

Git Committed

BY ROB HERNANDEZ
SR. SYSTEMS ENGINEER @NEBULAWORKS



What is git?

What is git?

```
~$ man git
Git is a fast, scalable, distributed revision control system with an
unusually rich command set that provides both high-level operations and
full access to internals.
...
```

What is git?

```
~$ man git
Git is a fast, scalable, distributed revision control system with an
unusually rich command set that provides both high-level operations and
full access to internals.
...
```

Its a way to manage a set of files over time

What can git manage?

What can git manage?

Text files

What can git manage?

Text files

Binary files

What can git manage?

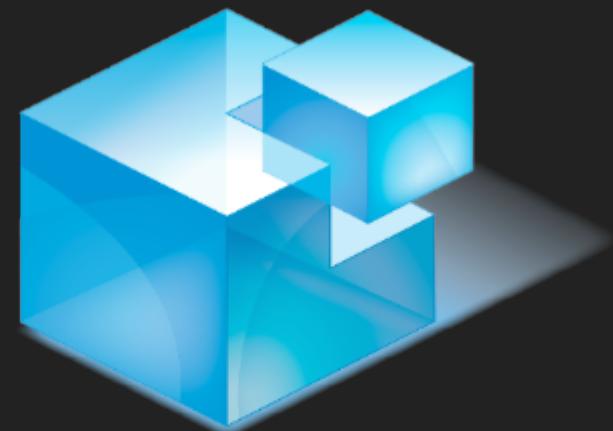
Text files

Binary files

Git can be used to manage any kind of file



My first PR



SALTSTACK

My fix

```
diff --git a/salt/modules/freebsdpkg.py b/salt/modules/freebsdpkg.py
index b3bfe1a3d9..63e0a4cb87 100644
--- a/salt/modules/freebsdpkg.py
+++ b/salt/modules/freebsdpkg.py
@@ -85,7 +85,7 @@ def refresh_db():
    __salt__['cmd.run']('portsnap update')

-def install(name, **kwargs):
+def install(name, *args, **kwargs):
    """
    Install the passed package

```

My fix

```
diff --git a/salt/modules/freebsdpkg.py b/salt/modules/freebsdpkg.py
index b3bfe1a3d9..63e0a4cb87 100644
--- a/salt/modules/freebsdpkg.py
+++ b/salt/modules/freebsdpkg.py
@@ -85,7 +85,7 @@ def refresh_db():
    __salt__['cmd.run']('portsnap update')

-def install(name, **kwargs):
+def install(name, *args, **kwargs):
    """
    Install the passed package

```

Straight forward enough

Resulted in



To get this merged required

To get this merged required

- fork the git project

To get this merged required

- fork the git project
- git clone down the fork

To get this merged required

- fork the git project
- git clone down the fork
- create a branch for my changes

To get this merged required

- fork the git project
- git clone down the fork
- create a branch for my changes
 - commit my changes

To get this merged required

- fork the git project
- git clone down the fork
- create a branch for my changes
 - commit my changes
 - push up my branch

To get this merged required

- fork the git project
- git clone down the fork
- create a branch for my changes
 - commit my changes
 - push up my branch
 - submit a pull request

This was a lot for just a oneline change

Where to start?

Whose making the changes?

Whose making the changes?

```
commit 28b1edb4a5a823c910a5d86ea3514ab982bcf3d1
Author: Rob Hernandez <rob@sarcasticadmin.com>
Date:   Fri Feb 1 12:16:59 2019 -0800
```

Updates to vim dictionary

Configure .gitconfig

Configure .gitconfig

Add your name and email

```
git config --global user.name "Rob Hernandez"  
git config --global user.email "rob@sarcasticadmin.com"
```

Configure .gitconfig

Add your name and email

```
git config --global user.name "Rob Hernandez"  
git config --global user.email "rob@sarcasticadmin.com"
```

config created in user's ~/.gitconfig

Configure .gitconfig

Add your name and email

```
git config --global user.name "Rob Hernandez"  
git config --global user.email "rob@sarcasticadmin.com"
```

config created in user's ~/.gitconfig

```
~$ cat ~/.gitconfig  
[user]  
name = Rob Hernandez  
email = rob@sarcasticadmin.com
```

