Git Introduction II - Remote Repositories

Luis Ibanez

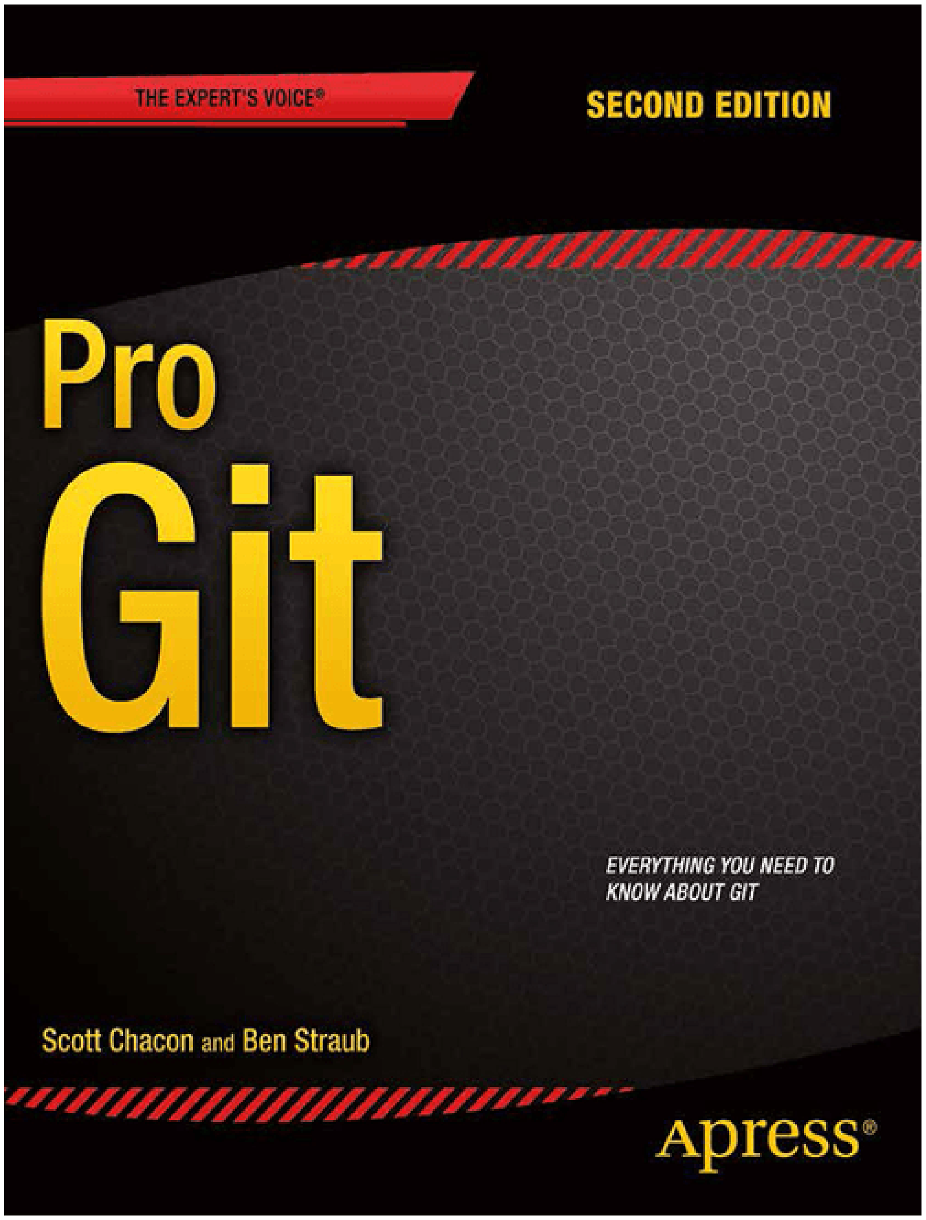
Git

Introduction

Part II

Created by Luis Ibanez / @luisibanez, modifed by Moorthy /@mskmoorthy and Wesley Turner / @wdturner

