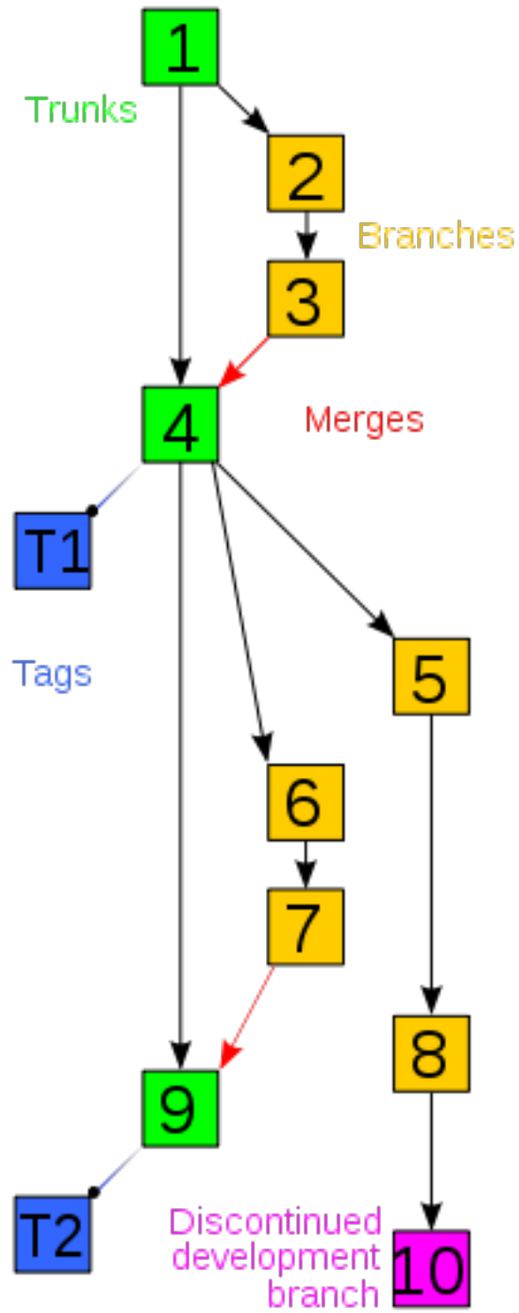
PHY410 / 505
Computational Physics 1

Salvatore Rappoccio

Version Control

- What about backing up what you did?
- Back in the “old days” we used to keep separate files as we backed things up
- This is fallen away considerably in favor of using **version control**
  - It’s basically a giant graph (as in “graph theory”) that keeps track of changes and allows you to go back to previous versions



https://en.wikipedia.org/wiki/Version_control#/media/File:Revision_controlled_project_visualization-2010-24-02.svg

Version Control

- There is a long history of version control tools
- In my own lifetime I have used:
  - CVS (concurrent versions system)
  - SVN (subversion)
  - git

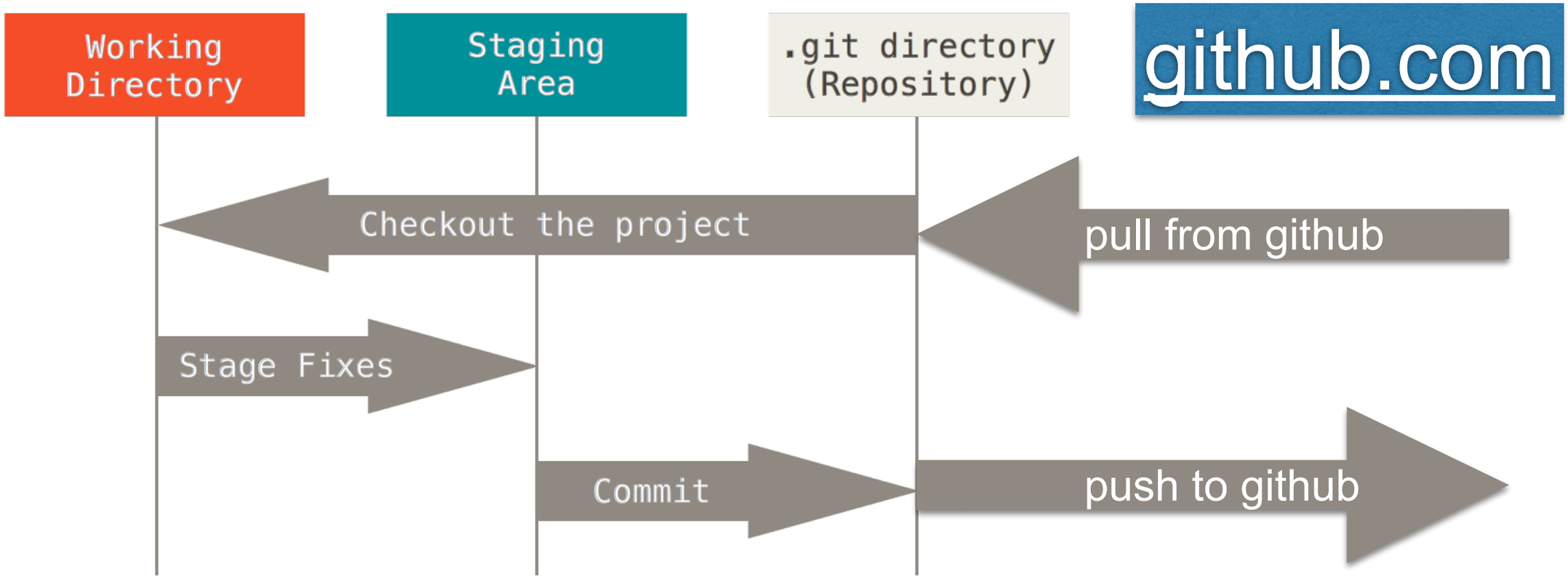
- The code in this class will be using git.
- As you go on, you should store your code with some versioning system so you can keep track of what changed, distribute your code to users, deploy bug fixes, etc.

- We will be using github.com for our purposes. You must get an account.

git

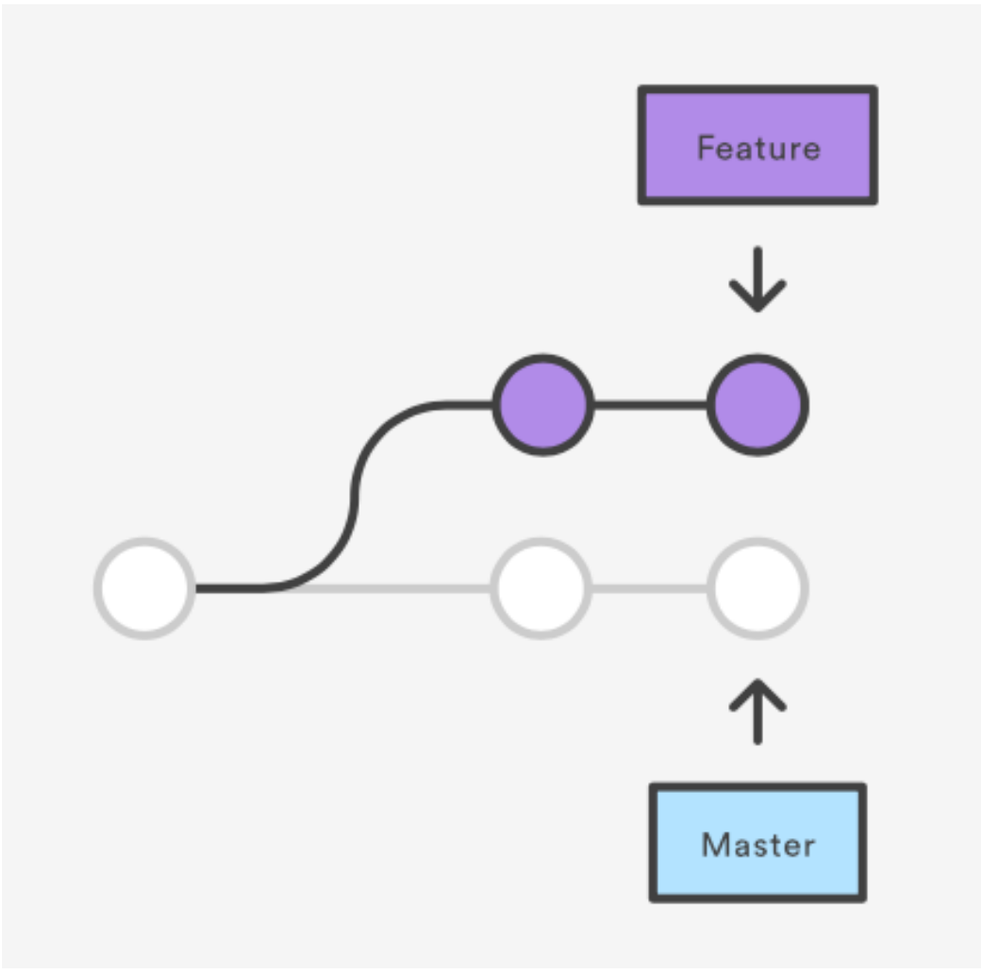
- Git and github tutorial :
  - https://try.github.io/levels/1/challenges/1
  - https://guides.github.com

- Basic idea:



git

- Why all these areas?
  - You have complete local control to “play around” and create your own branch structure without messing anyone else up
  - In parlance : “branching is easy”

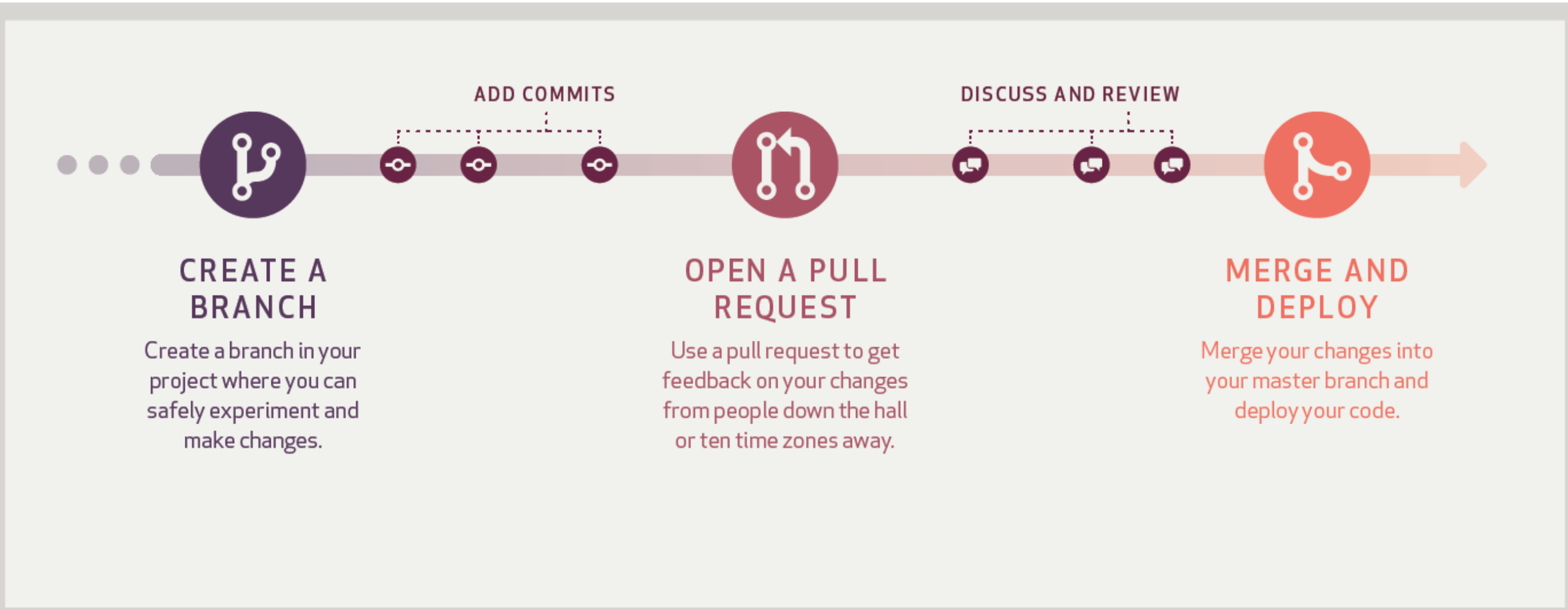


- Once you are sure things are working, you can update the central repository

- Also : you can suggest your changes for OTHER people’s projects!

git

- The github workflow:
  - https://guides.github.com/introduction/flow/



git

- Creating and using a git repository
  - https://guides.github.com/activities/hello-world/

- Outline:

1. Create a github account : https://github.com
2. Create a repository
3. Create a branch
4. Edits and commits
5. Pull request

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner **Repository name**

rappoccio / hello-world

Great repository names are short and memorable. Need inspiration? How about **super-duper-robot**.

Description (optional)

Hello world stuff.

Public
Anyone can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** | Add a license: **GNU General Public License v3.0**

Create repository

git

rappoccio / hello-world

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Pulse | Graphs | Settings

Hello world stuff.

Edit

1 commit | 1 branch | 0 releases | 1 contributor

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

rappoccio Initial commit — Latest commit 14c2599 just now

LICENSE	Initial commit	just now
README.md	Initial commit	just now

README.md

hello-world

Hello world stuff.

git

Branch created.

rappoccio / hello-world

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Pulse | Graphs | Settings

Hello world stuff.

Edit

1 commit | 2 branches | 0 releases | 1 contributor | GPL-3.0

Branch: readme-edits | New pull request | Create new file | Upload files | Find file | Clone or download

This branch is even with master. Pull request Compare

rappoccio Initial commit — Latest commit 14c2599 a minute ago

LICENSE	Initial commit	a minute ago
README.md	Initial commit	a minute ago

README.md

hello-world

Hello world stuff.

git

rappoccio / hello-world

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Pulse | Graphs | Settings

hello-world / README.md or cancel

Edit file | Preview changes | Spaces | 2 | Soft wrap

```
1  # hello-world
2  Hello world stuff.
3
4
5  and this is a strange question, amirite?
```

Commit changes

Finish README, amirite?

uriteya.

