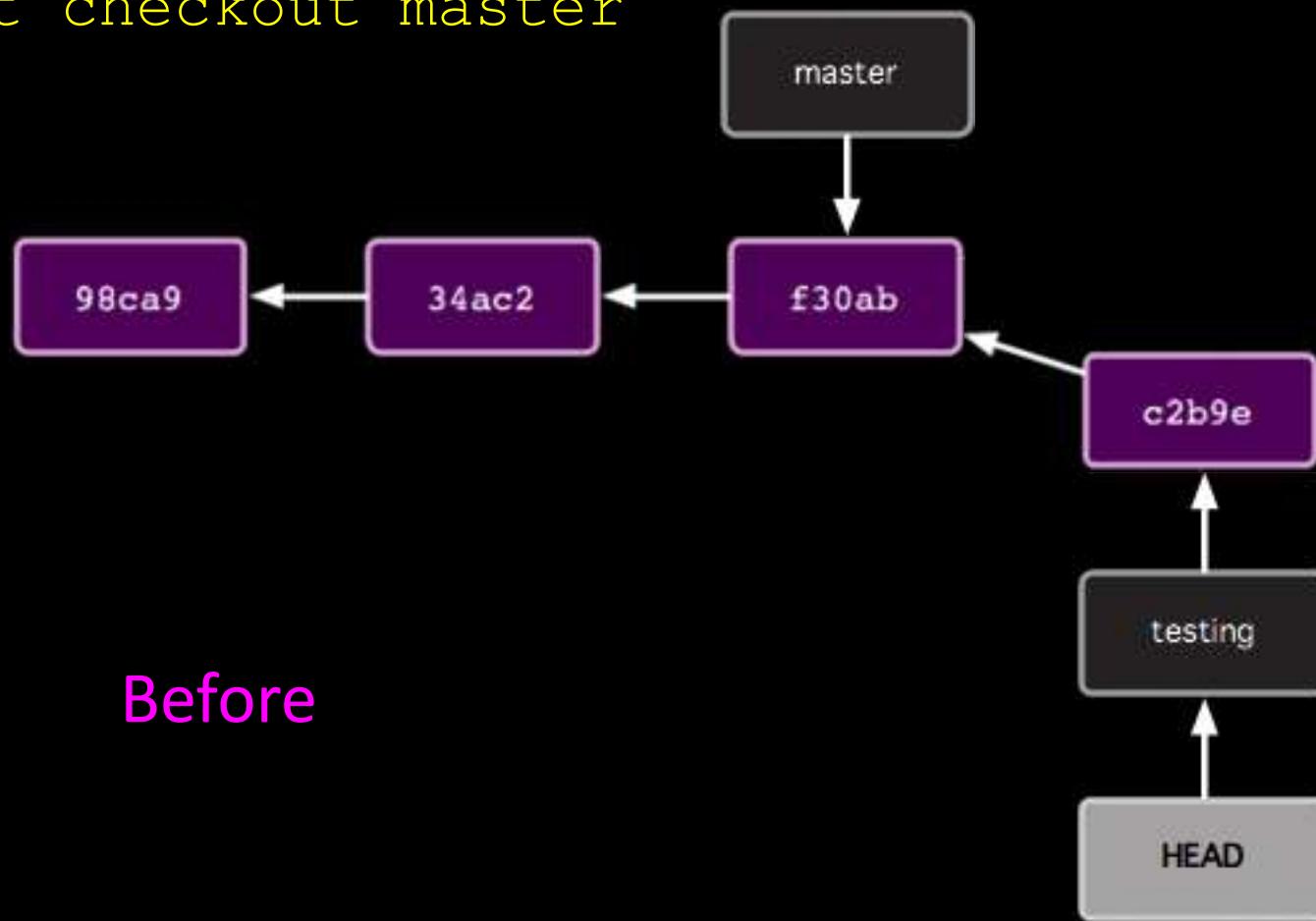
After

Common Workflow

1. Create temp local branch
2. Checkout temp branch
3. Edit/Add/Commit on temp branch
4. Checkout master branch
5. Pull to update master branch
6. Merge temp branch with updated master
7. Delete temp branch
8. Push to update server repos

