GIT Overview

burt.walsh@ast.myflorida.com

GIT

- Distributed version control system
- You work in a local repository
- You push changes to a remote/shared repository
- Repositories contain branches (which have a series of changes/commits)
- Branches are groups of changes/commits

Creating a Local Repository

- Determine a directory in which you will work

- Create the directory and change directory to that directory

- Perform (git init) in the directory to create a new "local" repository

- The creation of a repository results in a .git directory being created

```
Microsoft Windows [Version 10.0.16299.192]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\liam>mkdir \new_project

C:\Users\liam>cd \new_project

C:\new_project>git init
Initialized empty Git repository in C:/new_project/.git/

C:\new_project>dir .git
 Volume in drive C is Windows
 Volume Serial Number is 62F1-03A3

 Directory of C:\new_project\.git

01/26/2018  07:01 PM               130 config
01/26/2018  07:01 PM                73 description
01/26/2018  07:01 PM                23 HEAD
01/26/2018  07:01 PM    <DIR>          hooks
01/26/2018  07:01 PM    <DIR>          info
01/26/2018  07:01 PM    <DIR>          objects
01/26/2018  07:01 PM    <DIR>          refs
               3 File(s)            226 bytes
               4 Dir(s)   8,975,609,856 bytes free

C:\new_project>
```

Creating branch and your first commit

- Create a branch in the local repository
- (git checkout –b myfirstbranch)
- Create and edit a new file (Ok.txt in our case with one line of text)
- Do a git status to see what file changes have not been committed to the branch (git status)
- Add the file to the next commit (git add Ok.txt)
- Commit the file (git commit –m "my first commit")
- See the commit with (git log)

```
C:\new_project>
C:\new_project>
C:\new_project>git checkout -b myfirstbranch
Switched to a new branch 'myfirstbranch'

C:\new_project>notepad Ok.txt

C:\new_project>git status
On branch myfirstbranch

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        Ok.txt

nothing added to commit but untracked files present (use "git add" to track)

C:\new_project>git add Ok.txt

C:\new_project>git status
On branch myfirstbranch

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   Ok.txt


C:\new_project>git commit -m "myfirstcommit"
[myfirstbranch (root-commit) a075381] myfirstcommit
 1 file changed, 1 insertion(+)
 create mode 100644 Ok.txt

C:\new_project>git log
commit a0753811573962932 6f1f5ce259f16d32cea5422 (HEAD -> myfirstbranch)
Author: liamwalsh3205 walsh <walshburt@gmail.com>
Date:   Sat Jan 27 02:53:25 2018 -0500

    myfirstcommit
```

Ok - Notepad

File   Edit   Format   View   Help

This is my OK file

Type here to search

Mail

2:54 AM
1/27/2018

What is a branch? what is a commit?

- A branch is a series of commits
- A commit is a delta of the "added" changed files or removed files from the last (commit) state of the branch
- We add files to or remove files from the perspective commit (stage for the next commit)
- We then execute the commit against the branch (git commit –m "message")
- (git log –p) all commits and the actual file changes

```
Command Prompt

C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>git branch
* myfirstbranch

C:\new_project>git log -p
commit a07538115739629326f1f5ce259f16d32cea5422 (HEAD -> myfirstbranch)
Author: liamwalsh3205 walsh <walshburt@gmail.com>
Date:   Sat Jan 27 02:53:25 2018 -0500

    myfirstcommit

diff --git a/Ok.txt b/Ok.txt
new file mode 100644
index 0000000..5d8f52b
--- /dev/null
+++ b/Ok.txt
@@ -0,0 +1 @@
+This is my OK file
\ No newline at end of file

C:\new_project>
```

Review

- A branch is a series of commits

- A commit is a delta of the added, removed or just changed files from the last (commit) state of the branch

- We add or remove files from the perspective commit (stage for the next commit)

- We then execute the commit against the branch

Adding a second commit to the branch

- We create a new branch mysecondbranch (git checkout –b mysecondbranch)
- We remove the Ok.txt file from the commit (git rm Ok.txt)
- We create a Y.txt file
- We add the Y.txt file to the commit (git add Y.txt)
- We execute the commit against the branch (git commit –m "commit against second branch")

```
Command Prompt

C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>git checkout -b mysecondbranch
Switched to a new branch 'mysecondbranch'

C:\new_project>git status
On branch mysecondbranch
nothing to commit, working tree clean

C:\new_project>notepad Y.txt

C:\new_project>git status
On branch mysecondbranch
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        Y.txt

nothing added to commit but untracked files present (use "git add" to track)

C:\new_project>git add Y.txt

C:\new_project>git rm Ok.txt
rm 'Ok.txt'

C:\new_project>git status
On branch mysecondbranch
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        deleted:    Ok.txt
        new file:   Y.txt


C:\new_project>git commit -m "commit against second branch"
[mysecondbranch 43f0d0d] commit against second branch
 2 files changed, 2 insertions(+), 1 deletion(-)
 delete mode 100644 Ok.txt
 create mode 100644 Y.txt

C:\new_project>
```

