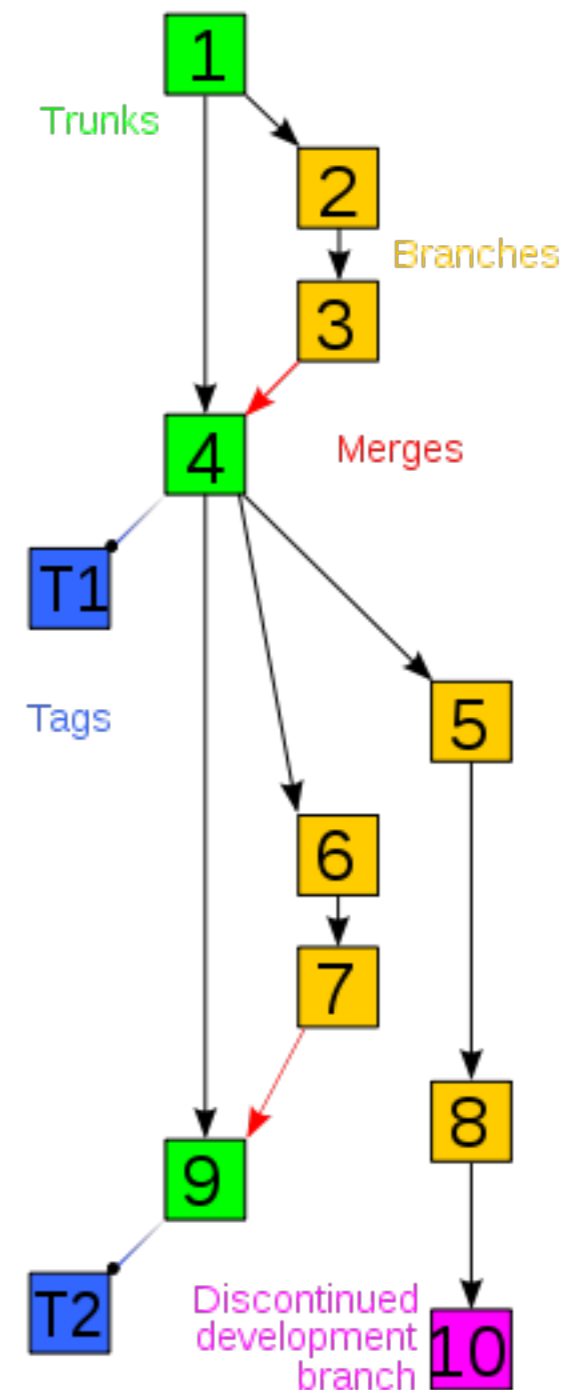


PHY410 / 505  
Computational Physics 1

**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



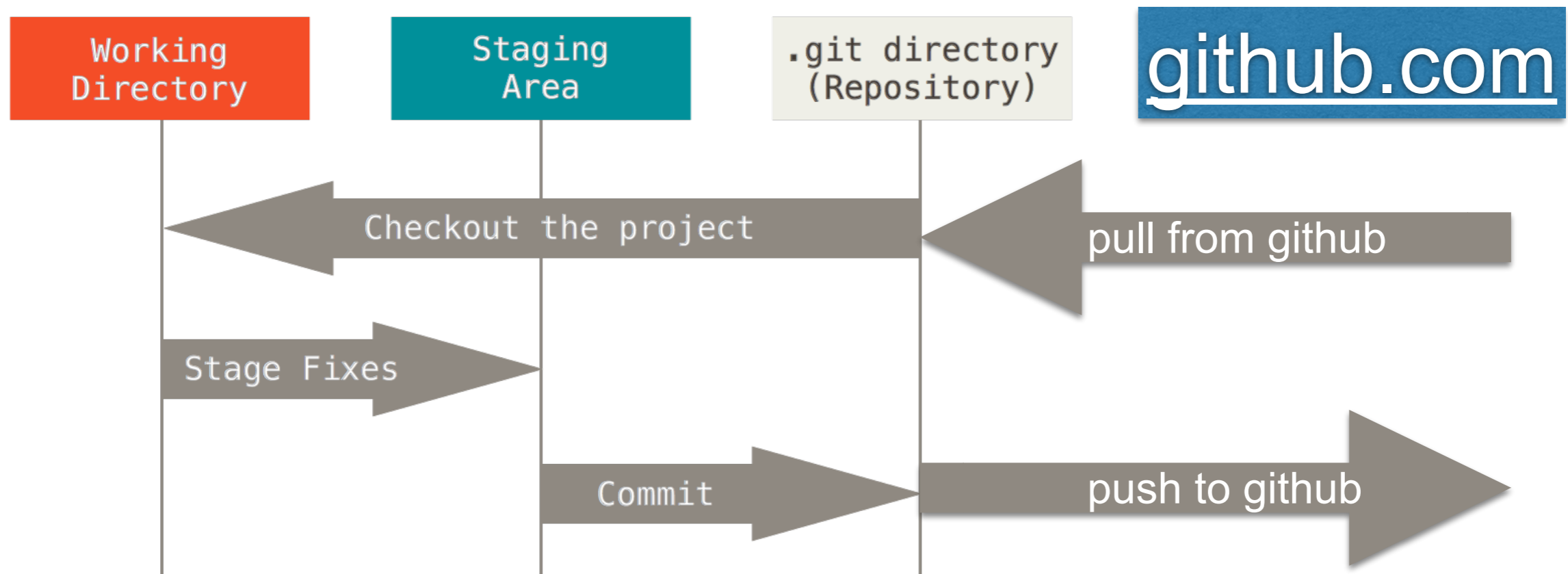
[https://en.wikipedia.org/wiki/Version\\_control#/media/File:Revision\\_controlled\\_project\\_visualization-2010-24-02.svg](https://en.wikipedia.org/wiki/Version_control#/media/File:Revision_controlled_project_visualization-2010-24-02.svg)