Commit directly to the `readme-edits` branch.

Create a **new branch** for this commit and start a pull request. Learn more about pull requests.

Commit changes | Cancel

git

rappoccio / hello-world

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Pulse | Graphs | Settings

Branch: readme-edits | hello-world / README.md | Find file | Copy path

rappoccio Finish README, amirite? 4bfb568 just now

1 contributor

6 lines (3 sloc) | 78 Bytes | Raw | Blame | History

hello-world

Hello world stuff.

and this is a strange question, amirite?

git

rappoccio / hello-world

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Pulse | Graphs | Settings

Comparing changes

Pull request for “readme-edits” onto “master”

Choose two branches to see what’s changed or to start a new pull request. If you need to, you can also compare across forks.

base: master ... compare: readme-edits ✓ Able to merge. These branches can be automatically merged.

Things are OK, you can merge

Create pull request Discuss and review the changes in this comparison with others.

1 commit | 1 file changed | 0 commit comments | 1 contributor

Commits on Jan 27, 2017

rappoccio Finish README, amirite? ... 4bfb568

Here are the commits

Showing 1 changed file with 3 additions and 0 deletions. Unified | Split

3 README.md View

Here are the diffs

```
...  @@ -1,2 +1,5 @@
1    # hello-world
2    Hello world stuff.
1    # hello-world
2    Hello world stuff.
3    +
4    +
5    +and this is a strange question, amirite?
```

No commit comments for this range

git

rappoccio / hello-world

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Pulse | Graphs | Settings

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks.

base: master ... compare: readme-edits ✓ Able to merge. These branches can be automatically merged.

Create pull request Discuss and review the changes in this comparison with others.

1 commit | 1 file changed | 0 commit comments | 1 contributor

Commits on

rappoc

Everything good, clicky clicky

Showing 1 changed file with 3 additions and 0 deletions. Unified | Split

3 README.md View

```
@@ -1,2 +1,5 @@
 # hello-world
 Hello world stuff.
+
+
+and this is a strange question, amirite?
```

No commit comments for this range

git

rappoccio / hello-world

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Pulse | Graphs | Settings

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.

base: master ... compare: readme-edits ✓ Able to merge. These branches can be automatically merged.

Short summary

Finish README, amirite?

Write | Preview

uriteya.

Detailed summary

Attach files by dragging & dropping or selecting them.

Styling with Markdown is supported

Create pull request

Reviewers

No reviews— request one

Assignees

No one—assign yourself

Labels

None yet

Milestone

No milestone

git

rappoccio / hello-world

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Pulse | Graphs | Settings

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.

base: master ... compare: readme-edits ✓ Able to merge. These branches can be automatically merged.

Finish README, amirite?

Write | Preview

uriteya.

Attach files by dragging & dropping or selecting them.

Styling with Markdown is supported

Create pull request

Reviewers

No reviews— request one

Assignees

No one—assign yourself

Labels

None yet

Everything good, clicky clicky

git

rappoccio / hello-world

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 1 | Projects 0 | Wiki | Pulse | Graphs | Settings

Finish README, amirite? #1

Open **rappoccio** wants to merge 1 commit into `master` from `readme-edits`

Conversation 0 | Commits 1 | Files changed 1

rappoccio commented just now — Owner

uriteya.

Finish README, amirite? ··· 4bfb568

Add more commits by pushing to the **readme-edits** branch on **rappoccio/hello-world**.

This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request | You can also open this in GitHub Desktop or view command line instructions.

Write | Preview

Leave a comment

Attach files by dragging & dropping or selecting them.

Styling with Markdown is supported — Close pull request | Comment

You can now request reviews from specific people you work with.
Learn more.

Got it!

Edit

Reviewers
No reviews— request one

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

1 participant

In real life, this is where the review happens. Add text, figures, descriptions, proofs, pleadings, lawsuits, anime cartoons, grocery lists, cat vids, whatever.

git

rappoccio / hello-world

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 1 | Projects 0 | Wiki | Pulse | Graphs | Settings

Finish README, amirite? #1

Open **rappoccio** wants to merge 1 commit into `master` from `readme-edits`

Conversation 0 | Commits 1 | Files changed 1

You can now request reviews from specific people you work with.
Learn more.

Got it!

rappoccio commented just now

uriteya.

Finish README, amirite? ...

Add more commits by pushing to the **readme-edits** branch on **rappoccio/hello-world**.

This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request | You can also open this in GitHub Desktop or view command line instructions.

Write | Preview

Leave a comment

Attach files by dragging & dropping or selecting them.

Styling with Markdown is supported

Close pull request | Comment

Labels
None yet

Projects
None yet

Milestone
No milestone

Notifications
Unsubscribe
You're receiving notifications because you authored the thread.

1 participant

But for us,
Everything good, clicky clicky

git

rappoccio / hello-world

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 1 | Projects 0 | Wiki | Pulse | Graphs | Settings

Finish README, amirite? #1

Merged **rappoccio** merged 1 commit into `master` from `readme-edits` 2 minutes ago

Conversation 0 | Commits 1 | Files changed 1

rappoccio commented 3 minutes ago — Owner

uriteya.

Finish README, amirite? ... 4bfb568

rappoccio merged commit `0473ddc` into `master` 2 minutes ago — Revert

Avoid bugs by automatically running your tests.

Continuous integration can help catch bugs by running your tests automatically. Merge your code with confidence using one of our continuous integration providers. Learn more

Pull request successfully merged and closed

You're all set—the `readme-edits` branch can be safely deleted. Delete branch

Yay! Purple = good!

Attach files by dragging & dropping or selecting them.

Styling with Markdown is supported — Comment

You can now request reviews from specific people you work with. Learn more.

Got it!

Reviewers

No reviews— request one

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Notifications

Unsubscribe

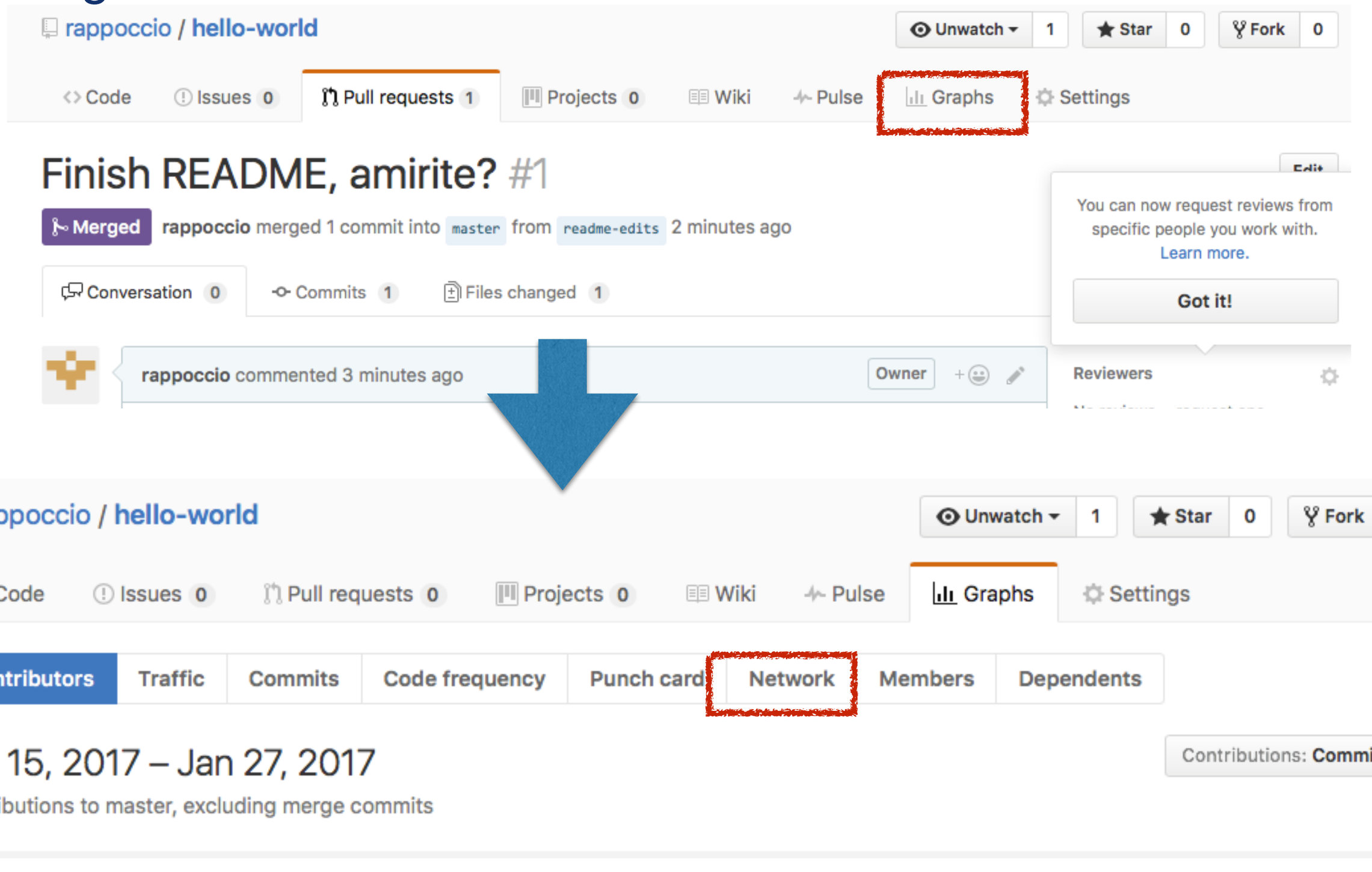
ceiving notifications you modified the se state.

