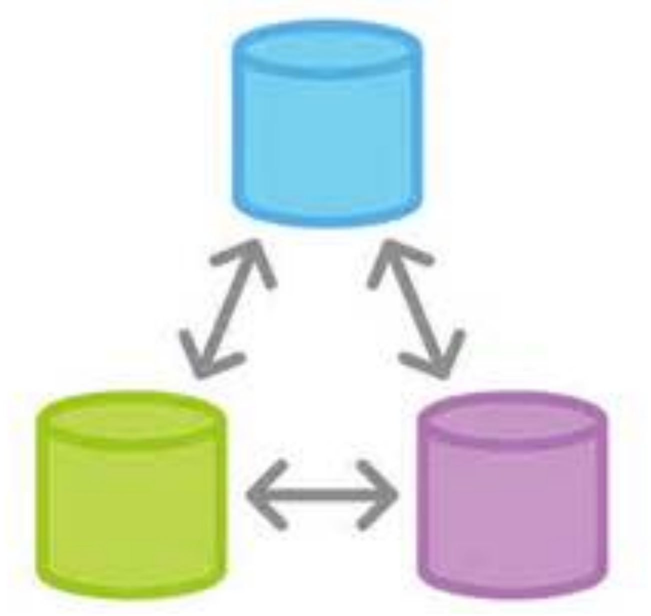
Lecture 7
More Remotes and Working with Github

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

Sign in on the attendance sheet!

Today

- Review of basic remotes
- More practical explanation of remotes with GitHub
- Practice using GitHub



Remote Tracking Branches

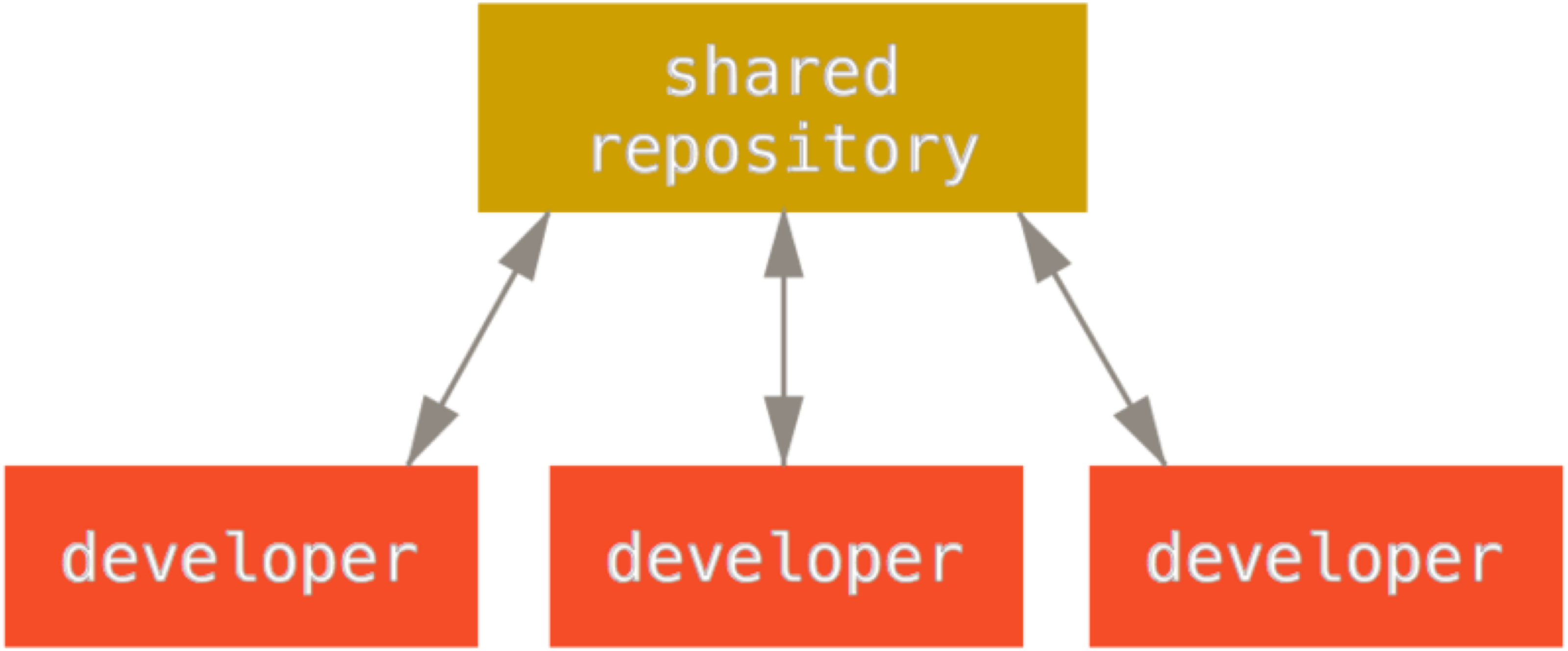
- Usually, we create local branches that have the same names as remote branches.
- Local branches can track remote branches. Git will tell you when they are “ahead”, “behind”, “diverged”, or “up-to-date”.
- Use git push -u <remote> <branch> to setup.