Switching between branches

- Branch is normally a work product such as a fix or enhancement
- We will have a current branch
- We can switch between branches (git checkout <branch>)
- The switch results in the files being displayed in the directory in accordance with the branches commits

```
Command Prompt                                                                    ─  □  ✕

C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>git branch
  myfirstbranch
* mysecondbranch

C:\new_project>dir
 Volume in drive C is Windows
 Volume Serial Number is 62F1-03A3

 Directory of C:\new_project

01/27/2018  03:01 AM    <DIR>          .
01/27/2018  03:01 AM    <DIR>          ..
01/27/2018  03:01 AM                23 Y.txt
               1 File(s)             23 bytes
               2 Dir(s)   8,714,502,144 bytes free

C:\new_project>git checkout myfirstbranch
Switched to branch 'myfirstbranch'

C:\new_project>dir
 Volume in drive C is Windows
 Volume Serial Number is 62F1-03A3

 Directory of C:\new_project

01/27/2018  03:07 AM    <DIR>          .
01/27/2018  03:07 AM    <DIR>          ..
01/27/2018  03:07 AM                18 Ok.txt
               1 File(s)             18 bytes
               2 Dir(s)   8,714,498,048 bytes free

C:\new_project>_
```
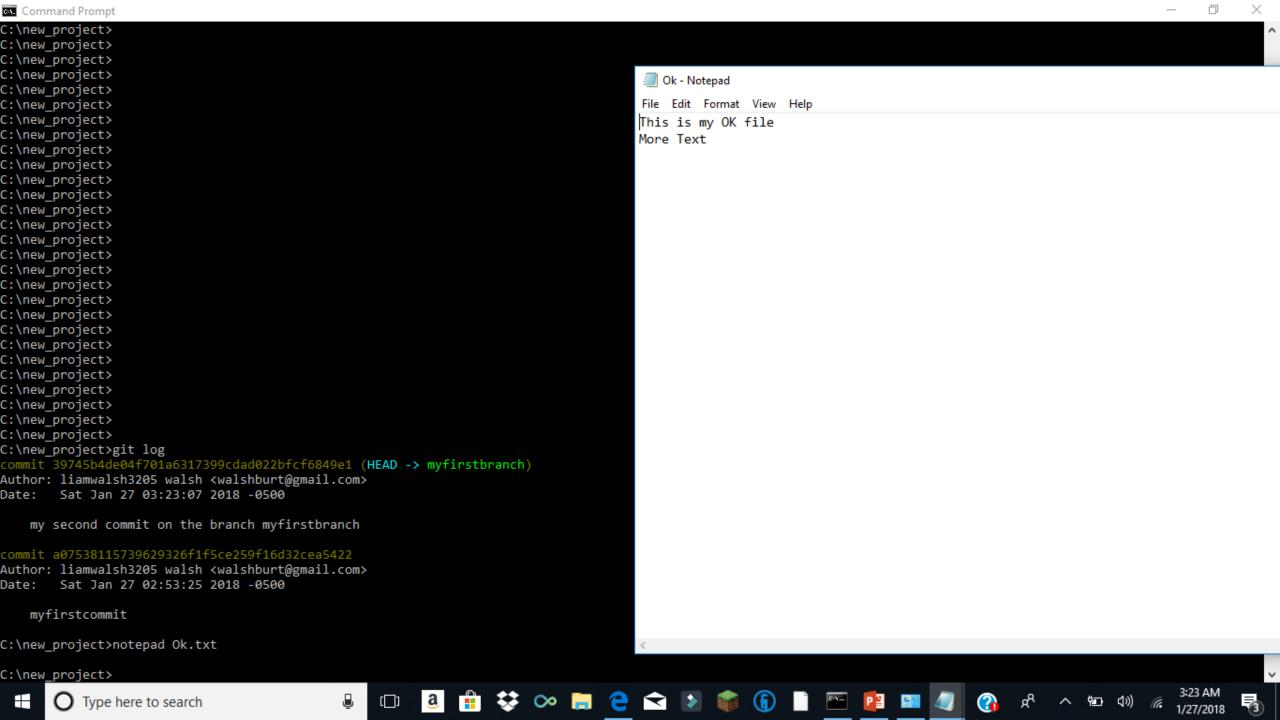
Each group of changes requires a commit

- We update files and then do a (git add <file name>)
- If we have to remove a file we do a (git rm <file name>)
- git add is for a change to a file including the creation of a new file
- Again these changes delta are committed against the current state of the branch