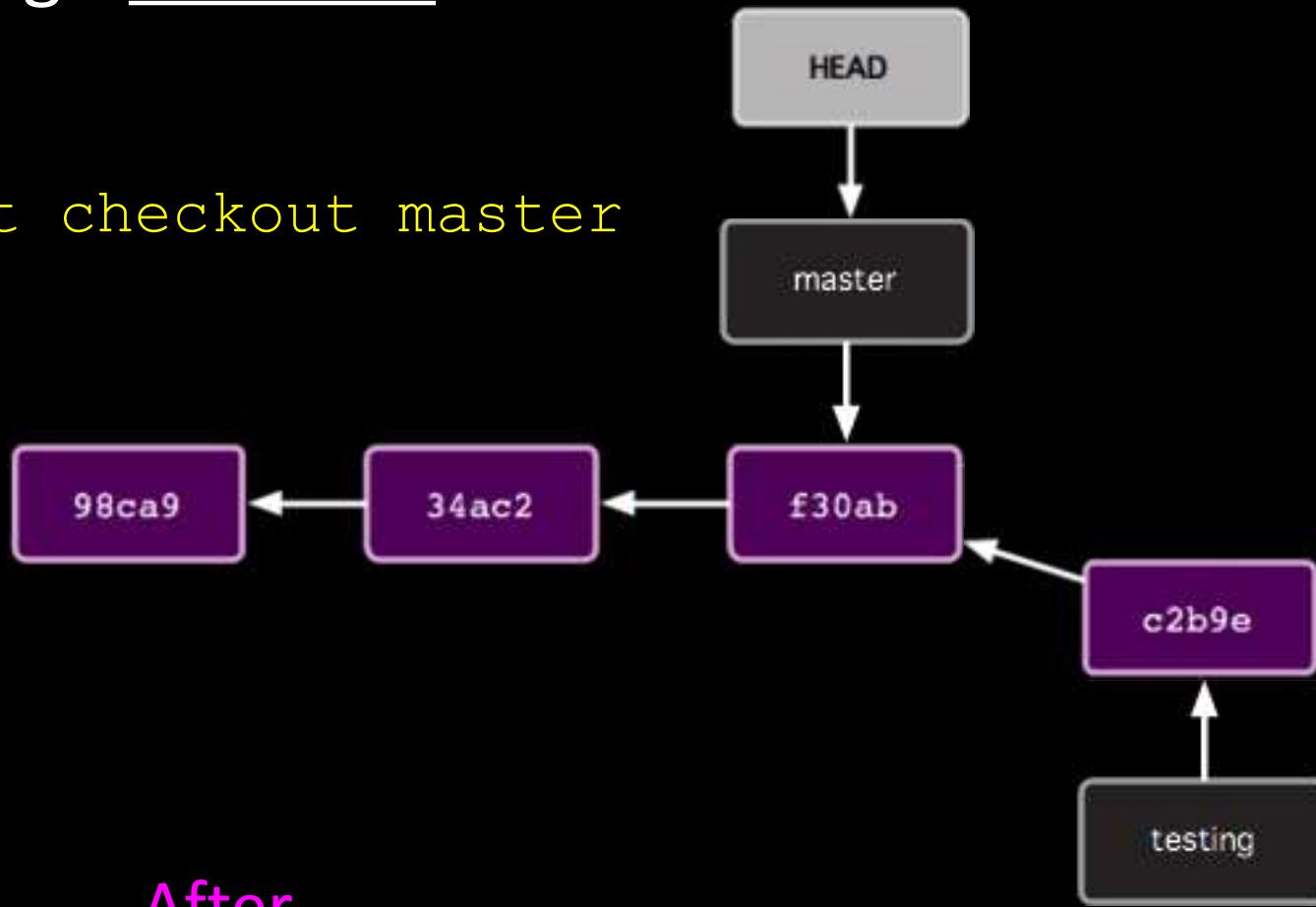
How git checkout works

```
$ git checkout master
```