Centralized Git Workflow

Scenario: We want to contribute a change to a file in a repository on GitHub

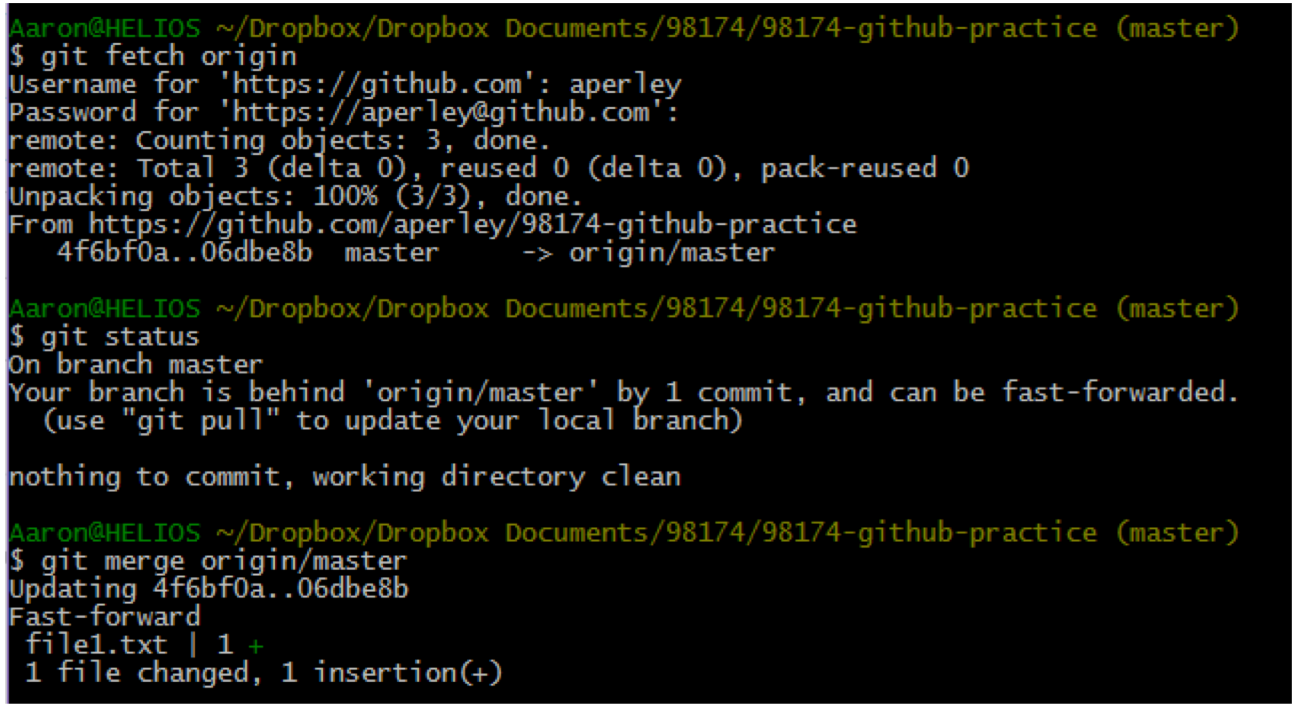
Centralized Git Workflow

WHAT?! I thought Git was a **Distributed** Version Control System!



Centralized Git Workflow

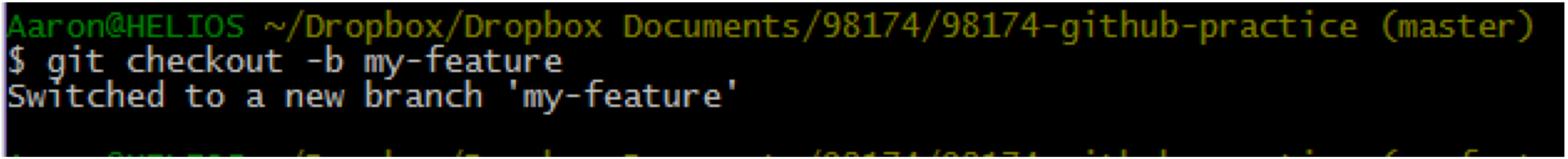
Step 1: Make sure the master branch in your repository is up to date with origin/master



```
Aaron@HELIOS ~/Dropbox/Dropbox Documents/98174/98174-github-practice (master)
$ git fetch origin
Username for 'https://github.com': aperley
Password for 'https://aperley@github.com':
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/aperley/98174-github-practice
   4f6bf0a..06dbe8b  master     -> origin/master

Aaron@HELIOS ~/Dropbox/Dropbox Documents/98174/98174-github-practice (master)
$ git status
On branch master
Your branch is behind 'origin/master' by 1 commit, and can be fast-forwarded.
  (use "git pull" to update your local branch)

nothing to commit, working directory clean

Aaron@HELIOS ~/Dropbox/Dropbox Documents/98174/98174-github-practice (master)
$ git merge origin/master
Updating 4f6bf0a..06dbe8b
Fast-forward
 file1.txt | 1 +
 1 file changed, 1 insertion(+)
```

Centralized Git Workflow

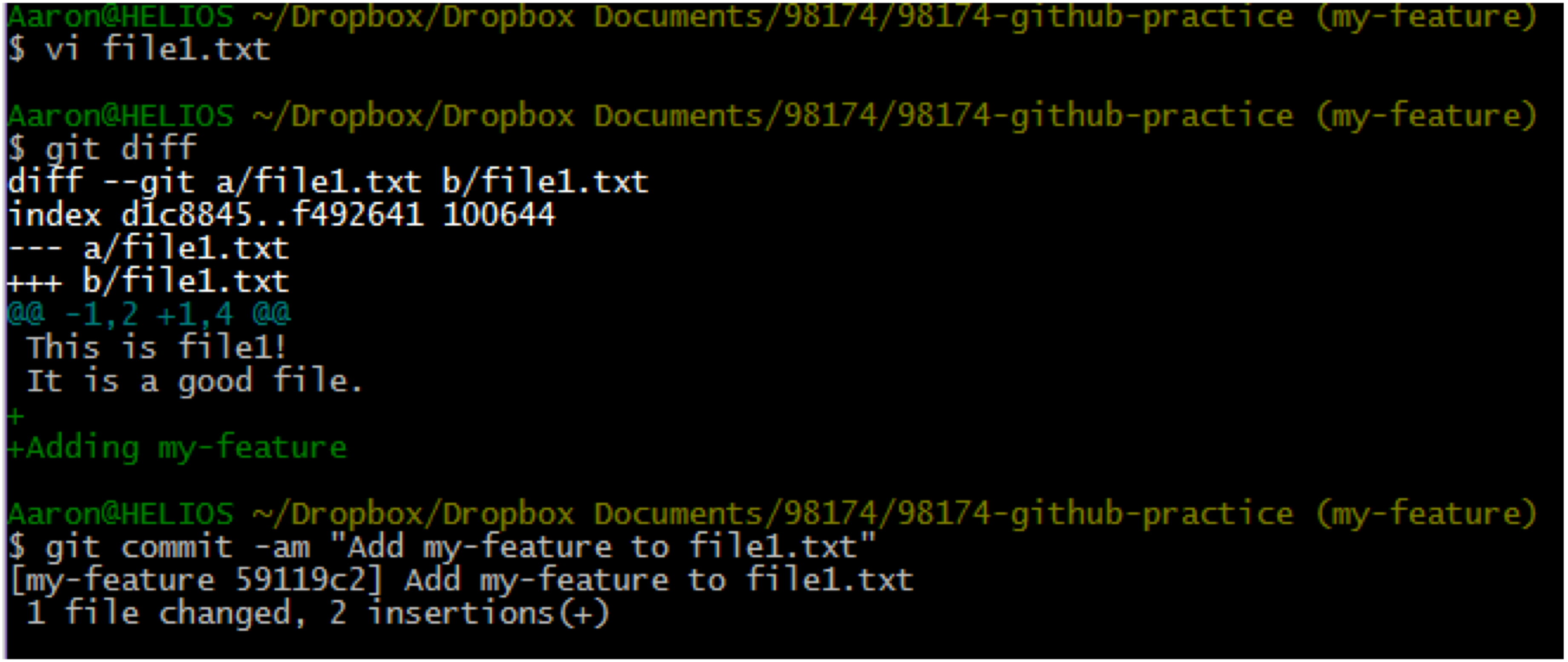
Step 2: Create a new “topic branch” from master



```
Aaron@HELIOS ~/Dropbox/Dropbox Documents/98174/98174-github-practice (master)
$ git checkout -b my-feature
Switched to a new branch 'my-feature'
```