1 participant

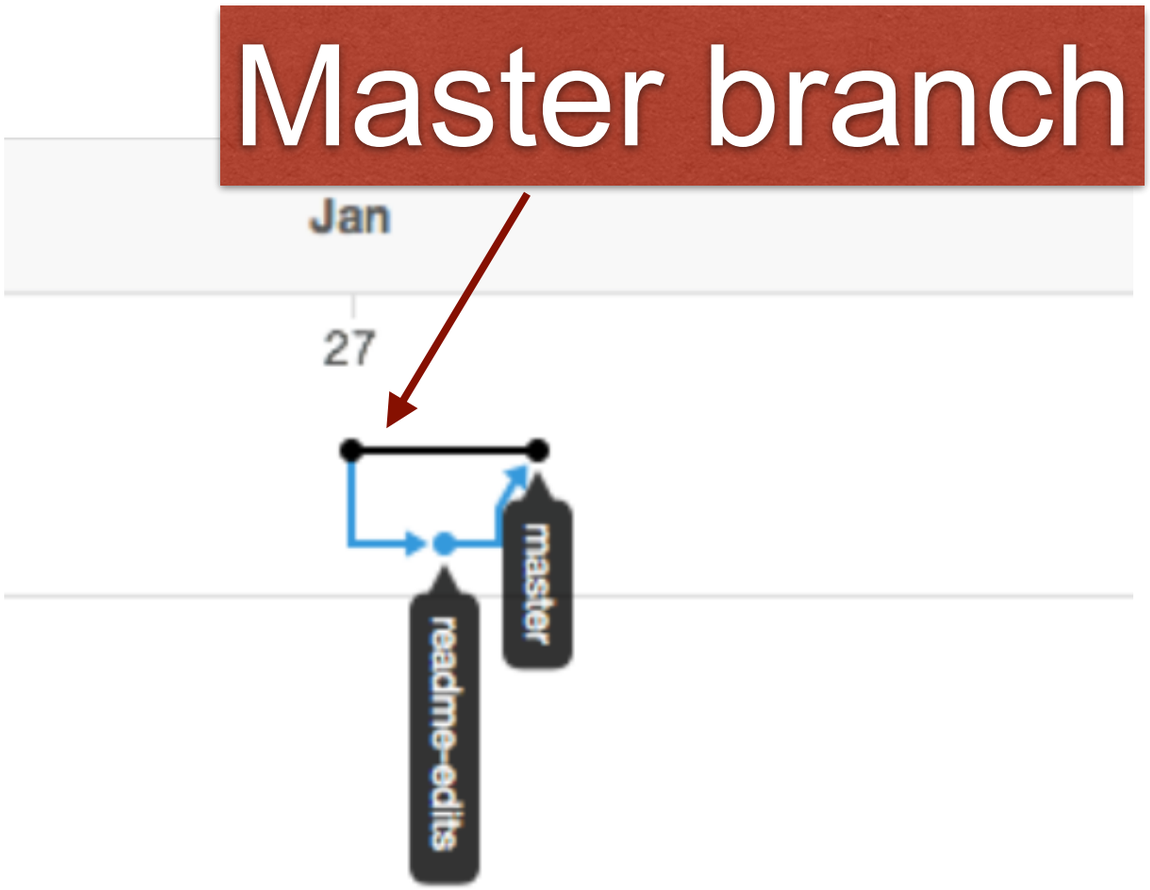
Lock conversation

git

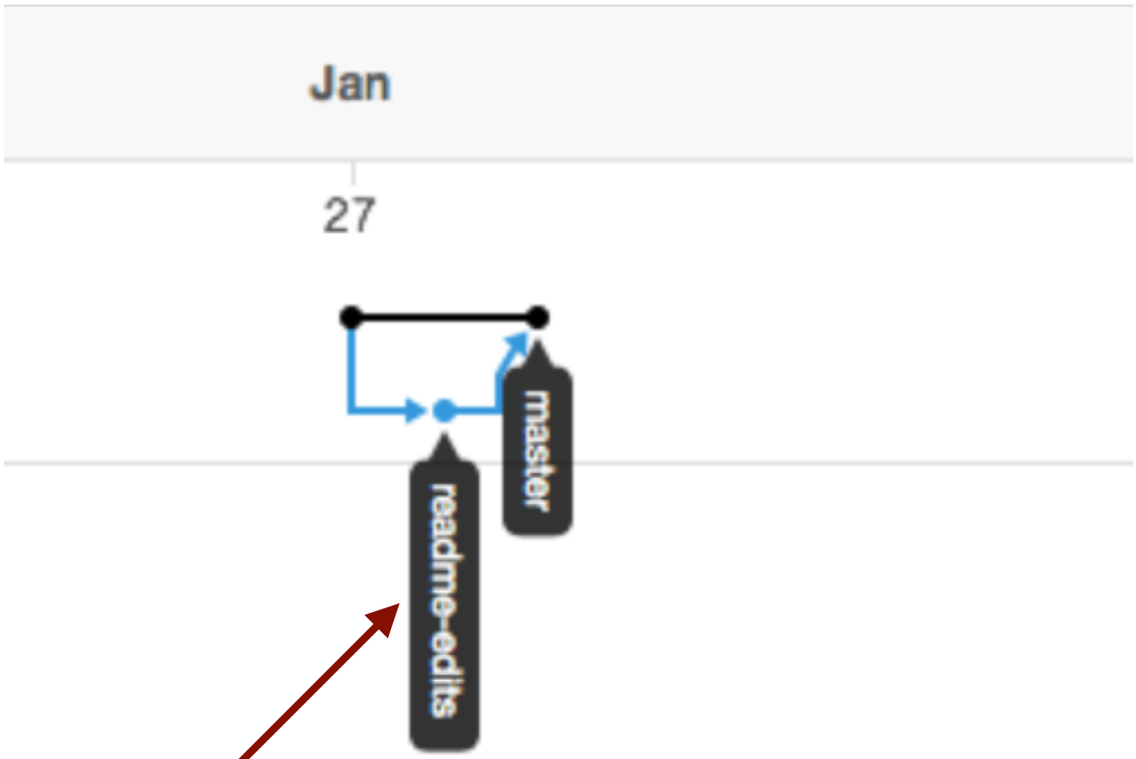
- But what did we do? You can check all sorts of nice things in github:



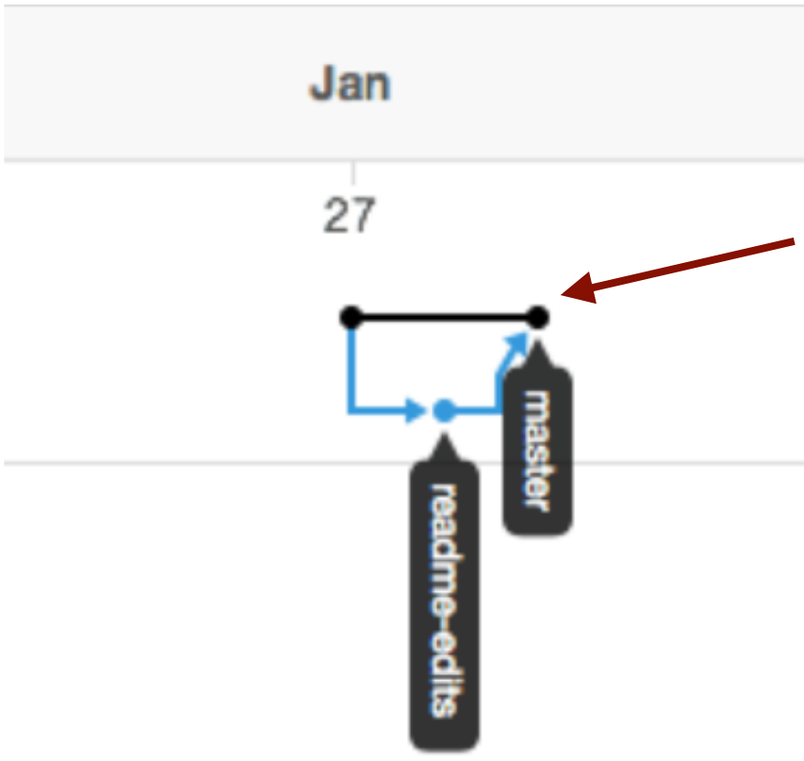
git



git



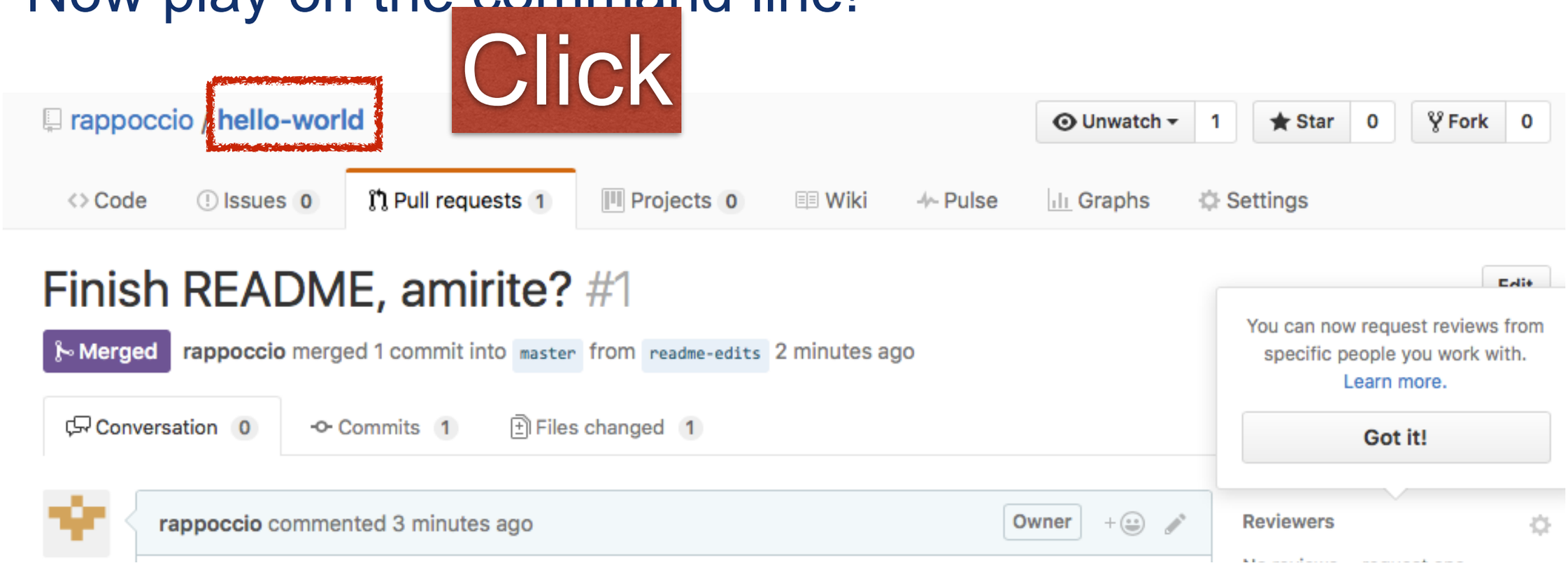
Topic branch



Topic branch merged into Master branch

git

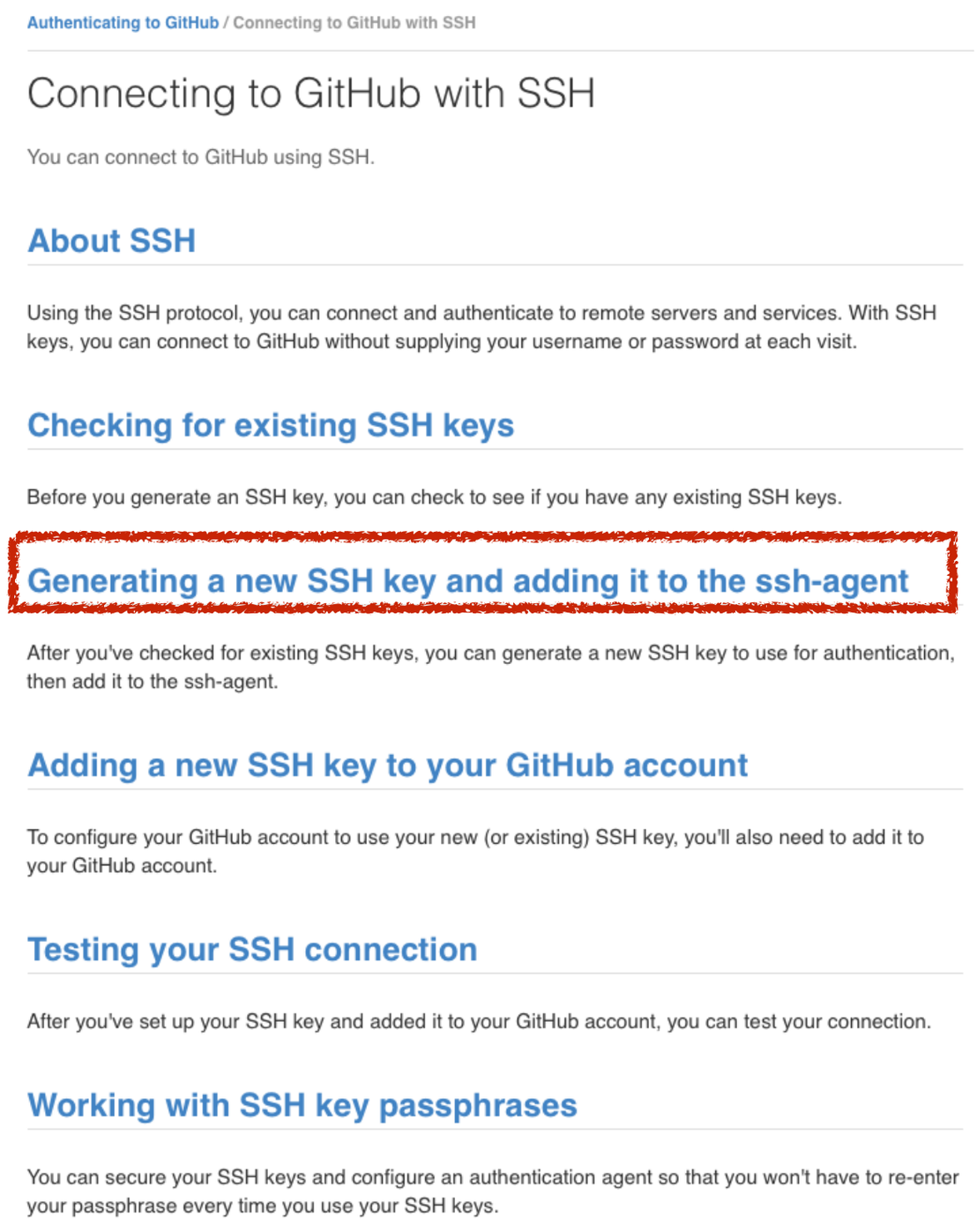
- Now play on the command line!



git

- Nice interactive app to practice here :
  - https://try.github.io/levels/1/challenges/1

- Before we do anything, need to get ssh set up for github :
  - https://help.github.com/articles/connecting-to-github-with-ssh/
-



Authenticating to GitHub / Connecting to GitHub with SSH

Connecting to GitHub with SSH

You can connect to GitHub using SSH.

About SSH

Using the SSH protocol, you can connect and authenticate to remote servers and services. With SSH keys, you can connect to GitHub without supplying your username or password at each visit.

Checking for existing SSH keys

Before you generate an SSH key, you can check to see if you have any existing SSH keys.

Generating a new SSH key and adding it to the ssh-agent

After you've checked for existing SSH keys, you can generate a new SSH key to use for authentication, then add it to the ssh-agent.

Adding a new SSH key to your GitHub account

To configure your GitHub account to use your new (or existing) SSH key, you'll also need to add it to your GitHub account.

Testing your SSH connection

After you've set up your SSH key and added it to your GitHub account, you can test your connection.

Working with SSH key passphrases

You can secure your SSH keys and configure an authentication agent so that you won't have to re-enter your passphrase every time you use your SSH keys.

git

- Need to utilize ssh for security :

```
$ ssh-keygen -t rsa -b 4096 -C "rappoccio@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/nsm/home/srrappoc/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /nsm/home/srrappoc/.ssh/id_rsa.
Your public key has been saved in /nsm/home/srrappoc/.ssh/id_rsa.pub.
The key fingerprint is:
0c:6f:ef:d1:e2:5f:c3:0a:fc:af:8a:45:80:e9:e8:e4 rappoccio@gmail.com
The key's randomart image is:
+--[ RSA 4096]----+
|                 |
|        o        |
|       + .       |
|      o + .      |
|     o . S .     |
|    +   . + . .  |
|     E     B . +  |
|         = = o . |
|        . +o=o.  |
+-----------------+
```

git

- Enable the ssh agent :
- bash :

```
eval “$(ssh-agent -s)”
```

- tcsh / csh :

```
eval `ssh-agent`
```

- Then add your key:

```
$ ssh-add ~/.ssh/id_rsa
Enter passphrase for /nsm/home/srrappoc/.ssh/id_rsa:
Identity added: /nsm/home/srrappoc/.ssh/id_rsa (/nsm/home/srrappoc/.ssh/id_rsa)
```

- And add your key to the github account:
  - https://help.github.com/articles/adding-a-new-ssh-key-to-your-github-account/

Adding a new SSH key to your GitHub account

MAC | WINDOWS | LINUX

To configure your GitHub account to use your new (or existing) SSH key, you'll also need to add it to your GitHub account.