# 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](https://github.com) for our purposes. You must get an account.

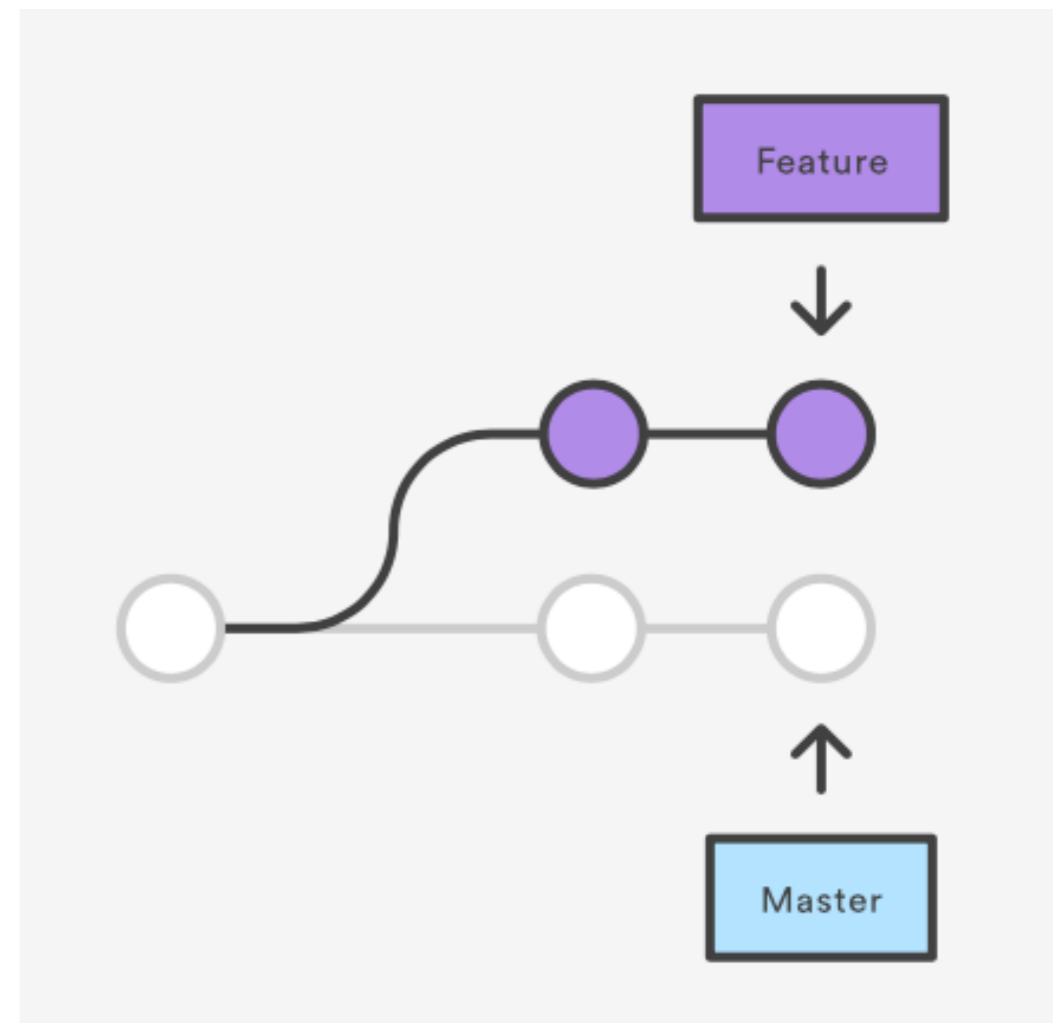
# git

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



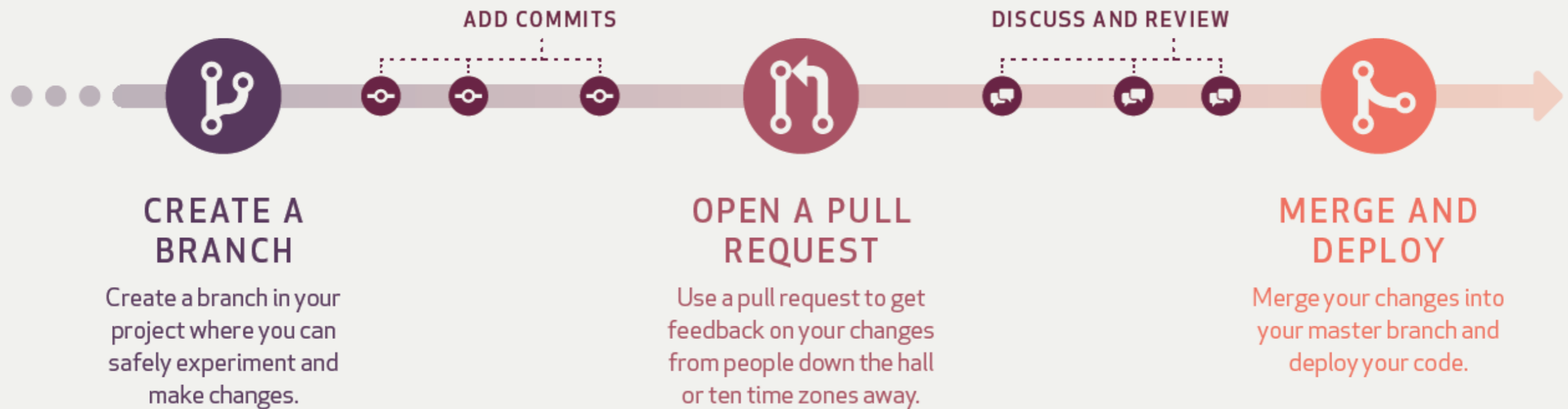
# git

- 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

 rappoccio ▾


Repository name


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

1

★ Star 0

0

Fork 0

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?

uriteva.

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.

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

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

3 README.md

|     |                    |   |
|-----|--------------------|---|
| ... | @@ -1,2 +1,5 @@    |   |
| 1   | # hello-world      | 1 # hello-world                             |
| 2   | Hello world stuff. | 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

# Everything good, clicky clicky

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

```
3 README.md
```

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

No commit comments for this range

## 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

AA B i “ <> 🔗 ☰ ☷ ✓ ↶ @ 📌

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

Reviewers: No reviews— request one

Assignees: No one—assign yourself

Labels: None yet

Create pull request

Everything good, clicky clicky



# git

The screenshot shows a GitHub pull request interface. At the top, the repository name 'rappoccio / hello-world' is displayed with options to 'Unwatch', 'Star', and 'Fork'. Below this, navigation tabs for 'Code', 'Issues', 'Pull requests', 'Projects', 'Wiki', 'Pulse', 'Graphs', and 'Settings' are visible. The main title of the pull request is 'Finish README, amirite? #1', and it indicates that 'rappoccio wants to merge 1 commit into master from readme-edits'. A notification box on the right states, 'You can now request reviews from specific people you work with. Learn more.' with a 'Got it!' button. The pull request details show '0 Reviews' and '0 Assignees'. A status message indicates 'This branch has no conflicts with the base branch' and offers a 'Merge pull request' button. At the bottom, there is a 'Write' section for comments with a text area and a 'Comment' button.

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

# git

The screenshot shows a GitHub pull request titled "Finish README, amirite? #1". The pull request is from the "readme-edits" branch to the "master" branch. A green "Merge pull request" button is highlighted with a red dashed border. The interface includes navigation tabs for Code, Issues, Pull requests, Projects, Wiki, Pulse, Graphs, and Settings. A notification box at the top right says "You can now request reviews from specific people you work with. Learn more." Below the pull request title, there are tabs for Conversation, Commits, and Files changed. A comment from "rappoccio" is visible. At the bottom, there is a comment input field and a "Comment" button. The right sidebar shows settings for Labels, Projects, Milestone, and Notifications, along with an "Unsubscribe" button and a "1 participant" section.

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  
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

1 participant

Lock conversation

# git

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

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

You can now request reviews from specific people you work with.  
[Learn more.](#)

Got it!

Reviewers

rappoccio / hello-world

Unwatch 1 Star 0 Fork 0

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

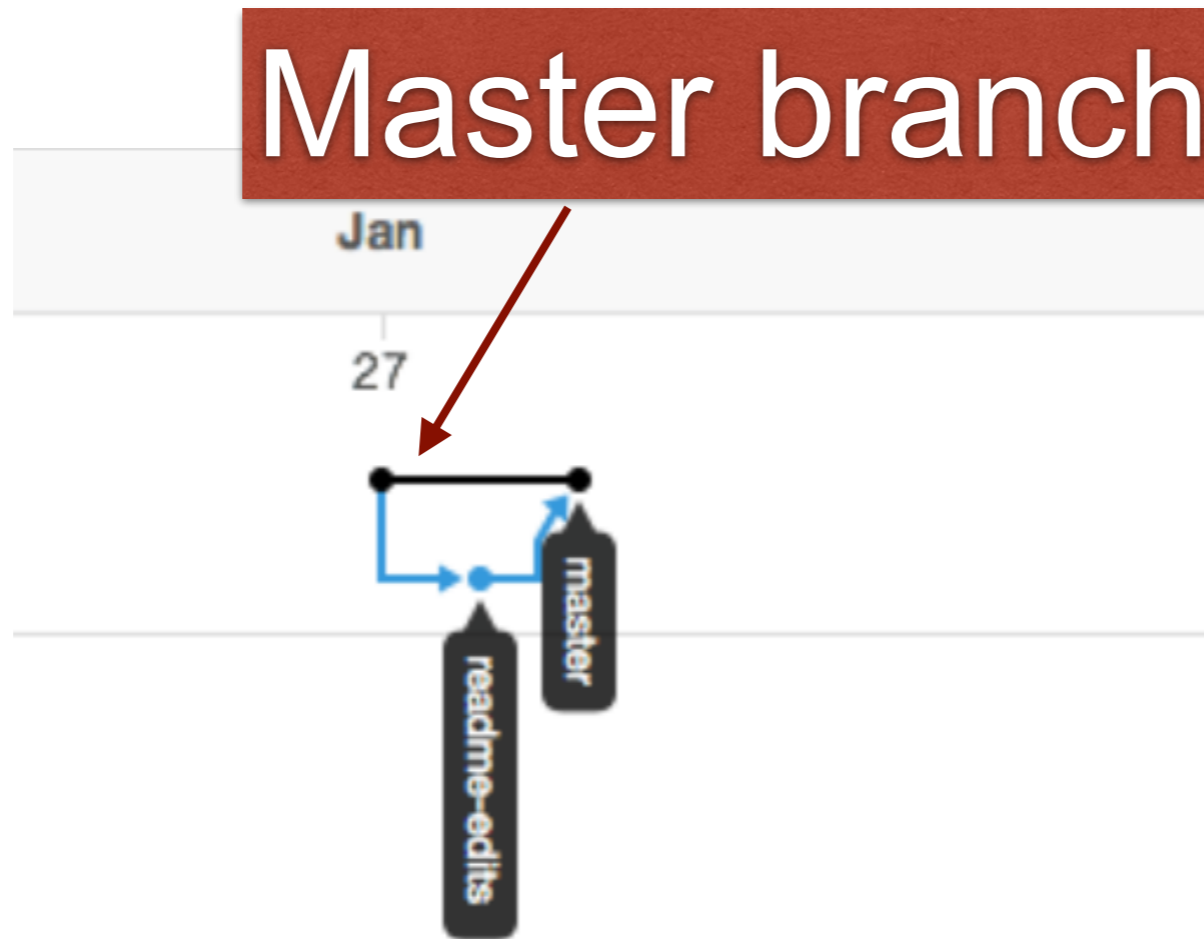
**Contributors** Traffic Commits Code frequency Punch card **Network** Members Dependents