Centralized Git Workflow

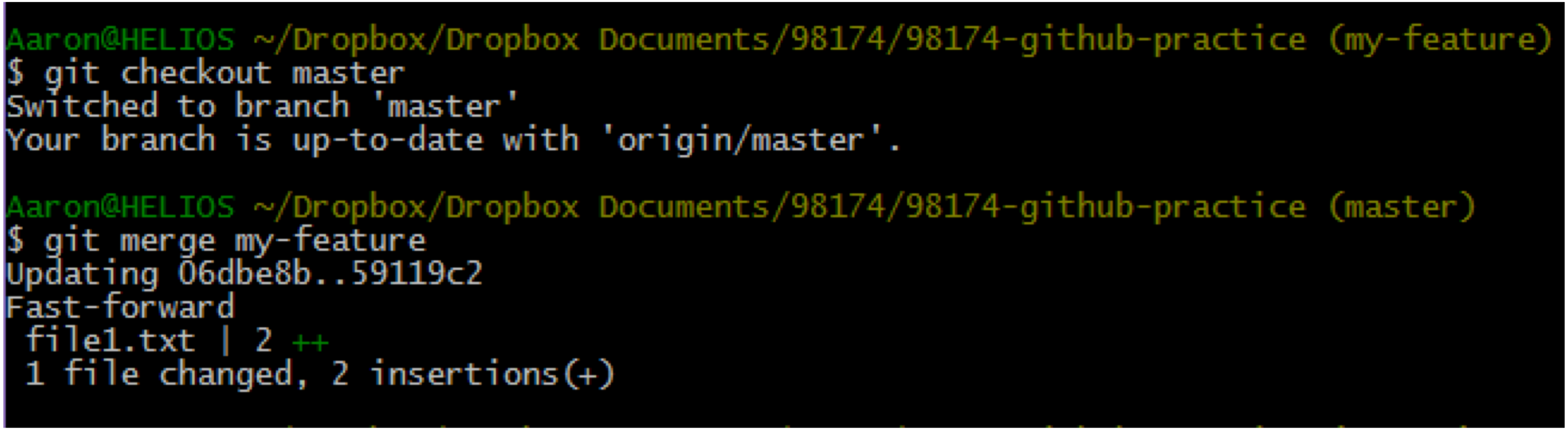
Step 3: Work and make some commits on that branch



```
Aaron@HELIOS ~/Dropbox/Dropbox Documents/98174/98174-github-practice (my-feature)
$ vi file1.txt

Aaron@HELIOS ~/Dropbox/Dropbox Documents/98174/98174-github-practice (my-feature)
$ git diff
diff --git a/file1.txt b/file1.txt
index d1c8845..f492641 100644
--- a/file1.txt
+++ b/file1.txt
@@ -1,2 +1,4 @@
 This is file1!
 It is a good file.
+
+Adding my-feature

Aaron@HELIOS ~/Dropbox/Dropbox Documents/98174/98174-github-practice (my-feature)
$ git commit -am "Add my-feature to file1.txt"
[my-feature 59119c2] Add my-feature to file1.txt
 1 file changed, 2 insertions(+)
```

Centralized Git Workflow

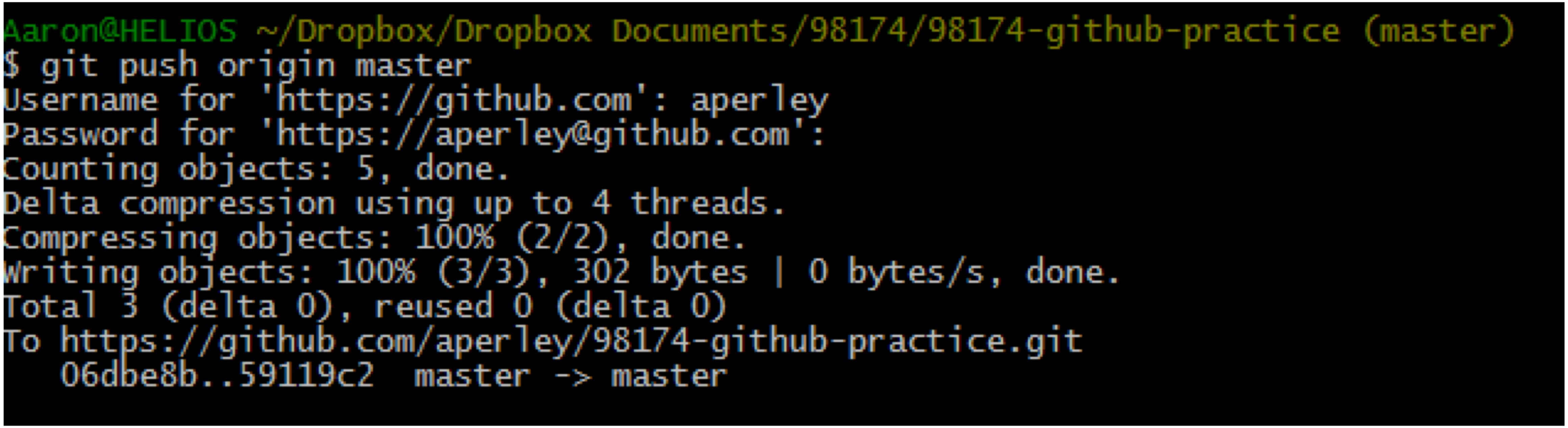
Step 4: Merge the branch back into master



```
Aaron@HELIOS ~/Dropbox/Dropbox Documents/98174/98174-github-practice (my-feature)
$ git checkout master
Switched to branch 'master'
Your branch is up-to-date with 'origin/master'.

Aaron@HELIOS ~/Dropbox/Dropbox Documents/98174/98174-github-practice (master)
$ git merge my-feature
Updating 06dbe8b..59119c2
Fast-forward
 file1.txt | 2 ++
 1 file changed, 2 insertions(+)
```

Centralized Git Workflow

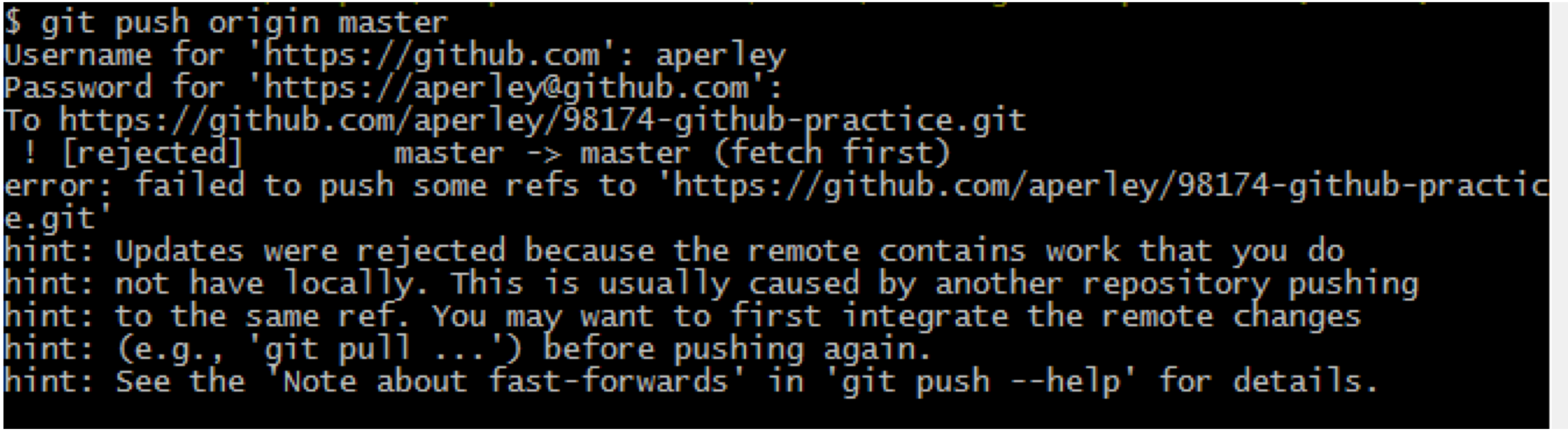
Step 5: Push master to origin/master



```
Aaron@HELIOS ~/Dropbox/Dropbox Documents/98174/98174-github-practice (master)
$ git push origin master
Username for 'https://github.com': aperley
Password for 'https://aperley@github.com':
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 302 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/aperley/98174-github-practice.git
   06dbe8b..59119c2  master -> master
```