Before adding a new SSH key to your GitHub account, you should have:

- Checked for existing SSH keys
- Generated a new SSH key and added it to the ssh-agent

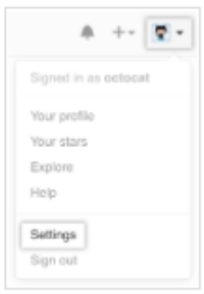
1. Copy the SSH key to your clipboard.

   If your SSH key file has a different name than the example code, modify the filename to match your current setup. When copying your key, don't add any newlines or whitespace.

   ```
   $ pbcopy < ~/.ssh/id_rsa.pub
   # Copies the contents of the id_rsa.pub file to your clipboard
   ```

   Tip: If `pbcopy` isn't working, you can locate the hidden `.ssh` folder, open the file in your favorite text editor, and copy it to your clipboard.

2. In the upper-right corner of any page, click your profile photo, then click **Settings**.



3. In the user settings sidebar, click **SSH and GPG keys**.

4. Click **New SSH key** or **Add SSH key**.

5. In the "Title" field, add a descriptive label for the new key. For example, if you're using a

git

- Copy and paste your key :

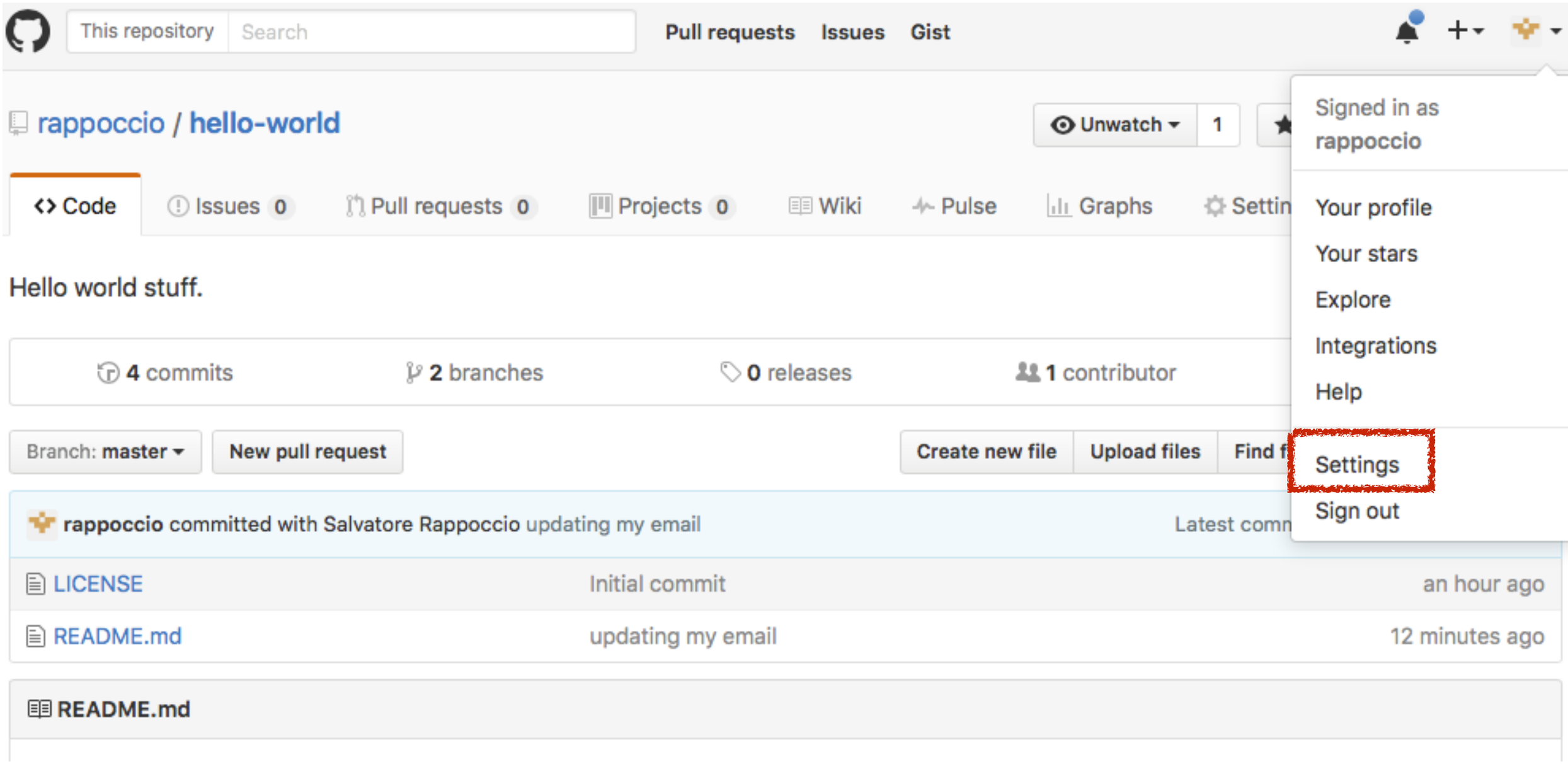
```
$ cat ~/.ssh/id_rsa.pub
```

- Will look something like this (but longer)(copy the whole thing):

```
ssh-rsa blablablablabla= rappoccio@gmail.com
```

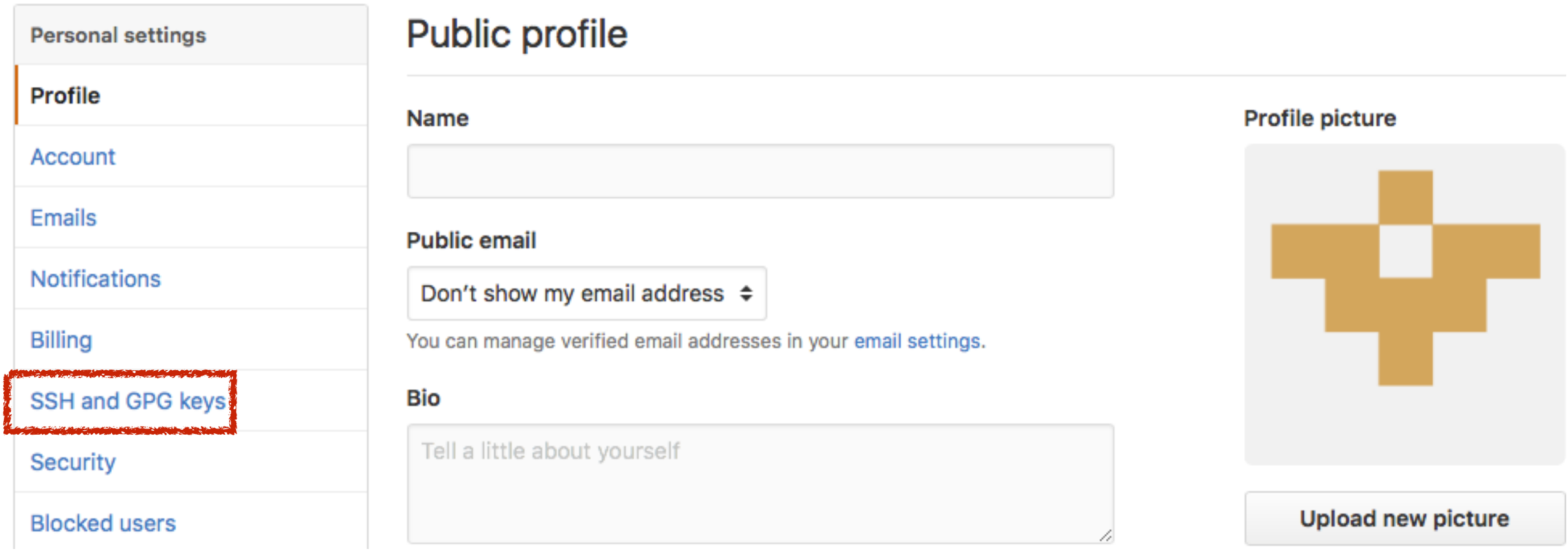
git

- Then select your “settings” :



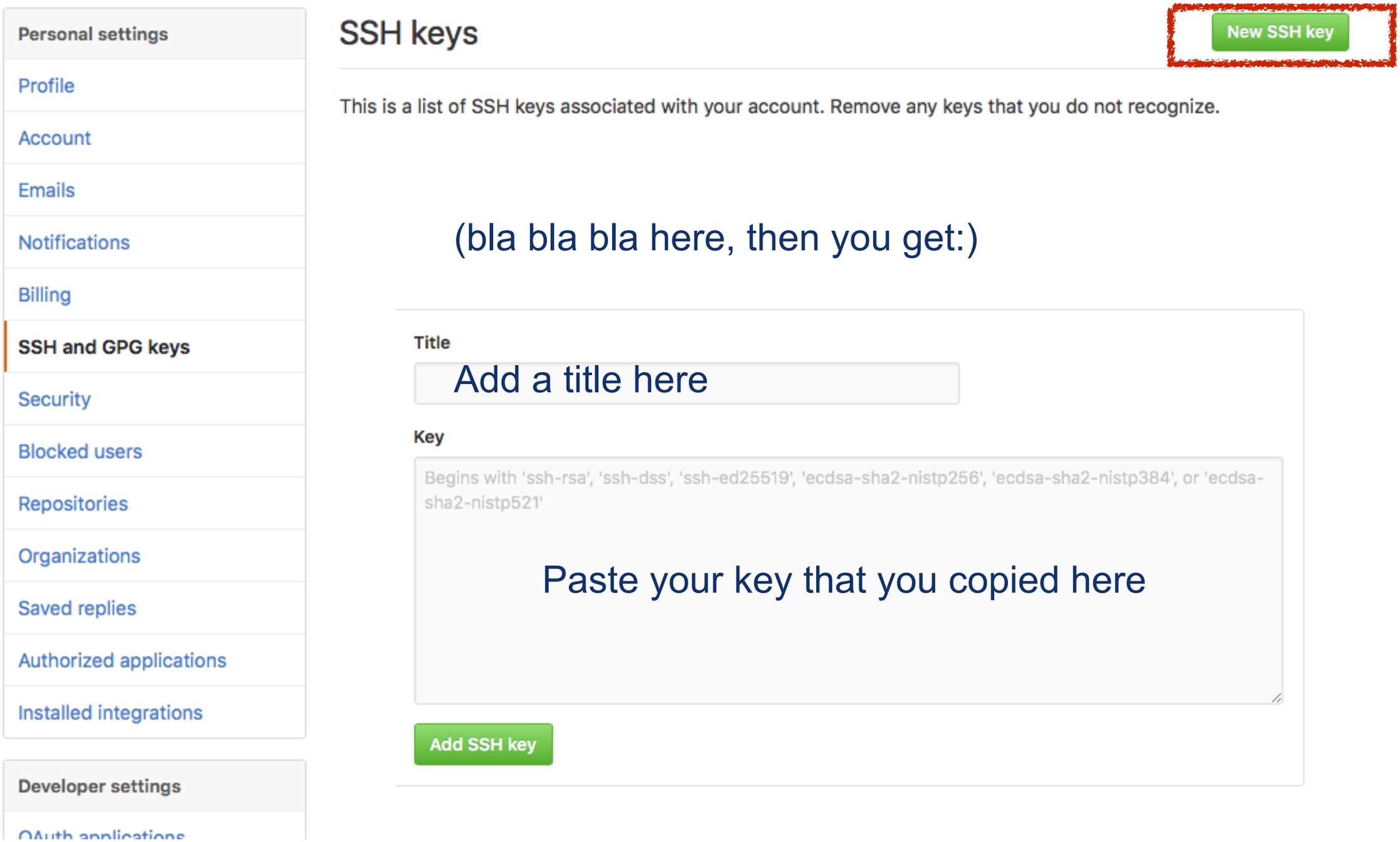
git

- Then your “SSH and GPG keys”



git

- Then “add new ssh key”



git

rappoccio / hello-world

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Pulse | Graphs | Settings

Hello world stuff.

Edit

Click

3 commits | 2 branches | 0 releases | 1 | GPL-3.0

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

rappoccio committed on GitHub Merge pull request #1 from rappoccio/readme-edits ... Latest commit 0473ddc 7 minutes ago

LICENSE	Initial commit	19 minutes ago
README.md	Finish README, amirite?	14 minutes ago

README.md

hello-world

Hello world stuff.

and this is a strange question, amirite?

git

rappoccio / hello-world

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0

Copy repository name to clipboard

Hello world stuff.

Edit

3 commits | 2 branches

Branch: master | New pull request

Create new file | Upload files | Find file | Clone or download

rappoccio committed on GitHub Merge pull request #1 from rappoccio/readme-edits ...

LICENSE	Initial commit
README.md	Finish README, amirite?