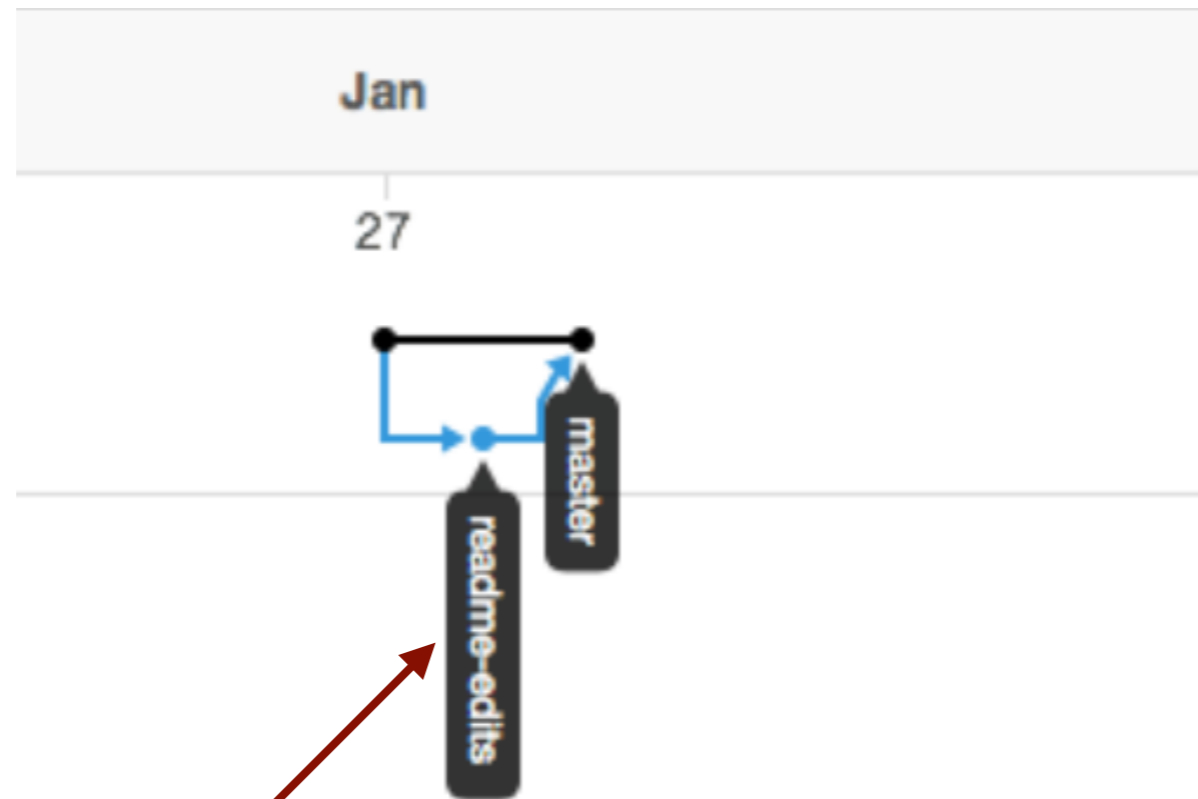
Jan 15, 2017 – Jan 27, 2017

Contributions: **Commits**

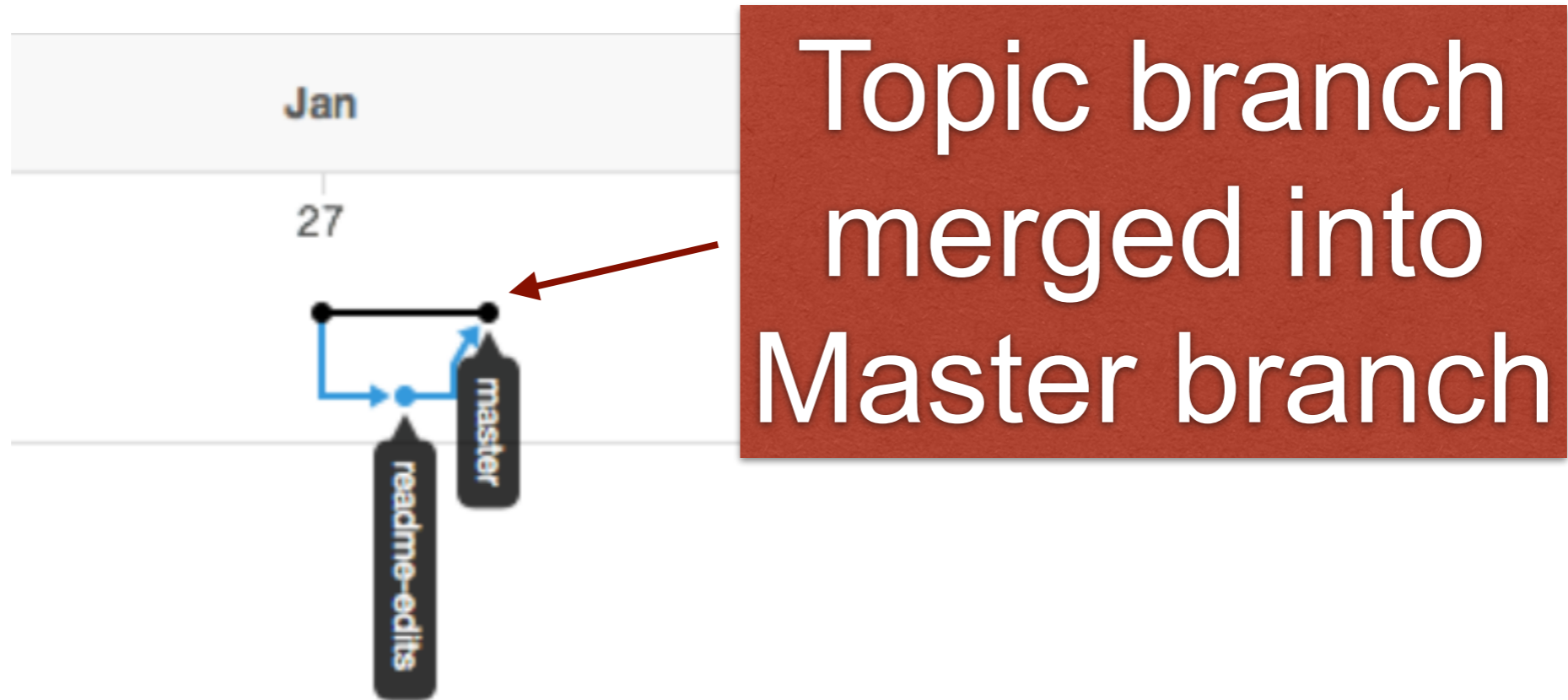
Contributions to master, excluding merge commits

# Master branch





Topic branch



# git

- Now play on the command line!

The screenshot shows a GitHub pull request interface. At the top, the repository name 'rappoccio / hello-world' is displayed, with 'hello-world' highlighted by a red dashed box. A large red box with the word 'Click' is overlaid on the 'Pull requests' tab, which is also highlighted with an orange underline. The navigation bar includes 'Code', 'Issues 0', 'Pull requests 1', 'Projects 0', 'Wiki', 'Pulse', 'Graphs', and 'Settings'. The pull request title is 'Finish README, amirite? #1'. A 'Merged' badge is present, along with the text 'rappoccio merged 1 commit into master from readme-edits 2 minutes ago'. Below this, there are statistics for 'Conversation 0', 'Commits 1', and 'Files changed 1'. A comment from 'rappoccio' is visible, stating 'commented 3 minutes ago'. On the right side, a notification box says 'You can now request reviews from specific people you work with. Learn more.' with a 'Got it!' button. Below the notification, the 'Reviewers' section is partially visible.