Adding a second commit

- We add text to our Ok.txt file

- We (git add Ok.txt) to stage the file changes to the commit

- We commit the staged changes to the branch (git commit –m "the second commit on my branch myfirstbranch"
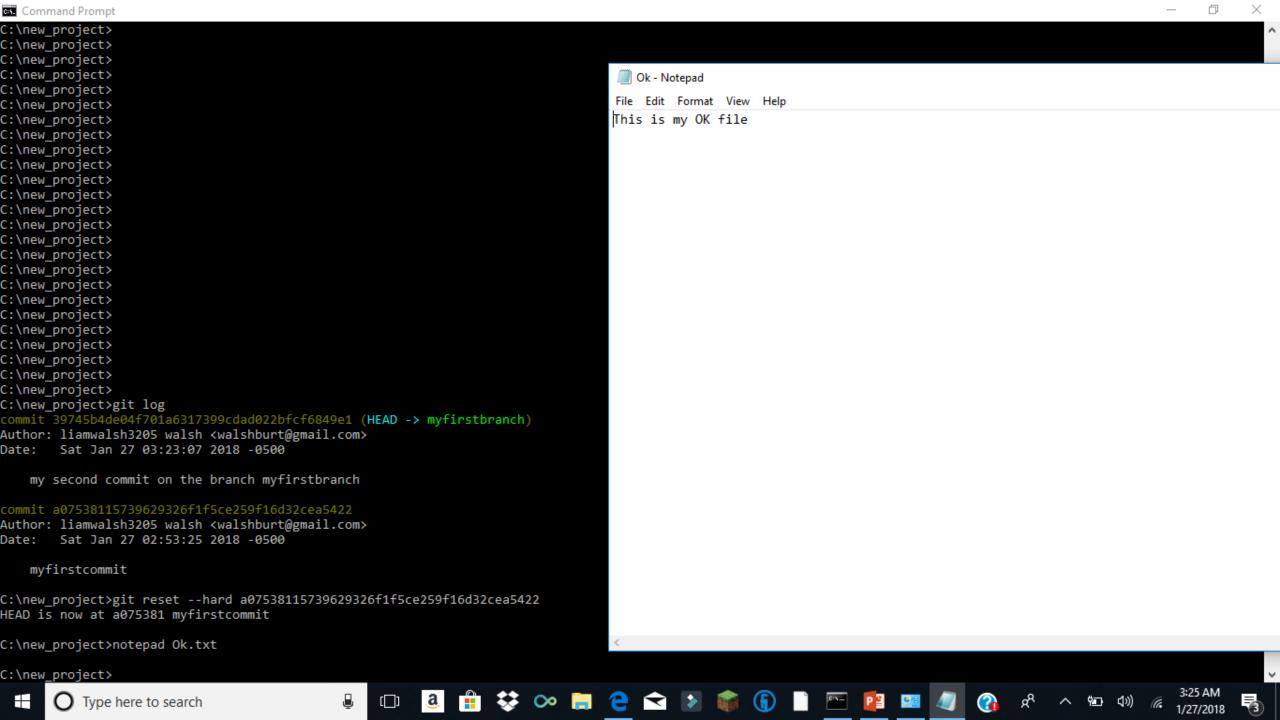
```
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>git log
commit 39745b4de04f701a6317399cdad022bfcf6849e1 (HEAD -> myfirstbranch)
Author: liamwalsh3205 walsh <walshburt@gmail.com>
Date:   Sat Jan 27 03:23:07 2018 -0500

    my second commit on the branch myfirstbranch

commit a07538115739629326f1f5ce259f16d32cea5422
Author: liamwalsh3205 walsh <walshburt@gmail.com>
Date:   Sat Jan 27 02:53:25 2018 -0500

    myfirstcommit

C:\new_project>notepad Ok.txt

C:\new_project>
```

Ok - Notepad

File Edit Format View Help

```
This is my OK file
More Text
```

Viewing the actual changes

- (git log –p) shows the actual file changes

```
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>git log -p
commit 8e8bad57bd1b052bf224951cd231e7500e9f7424 (HEAD -> myfirstbranch)
Author: liamwalsh3205 walsh <walshburt@gmail.com>
Date:   Sat Jan 27 03:12:59 2018 -0500

    my second commit on the branch myfirstbranch

diff --git a/Ok.txt b/Ok.txt
index 5d8f52b..78faed2 100644
--- a/Ok.txt
+++ b/Ok.txt
@@ -1 +1,2 @@
-This is my OK file
\ No newline at end of file
+This is my OK file
+More Text
\ No newline at end of file

commit a07538115739629326f1f5ce259f16d32cea5422
Author: liamwalsh3205 walsh <walshburt@gmail.com>
Date:   Sat Jan 27 02:53:25 2018 -0500

    myfirstcommit

diff --git a/Ok.txt b/Ok.txt
new file mode 100644
index 0000000..5d8f52b
--- /dev/null
+++ b/Ok.txt
@@ -0,0 +1 @@
+This is my OK file
\ No newline at end of file

C:\new_project>
```

Rolling back a change

- (git log) will show the commits
- (git reset <commit>) will put the HEAD pointer of the current branch at the indicated commit
- The HEAD pointer just indicates which commit is the most recent

```
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>
C:\new_project>git log
commit 39745b4de04f701a6317399cdad022bfcf6849e1 (HEAD -> myfirstbranch)
Author: liamwalsh3205 walsh <walshburt@gmail.com>
Date:   Sat Jan 27 03:23:07 2018 -0500

    my second commit on the branch myfirstbranch

commit a07538115739629326f1f5ce259f16d32cea5422
Author: liamwalsh3205 walsh <walshburt@gmail.com>
Date:   Sat Jan 27 02:53:25 2018 -0500

    myfirstcommit

C:\new_project>git reset --hard a07538115739629326f1f5ce259f16d32cea5422
HEAD is now at a075381 myfirstcommit

C:\new_project>notepad Ok.txt

C:\new_project>
```

Ok - Notepad

File   Edit   Format   View   Help

This is my OK file

Type here to search

3:25 AM
1/27/2018

Review

- Repositories contain branches (git init creates a repo in a directory)
- Branches contain commits
- The HEAD of a branch is the most recent commit
- You can create branches (git checkout –b <branch name>)
- You can switch between branches (git checkout <branch name>)
- You add (stage) changed files, removed file to a commit and execute the commit against the branch (git commit –m " my message")
- Commits store deltas against the branch