Clone with SSH | Use HTTPS

Use an SSH key and passphrase from account.

git@github.com:rappoccio/hello-world.git

Open in Desktop | Download ZIP

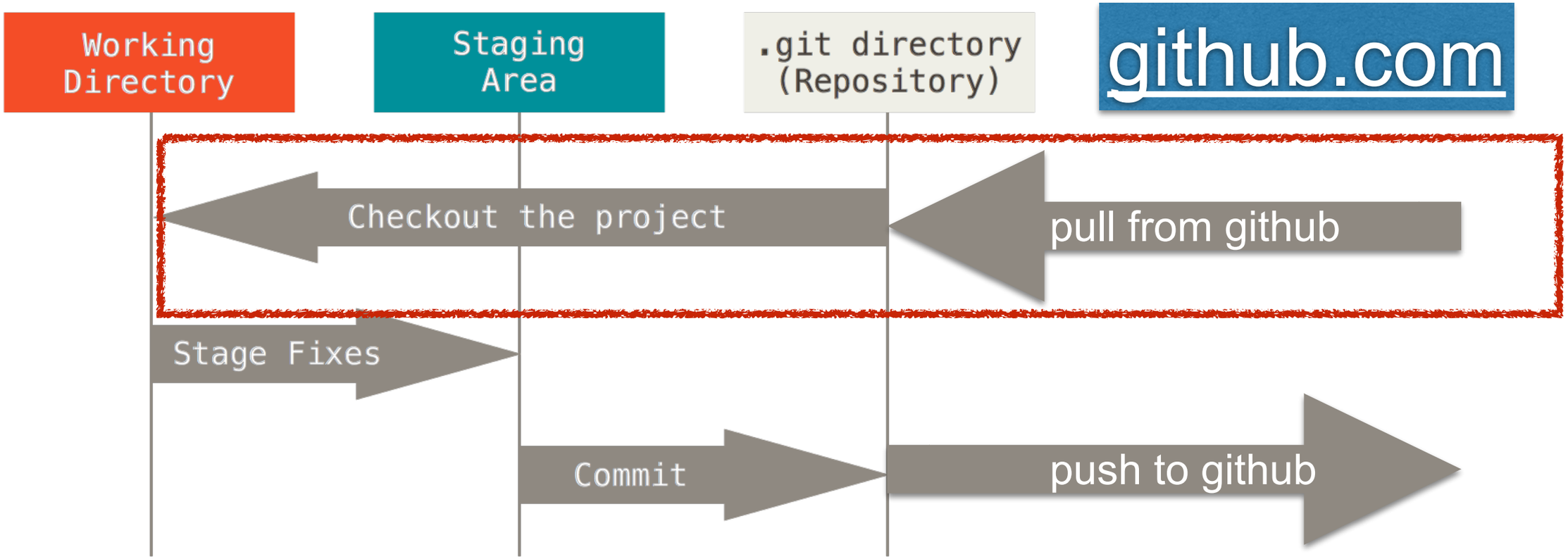
README.md

hello-world

Hello world stuff.

and this is a strange question, amirite?

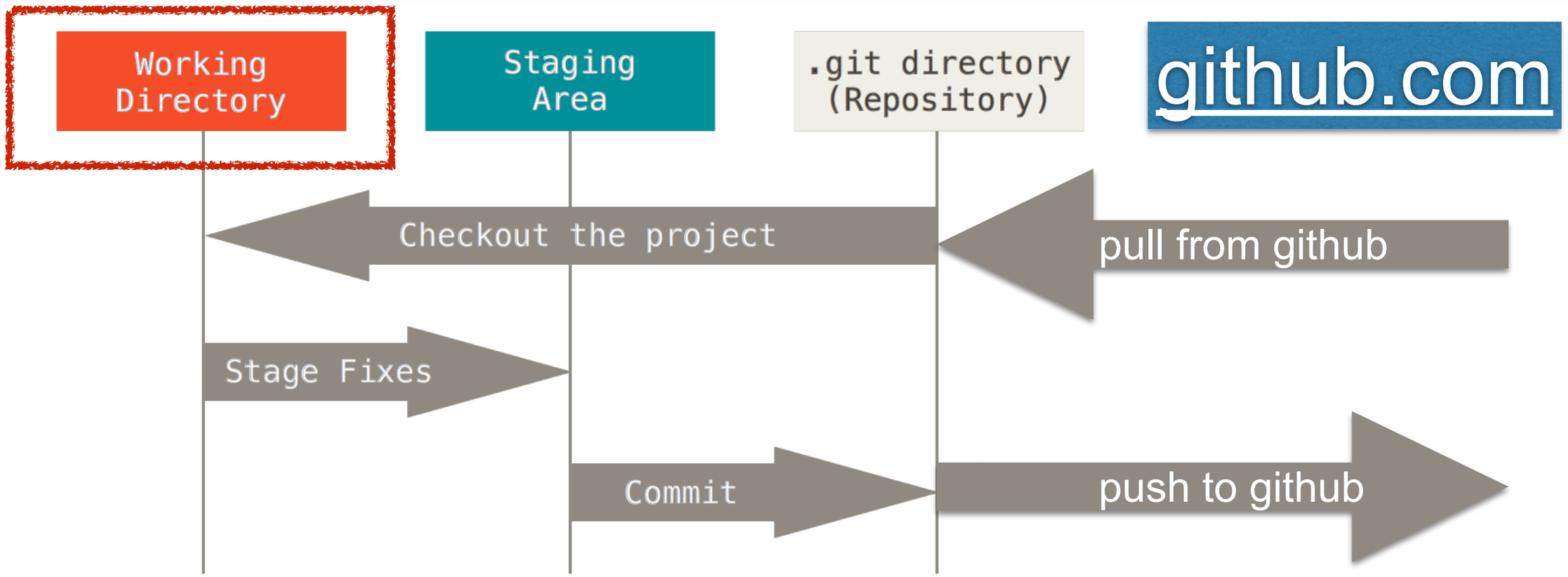
git



- git clone : pulls a repository from github.com, updates working directory in one step

```
$ mkdir working
$ cd working
/nsm/home/srrappoc/working
$ git clone https://github.com/rappoccio/hello-world.git
Initialized empty Git repository in /nsm/home/srrappoc/working/hello-world/.git/
remote: Counting objects: 8, done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (8/8), done.
$
```

git



- Go to the “hello-world” directory, and edit “README.md”.
- Add some text:

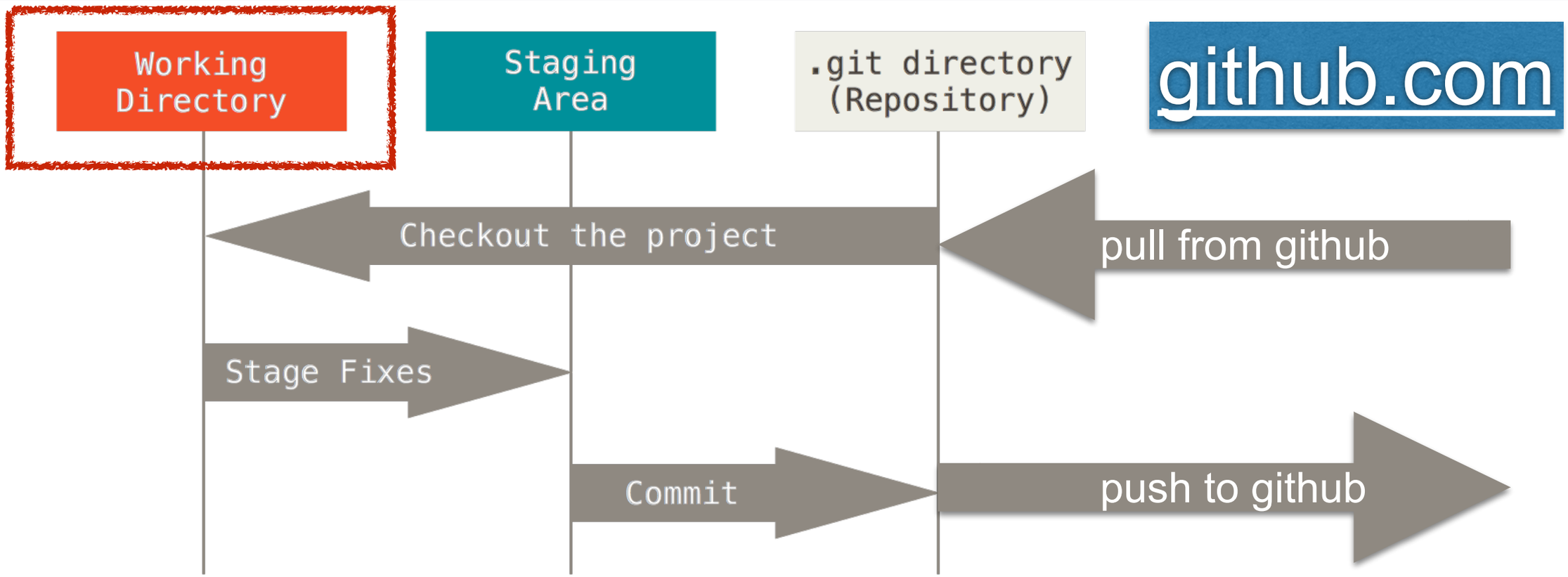
```
# hello-world
Hello world stuff.


and this is a strange question, amirite?


Wow! Works on the command line too!
```

- This updates the WORKING directory

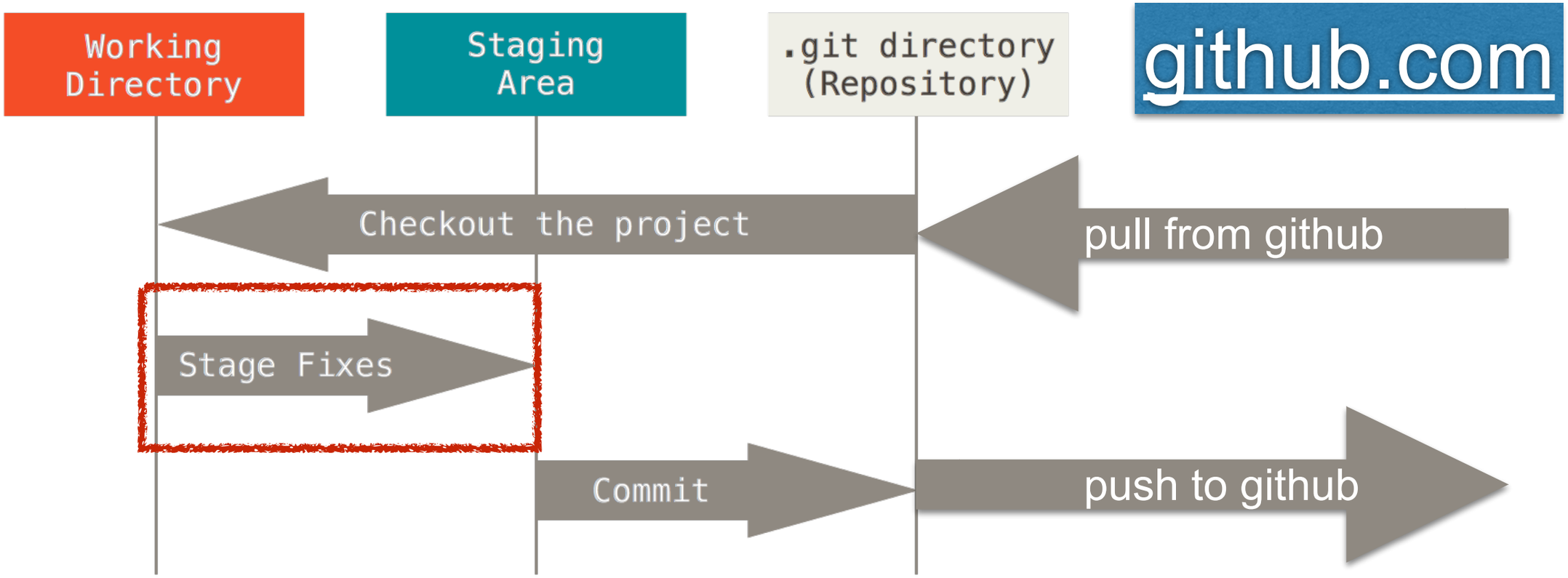
git



- git status : check your working directory:

```
$ git status
# On branch master
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#  modified:   README.md
#
```