Reading Material


THE EXPERT'S VOICE®
SECOND EDITION
Pro Git
EVERYTHING YOU NEED TO KNOW ABOUT GIT
Scott Chacon and Ben Straub
Apress®

Pro Git

by Scott Chacon and Ben Straub

Chapters 3.6 and 5.1

Complete First

Git-Introduction-Part-I Exercises

Clone Repository

```
git clone git@github.com:wdturner/OSSStory.git
```

- github username = "wdturner"
- repository name = "OSSStory"

Go Inside the Directory

```
cd OSSStory
```

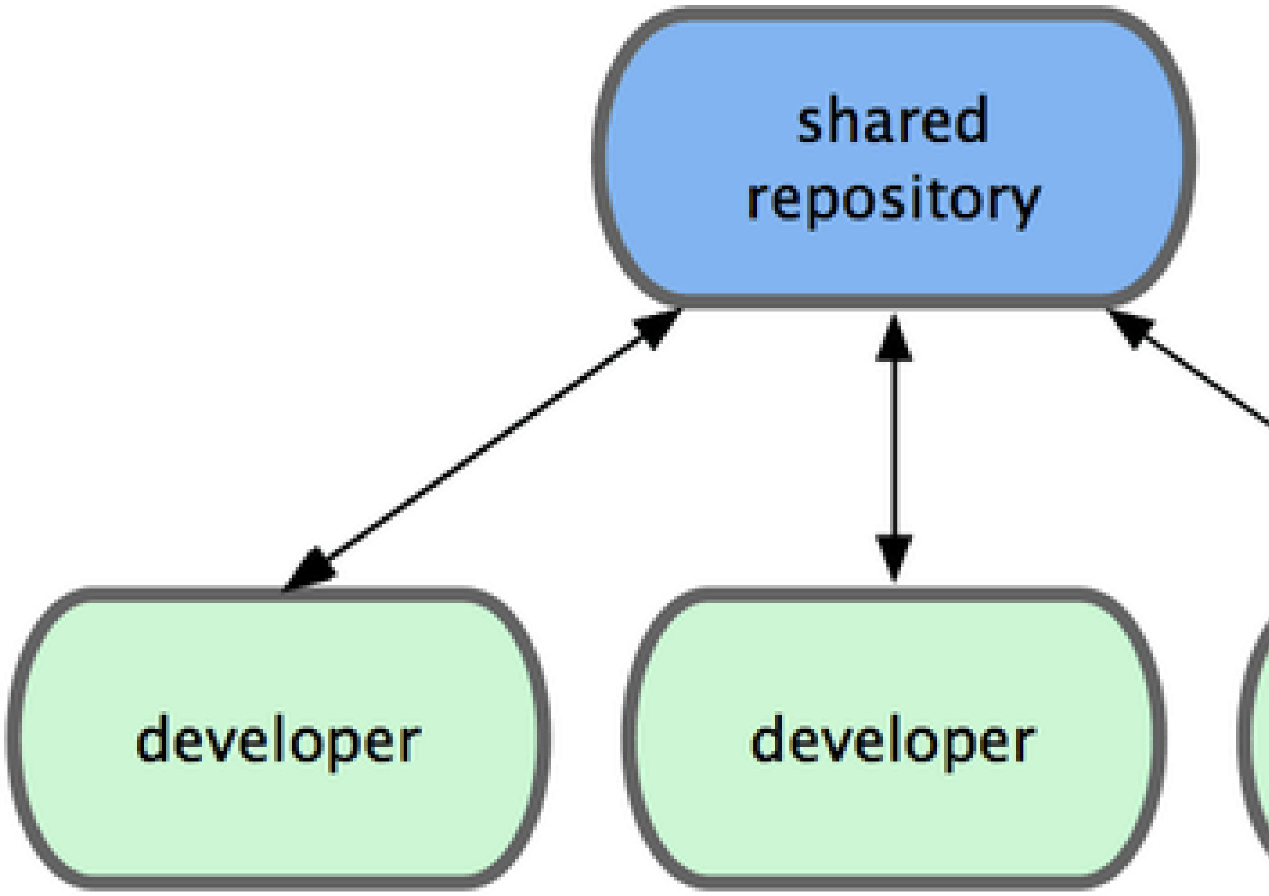
See the remote repositories

```
git remote -v
```