Without a .gitconfig
Unable to identify the author

Without a .gitconfig

Unable to identify the author

```
commit e0c03273720e088b157f53611ba4483b18d4cba6 (HEAD -> master)
Author: Minion <minion@thinkpad230x.local>
Date:   Sun Feb 24 12:15:31 2019 -0800
```

Adding a new file

Without a .gitconfig

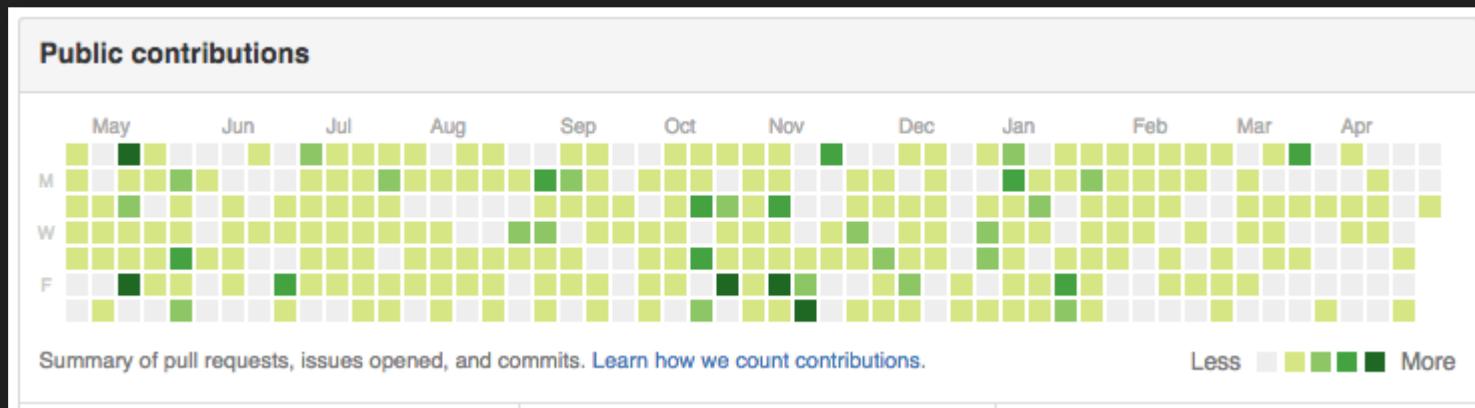
Unable to identify the author

```
commit e0c03273720e088b157f53611ba4483b18d4cba6 (HEAD -> master)
Author: Minion <minion@thinkpad230x.local>
Date:   Sun Feb 24 12:15:31 2019 -0800
```