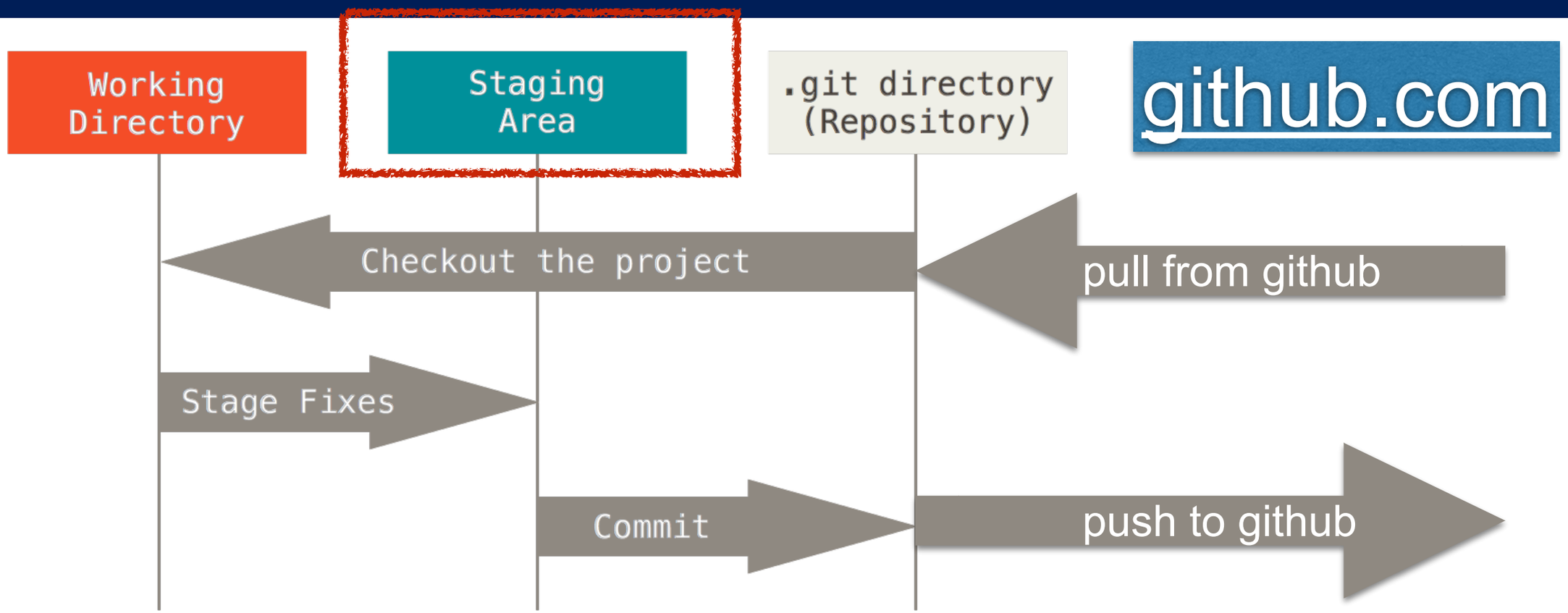
git



- git add : add files to be staged:

```
$ git add README.md
```

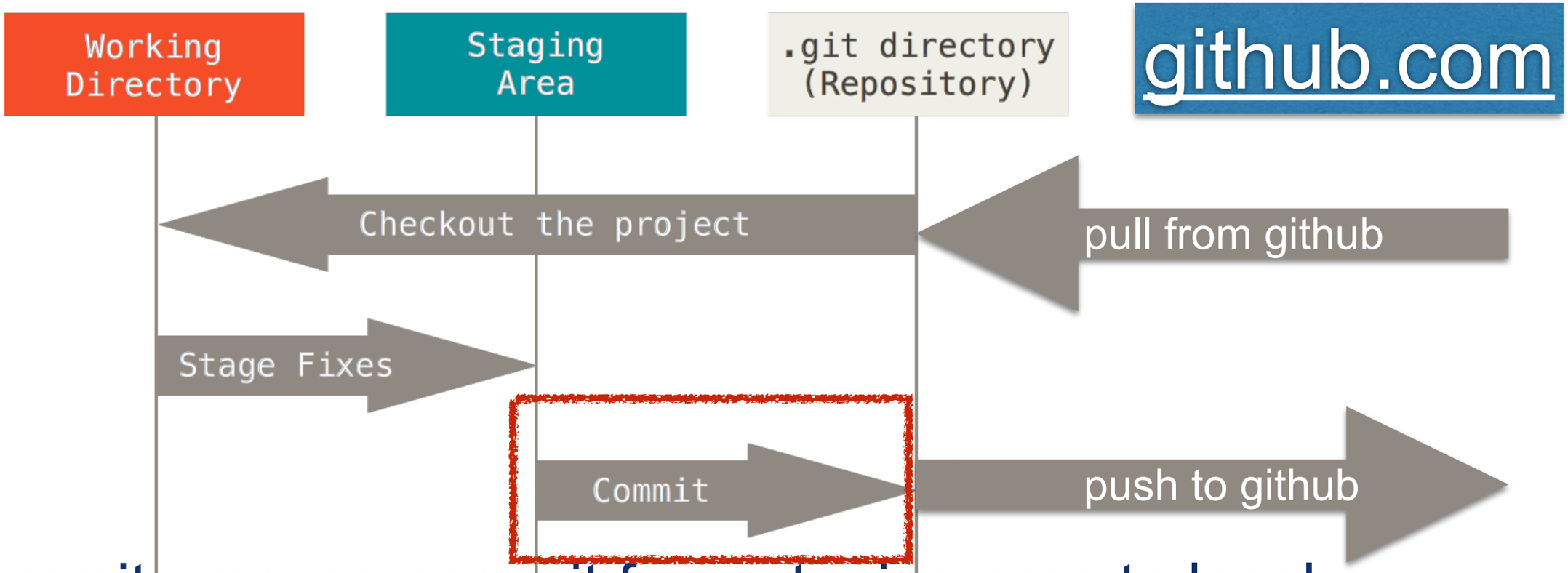
git



- Now updated README.md is in the staging area:

```
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#  modified:   README.md
#
```

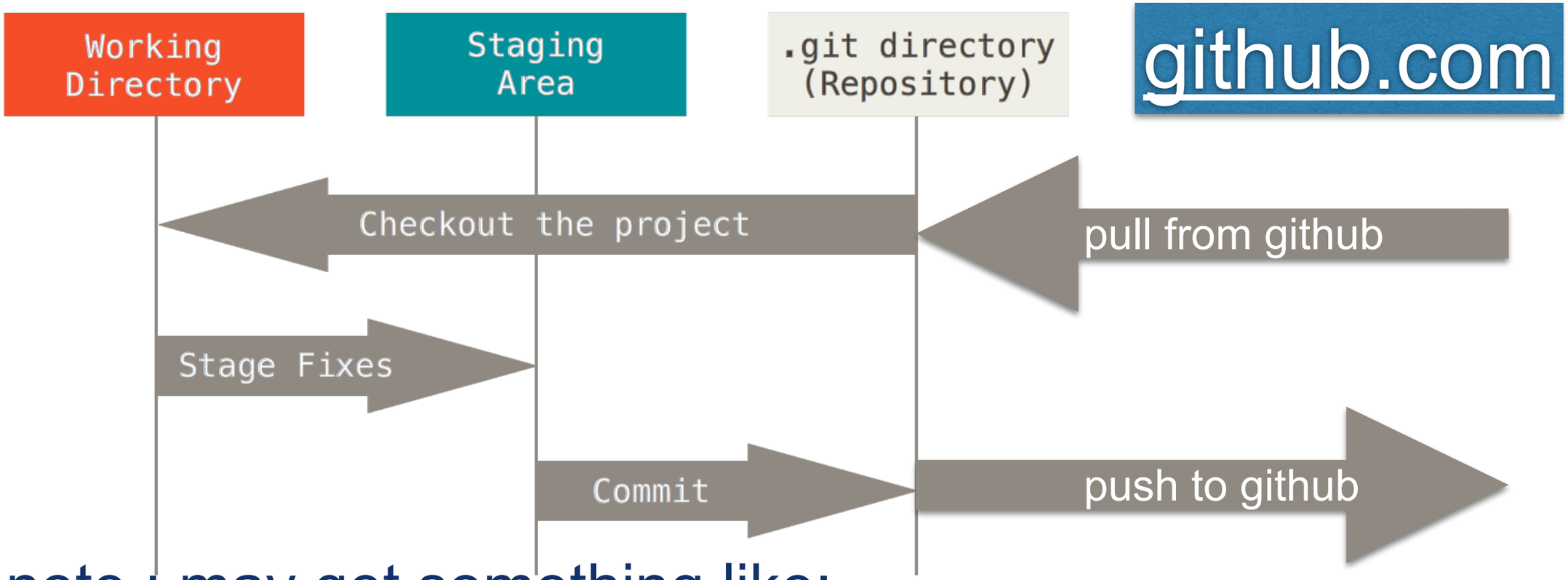
git



- git commit : moves commit from staging area to local repository

```
$ git commit -m"Adding updated README.md"
[master eab61b9] Adding updated README.md
 1 files changed, 3 insertions(+), 0 deletions(-)
```

git



- Side note : may get something like:

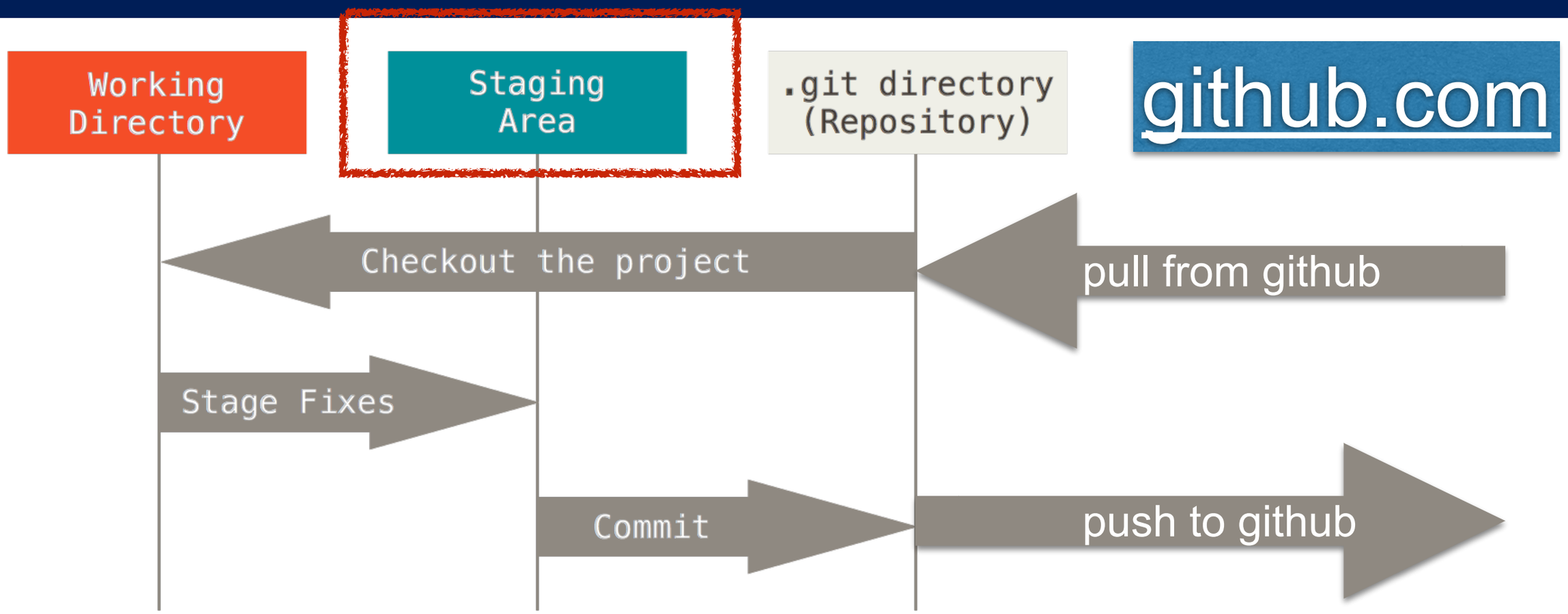
```
our name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

If the identity used for this commit is wrong, you can fix it with:

    git commit --amend --author='Your Name <you@example.com>'
```

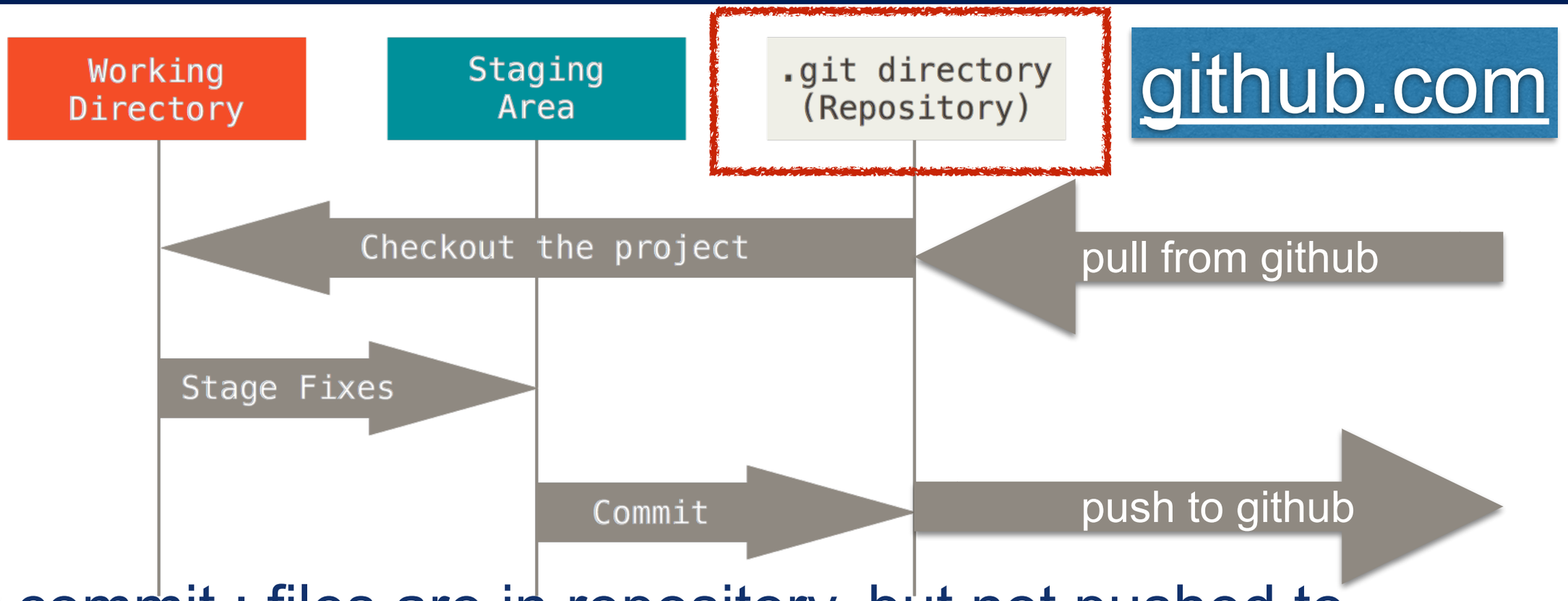
git



- Can fix :

```
$ git config --global user.name "Salvatore Rappoccio"
$ git config --global user.email srrappoc@buffalo.edu
$ git commit --amend --author='Salvatore Rappoccio <srrappoc@buffalo.edu>'
[master 46e836f] updating my email
 1 files changed, 3 insertions(+), 0 deletions(-)
```