- "-v" is for "verbose"

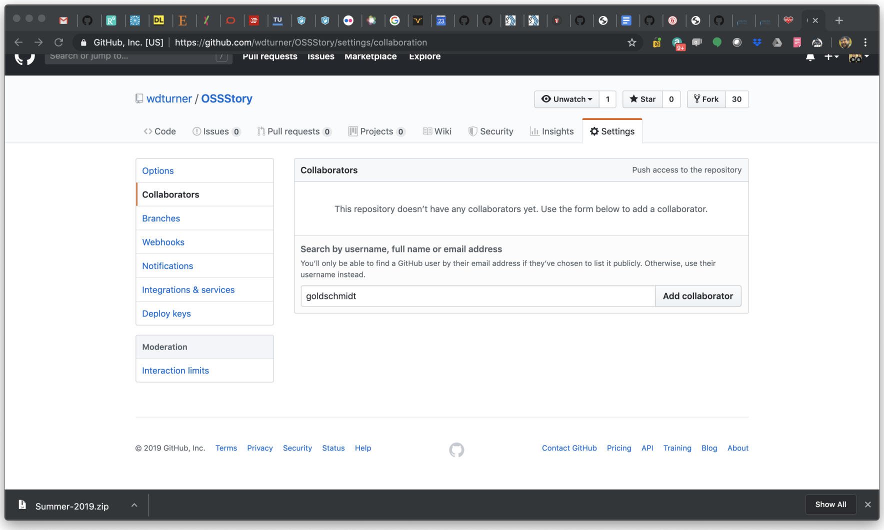
At this point we have a

Shared Repository



Courtesy of git-scm Book.

We can allow unfettered access by adding collaborators . . .


https://github.com/wdturner/OSSStory/settings/collaboration

wdturner / OSSStory

Unwatch 1 Star 0 Fork 30

Code · Issues 0 · Pull requests 0 · Projects 0 · Wiki · Security · Insights · Settings

Options
Collaborators
Branches
Webhooks
Notifications
Integrations & services
Deploy keys

Moderation
Interaction limits

Collaborators Push access to the repository

This repository doesn't have any collaborators yet. Use the form below to add a collaborator.

Search by username, full name or email address

You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead.

goldschmidt [Add collaborator]