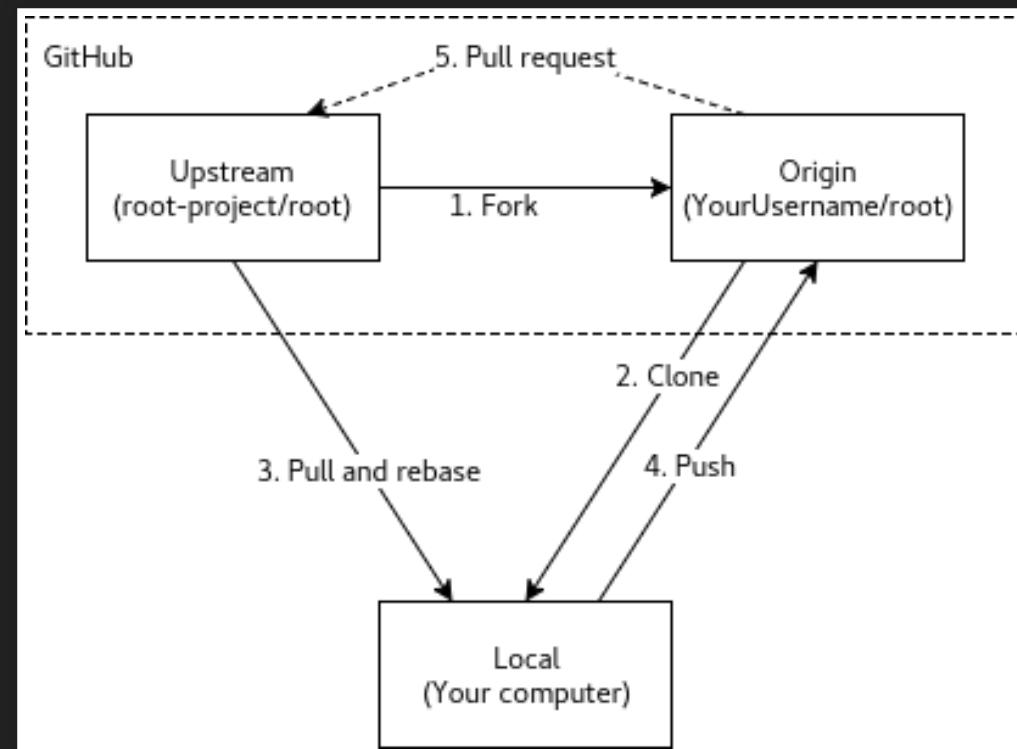
Adding a new file

uses the info about the current user and system

Bonus! Remotes will track contributions



A simple git workflow



Find a repo

 [firstcontributions / first-contributions](#)

 Watch ▾ 99  Star 5,993  Fork 15,581

 Code  Issues 7  Pull requests 748  Projects 2  Wiki  Insights

 Help beginners to contribute to open source projects <https://firstcontributions.github.io>

[open-source](#) [tutorial](#) [contribution](#) [community](#)

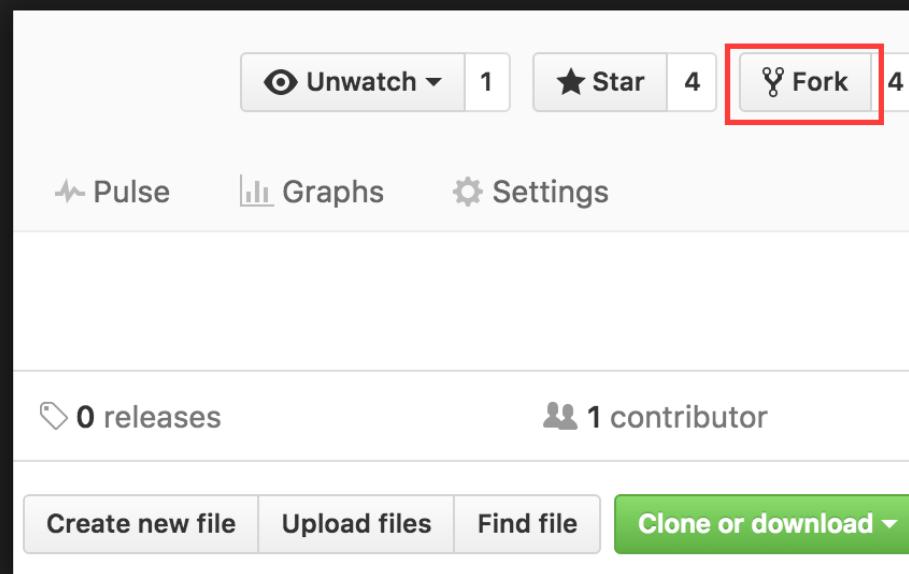
 32,874 commits  2 branches  0 releases  10,722 contributors  MIT