- 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 .
|          . +0=0.
+-----+

```

# 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](#)

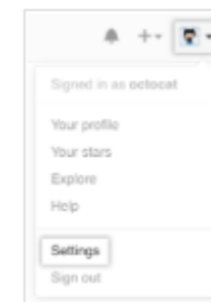
1 Copy the SSH key to your clipboard.

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” :

The screenshot shows the GitHub interface for the repository 'rappoccio / hello-world'. The user is signed in as 'rappoccio'. The user menu is open, showing options like 'Your profile', 'Your stars', 'Explore', 'Integrations', 'Help', 'Settings', and 'Sign out'. The 'Settings' option is highlighted with a red dashed box. The repository page shows 'Hello world stuff.' with 4 commits, 2 branches, 0 releases, and 1 contributor. The current branch is 'master'. The commit history shows two commits: 'Initial commit' (LICENSE) and 'updating my email' (README.md).

This repository Search

Pull requests Issues Gist

rappoccio / hello-world

Unwatch 1

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

Hello world stuff.

4 commits 2 branches 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find files

rappoccio committed with Salvatore Rappoccio updating my email Latest commit

| File      | Commit            | Time           |
|-----------|-------------------|----------------|
| LICENSE   | Initial commit    | an hour ago    |
| README.md | updating my email | 12 minutes ago |

README.md

- Then your “SSH and GPG keys”

Personal settings

**Profile**

Account

Emails

Notifications

Billing

**SSH and GPG keys**

Security

Blocked users

## Public profile

**Name**

**Public email**

Don't show my email address ▾

You can manage verified email addresses in your [email settings](#).

**Bio**

Tell a little about yourself

**Profile picture**



**Upload new picture**

- Then “add new ssh key”

|                         |
|-------------------------|
| Personal settings       |
| Profile                 |
| Account                 |
| Emails                  |
| Notifications           |
| Billing                 |
| <b>SSH and GPG keys</b> |
| Security                |
| Blocked users           |
| Repositories            |
| Organizations           |
| Saved replies           |
| Authorized applications |
| Installed integrations  |
| Developer settings      |
| Auth applications       |

## SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

(bla bla bla here, then you get:)

Title

Add a title here

Key

Begins with 'ssh-rsa', 'ssh-dss', 'ssh-ed25519', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', or 'ecdsa-sha2-nistp521'

Paste your key that you copied here

Add 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?

README.md

Clone with SSH

Use HTTPS

Use an SSH key and passphrase from account.

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



Open in Desktop

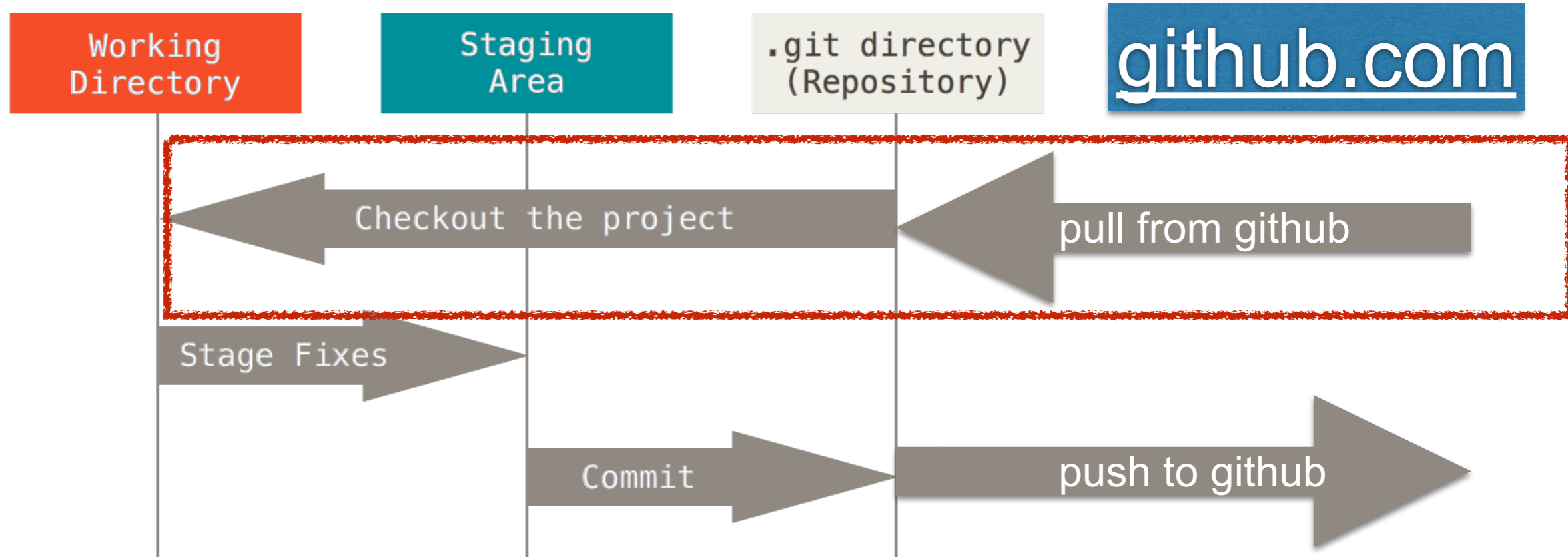
Download ZIP

## hello-world

Hello world stuff.

and this is a strange question, amirite?

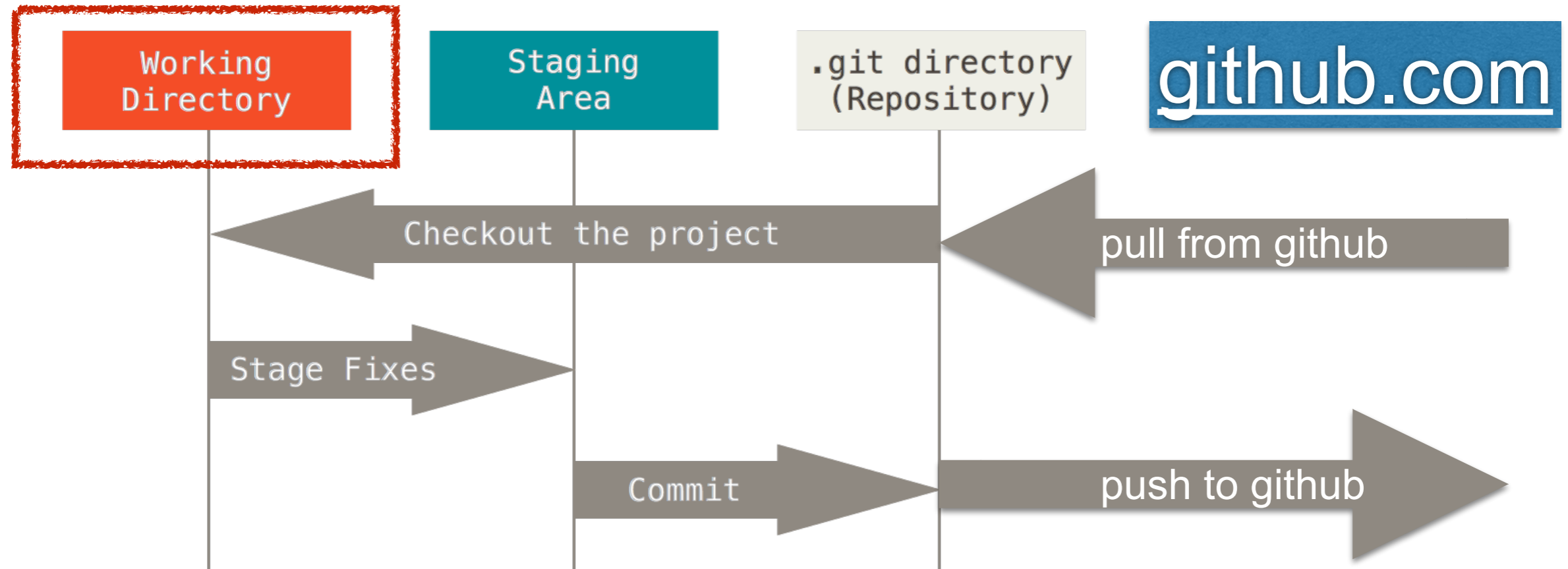
# git



- `git clone` : pulls a repository from [github.com](https://github.com), updates working directory in one step

```
$ mkdir working
$ cd working
/nsm/home/srappoc/working
$ git clone https://github.com/rappoccio/hello-world.git
Initialized empty Git repository in /nsm/home/srappoc/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:

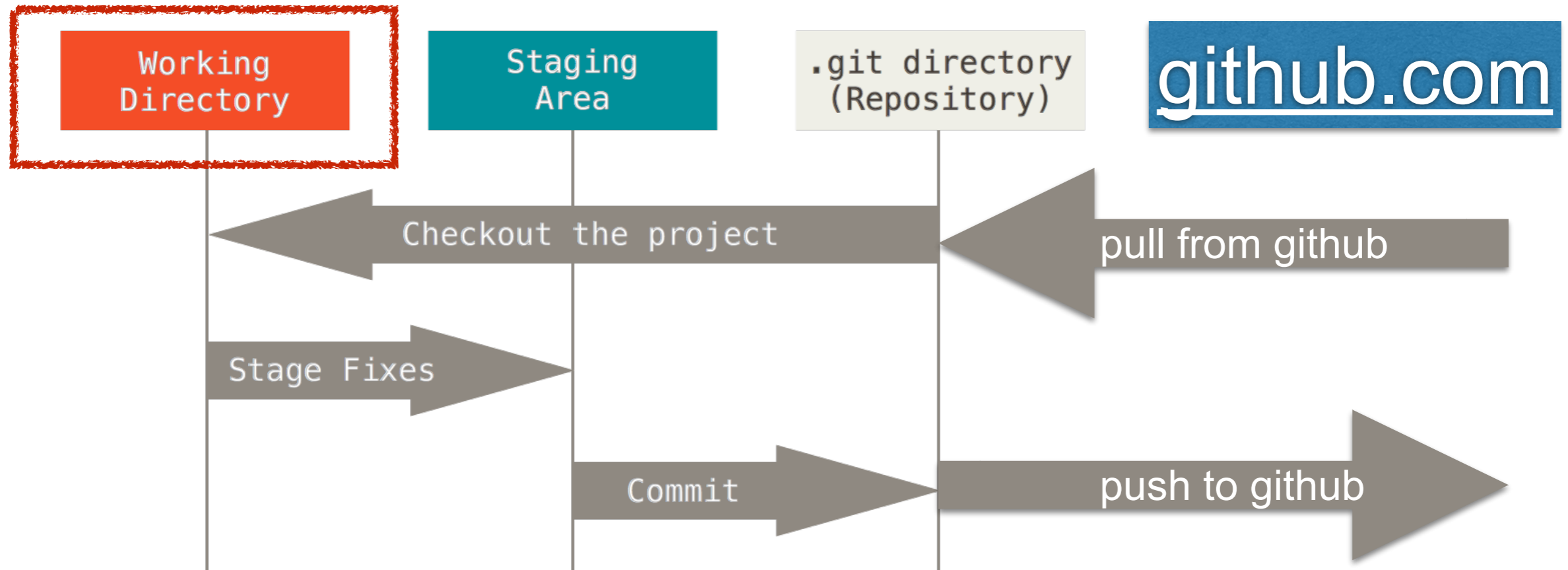
```
# hello-world
Hello world stuff.

and this is a strange question, amirite?

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

- This updates the **WORKING** directory

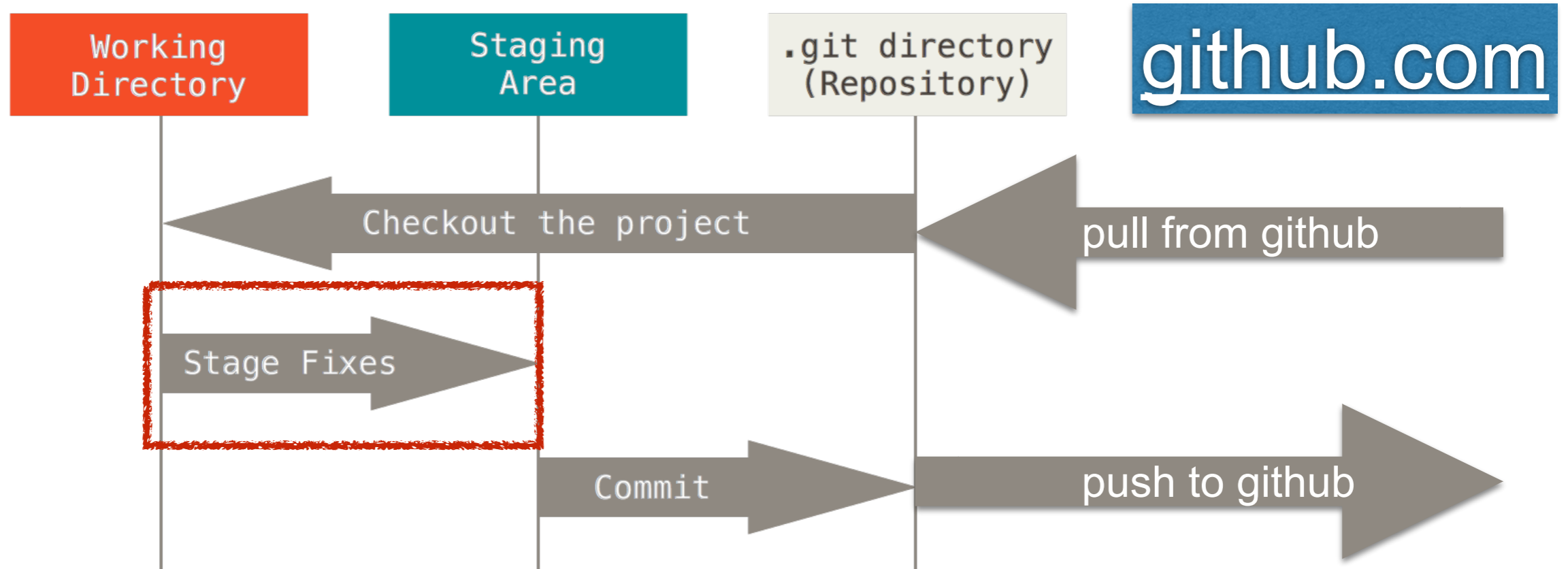
# git



- `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
#
```