© 2019 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

Summer-2019.zip Show All

This configuration

is used for a

Centralized Workflow

Git enables

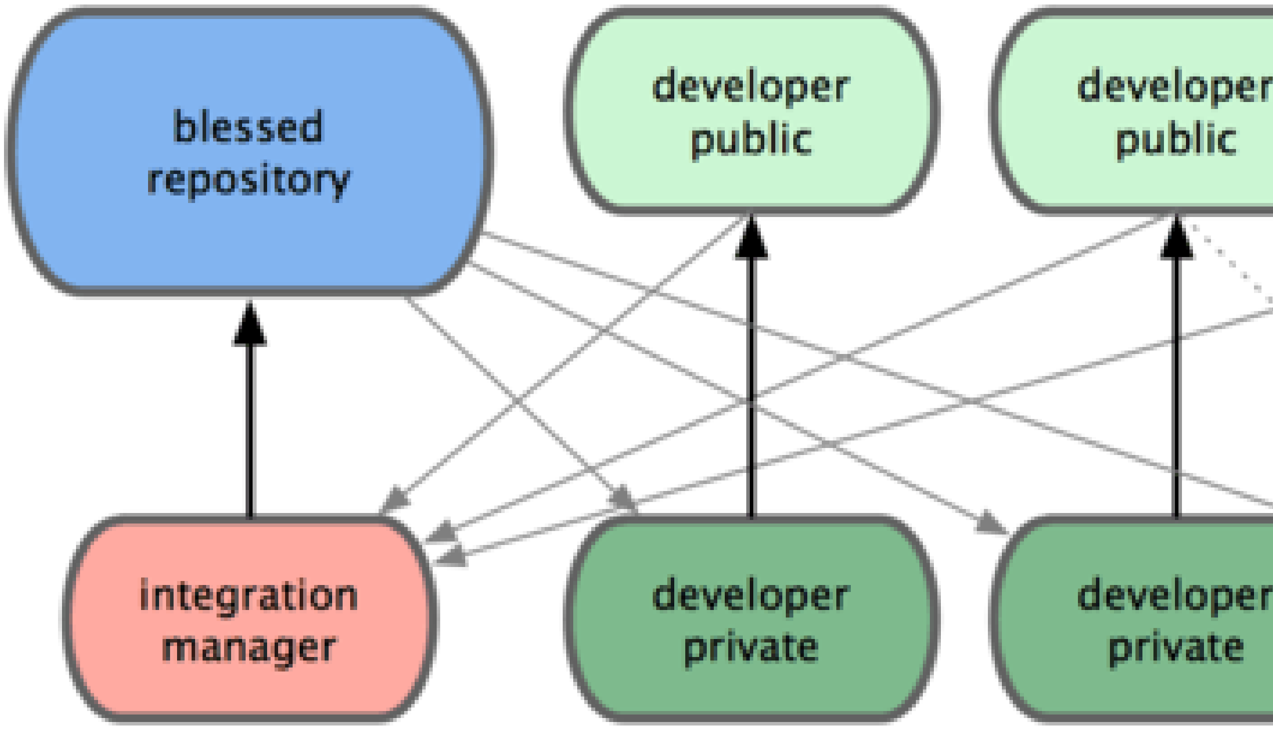
other configurations...

to support

other workflows...

For example

Integration Manager Workflow



Courtesy of git-scm Book.

Let’s create a workflow!

You already have

local private

repositories

Let’s create

your public

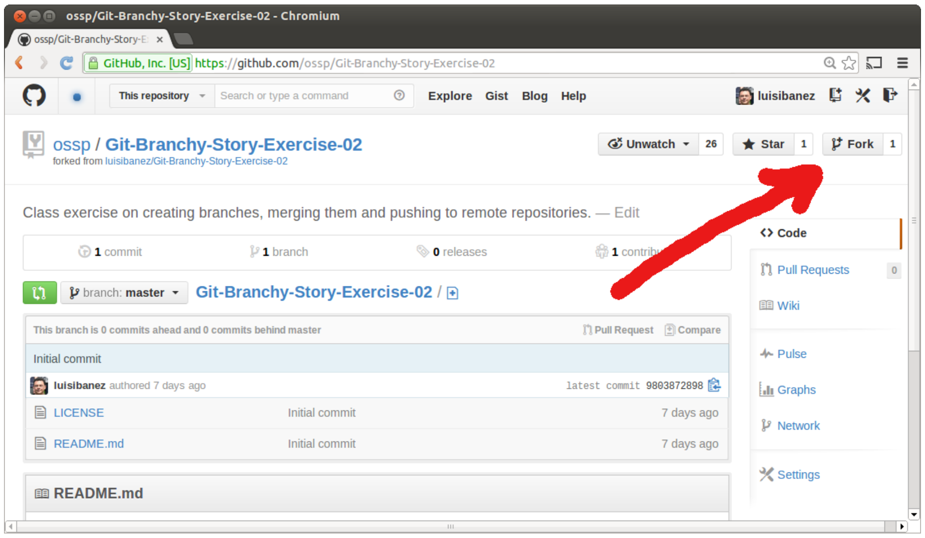
repositories

We do this

by forking

in Github

Fork the Repository



Class exercise on creating branches, merging them and pushing to remote repositories. — Edit