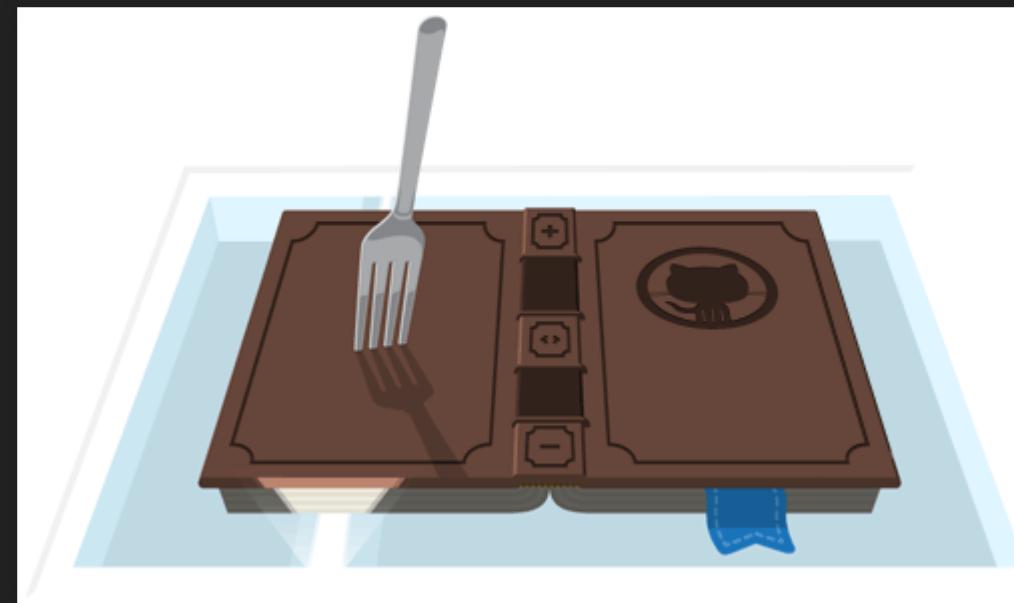
Branch: [master](#) ▾ [New pull request](#) [Create new file](#) [Upload files](#) [Find File](#) [Clone or download](#) ▾

| File | Description | Time Ago |
|--|--|-----------------------------------|
|  Roshanjossey | Merge pull request #15331 from whoismaruf/add-maruf ... | Latest commit b011fac an hour ago |
|  .github | Update contributing guide to be tutorial specific | 2 months ago |
|  additional-material | Renamed file extension to match the lowercase extension used everywhe... | a month ago |
|  assets | Increased contrast of red color for git-status image | 3 months ago |
|  translations | Merge branch 'master' into fix-typos-in-german-translation | a month ago |
|  .gitignore | added sometemp files, eslint, & spec files for Vim & sublime to .giti... | 7 months ago |
|  CODE_OF_CONDUCT.md | spelling mistakes fixed! | 3 months ago |