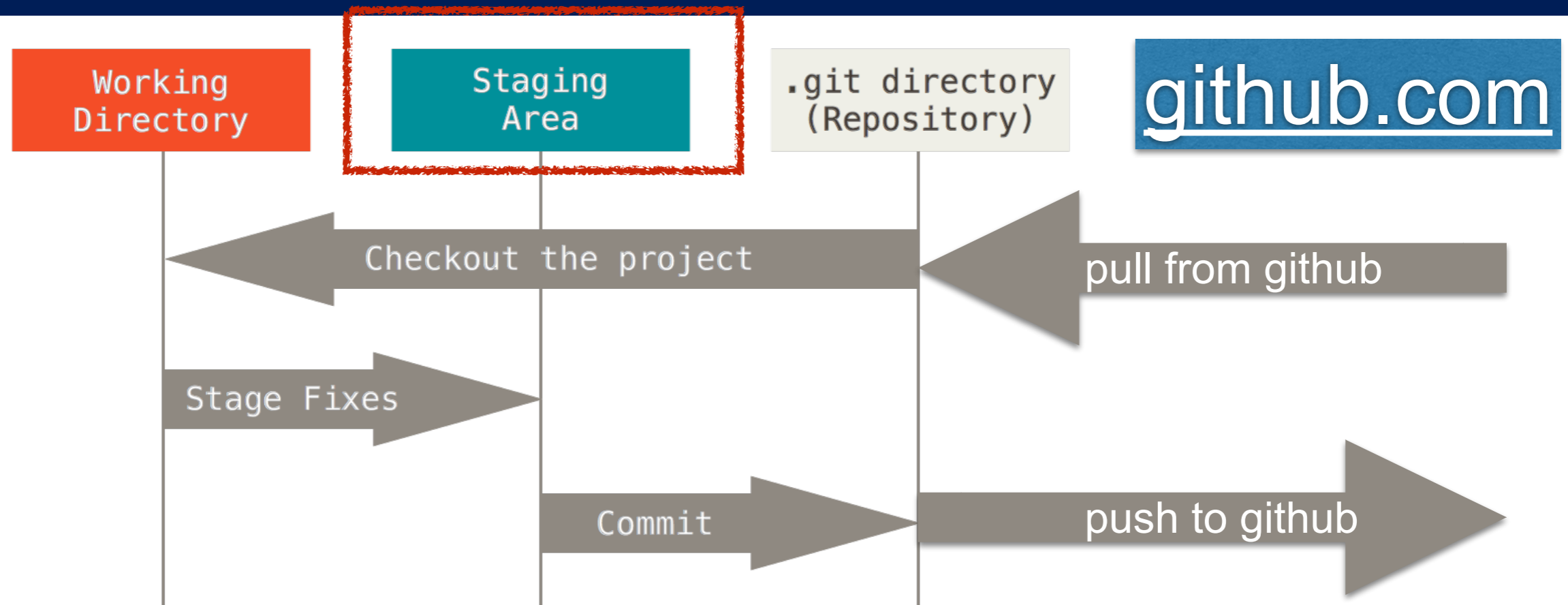
# git



- `git add` : add files to be staged:

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

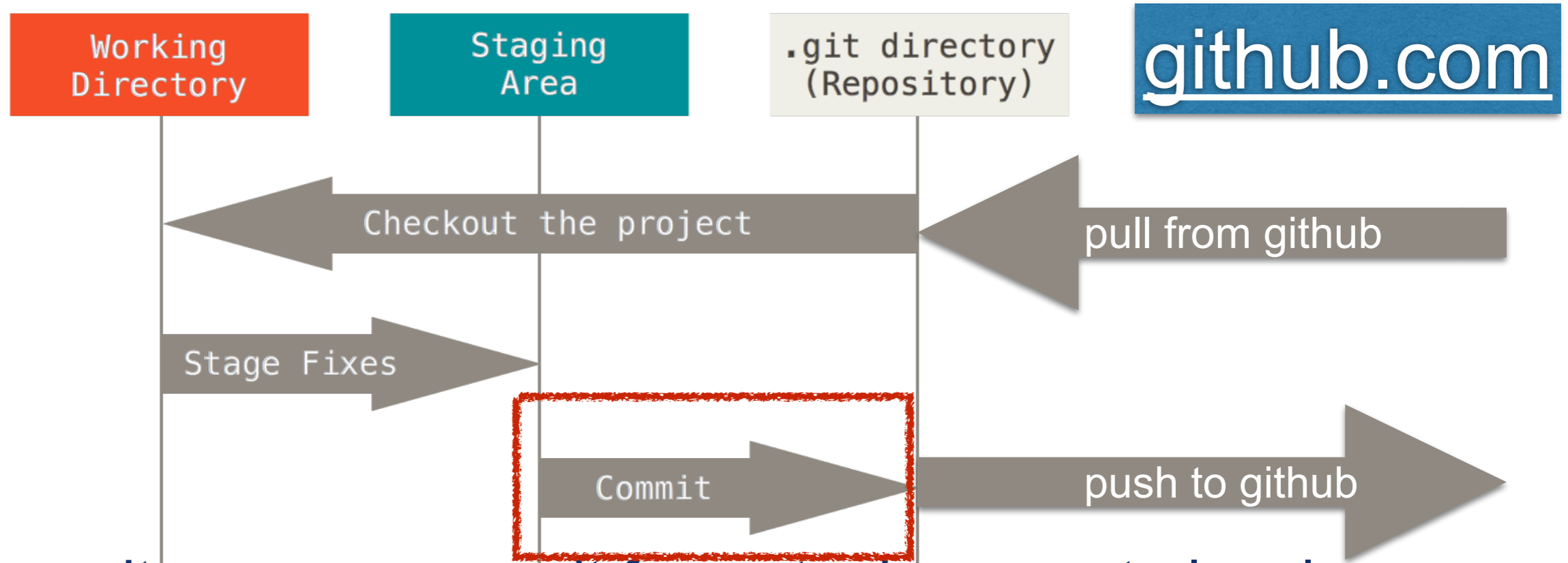
# git



- 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

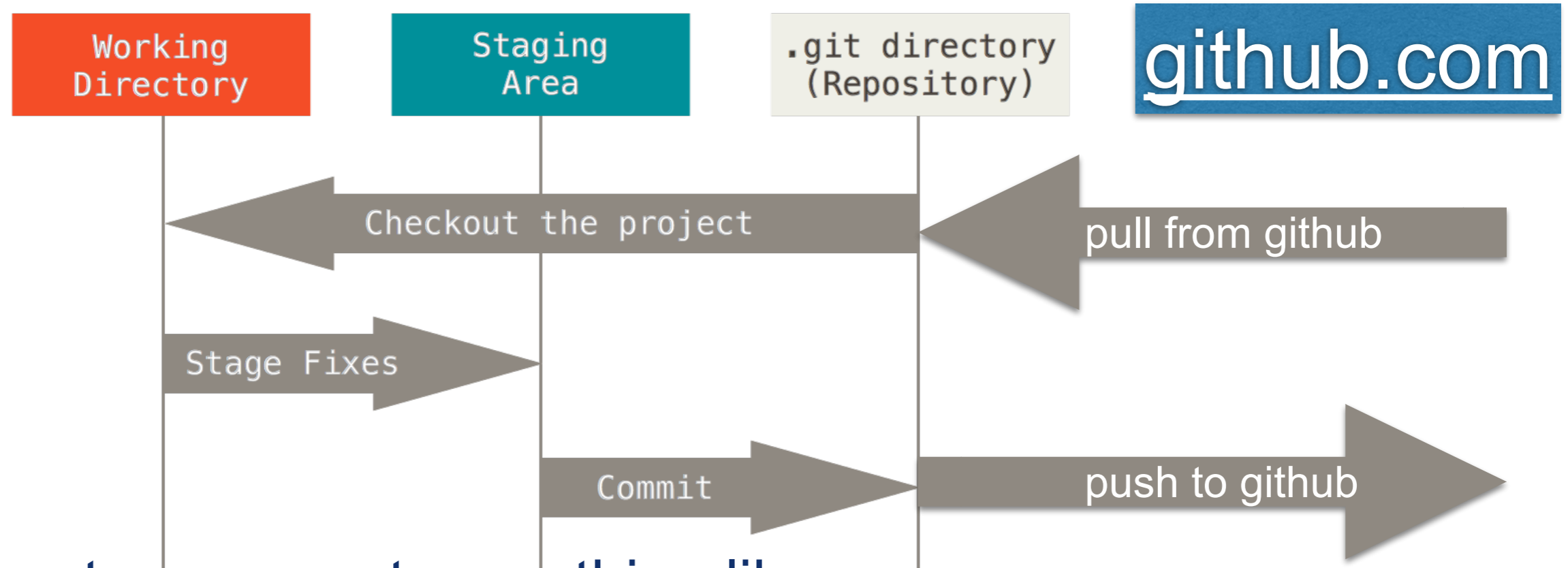


- **git commit** : moves commit from staging area to local repository

```
$ 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:**

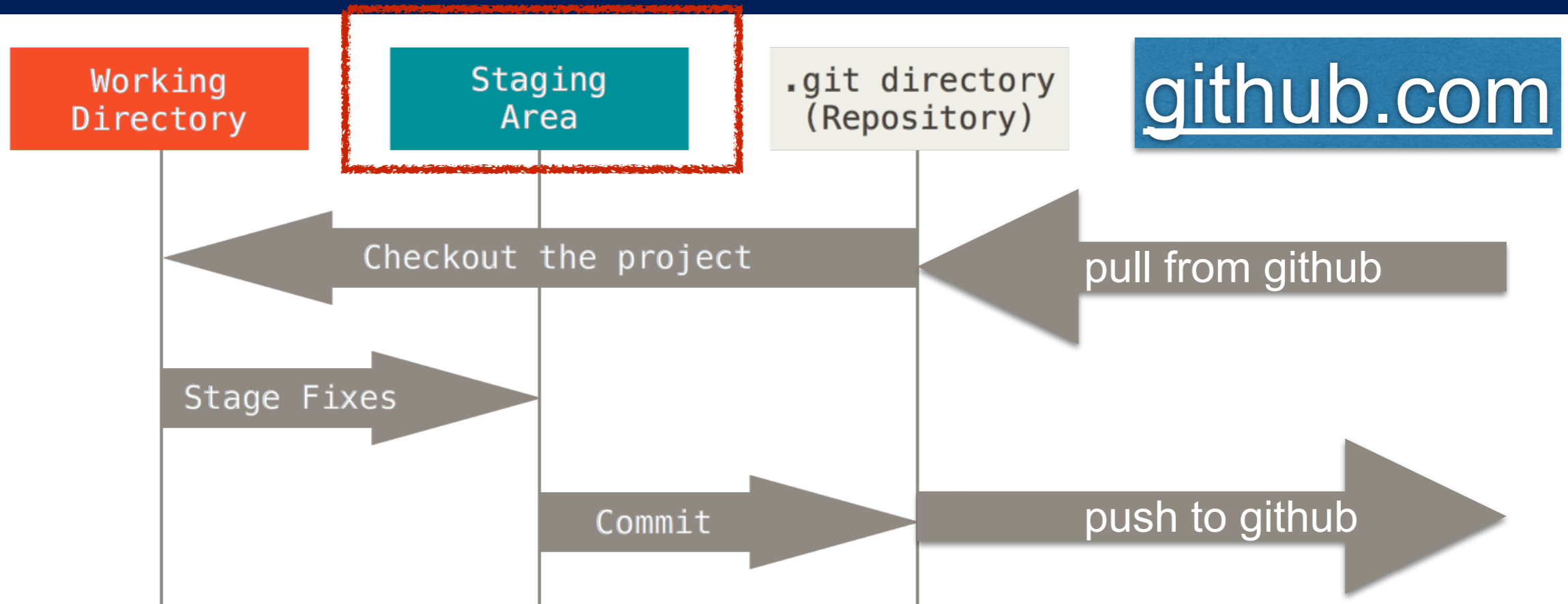
our name and email address were configured automatically based on your username and hostname. Please check that they are accurate. You can suppress this message by setting them explicitly:

```
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>'
```

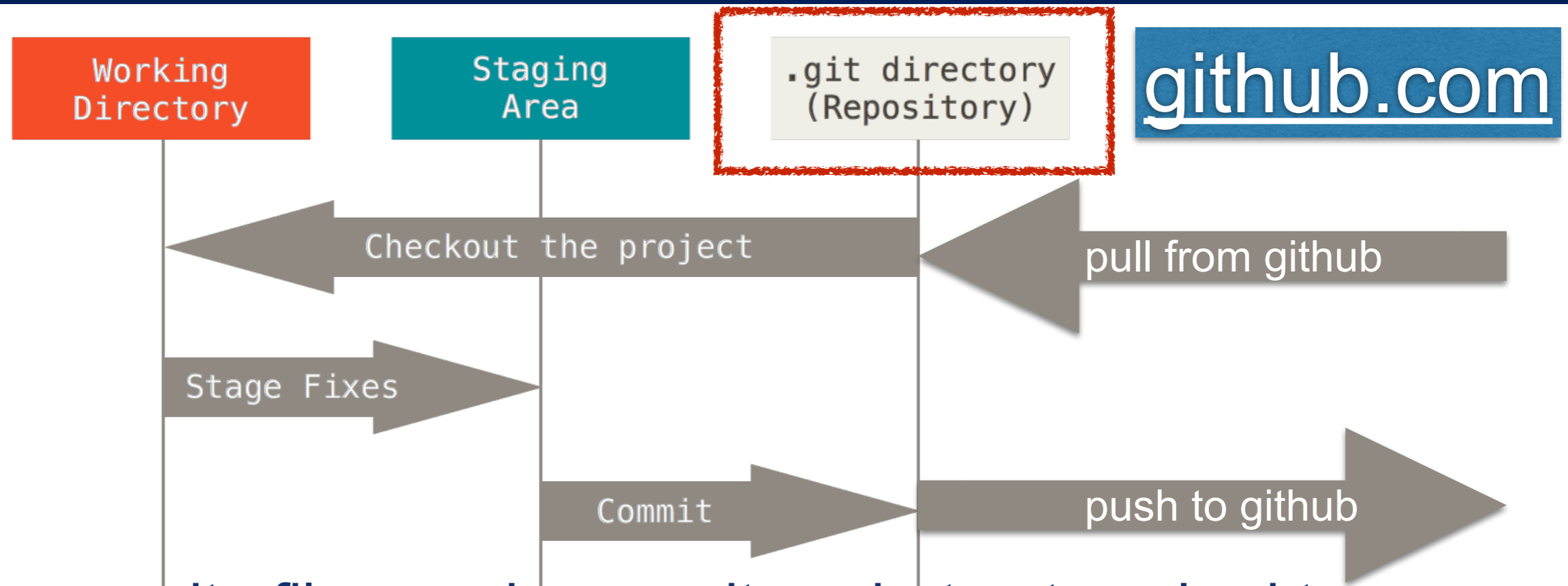
# git



- 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

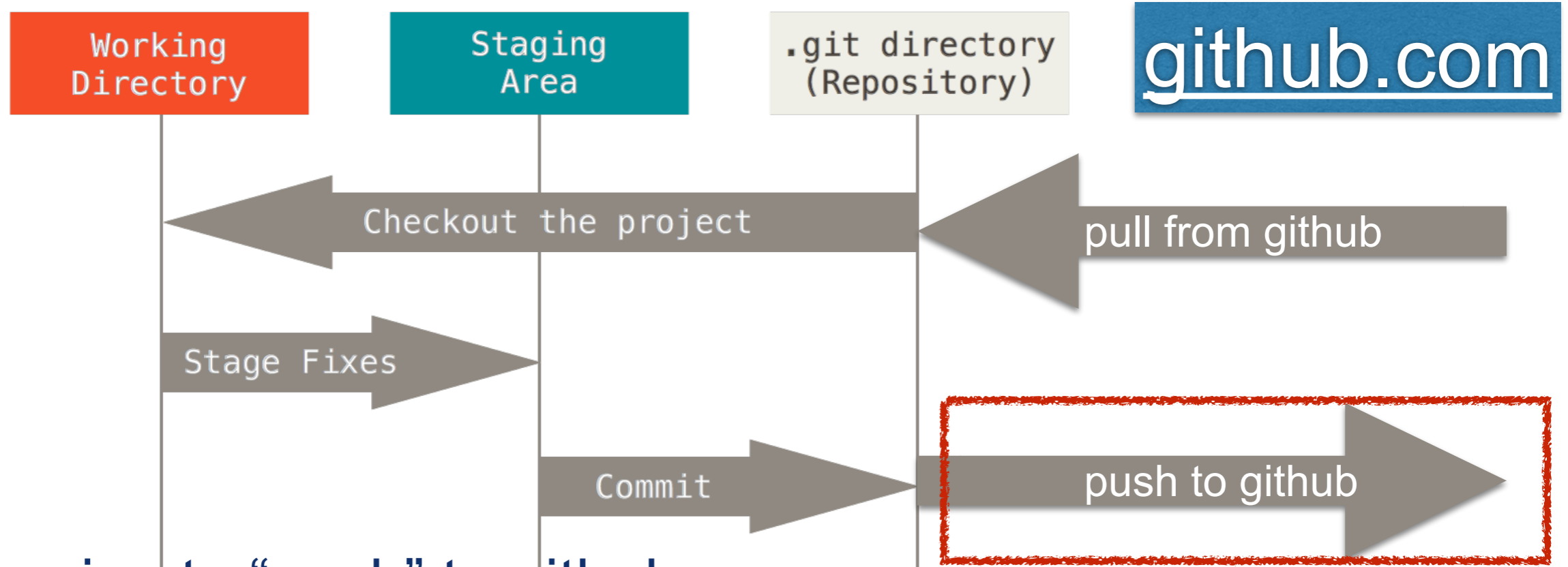


- After commit : files are in repository, but not pushed to “origin”