How git checkout works

```
$ git checkout master
```



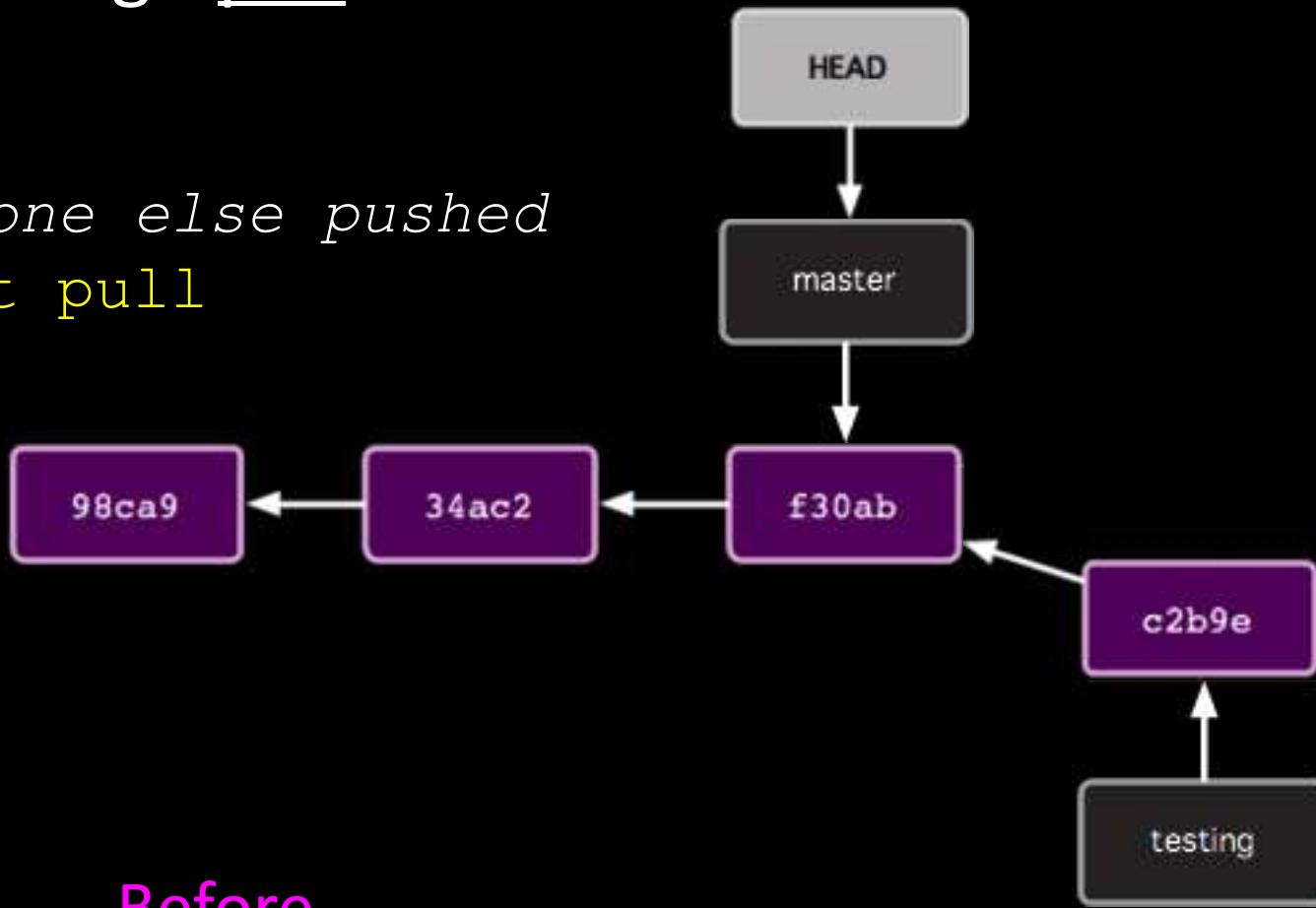
After

Common Workflow

1. Create temp local branch
2. Checkout temp branch
3. Edit/Add/Commit on temp branch
4. Checkout master branch
5. Pull to update master branch
6. Merge temp branch with updated master
7. Delete temp branch
8. Push to update server repos