git

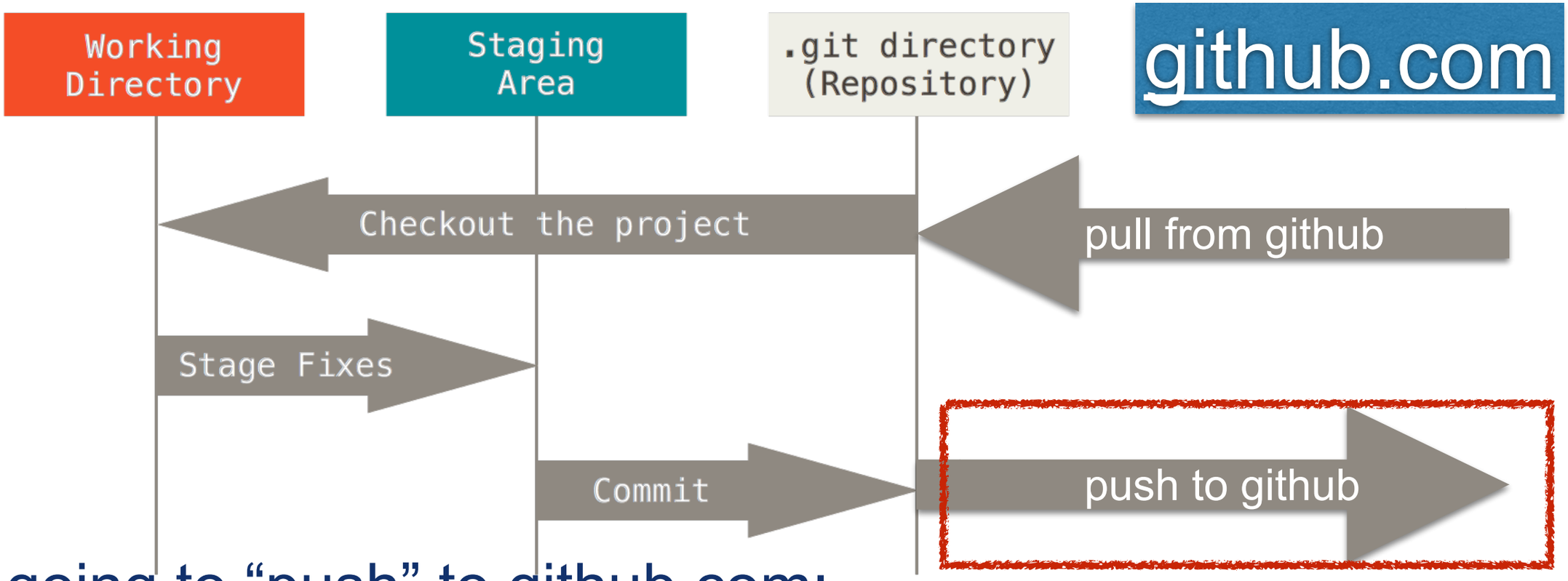


- After commit : files are in repository, but not pushed to “origin”

```
$ git status
# On branch master
# Your branch is ahead of 'origin/master' by 1 commit.
#
nothing to commit (working directory clean)
```

By default, the “remote” repository gets a name “origin”

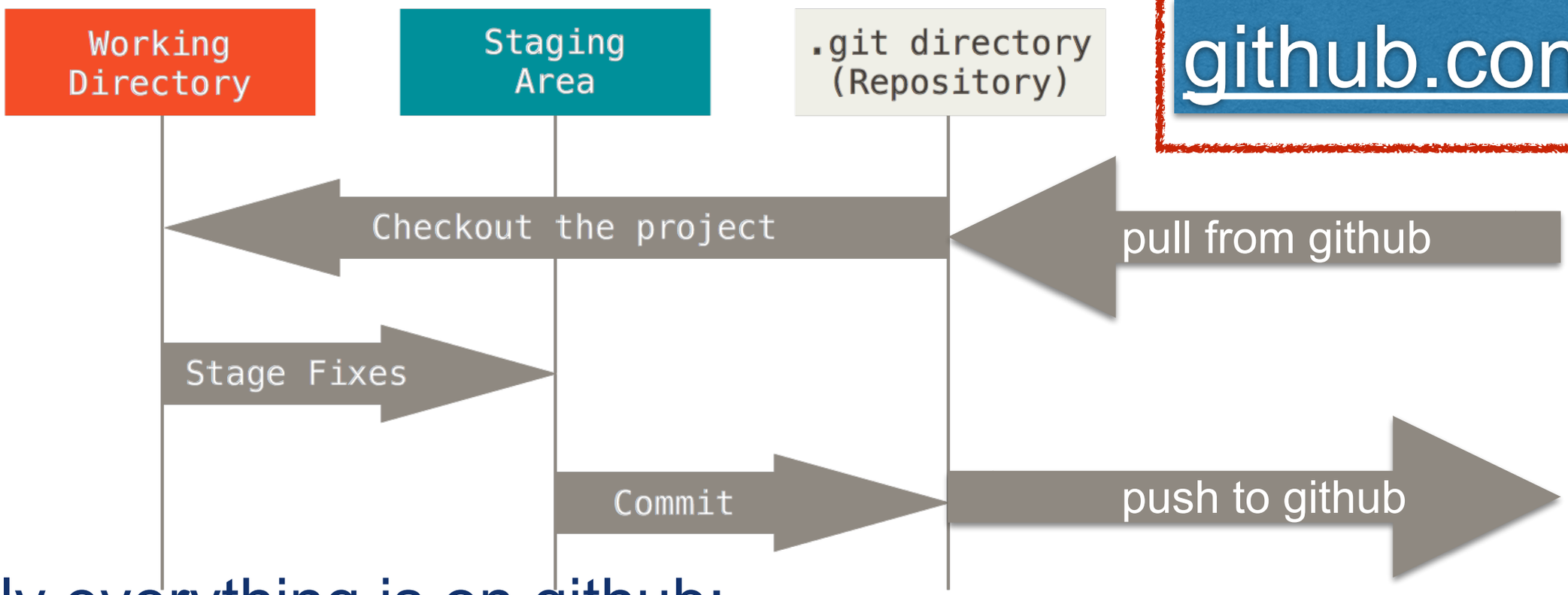
git



- Now going to “push” to github.com:

```
$ git push origin master
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 390 bytes, done.
Total 3 (delta 0), reused 0 (delta 0)
To git@github.com:rappoccio/hello-world.git
   0473ddc..0981661  master -> master
```

git



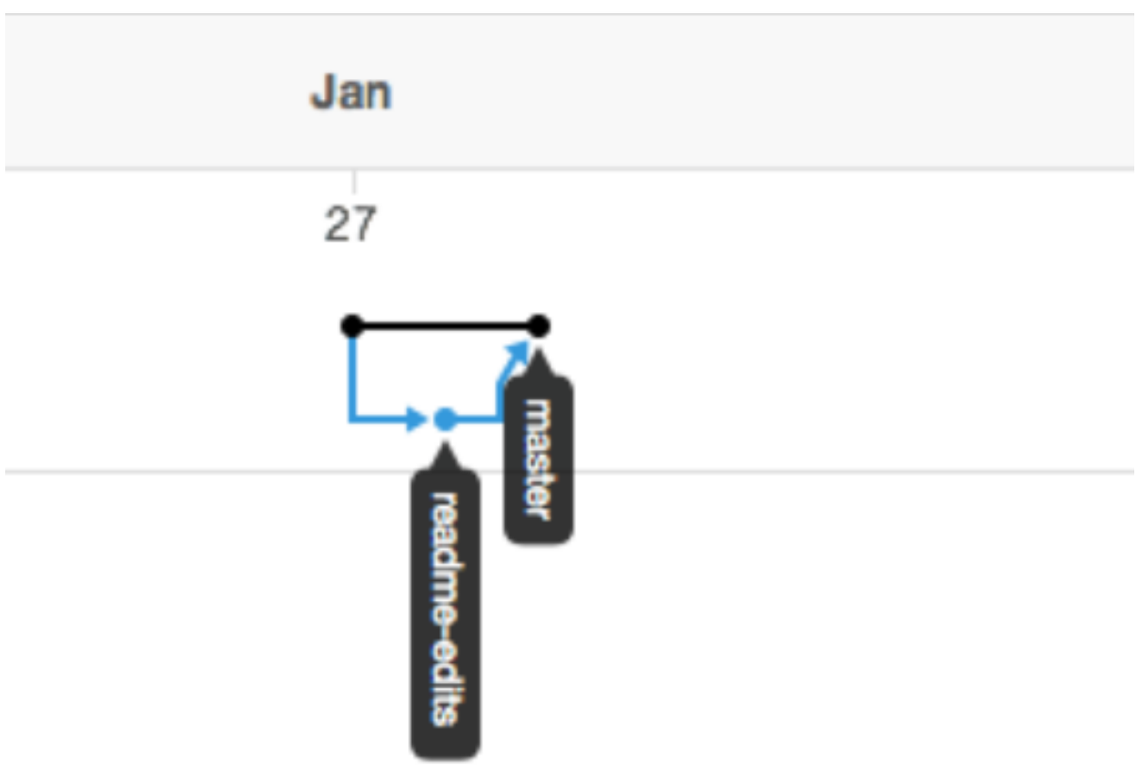
- Finally everything is on github:

```
$ git status
# On branch master
nothing to commit (working directory clean)
```

git

- Now check github again (main page, “Graphs”, “Network”)

Before



Now

git

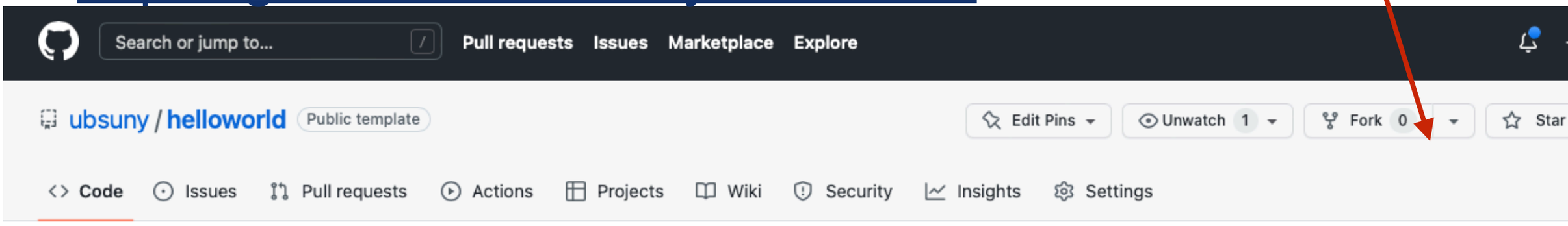
- Nice thing about git : all of the branching is basically delocalized :
  - You can branch your own projects
  - You can branch your own branches
  - You can branch OTHER people's projects
  - You can branch OTHER people's branches

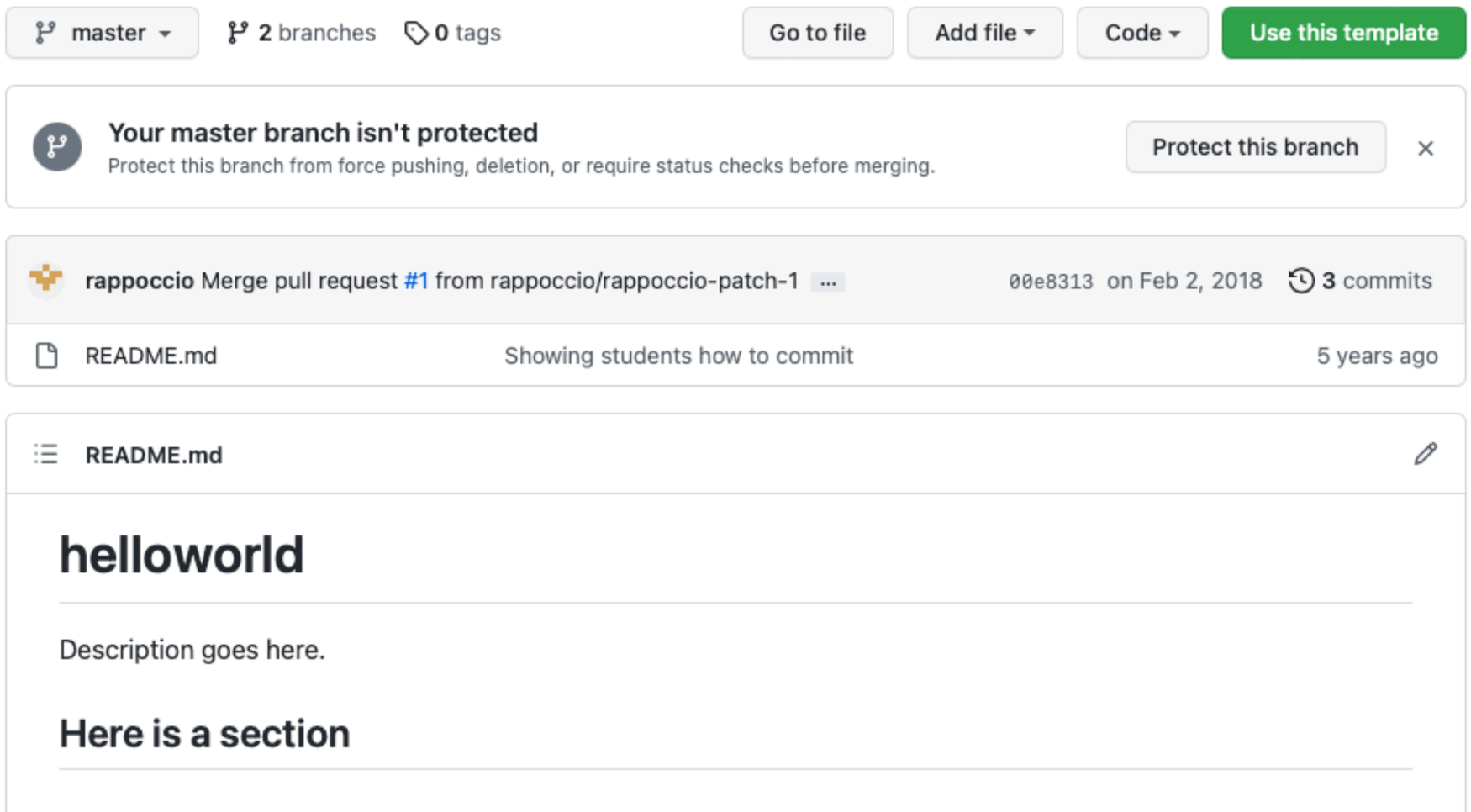
- Making a branch of someone's entire project is called a "fork"

git

- You will get forks for this class.
- Use the “helloworld” as an example:
  - https://github.com/ubsuny/helloworld