Remotes and Using GIT with a team

- A remote repo is just a repository that your local repo knows about
- Normally the master code for the group is stored on the remote server in a remote repo
- You fetch code/branches/commits from the remote
- You push you local changes to the remote
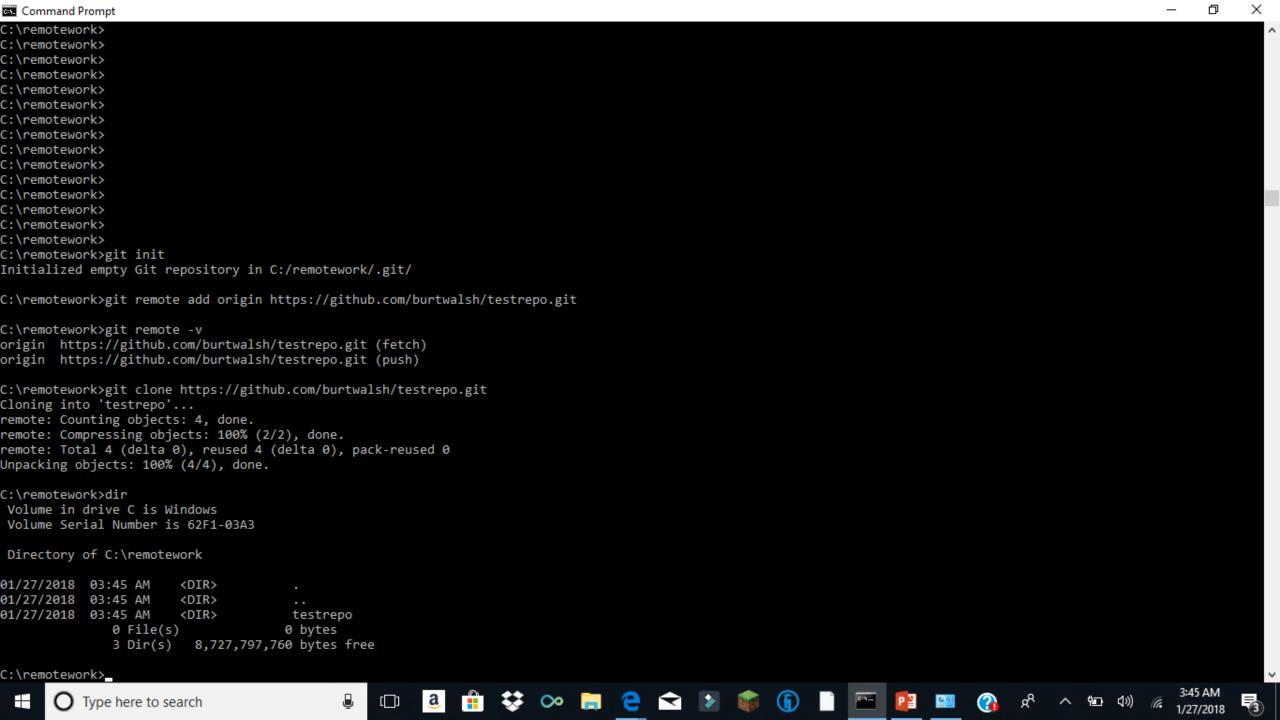- Merges (merge pull request) occur on the remote

Remote repo

- Let us assume an admin created a remote repo (project) for us
- The repo has a master branch
- In our case the branch has one commit which added two files

C:\remotewor
C:\remotewor
C:\remotewor
C:\remotewor
C:\remotewor
C:\remotewor
C:\remotewor
C:\remotewor
C:\remotewor
C:\remotewor
C:\remotewor
C:\remotewor
C:\remotewor
C:\remotewor
C:\remotewor
C:\remotewor
C:\remotewor
C:\remotewor
C:\remotewor
C:\remotewor
C:\remotewor
C:\remotewor
C:\remotewor
C:\remotewor
C:\remotewor
C:\remotewor
C:\remotewor
C:\remotewor
C:\remotewor
Initialized

C:\remotewor

C:\remotewor

C:\remotewor
origin   http
origin   http

C:\remotewor

burtwalsh/testrepo

GitHub, Inc. [US] https://github.com/burtwalsh/testrepo

To see favorites here, select ⭐ then ☆, and drag to the Favorites Bar folder. Or import from another browser. **Import favorites**

This repository | Search

Pull requests   Issues   Marketplace   Explore

burtwalsh / **testrepo**

👁 Watch ▾ 0   ★ Star 0   ⑂ Fork 0

<> Code | ⓘ Issues 0 | Pull requests 0 | Projects 0 | 📖 Wiki | 📊 Insights | ⚙ Settings

No description, website, or topics provided.

Add topics

Edit

⊙ 1 commit   1 branch   ◌ 0 releases   👥 1 contributor

Branch: master ▾ | New pull request

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

burtwalsh my first commit

Clone with HTTPS ⑦   Use SSH

Use Git or checkout with SVN using the web URL.

One.txt   my first commit

https://github.com/burtwalsh/testrepo.git   📋

Two.txt   my first commit

Open in Desktop | Download ZIP

Help people interested in this repository understand your project by adding a README.

© 2018 GitHub, Inc.   Terms   Privacy   Security   Status   Help   Contact GitHub   API   Training   Shop   Blog   About

Type here to search

3:43 AM
1/27/2018

Working Locally with the Remote

- We need to create a local repository (repo)

- We need to set the remote repo with which we will exchange changes (commits) branches (git remote add origin https://github.com/burtwalsh/testrepo)

- We need to clone (copy all code, branches and associated commits) from the remote to our local repo (https://github.com/burtwalsh/testrepo)