How git pull works

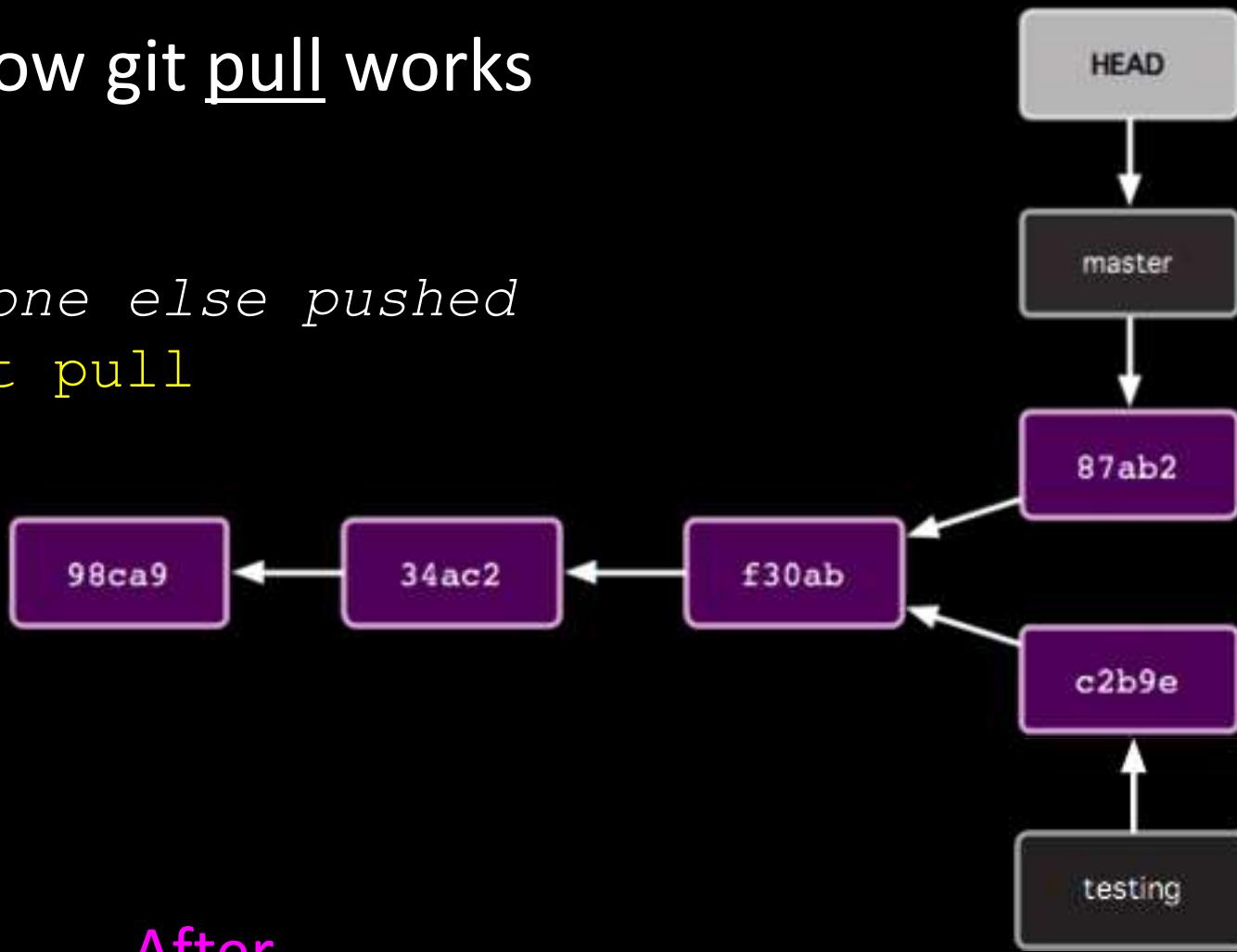
Someone else pushed
\$ git pull



Before

How git pull works

Someone else pushed
\$ git pull



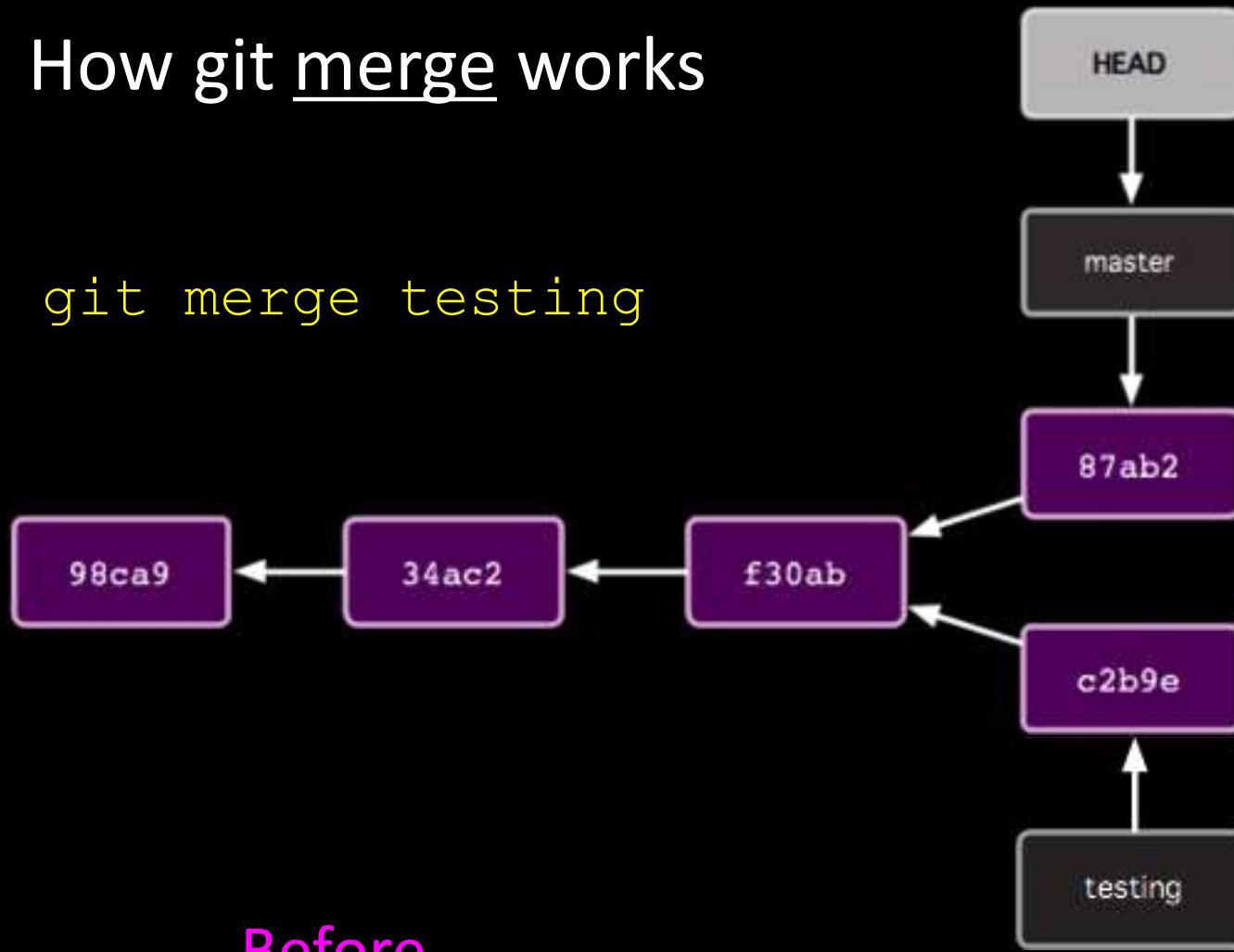
After

Common Workflow

1. Create temp local branch
2. Checkout temp branch
3. Edit/Add/Commit on temp branch
4. Checkout master branch
5. Pull to update master branch
6. Merge temp branch with updated master
7. Delete temp branch
8. Push to update server repos