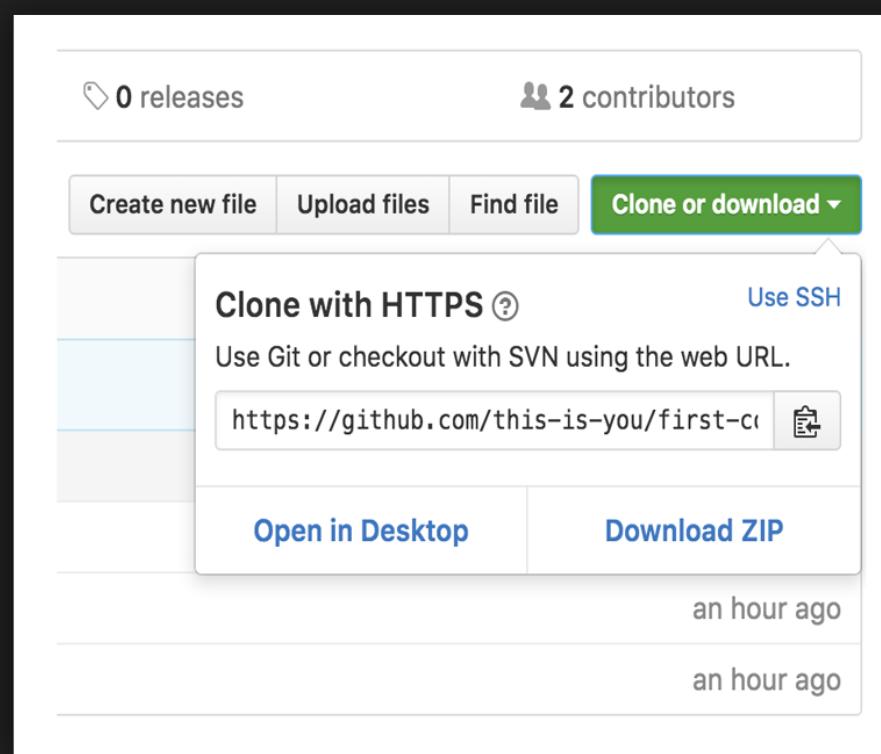
First Contributors on Github

Fork the project repo

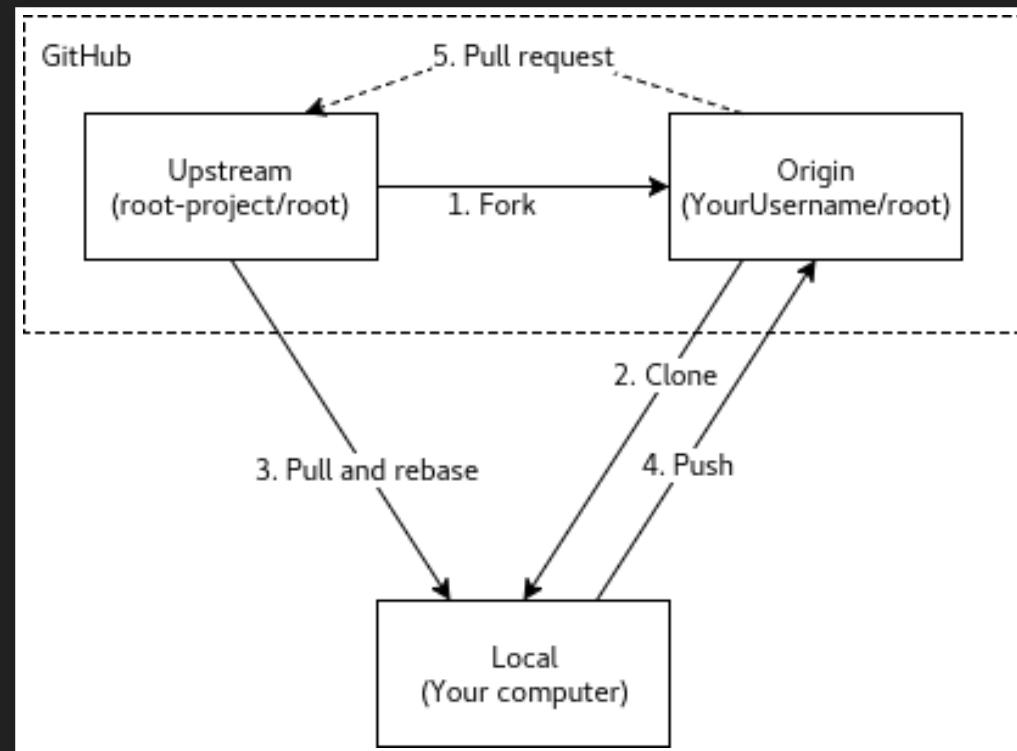




Get the URL or SSH string of fork



git workflow



Clone the repo down

Clone the repo down

```
~$ git clone git@github.com:sarcasticadmin/first-contributions.git
```

Clone the repo down

```
~$ git clone git@github.com:sarcasticadmin/first-contributions.git
```

Navigate into the repo

Clone the repo down

```
~$ git clone git@github.com:sarcasticadmin/first-contributions.git
```

Navigate into the repo

```
~$ cd first-contributions
```

What did git clone do?

What did git clone do?

Retrieved the source code

```
~$ tree -L 1
.
|-- CODE_OF_CONDUCT.md
|-- Contributors.md
|-- LICENSE
|-- README.md
|-- additional-material
|-- assets
|-- github-desktop-old-version-tutorial.md
|-- github-desktop-tutorial.md
|-- github-windows-vs-code-tutorial.md
|-- github-windows-vs2017-tutorial.md
|-- gitkraken-tutorial.md
|-- package-lock.json
|-- translations
```

Additionally git clone

Additionally git clone

Adds a remote

```
~$ git remote -v
origin  git@github.com:sarcasticadmin/first-contributions.git (fetch)
origin  git@github.com:sarcasticadmin/first-contributions.git (push)
```

Additionally git clone

Adds a remote

```
~$ git remote -v  
origin  git@github.com:sarcasticadmin/first-contributions.git (fetch)  
origin  git@github.com:sarcasticadmin/first-contributions.git (push)
```