- No (git init) needed in this case!

```
Command Prompt

C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>git init
Initialized empty Git repository in C:/remotework/.git/

C:\remotework>git remote add origin https://github.com/burtwalsh/testrepo.git

C:\remotework>git remote -v
origin  https://github.com/burtwalsh/testrepo.git (fetch)
origin  https://github.com/burtwalsh/testrepo.git (push)

C:\remotework>git clone https://github.com/burtwalsh/testrepo.git
Cloning into 'testrepo'...
remote: Counting objects: 4, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), done.

C:\remotework>dir
 Volume in drive C is Windows
 Volume Serial Number is 62F1-03A3

 Directory of C:\remotework

01/27/2018  03:45 AM    <DIR>          .
01/27/2018  03:45 AM    <DIR>          ..
01/27/2018  03:45 AM    <DIR>          testrepo
               0 File(s)              0 bytes
               3 Dir(s)   8,727,797,760 bytes free

C:\remotework>
```
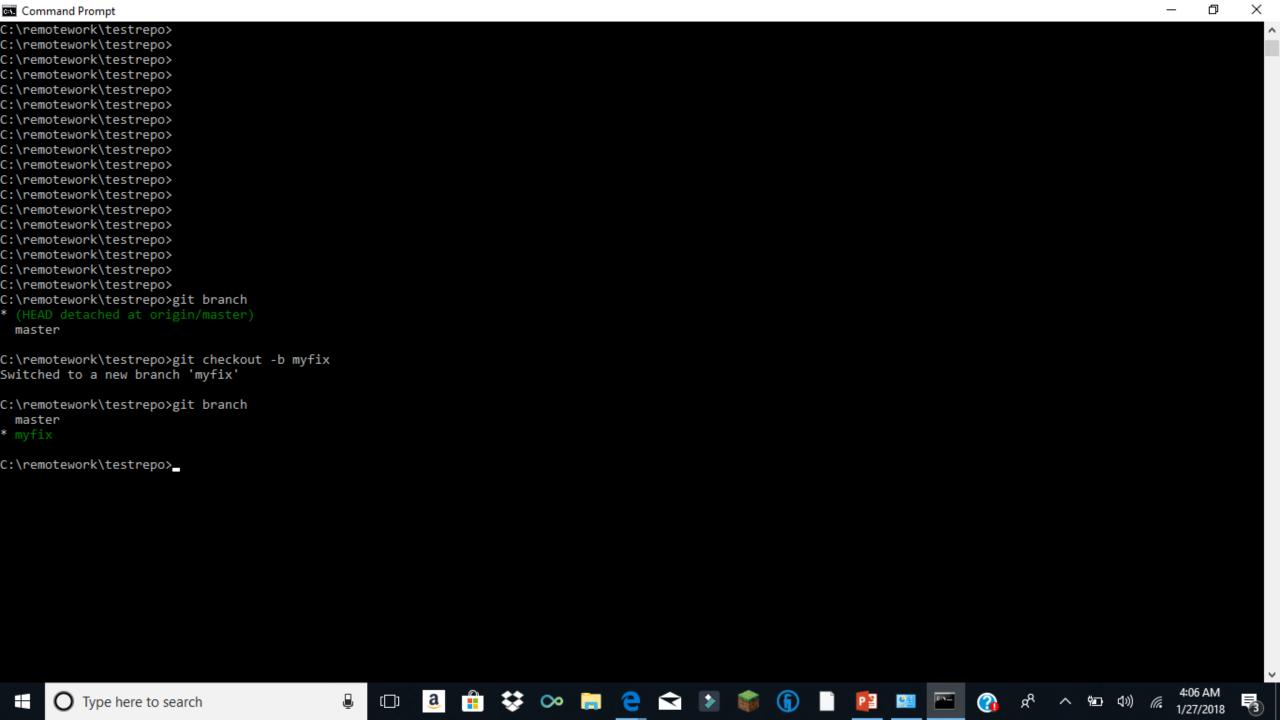
Looking at the remote

- We see a subdirectory called testrepo
- This directory has a .git subdirectory
- Checkout the origin/master which is a branch from the remote that we now have locally
- HEAD is at the most recent commit (that we know about) for the remote master branch

```
Command Prompt                                                                    ☐  ⧉  ✕

C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>
C:\remotework>cd testrepo

C:\remotework\testrepo>dir
 Volume in drive C is Windows
 Volume Serial Number is 62F1-03A3

 Directory of C:\remotework\testrepo

01/27/2018  03:45 AM    <DIR>          .
01/27/2018  03:45 AM    <DIR>          ..
01/27/2018  03:45 AM                 3 One.txt
01/27/2018  03:45 AM                 3 Two.txt
               2 File(s)              6 bytes
               2 Dir(s)   8,727,789,568 bytes free

C:\remotework\testrepo>git checkout origin/master
Note: checking out 'origin/master'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

  git checkout -b <new-branch-name>

HEAD is now at 77e4f21 my first commit

C:\remotework\testrepo>
```
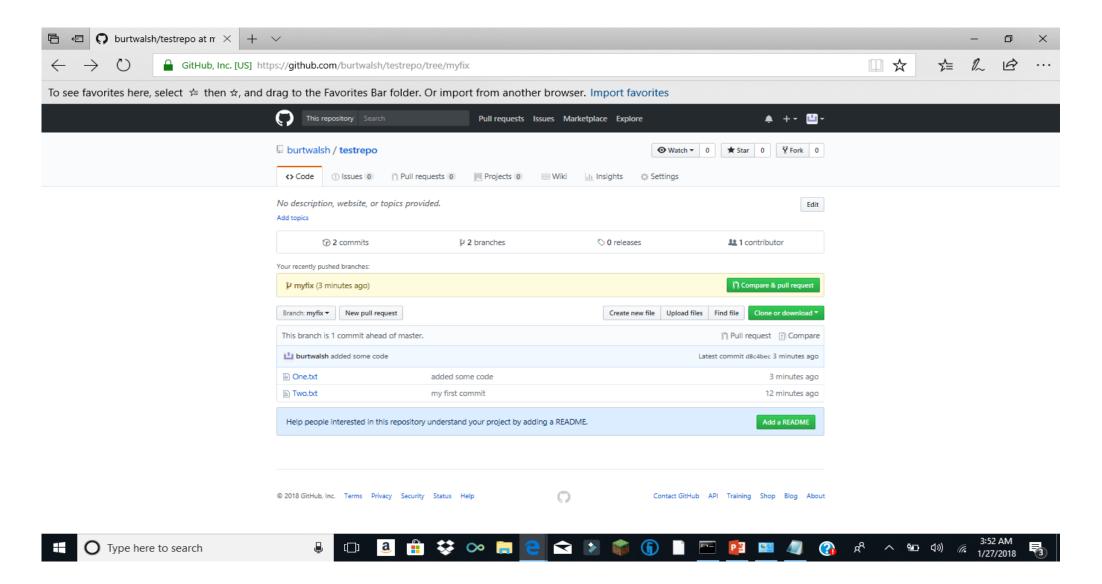
Create a Local Working branch