How git merge works

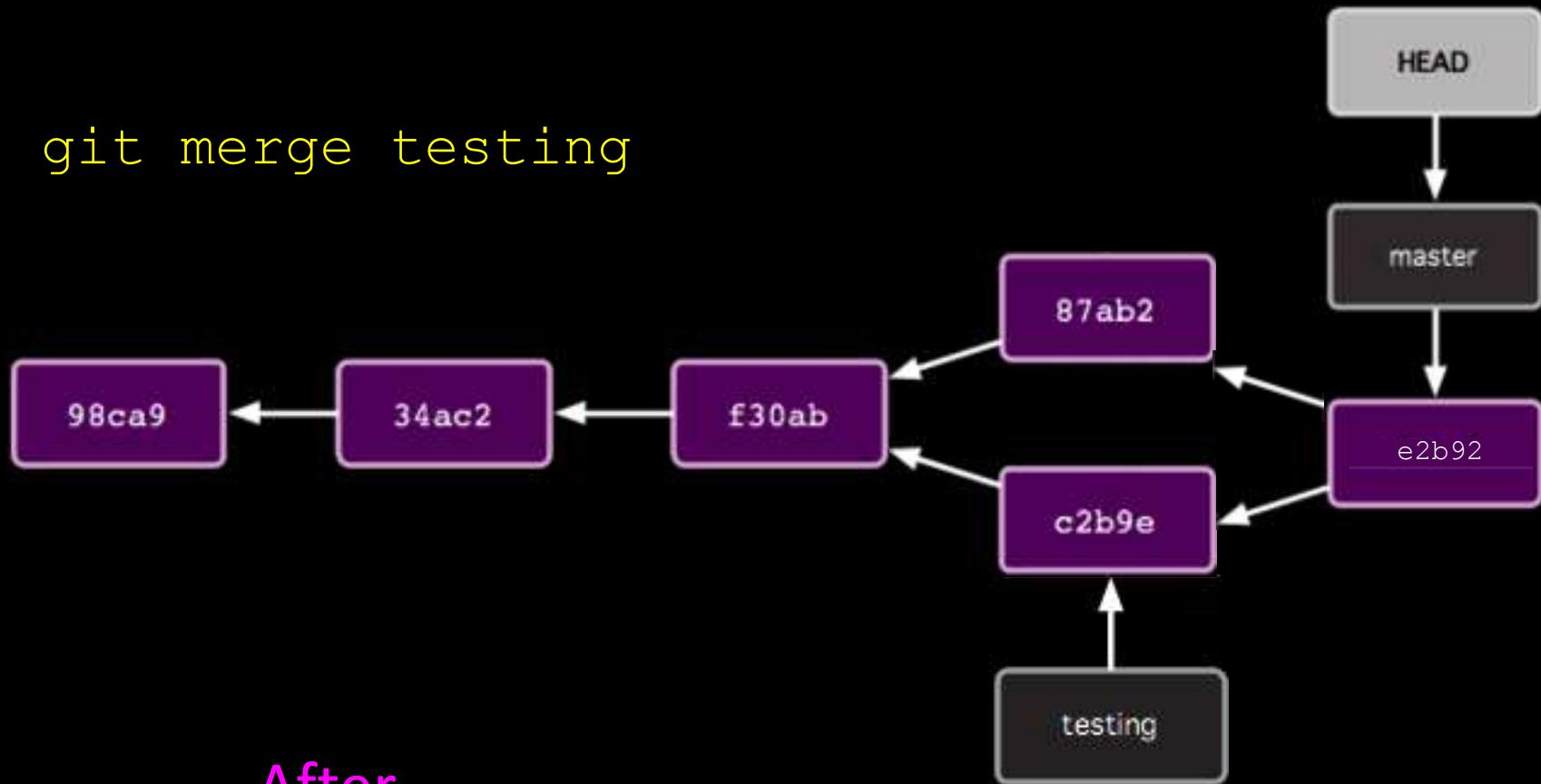
```
$ git merge testing
```



Before

How git merge works

```
$ git merge testing
```



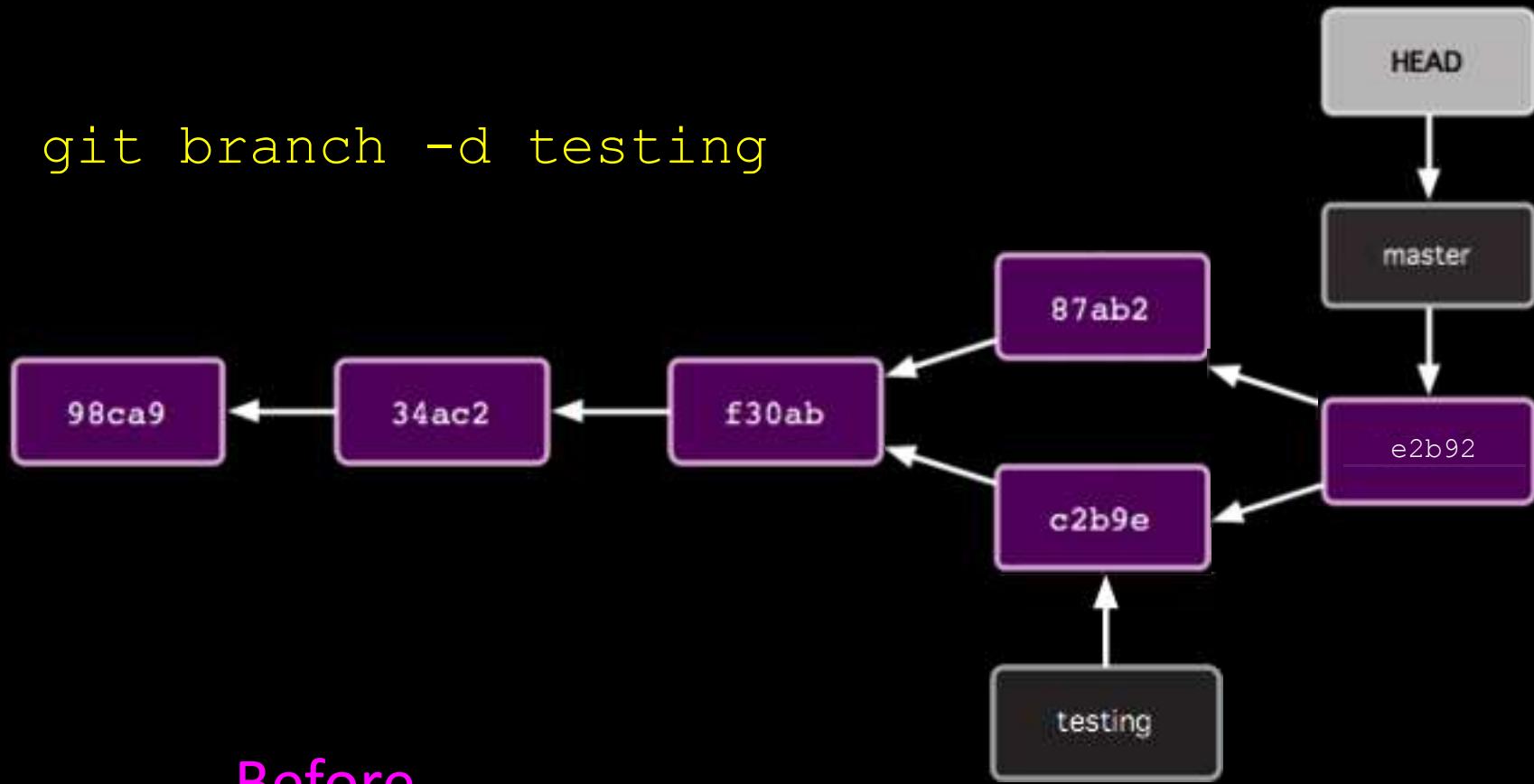
After

Common Workflow

1. Create temp local branch
2. Checkout temp branch
3. Edit/Add/Commit on temp branch
4. Checkout master branch
5. Pull to update master branch
6. Merge temp branch with updated master
7. Delete temp branch
8. Push to update server repos