Sets the default branch

```
~$ git status  
On branch master  
nothing to commit, working tree clean
```

default branch? master?

default branch? master?



the fattest and longest span is the trunk(master), not the branches

Create a branch for your changes

Create a branch for your changes

```
~$ git checkout -b mychanges-1
```

Create a branch for your changes

```
~$ git checkout -b mychanges-1
```

```
~$ git status  
On branch mychanges-1  
nothing to commit, working tree clean
```

Create a branch for your changes

```
~$ git checkout -b mychanges-1
```

```
~$ git status  
On branch mychanges-1  
nothing to commit, working tree clean
```

We're now on our new branch

Lets modify the branch

Lets modify the branch

```
~$ cat << EOF >> Contributors.md
- [Rob Hernandez](https://github.com/sarcasticadmin)
EOF
```

Lets modify the branch

```
~$ cat << EOF >> Contributors.md
- [Rob Hernandez](https://github.com/sarcasticadmin)
EOF
```

```
~$ git status
```

```
On branch mychanges-1
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   Contributors.md

no changes added to commit (use "git add" and/or "git commit -a")
```

Lets modify the branch

```
~$ cat << EOF >> Contributors.md
- [Rob Hernandez](https://github.com/sarcasticadmin)
EOF
```

```
~$ git status
```

```
On branch mychanges-1
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   Contributors.md

no changes added to commit (use "git add" and/or "git commit -a")
```

git detected a change!

Stage the change

Stage the change

```
~$ git add Contributors.md
```

Stage the change

```
~$ git add Contributors.md
```

```
~$ git status
```

```
On branch mychanges-1
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   Contributors.md
```

Stage the change

```
~$ git add Contributors.md
```

```
~$ git status
```

```
On branch mychanges-1
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   Contributors.md
```

git staged the change!

Commit it to the branch

Commit it to the branch

```
~$ git commit -m "Add rob to Contributors list"
```

Commit it to the branch

```
~$ git commit -m "Add rob to Contributors list"
```

```
~$ git status  
On branch mychanges-1  
nothing to commit, working tree clean
```

Commit it to the branch

```
~$ git commit -m "Add rob to Contributors list"
```

```
~$ git status  
On branch mychanges-1  
nothing to commit, working tree clean
```

what happened to the change?

Check the git log

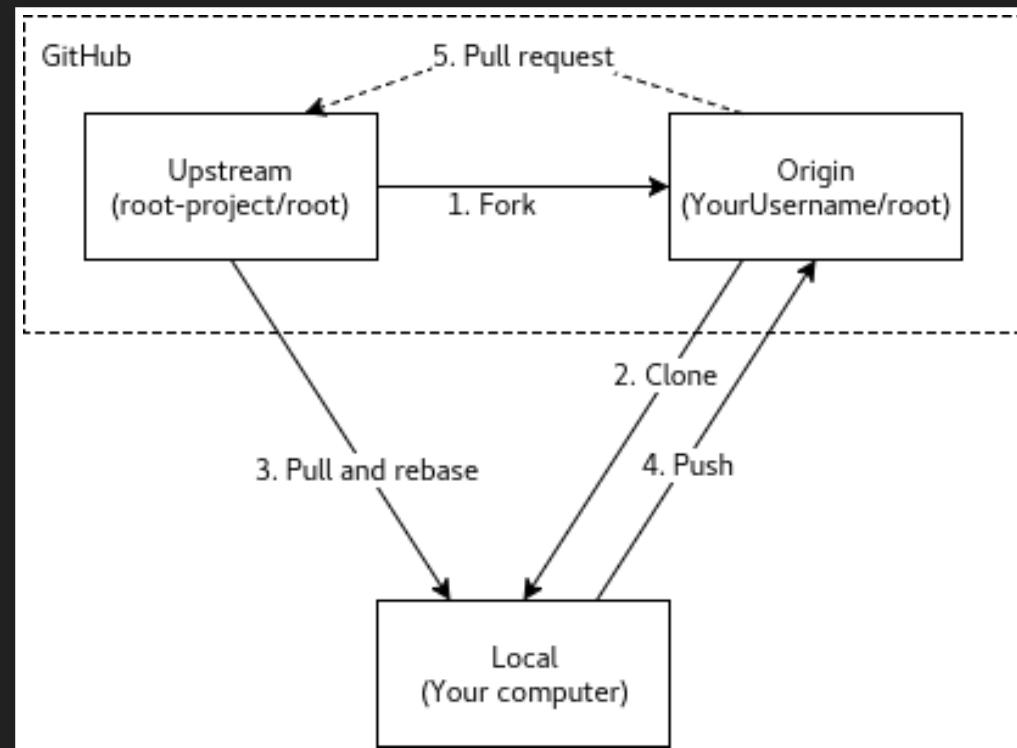
Check the git log

```
~$ git log
```

```
commit 350df3add021582457be51896ef00f7499738b19 (HEAD -> mychanges-1)
Author: Robert James Hernandez <rob@sarcasticadmin.com>
Date:   Wed Feb 20 20:19:19 2019 -0800
```

```
    Add rob to Contributors list
```