What if someone else pushes to master before I do?

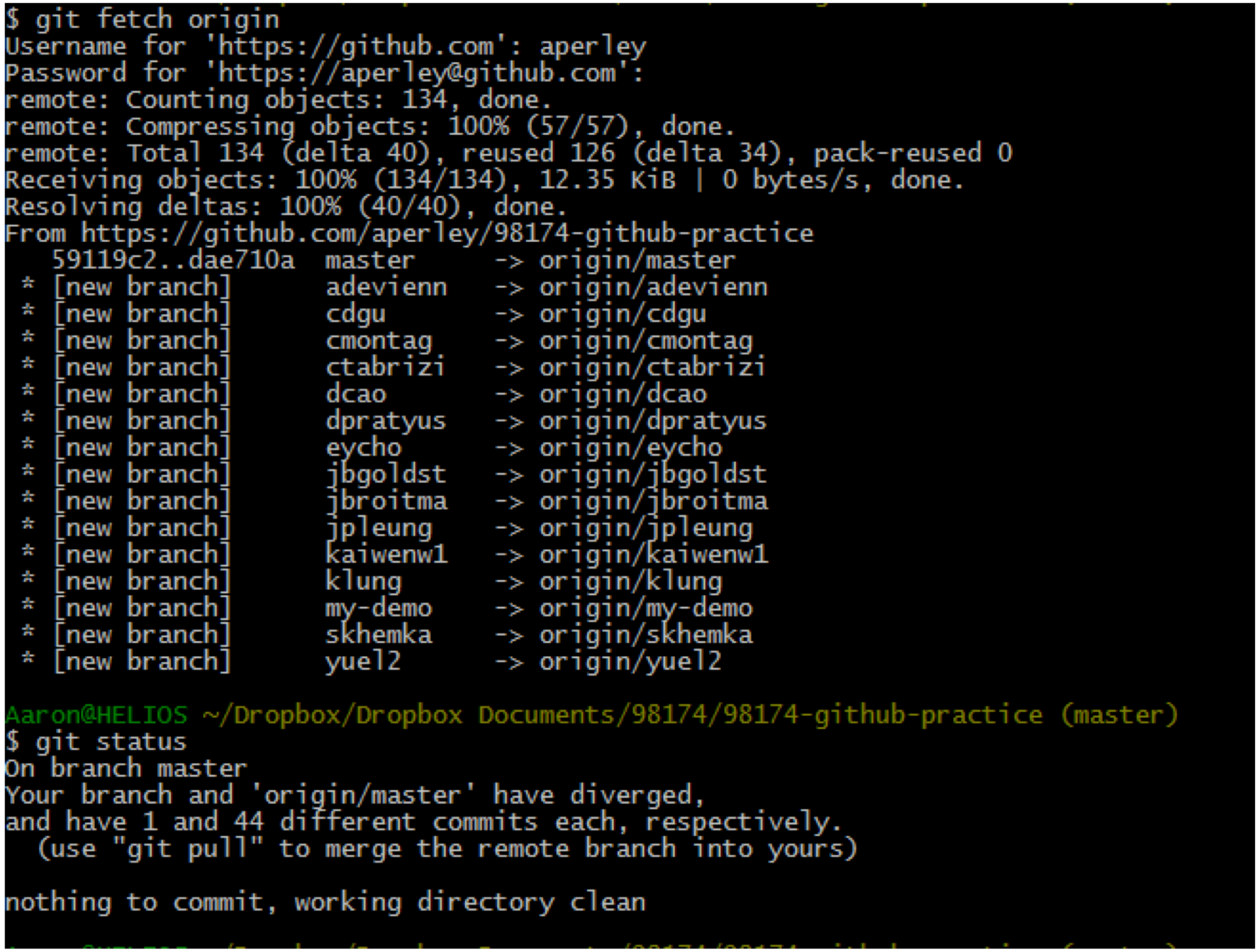
- Your push will be rejected:



```
$ git push origin master
Username for 'https://github.com': aperley
Password for 'https://aperley@github.com':
To https://github.com/aperley/98174-github-practice.git
 ! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'https://github.com/aperley/98174-github-practic
e.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

What if someone else pushes to master before I do?

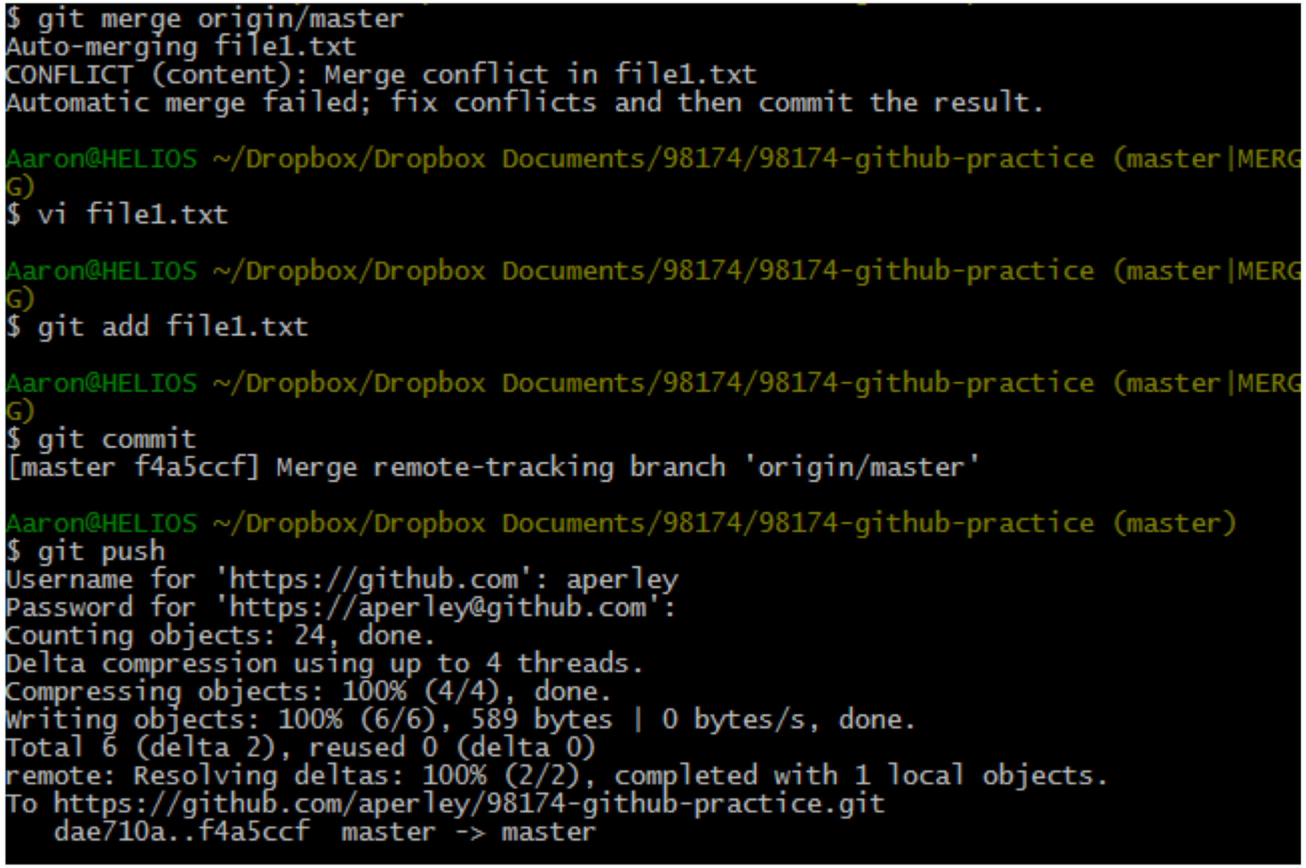
- git status will indicate that your branch and its remote tracking branch have **diverged**