- (git branch) now shows the remote branches
- (git checkout –b myfix) creates a local branch called myfix

```
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>git branch
* (HEAD detached at origin/master)
  master

C:\remotework\testrepo>git checkout -b myfix
Switched to a new branch 'myfix'

C:\remotework\testrepo>git branch
  master
* myfix

C:\remotework\testrepo>
```

Updating and pushing changes

- We will update the fetched One.txt file
- We will (git add One.txt) the changes
- We will commit (git commit –m "add some code") to our LOCAL myfix branch
- We will push the changes to the remote (origin) as a remote branch myfix
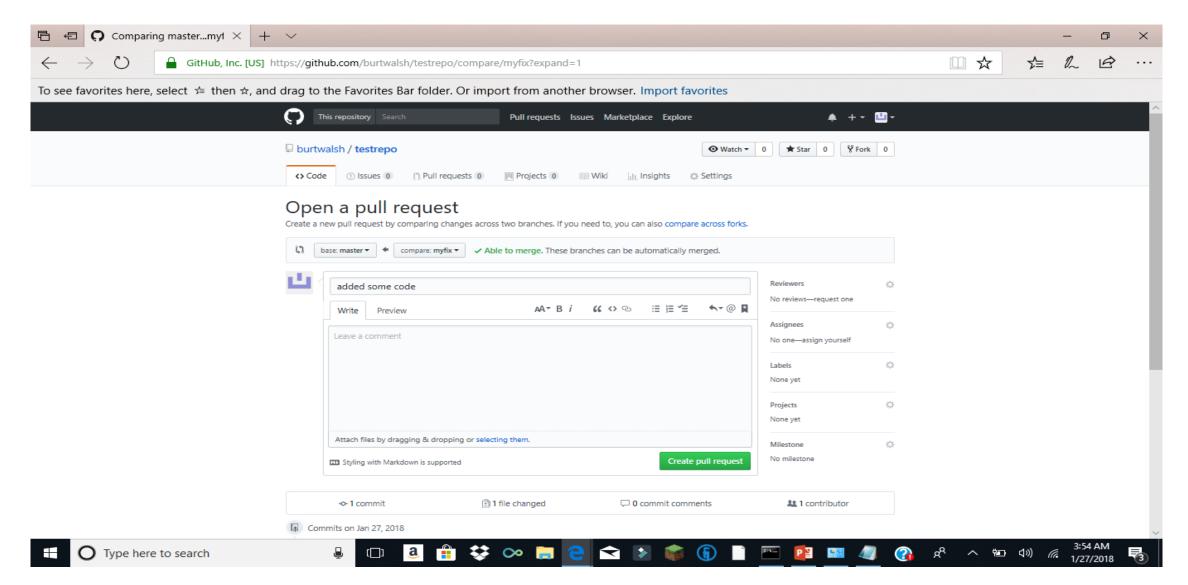- If the branch does not exist it will be created

```
Command Prompt

C:\remotework\testrepo>dir
 Volume in drive C is Windows
 Volume Serial Number is 62F1-03A3

 Directory of C:\remotework\testrepo

01/27/2018  03:45 AM    <DIR>          .
01/27/2018  03:45 AM    <DIR>          ..
01/27/2018  03:45 AM                 3 One.txt
01/27/2018  03:45 AM                 3 Two.txt
               2 File(s)              6 bytes
               2 Dir(s)   8,727,789,568 bytes free

C:\remotework\testrepo>notepad One.txt

C:\remotework\testrepo>git status
On branch myfix
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:   One.txt

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

C:\remotework\testrepo>git add One.txt

C:\remotework\testrepo>git commit -m "added some code"
[myfix d8c4bec] added some code
 1 file changed, 2 insertions(+), 1 deletion(-)

C:\remotework\testrepo>git push origin myfix
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 342 bytes | 68.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/burtwalsh/testrepo.git
 * [new branch]      myfix -> myfix

C:\remotework\testrepo>notepad One.txt

C:\remotework\testrepo>
```