How to delete branches

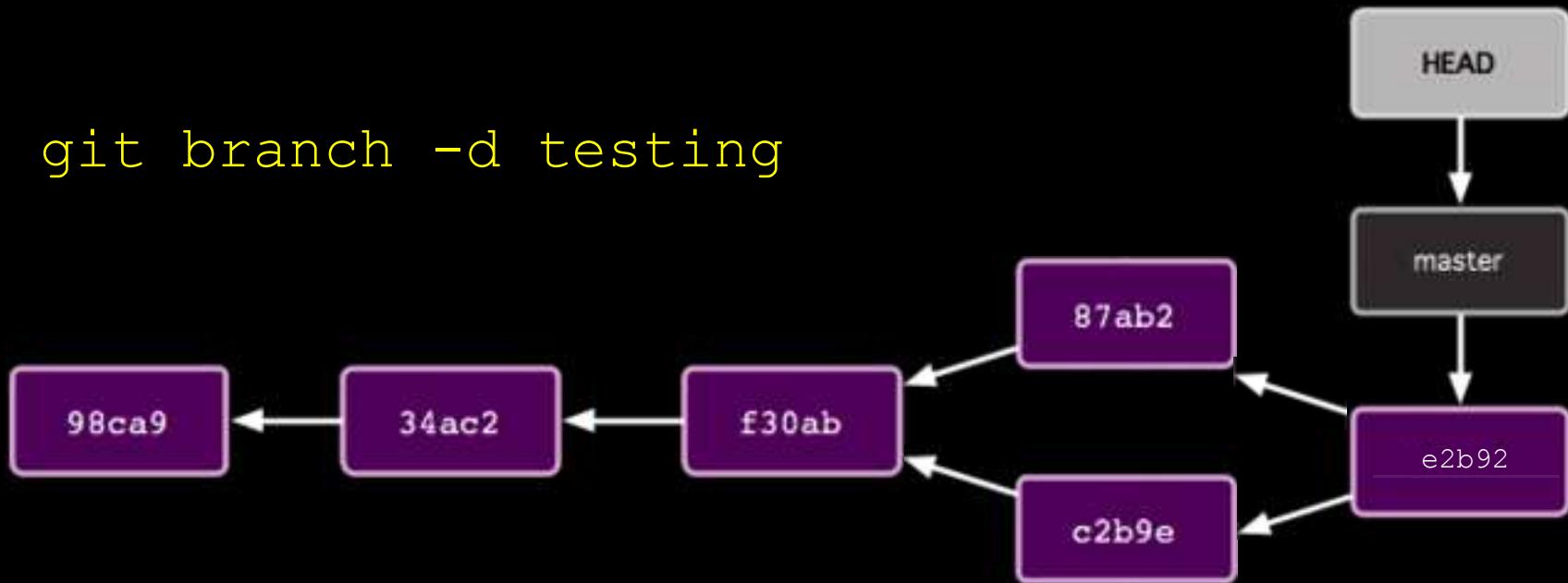
```
$ git branch -d testing
```



Before

How to delete branches

```
$ git branch -d testing
```



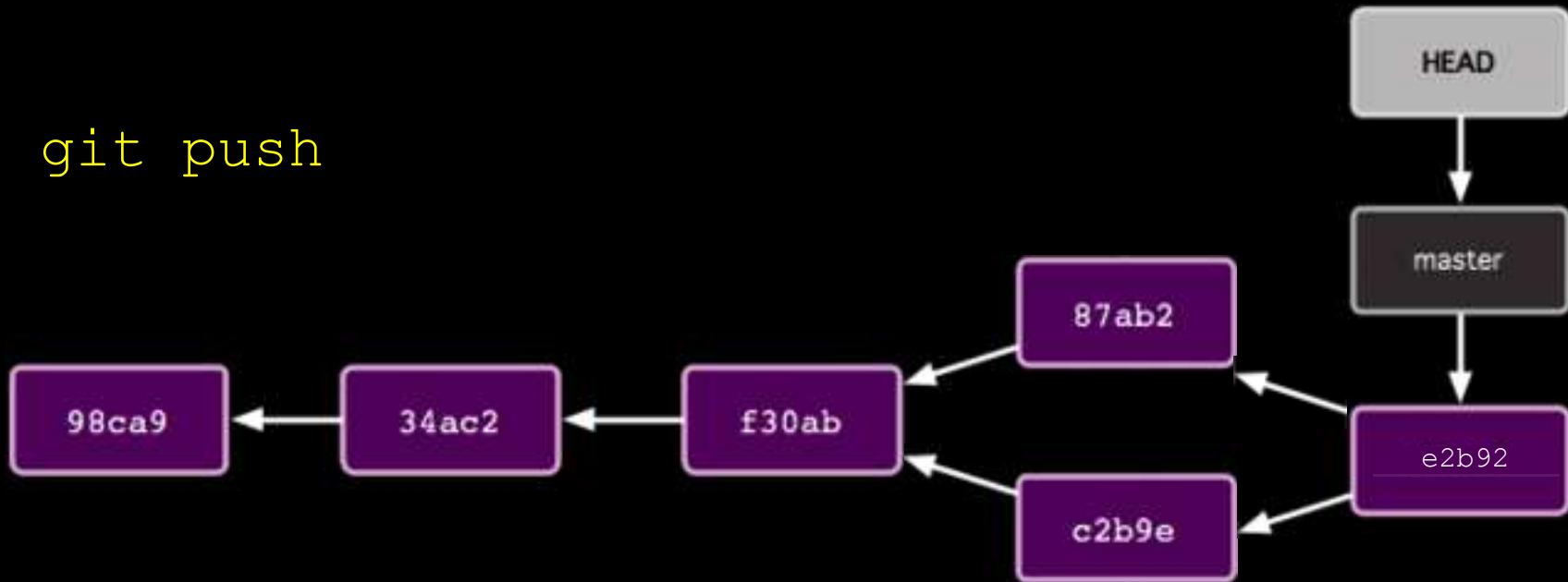
After

Common Workflow

1. Create temp local branch
2. Checkout temp branch
3. Edit/Add/Commit on temp branch
4. Checkout master branch
5. Pull to update master branch
6. Merge temp branch with updated master
7. Delete temp branch
8. Push to update server repos

How git push works

```
$ git push
```



Should update server repos
(if no one else has pushed commits to
master branch since last pull)

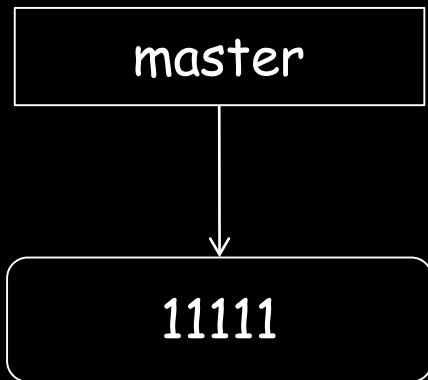
Tips

- git output contains lots of hints
 - git status is your friend!
- Merging may not be as easy as showed
 - E.g.: Multiple collabs updated same parts of file
- Pull before starting temp branch
- Team communication important !

Pop Quiz

- 5 questions
- Update diagram in each
 - Commit nodes
 - Branch nodes
- Based on actions of Alice and Bob
 - Collaborating via GitHub repo

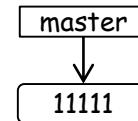
Start like this



Scott Fleming

SF 1

GitHub



Alice

Bob

Question 1

- Alice:
 - \$ git clone https://github.com/whatever.git
 - \$ cd whatever
- Bob:
 - \$ git clone https://github.com/whatever.git
 - \$ cd whatever

(include the HEAD node)

Question 2

- Alice:
 - `$ git branch myfix`
 - `$ git checkout myfix`
- (Alternatively)
 - `$ git checkout -b myfix`

Question 3

- Alice:
 - \$ rails generate scaffold User ...
 - \$ git add -A
 - \$ git commit -m "Added User" # 22222
- Bob:
 - \$ rails generate scaffold Micropost ...
 - \$ git add -A
 - \$ git commit -m "Added Micropost" # 33333

Question 4

- Bob:
 - git push

Question 5

- Alice:
 - git pull

Appendix

What if...