```
$ git fetch origin
Username for 'https://github.com': aperley
Password for 'https://aperley@github.com':
remote: Counting objects: 134, done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 134 (delta 40), reused 126 (delta 34), pack-reused 0
Receiving objects: 100% (134/134), 12.35 KiB | 0 bytes/s, done.
Resolving deltas: 100% (40/40), done.
From https://github.com/aperley/98174-github-practice
   59119c2..dae710a  master     -> origin/master
 * [new branch]      adevienn   -> origin/adevienn
 * [new branch]      cdgu       -> origin/cdgu
 * [new branch]      cmontag    -> origin/cmontag
 * [new branch]      ctabrizi   -> origin/ctabrizi
 * [new branch]      dcao       -> origin/dcao
 * [new branch]      dpratyus   -> origin/dpratyus
 * [new branch]      eycho      -> origin/eycho
 * [new branch]      jbgoldst   -> origin/jbgoldst
 * [new branch]      jbroitma   -> origin/jbroitma
 * [new branch]      jpleung    -> origin/jpleung
 * [new branch]      kaiwenw1   -> origin/kaiwenw1
 * [new branch]      klung      -> origin/klung
 * [new branch]      my-demo    -> origin/my-demo
 * [new branch]      skhemka    -> origin/skhemka
 * [new branch]      yuel2      -> origin/yuel2

Aaron@HELIOS ~/Dropbox/Dropbox Documents/98174/98174-github-practice (master)
$ git status
On branch master
Your branch and 'origin/master' have diverged,
and have 1 and 44 different commits each, respectively.
  (use "git pull" to merge the remote branch into yours)

nothing to commit, working directory clean
```

What if someone else pushes to master before I do?

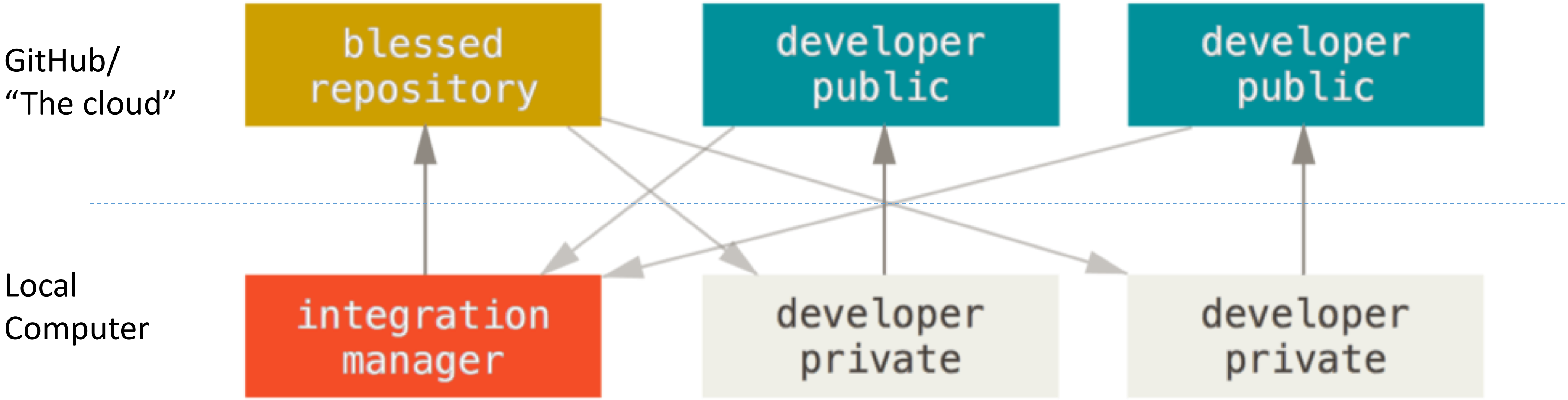
- We know how to fix diversions! `git merge`



```
$ git merge origin/master
Auto-merging file1.txt
CONFLICT (content): Merge conflict in file1.txt
Automatic merge failed; fix conflicts and then commit the result.

Aaron@HELIOS ~/Dropbox/Dropbox Documents/98174/98174-github-practice (master|MERG
G)
$ vi file1.txt

Aaron@HELIOS ~/Dropbox/Dropbox Documents/98174/98174-github-practice (master|MERG
G)
$ git add file1.txt

Aaron@HELIOS ~/Dropbox/Dropbox Documents/98174/98174-github-practice (master|MERG
G)
$ git commit
[master f4a5ccf] Merge remote-tracking branch 'origin/master'