One - Notepad

File Edit Format View Help

One
I added this Locally GOING TO PUSH it to the remote in a new branch
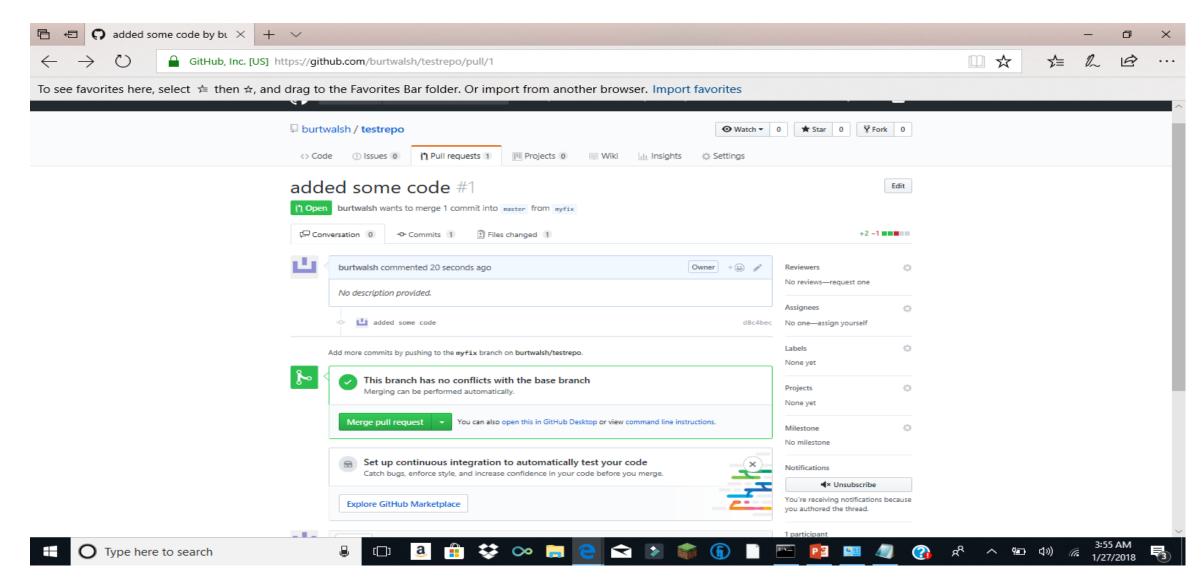
New branch on the remote

A Merge Pull Request

- A merge pull request is associated with a merge between two branches

- You select the base branch (on the remote) that you are going to merge your changes into

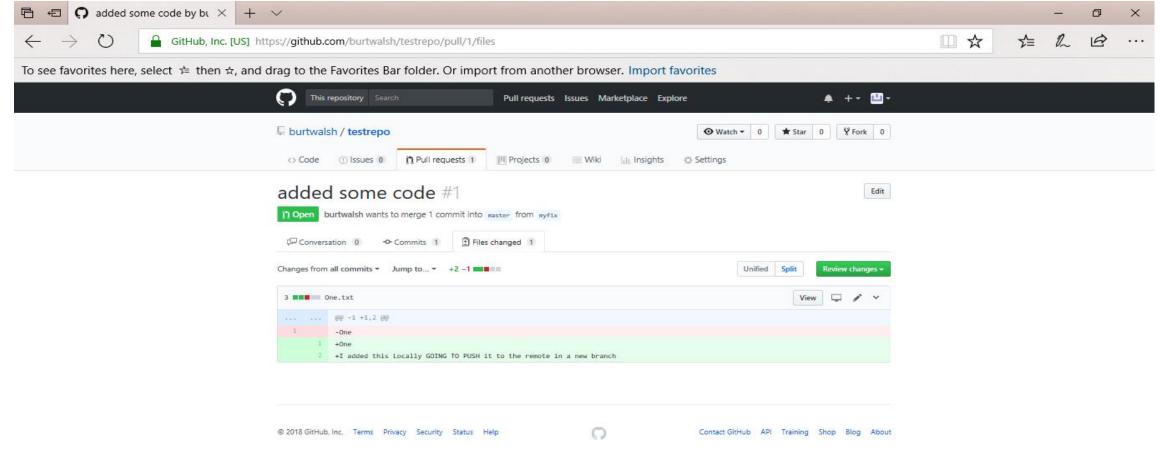- Your changes came up from the Local when you did the (git push)