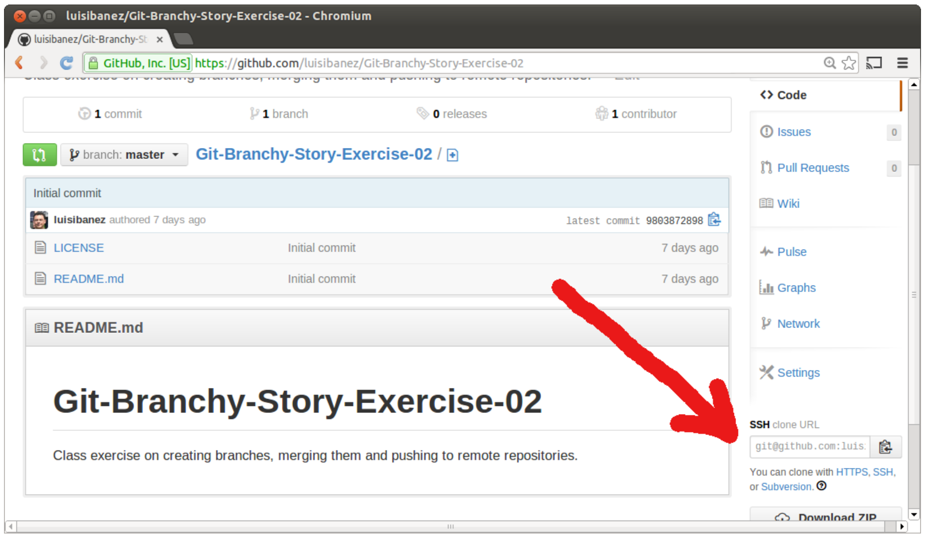
This branch is 0 commits ahead and 0 commits behind master

Initial commit

luisibanez authored 7 days ago — latest commit 9803872898

LICENSE	Initial commit	7 days ago
README.md	Initial commit	7 days ago

Copy the Fork Address


luisibanez/Git-Branchy-Story-Exercise-02 - Chromium
GitHub, Inc. [US] https://github.com/luisibanez/Git-Branchy-Story-Exercise-02
1 commit
1 branch
0 releases
1 contributor
branch: master
Git-Branchy-Story-Exercise-02
Code
Issues 0
Pull Requests 0
Wiki
Pulse
Graphs
Network
Settings
SSH clone URL
git@github.com:luis
Download ZIP

Initial commit
luisibanez authored 7 days ago latest commit 9803872898

LICENSE	Initial commit	7 days ago
README.md	Initial commit	7 days ago

README.md

Git-Branchy-Story-Exercise-02

Class exercise on creating branches, merging them and pushing to remote repositories.

You can clone with HTTPS, SSH, or Subversion.

Now go to your

local private

repository

Add the Remote repository as the new orgin

```
git remote set-url origin "your clone address"
```

- origin = local name for remote repository
- This is where you will do your personal work

See the remote repositories

```
git remote -v
```

Your Turn !