Aaron@HELIOS ~/Dropbox/Dropbox Documents/98174/98174-github-practice (master)
$ git push
Username for 'https://github.com': aperley
Password for 'https://aperley@github.com':
Counting objects: 24, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 589 bytes | 0 bytes/s, done.
Total 6 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local objects.
To https://github.com/aperley/98174-github-practice.git
   dae710a..f4a5ccf  master -> master
```

Is there a better way?

Integration-Manager Workflow



GitHub/
“The cloud”

blessed repository

developer public

developer public

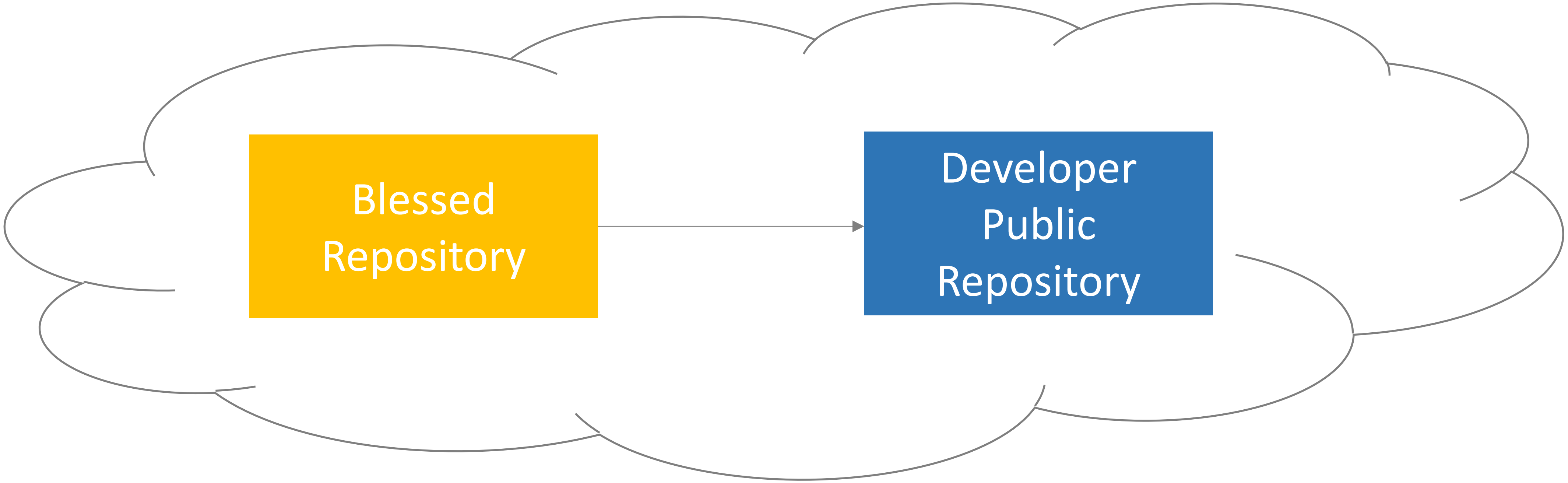
Local
Computer

integration manager

developer private

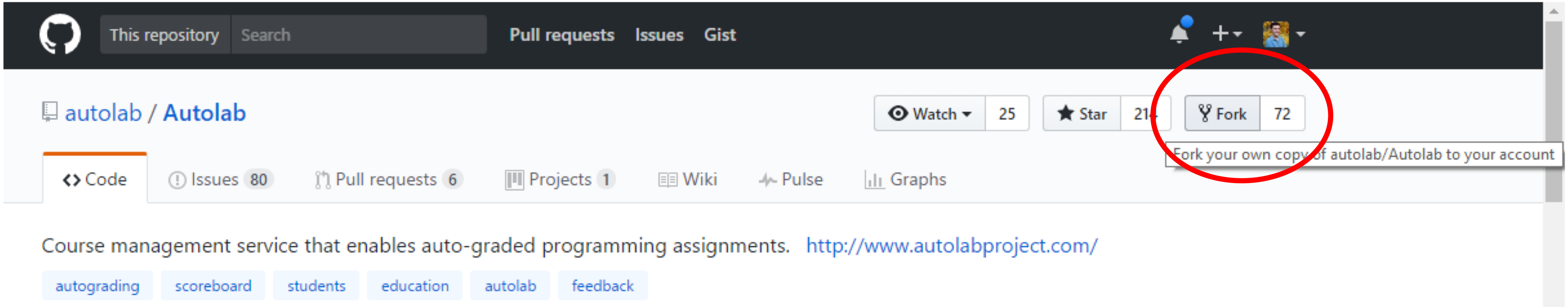
developer private

Step 1. **Fork** the public repository



Blessed Repository

Developer Public Repository

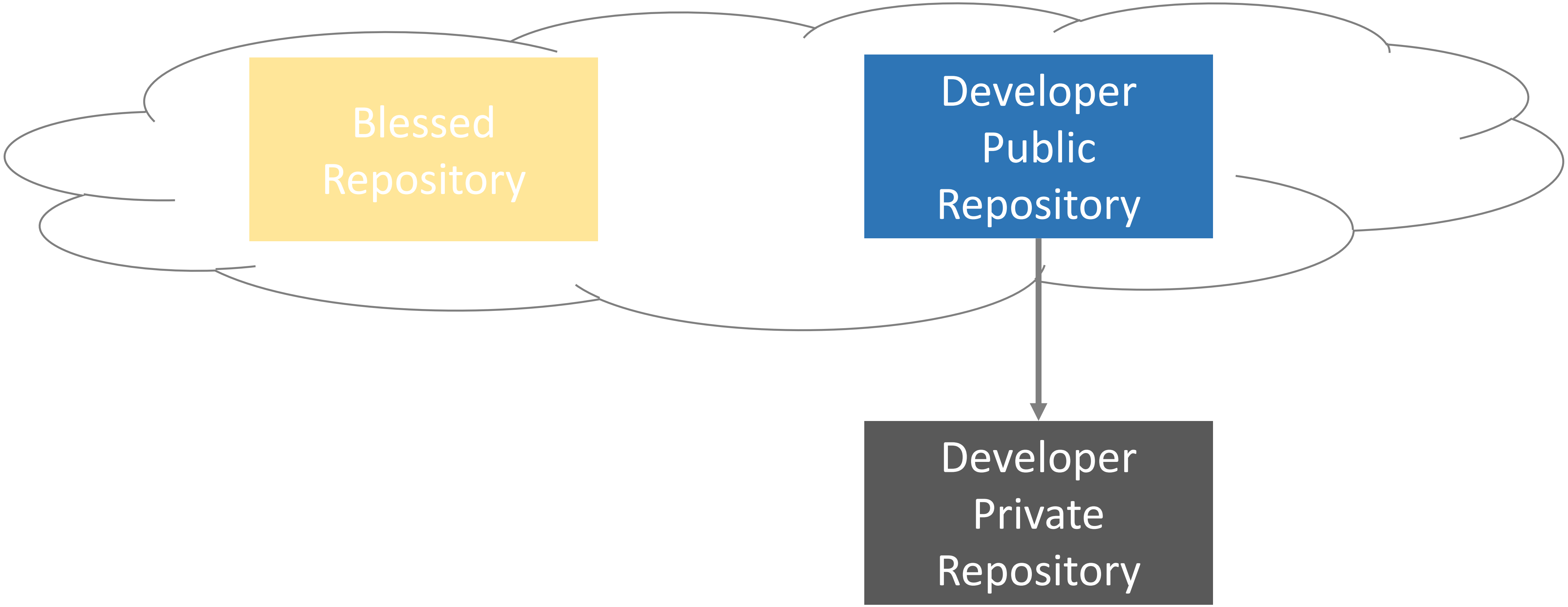


Fork your own copy of autolab/Autolab to your account

Course management service that enables auto-graded programming assignments. http://www.autolabproject.com/

Step 2. Clone your public repository