git workflow



Push up the commit

Push up the commit

```
~$ git push origin mychanges-1
```

Pre Pull Request

Pre Pull Request
check repo documentation

Pre Pull Request

check repo documentation

check existing issues/pull requests

Submit the PR

Submit the PR

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 repository: **firstcontributions/first-c...** ▾ base: **master** ▾ ←

head repository: **sarcasticadmin/first-con...** ▾ compare: **mychanges-1** ▾

A screenshot of a Notion workspace. At the top left is a small cartoon character icon. Next to it is a header bar with the title "Add rob to Contributors list". Below the header are two tabs: "Write" (which is highlighted with a light blue background) and "Preview". To the right of the tabs is a toolbar with various icons: bold (AA), italic (i), quote (‘’), link (🔗), code (code), list (list), table (table), emoji (emojis), user (user), and a share icon (share). Below the toolbar is a large text input field containing the placeholder text "Leave a comment".



It looks like this is your first time opening a pull request in this project! Be sure to review the [contributing guidelines](#) and [code of conduct](#).

Submit the PR

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 repository: **firstcontributions/first-c...** ↗ base: **master** ↘

head repository: **sarcasticadmin/first-con...** ↗ compare: **mychanges-1** ↘

✓ **Able to merge.** These branches can be automatically merged.

submit against the upstream master

Submit the PR

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 repository: **firstcontributions/first-c...** ↗ base: **master** ←

head repository: **sarcasticadmin/first-con...** ↗ compare: **mychanges-1** ↗

✓ **Able to merge.** These branches can be automatically merged.

submit against the upstream master

include a detailed description



BRILLIANT! WELL DONE!