```
$ 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”

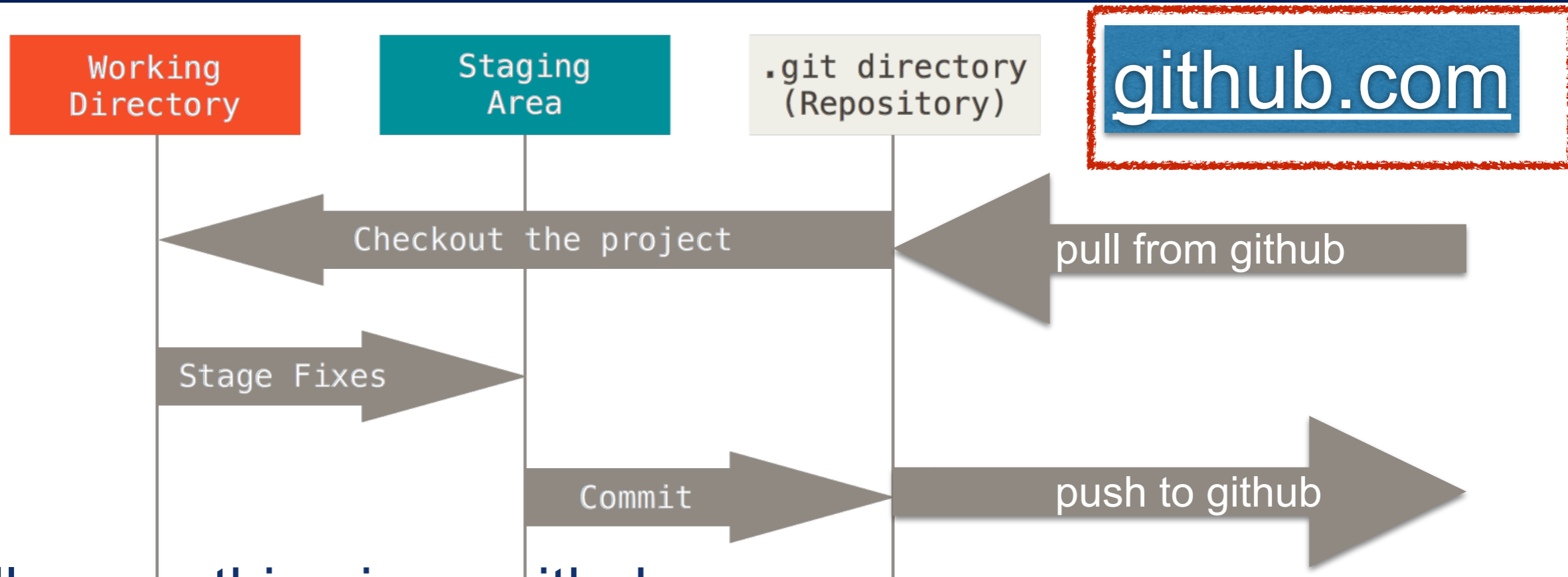
# git



- Now going to “push” to [github.com](https://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)
```

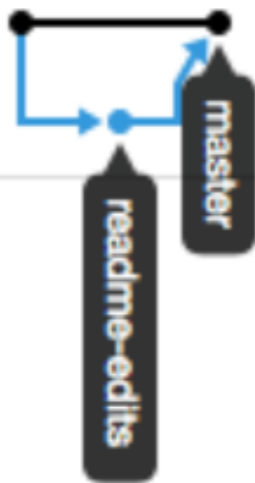
# git

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

## Before

Jan

27



## Now

Jan

27

New commit!



# 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**

The screenshot shows the GitHub interface for the repository 'ubsuny/helloworld'. At the top, there is a search bar and navigation links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below this, the repository name 'ubsuny/helloworld' is displayed as a 'Public template'. To the right of the repository name are buttons for 'Edit Pins', 'Unwatch' (with a count of 1), 'Fork' (with a count of 0), and 'Star'. A red arrow points from the 'FORK IT' text above to the 'Fork' button. Below the repository name, there are navigation links for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. The main content area shows the 'master' branch selected, with '2 branches' and '0 tags' indicated. There are buttons for 'Go to file', 'Add file', 'Code', and 'Use this template'. A notification banner states 'Your master branch isn't protected' with a 'Protect this branch' button. Below this, a pull request by 'rappoccio' is shown, merging pull request #1 from 'rappoccio/rappoccio-patch-1' on Feb 2, 2018, with 3 commits. The pull request includes a file named 'README.md' with the description 'Showing students how to commit' and a commit date of '5 years ago'. The main content area displays the 'README.md' file content, which includes the title 'helloworld', a description 'Description goes here.', and a section header 'Here is a section'. On the right side, there are sections for 'About', 'Releases', and 'Packages'. The 'About' section includes a description 'Description goes here.', a 'Readme' link, '0 stars', '1 watching', and '0 forks'. The 'Releases' section states 'No releases published' with a link to 'Create a new release'. The 'Packages' section states 'No packages published' with a link to 'Publish your first package'.



# git

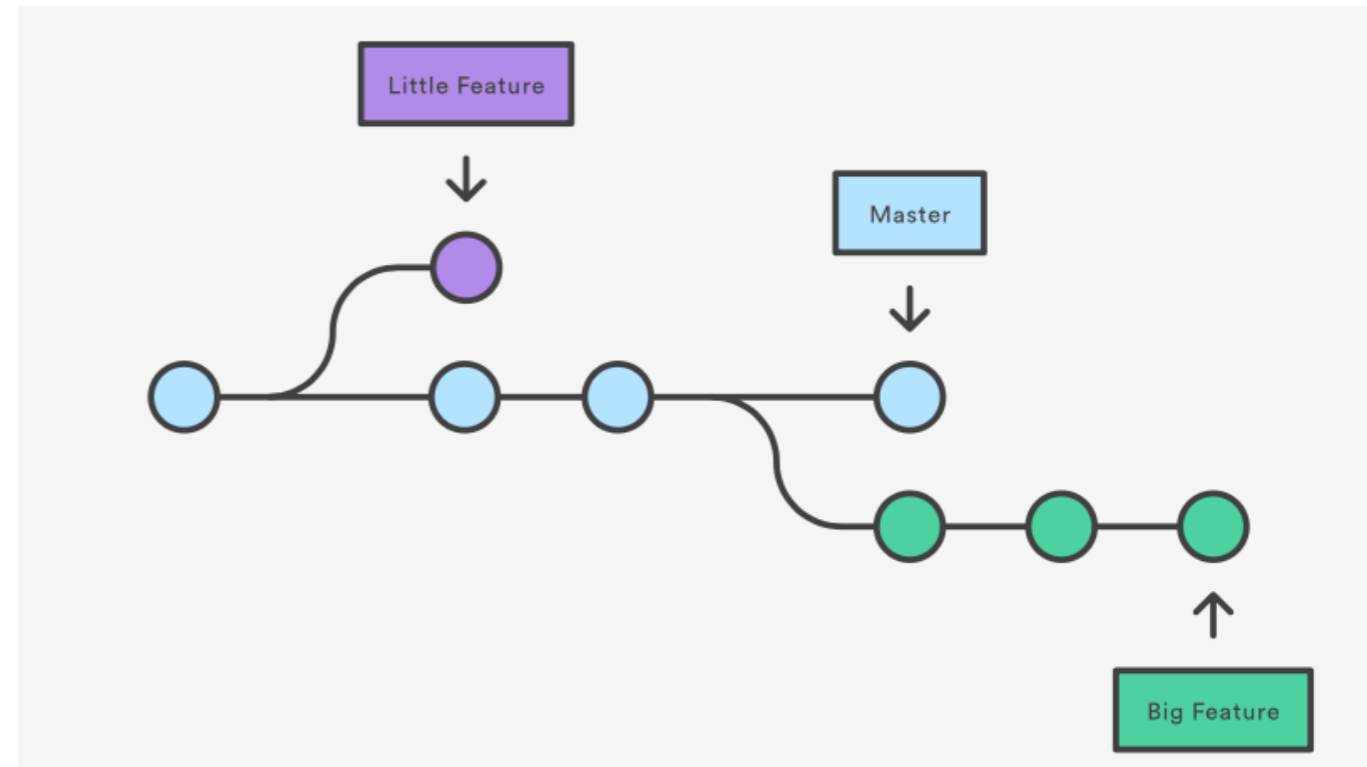
- Now you are able to clone your OWN repository:

```
$ 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](https://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

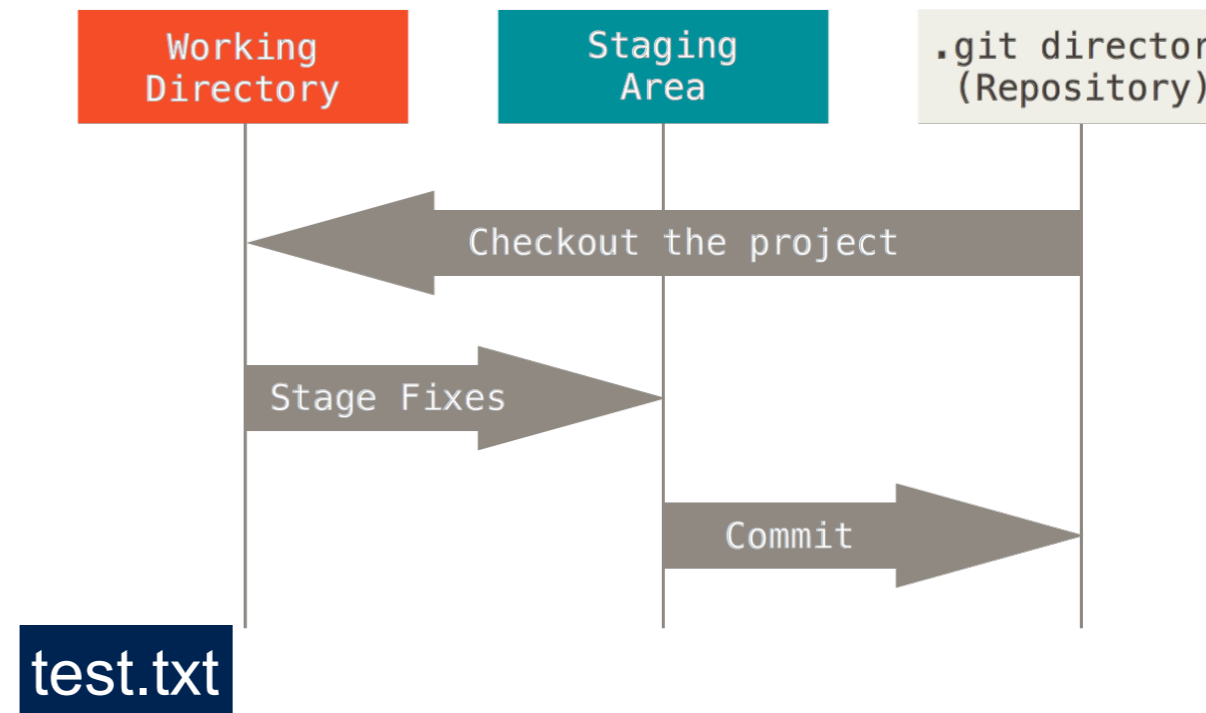
- So now let's get your HW1 set up. First make a directory, and then add a file to it. Finally do "git status", and this is what you should see:

```
$ 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"

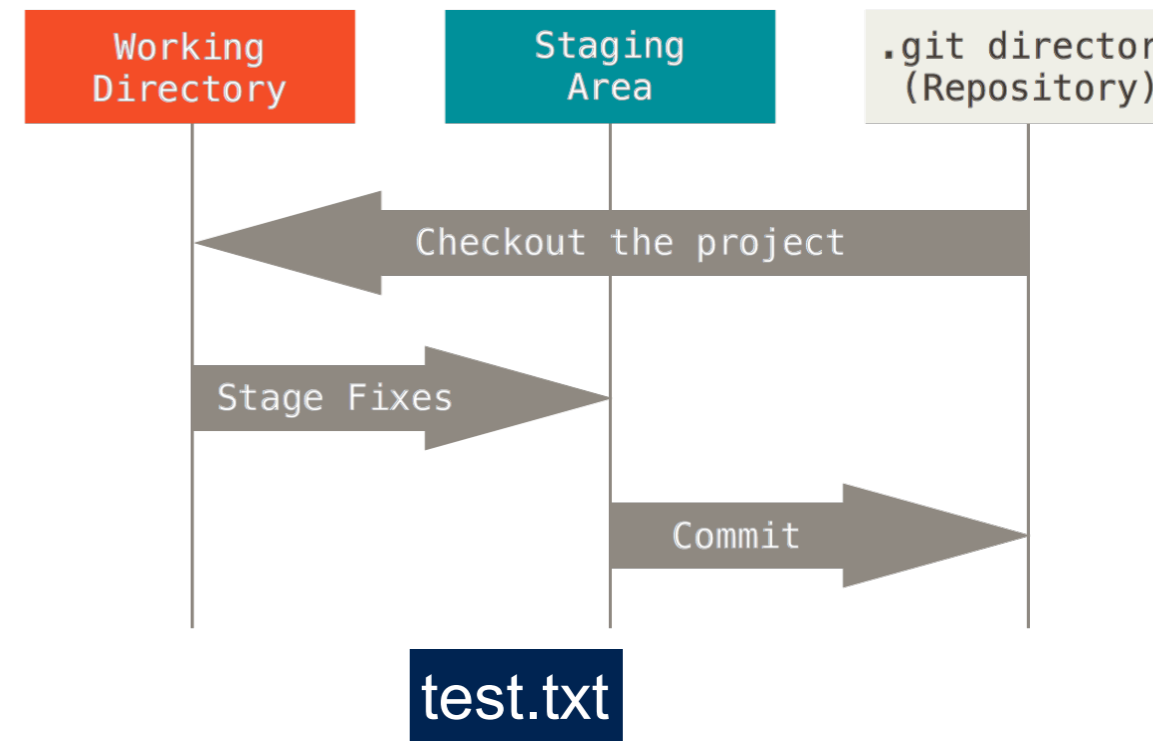
# git

- 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
```

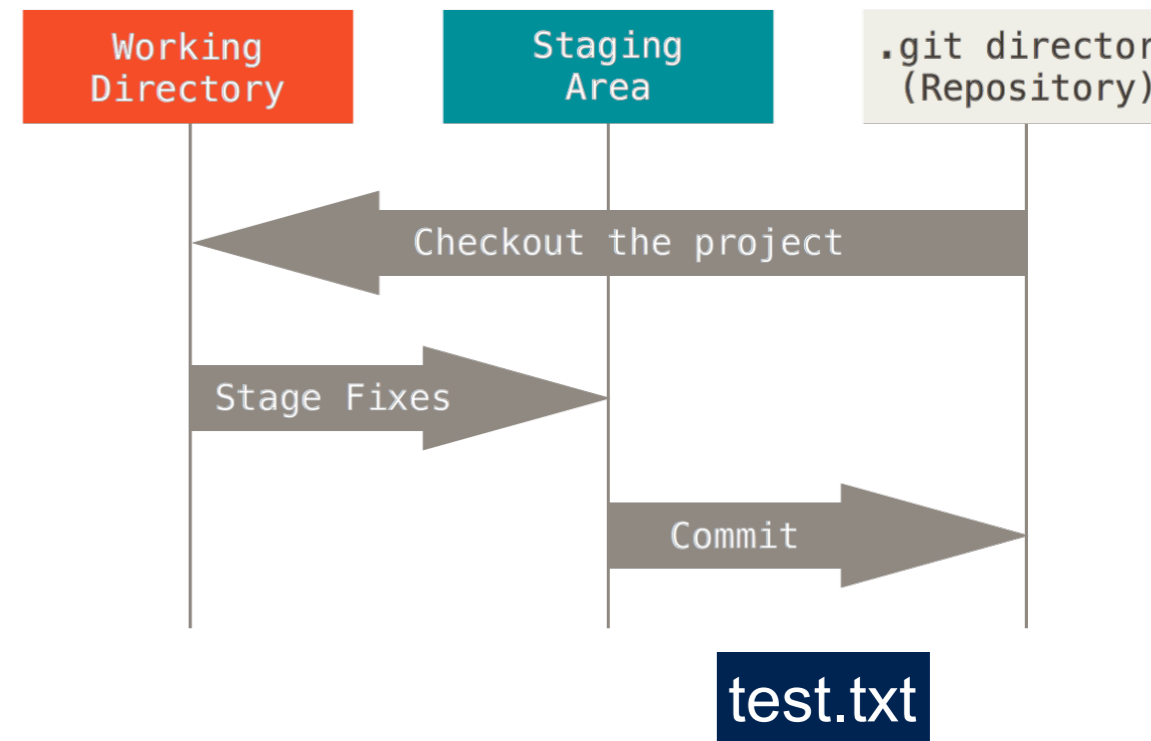
- And finally commit to your .git directory (local repository)

```
$ 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)