```
$ git clone https://github.com/aperley/Autolab.git
```

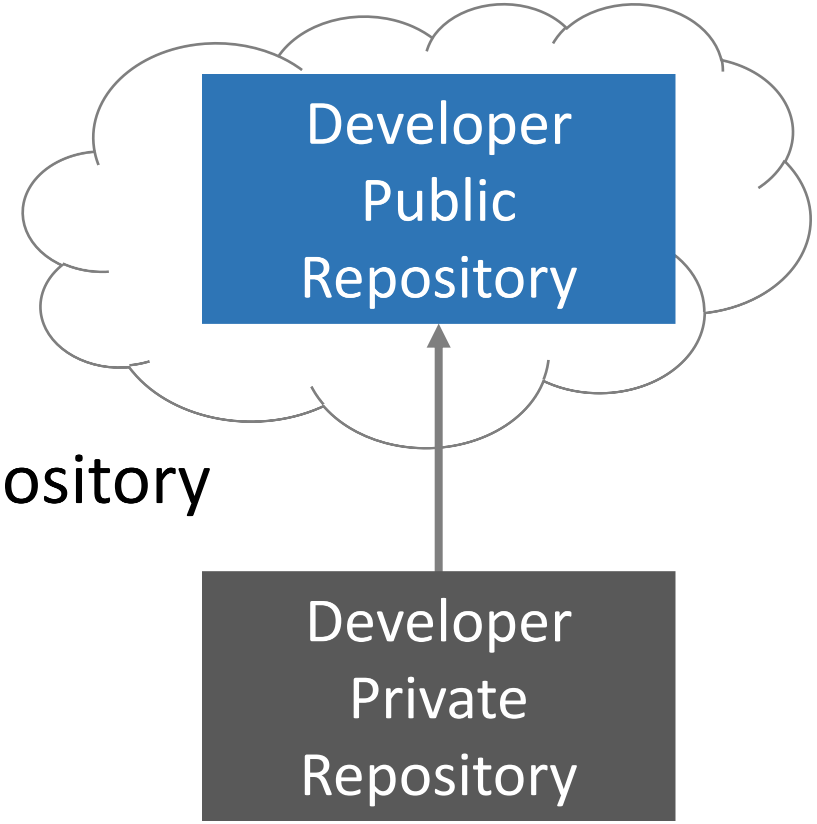


Step 3. Create a **feature branch** and make some commits

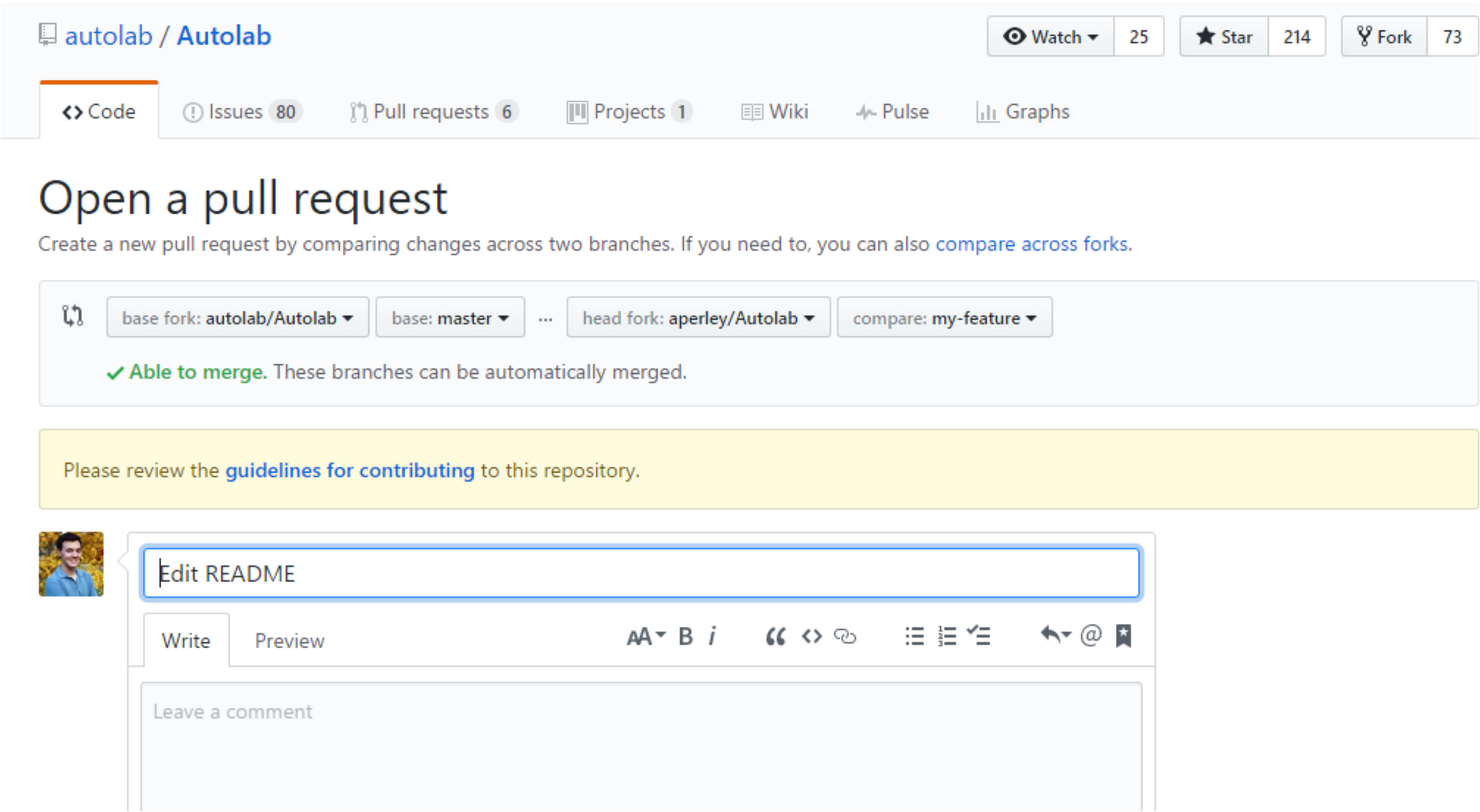
```
$ git checkout -b my-feature
$ <do some work>
$ git commit -am "add my feature"
```

Then **push** your feature branch to your public repository