Alice did this:

app/models/micropost.rb

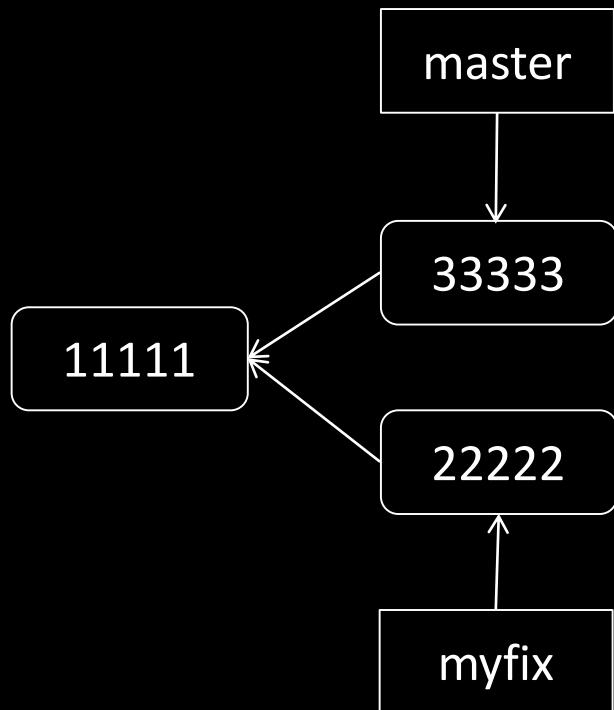
```
class Micropost < ActiveRecord::Base
  validates :content, length: { maximum: 140 }
end
```

Bob did this:

app/models/micropost.rb

```
class Micropost < ActiveRecord::Base
  validates :content, length: { maximum: 120 }
end
```

What if Alice did this?



\$ git checkout master
\$ git merge myfix

\$ git merge myfix
Auto-merging app/models/micropost.rb
Automatic merge failed; fix conflict and then commit result.

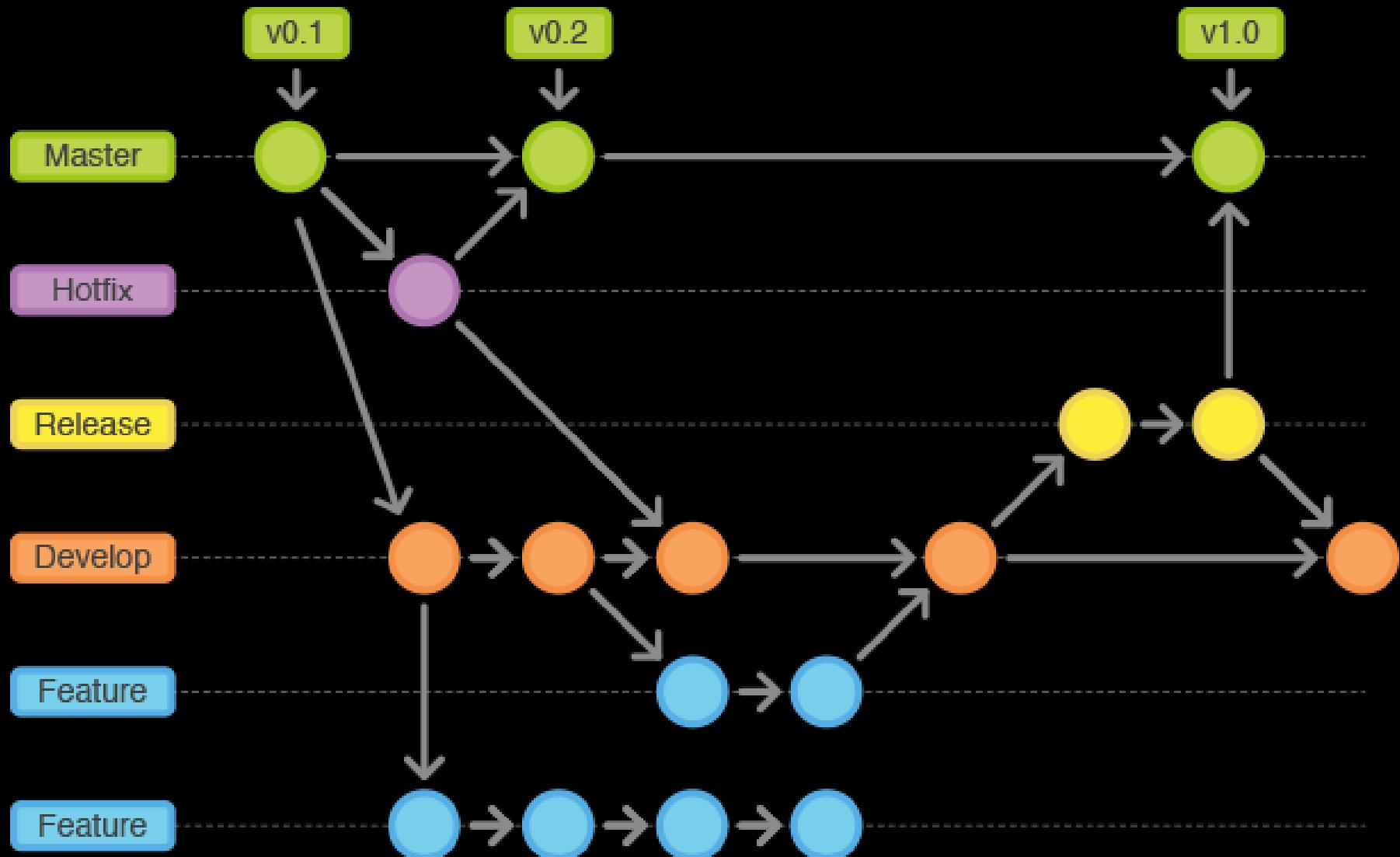
app/models/micropost.rb

```
class Micropost < ActiveRecord::Base
<<<<<< HEAD
  validates :content, length: { maximum: 140 }
=====
  validates :content, length: { maximum: 120 }
>>>>> myfix
end
```

To resolve:
Manually fix the file; git add and commit

Reality

Reality



End