FORK IT





git

- Now you are able to clone your OWN repository:

```
$ git clone git@github.com:dummyrappoccio/helloworld.git
Initialized empty Git repository in /nsm/home/srrappoc/working/helloworld/.git/
remote: Counting objects: 859, done.
remote: Compressing objects: 100% (446/446), done.
remote: Total 859 (delta 375), reused 859 (delta 375), pack-reused 0
Receiving objects: 100% (859/859), 3.04 MiB, done.
Resolving deltas: 100% (375/375), done.
$
```

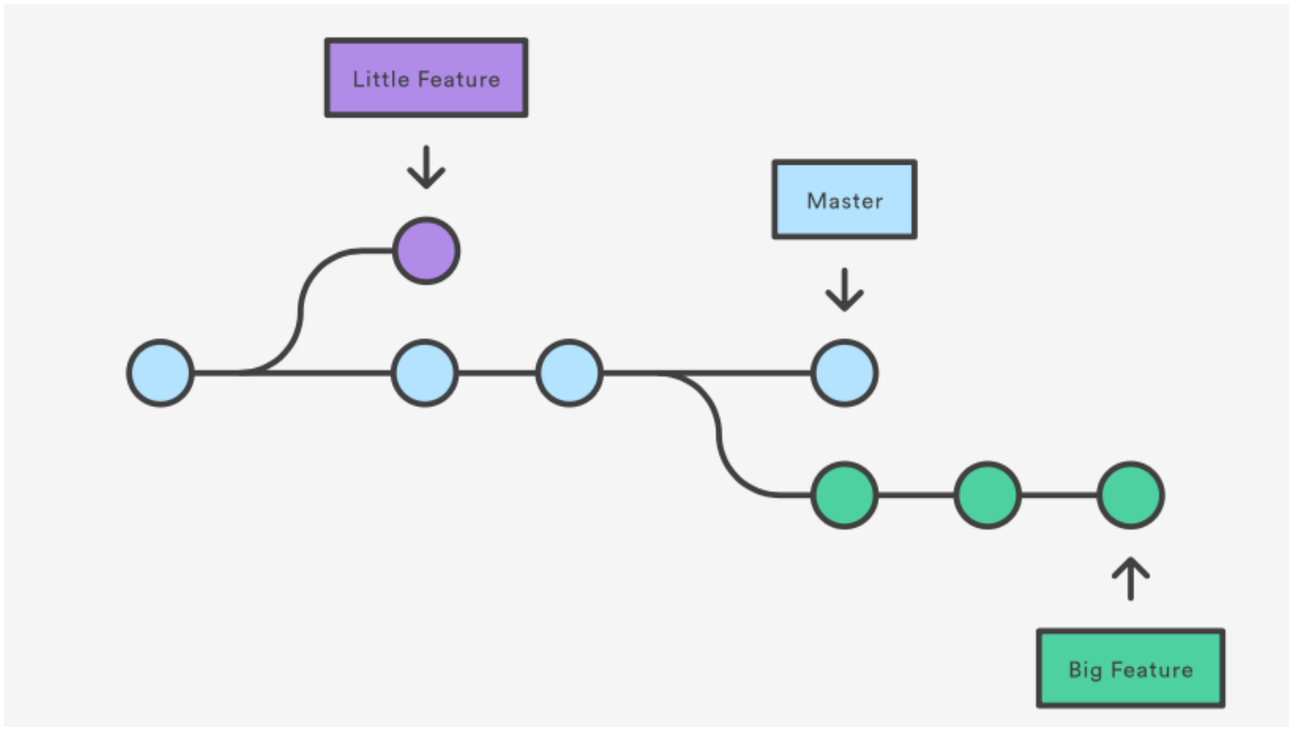
Your name will be here

git

- Let’s now create a branch

- First, a bit more on branches
  - https://www.atlassian.com/git/tutorials/using-branches

- Branch structure is a graph (as in graph theory)



- Your “main” trunk is called “master”
- The individual snapshots are leafs
- Topic branches are split from a specific leaf on ANY branch (including master, shown here)

git

- Let’s create a branch as an example

- We should be on a clean branch in our working directory, cloned from our fork on github.com

- First check status :

```
$ git status
# On branch master
nothing to commit (working directory clean)
```

- Now create the branch with:

```
$ git checkout -b test
Switched to a new branch ‘test'
```

git

- You can see the status of your branches here :

```
$ git branch -v
* HW1    fee04d9 Fixing question of running ARK4 or RK4
  master fee04d9 Fixing question of running ARK4 or RK4
```

- The starred one is the one you are on
- The second column is a short name for the commit you're interested in.
- The third column is the comment for that commit.

git

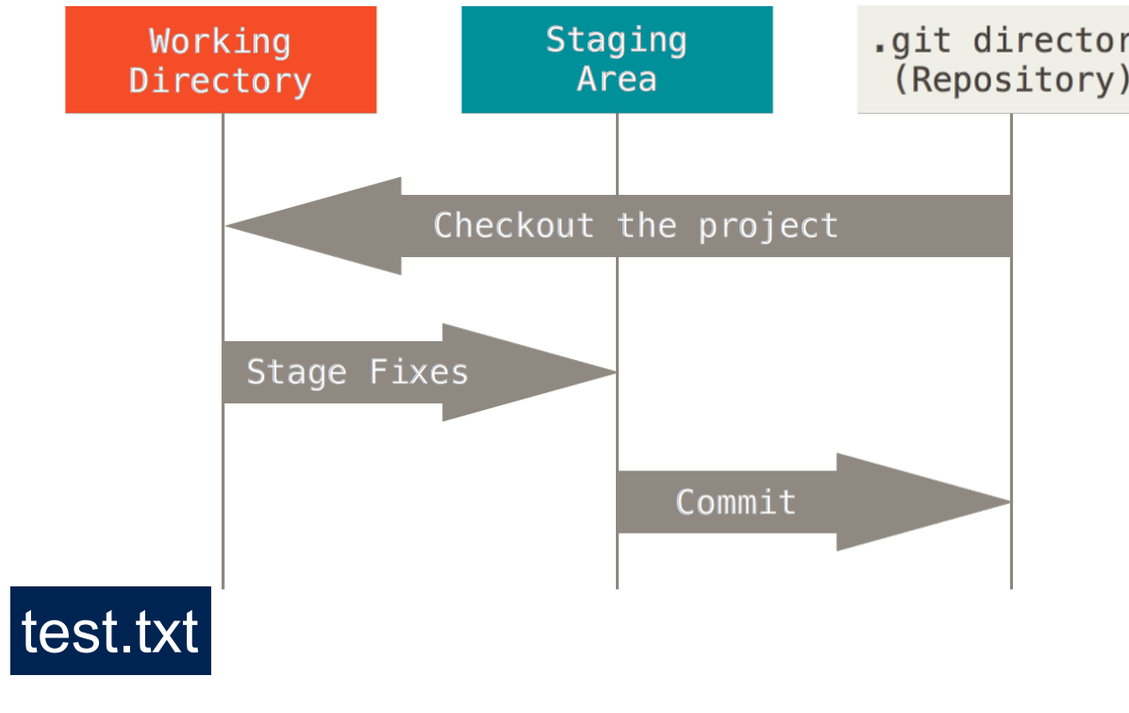
- So now let's get your HW1 set up. First make a directory, and then add a file to it. Finally do "git status", and this is what you should see:

```
$ mkdir Test
$ echo "Test goes here" > Test/test.txt
$ git status
# On branch test
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
# Test/
nothing added to commit but untracked files present (use "git add" to track)
```

- The important piece is that you see Test in your working area, but it is not staged "nothing added to commit but untracked files present"

git

- So what's happening here?
  - Presently "test.txt" is in your working directory

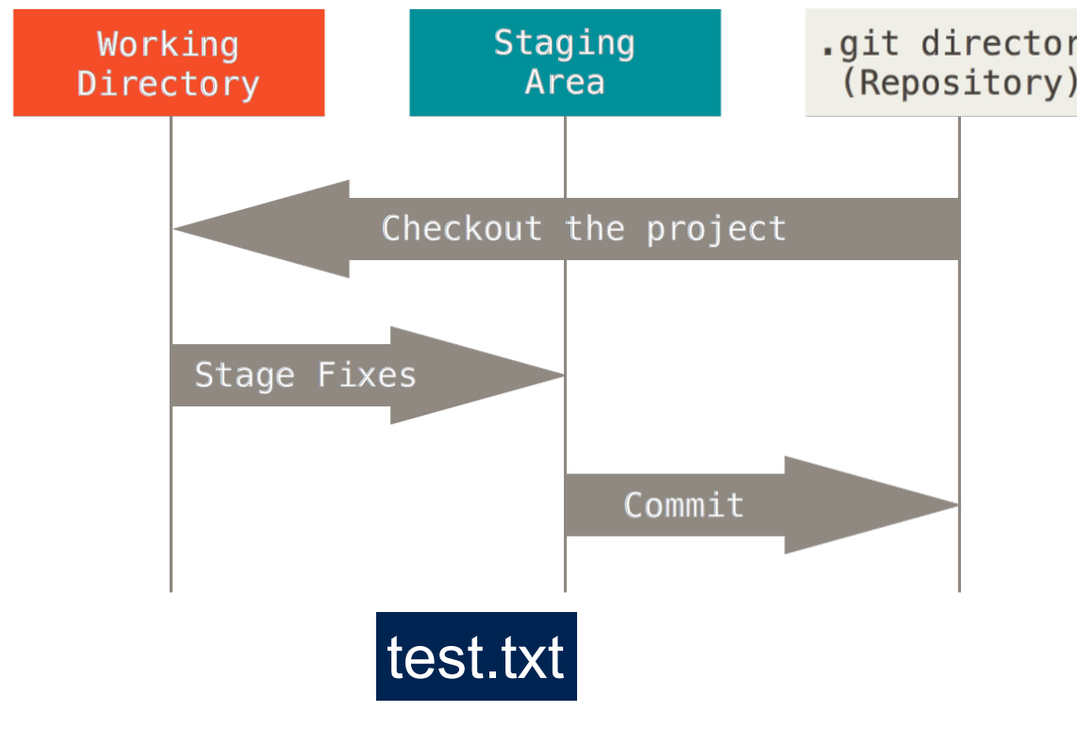


git

- So what's happening here?
  - Presently "test.txt" is in your working directory

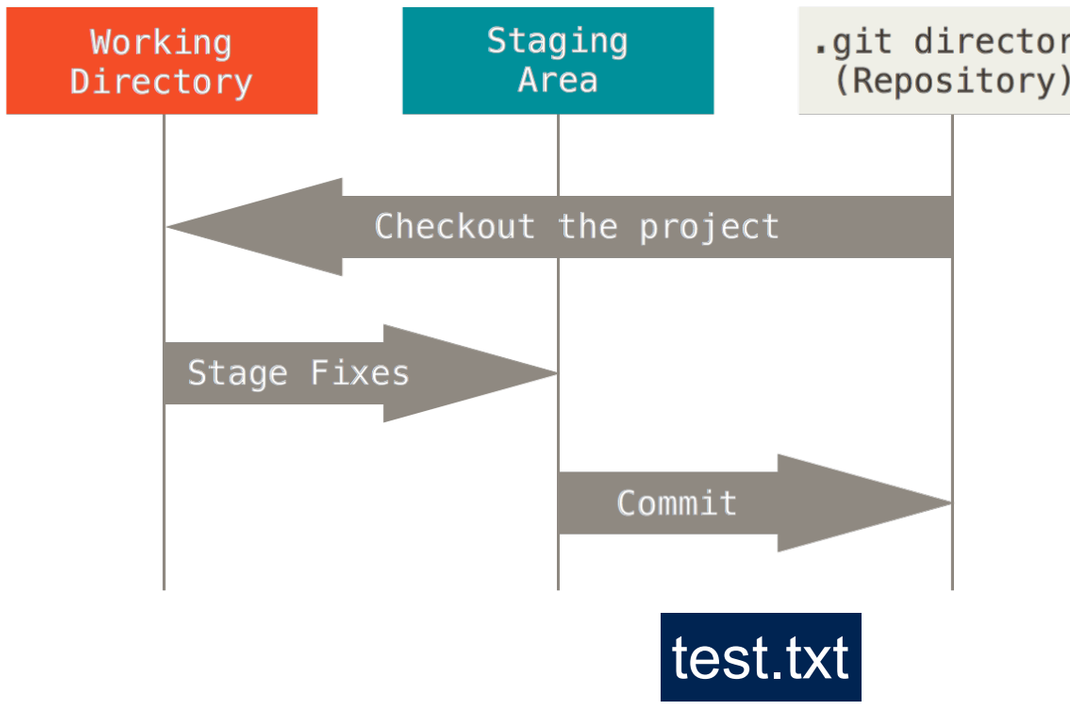
- Now add to your staging area:

```
$ git add Test/test.txt
```



git

- So what's happening here?
  - Presently "test.txt" is in your working directory



- Now add to your staging area:

```
$ git add Test/test.txt
```

- And finally commit to your .git directory (local repository)

```
$ git commit -m"Adding test" .
[HW1 d5f152a] Adding test
 1 files changed, 1 insertions(+), 0 deletions(-)
 create mode 100644 Test/test.txt
```

git

- What happened here? Check the status :

```
$ git status
# On branch test
nothing to commit (working directory clean)
$ git branch -v
* test   47d8940 Adding test
  master fee04d9 Showing students how to commit
```

Can now see that the commit on “test” branch is different from the commit on the “master” branch

For your Homework : Github Classroom!

- Your homework will therefore be :
  - Write up (latex, word, jupyter notebook, or some equivalent)
  - Code (do not submit binaries)