```
$ git push origin my-feature
```



Step 4. Create a **pull request**



autolab / Autolab

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 fork: autolab/Autolab base: master ... head fork: aperley/Autolab compare: my-feature

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

Please review the guidelines for contributing to this repository.

Edit README

The integration manager can inspect and **pull in** your changes

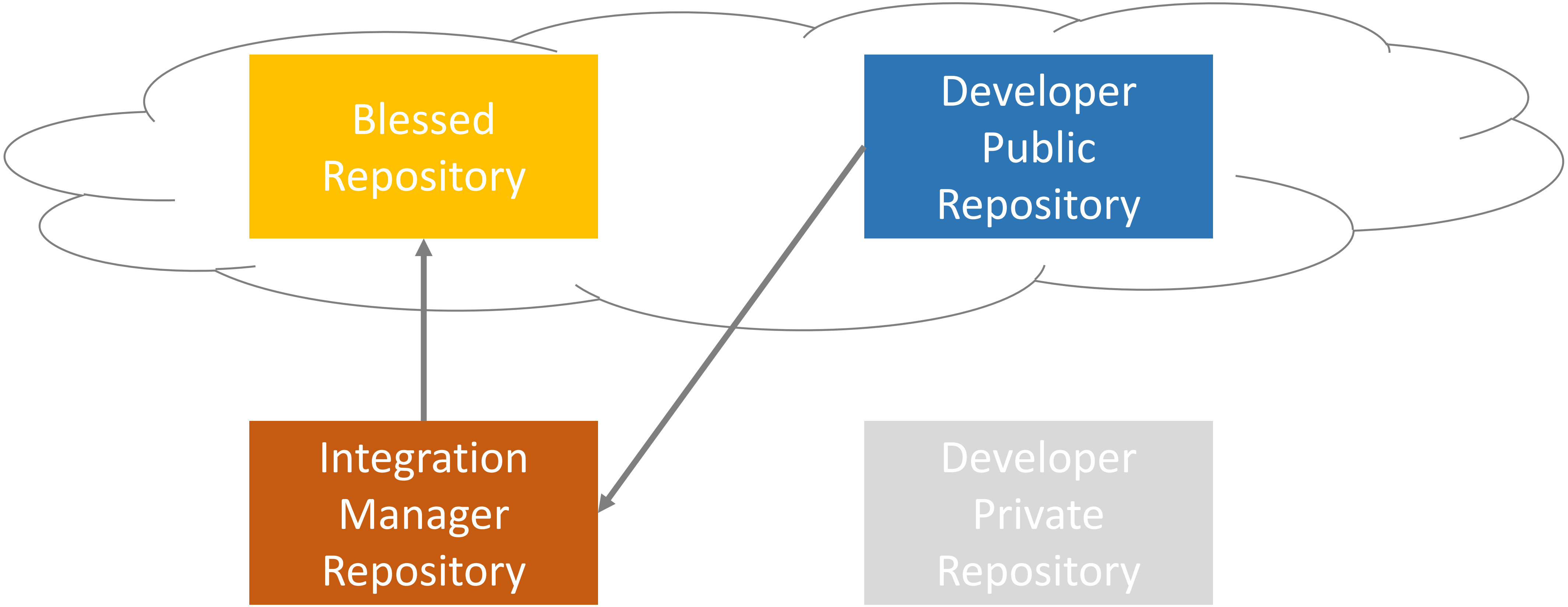
As the integration manager:

```
$ git remote add aperleys-fork
https://github.com/aperley/Autolab.git
$ git checkout aperleys-fork/my-feature
```

If it looks good:

```
$ git checkout master
$ git merge aperleys-fork/my-feature
$ git push origin master
```

The integration manager can inspect and **pull in** your changes

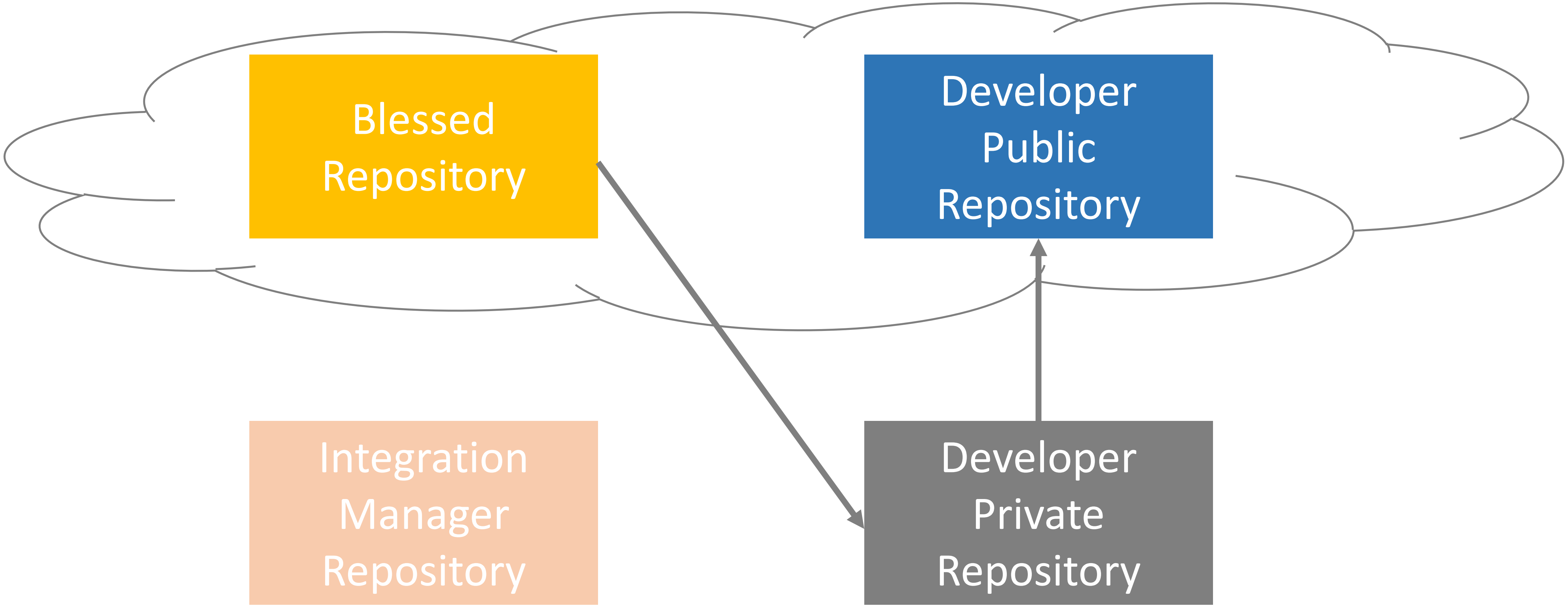

Blessed Repository
Developer Public Repository
Integration Manager Repository
Developer Private Repository

You need to keep your fork up to date

In the private developer repo

```
$ git remote add upstream
https://github.com/autolab/Autolab.git
$ git fetch upstream
$ git checkout master
$ git merge upstream/master
$ git push origin master
```

You need to keep your fork up to date



Blessed Repository

Developer Public Repository

Integration Manager Repository

Developer Private Repository

Activity/Homework

Create a fork of

https://github.com/ilanbiala/squirrel-story

Finish the story, push it to a branch named <ANDREWID> on your fork, and make a pull request to the blessed repository (ibiala/squirrel-story).