Add Remote Repositories

Work with your Team

- Ask team members for their fork address
- Add their address as a remote

- Name the remotes after their first names

Use HTTPS

For the address of your team mates

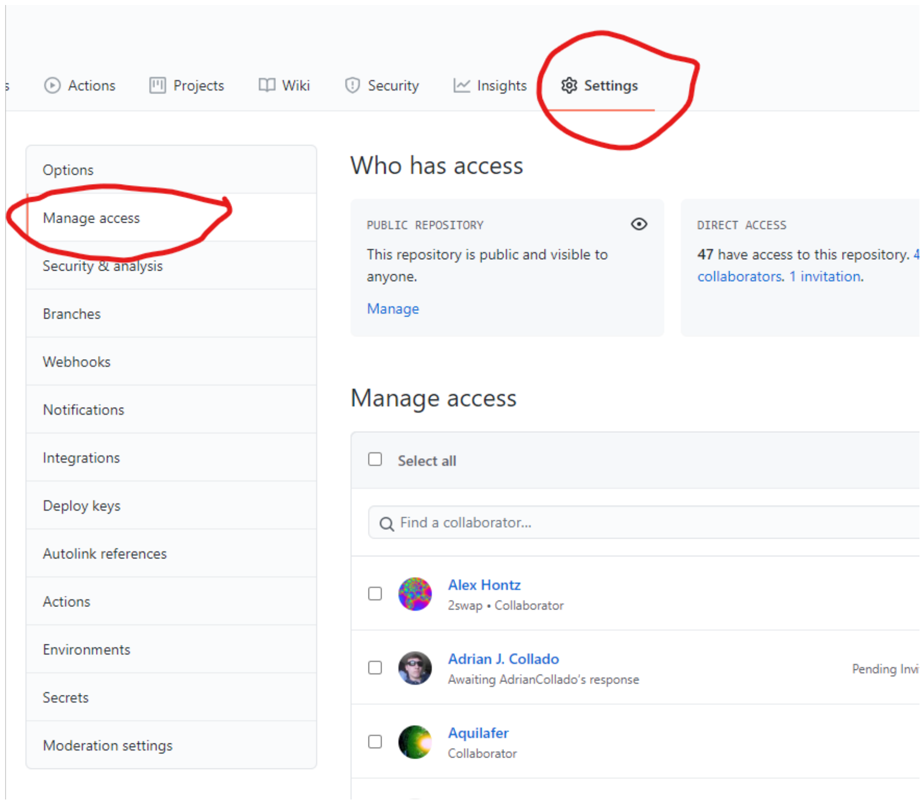
- Use: https://github.com/aaronsw/...
- Instead of: git@github.com:aaronsw/...

Now choose one of your table to host the Centralized Repository

```
git remote add upstream "address of centralized repository"
```

- upstream will be the centralized repository
- owner of the upstream will need to add everyone else as a collaborator

Add Collaborators



Who has access

PUBLIC REPOSITORY

This repository is public and visible to anyone.

Manage

DIRECT ACCESS

47 have access to this repository. 4 collaborators. 1 invitation.

Manage access

Select all

Find a collaborator...

Alex Hontz
2swap • Collaborator

Adrian J. Collado
Awaiting AdrianCollado's response
Pending Invi

Aquilafer
Collaborator

What does our workflow look like?

Now let's create

a team directory

to work together

One Team Member

Creates a Branch

```
git branch Table<i>
```

```
git checkout Table<i>
```

```
cd <Current Semester>
```

```
mkdir Table<i>
```

```
cd Table<i>
```

```
nano table<i>.Md #Or your editor of choice
```

```
cd ..
```

One Team Member

Commits Team Directory

```
git add table&lti>.Md
```

```
git commit -m "Started team directory"
```

merges to master and pushes it to upstream

```
git checkout master
```

```
git merge Table<i>
```

```
git push upstream master
```

Your push may fail...

with a message like this

```
 ! [rejected]        master -> master (non-fast-forward)
```

```
error: failed to push some refs to 'git@github.com:wdturner/OSSStory.git'
hint: Updates were rejected because a pushed branch tip is behind its remote
hint: counterpart. Check out this branch and merge the remote changes
hint: (e.g. 'git pull') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

Your push may fail...

with a message like rejected

This just means

than someone else

pushed before you

so now

you have to update

your local repository

Update With the Command

```
git pull upstream master
```

and then try the Command again

```
git push upstream master
```

You may have

to do it

multiple times

All Team Members

Update their local table .Md

```
git pull upstream master
git push origin master
```

All Team Members

Look at recent commits

```
git log
```

Note That

The log tells you:

- Who did it
- When it was done

All Team Members

Look at recent commits

Using short format

```
git log --oneline
```

Note That

Every commit

is identified by a hash

such as:

```
5d7d137a3592fe402a4bc282375c2d8bb878326c
```

You can use

the first characters

to refer

to this commit

You can use

"git show"

```
git show 5d7d137
```

to find out more

about this commit

See commit details

```
git show 5d7d137
```

Each Team Member

- Picks a "character" name
- Goes into the team directory
- Creates a file with that name
- and extension ".md"

Each Team Member

- Inside the file write one sentence
- Describing the state of the character

Each Team Member

- Git add the file
- Git commit
- Git push to upstream

Your push may fail...

with a message like this

```
 ! [rejected]        master -> master (non-fast-forward)
error: failed to push some refs to 'git@github.com:wdturner/OSSStory.git'
hint: Updates were rejected because a pushed branch tip is behind its remote
hint: counterpart. Check out this branch and merge the remote changes
hint: (e.g. 'git pull') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

This just means

than someone else

pushed before you

so now

you have to update

your local repository

Update With the Command

```
git pull upstream master
```

and then try the Command again

```
git push upstream master
```

You may have

to do it

multiple times

Each Team Member

- Edits the file of her/his character
- Changes the state of the character

Each Team Member

- Git create branch
- Git select branch
- Git add
- Git commit into branch

Use the Commands

```
git branch MyNewBranch
git checkout MyNewBranch
vim myCharacter.md
git add myCharacter.md
git commit
```

The commit message

Must explain "how" and "why"

the character changed state

Each Team Member

- Git move to master branch
- Git update master from upstream
- Git merge branch into master
- Git push master to upstream

Use the Commands

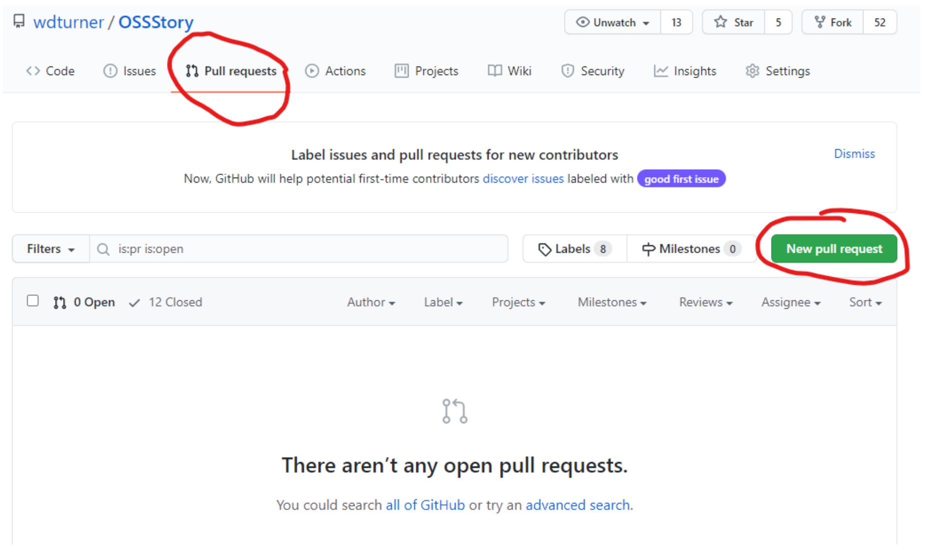
```
git checkout master
git pull upstream master
git merge --no-ff MyNewBranch
git push upstream master
```

Repeat the push until it succeds

```
git pull upstream master
git push upstream master
# When you're done, end by updating your personal repository
git push origin master
```

Finally, one person at the table

- Do a Pull Request to wdturner/OSSStory



Label issues and pull requests for new contributors

Now, GitHub will help potential first-time contributors discover issues labeled with good first issue

There aren't any open pull requests.

You could search all of GitHub or try an advanced search.

This shows 2 Workflows

- Your table is a "Collective Workflow"
- The Pull Request back to the wdturner repository is a "Blessed Repository"

THE END

BY Luis Ibanez,

modified

by Moorthy and Wes Turner