Merged upstream

Merged upstream

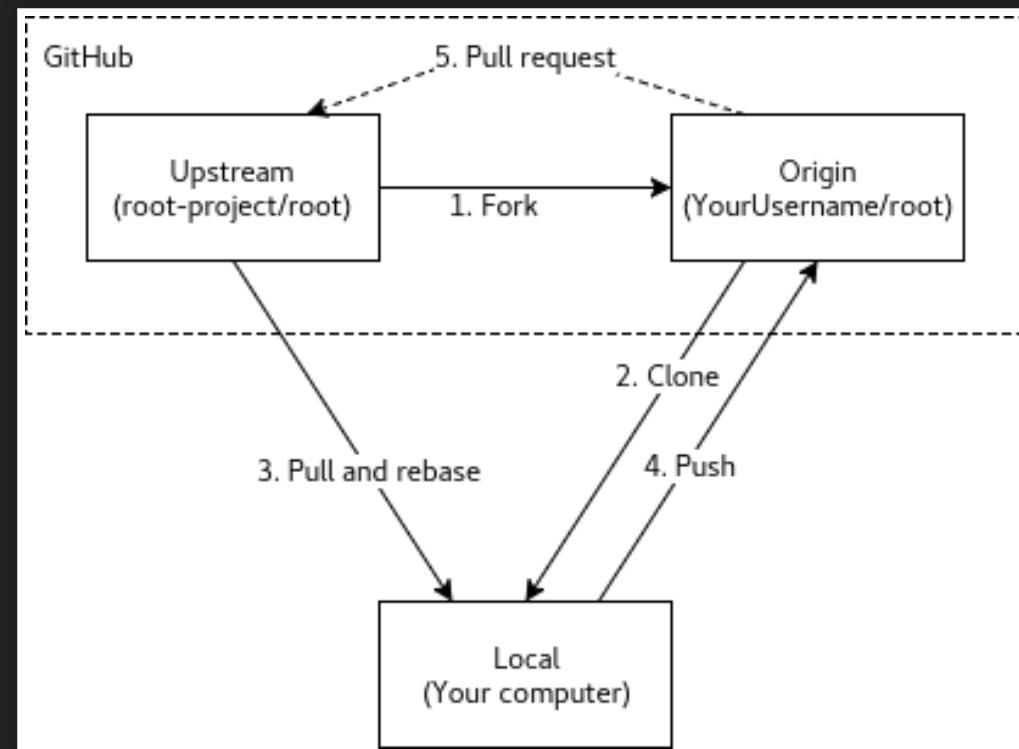


Other considerations

1. Pull updates from upstream repo

2. Merge conflict

Pull updates from upstream



Configure upstream

Configure upstream

```
~$ git add remote upstream https://github.com/firstcontributions/first-contribut
```

Configure upstream

```
~$ git add remote upstream https://github.com/firstcontributions/first-contribut
```

```
~$ git remote -v
origin  git@github.com:sarcasticadmin/first-contributions.git (fetch)
origin  git@github.com:sarcasticadmin/first-contributions.git (push)
upstream      https://github.com/firstcontributions/first-contributions.git (f
upstream      https://github.com/firstcontributions/first-contributions.git (p
```

Update master upstream

Update master upstream

```
~$ git checkout master  
~$ git pull --rebase upstream master
```

Update master upstream

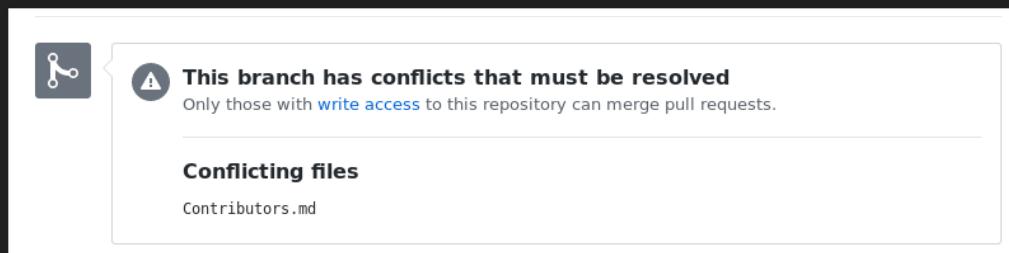
```
~$ git checkout master  
~$ git pull --rebase upstream master
```

Always keep local master identical to upstream master

1. Pull updates from upstream repo

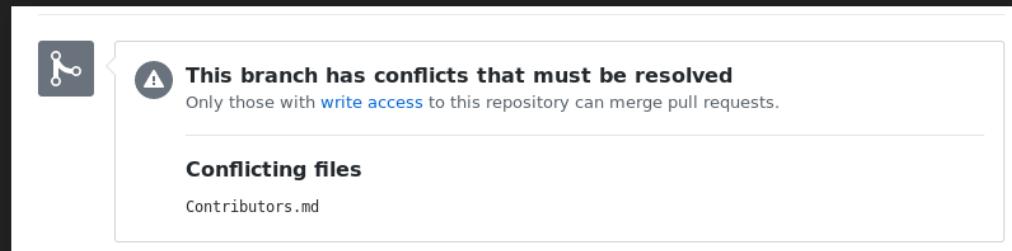
Other considerations 2. Merge conflict

Merge conflict



Merge conflict

Now what?



Other considerations

1. Pull updates from upstream repo

2. Merge conflict



Example repo
First Contributors on Github

Thank you

ROB HERNANDEZ @SARCASTICADMIN

GITHUB: SARCASTICADMIN