Click Compare and Pull

Click Create Merge pull Request
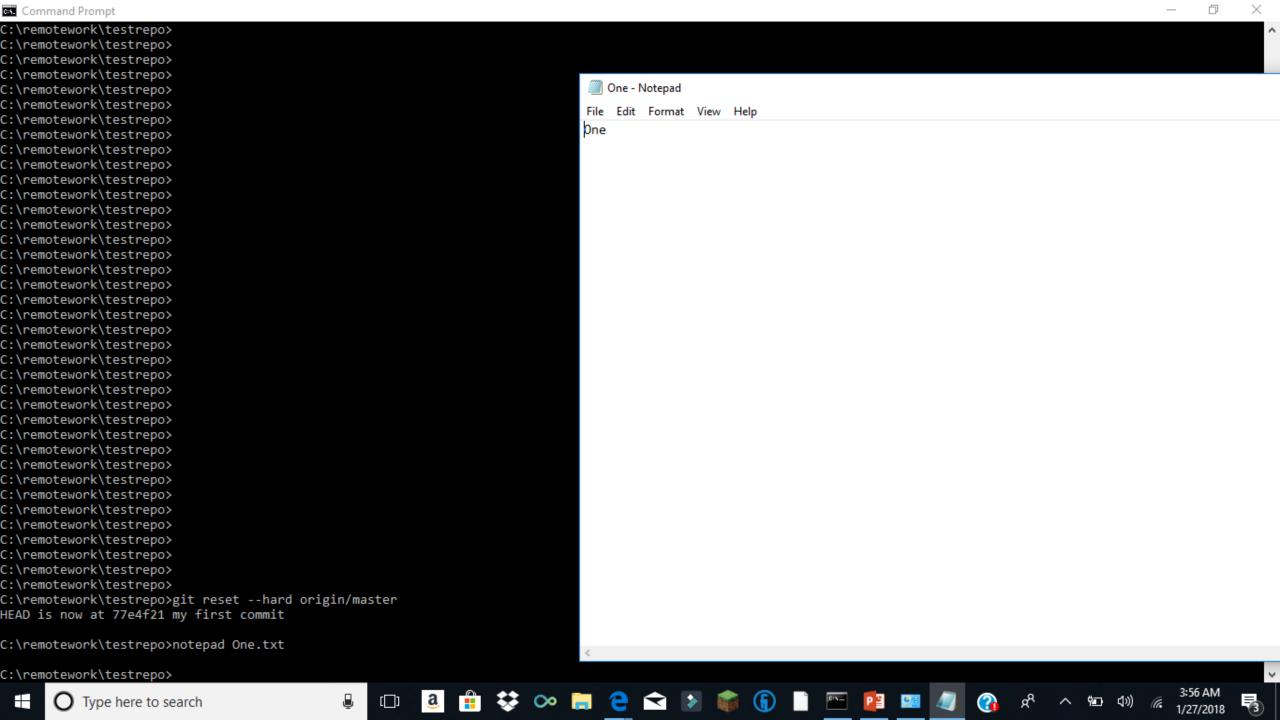
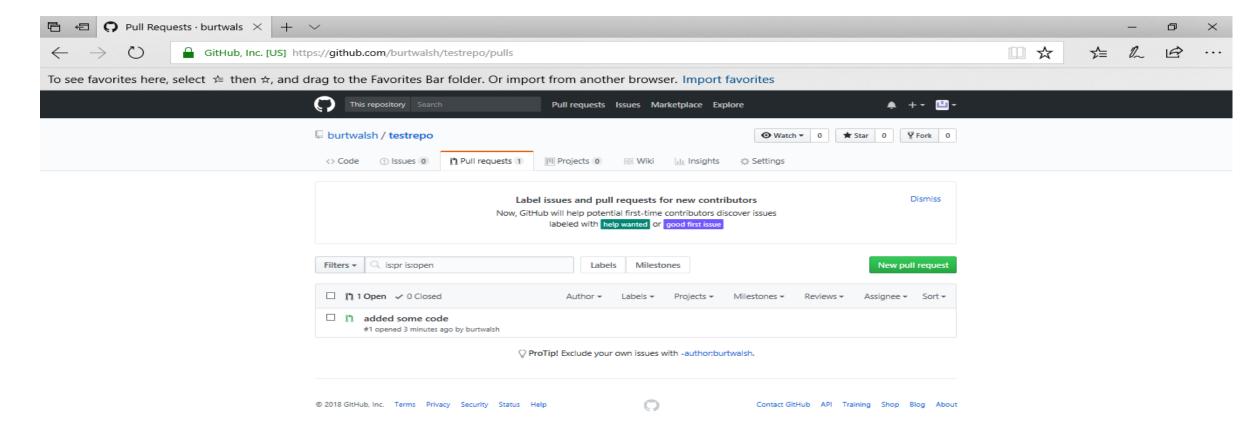# Looking at What is to be Merged

Back at the Local Repo

- On the remote we have not merged YET
- You could (git fetch)
- (git reset –hard origin/master) to see the current state of the remote master branch
- Look at the One.txt file and see it has not yet been merged at the remote
- REMEMBER THIS WHOLE PROCESS STARTED FROM THE REMOTE MASTER

Command Prompt

```
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>git reset --hard origin/master
HEAD is now at 77e4f21 my first commit

C:\remotework\testrepo>notepad One.txt

C:\remotework\testrepo>
```

One - Notepad

File   Edit   Format   View   Help

One

Type here to search

3:56 AM
1/27/2018

Back at the Remote

- Another user will review and hopefully MERGE my changes!
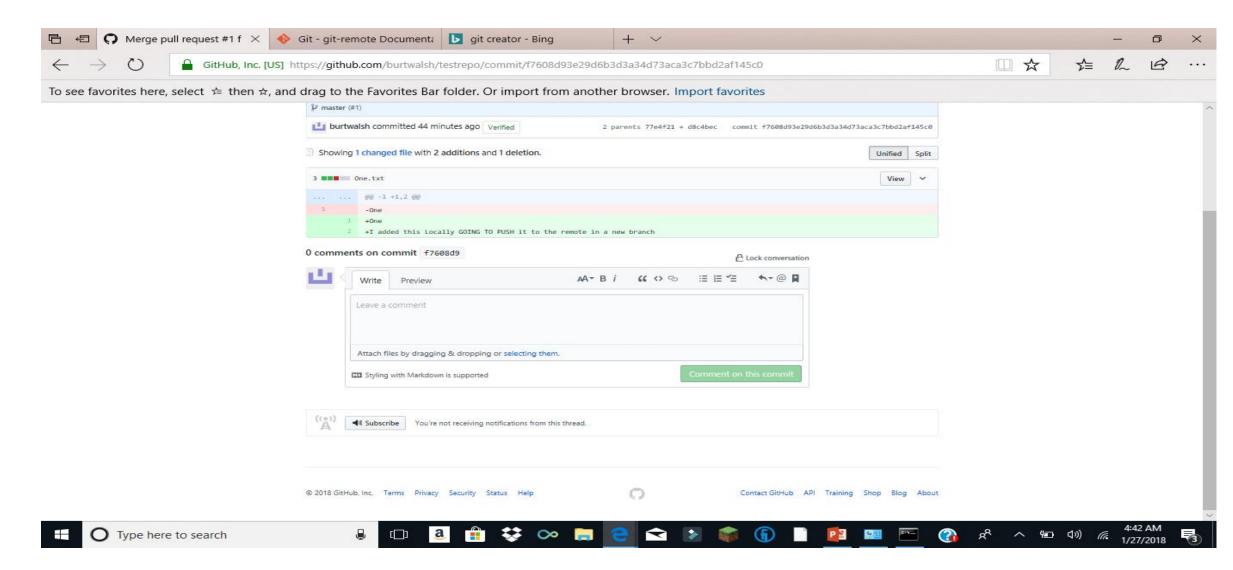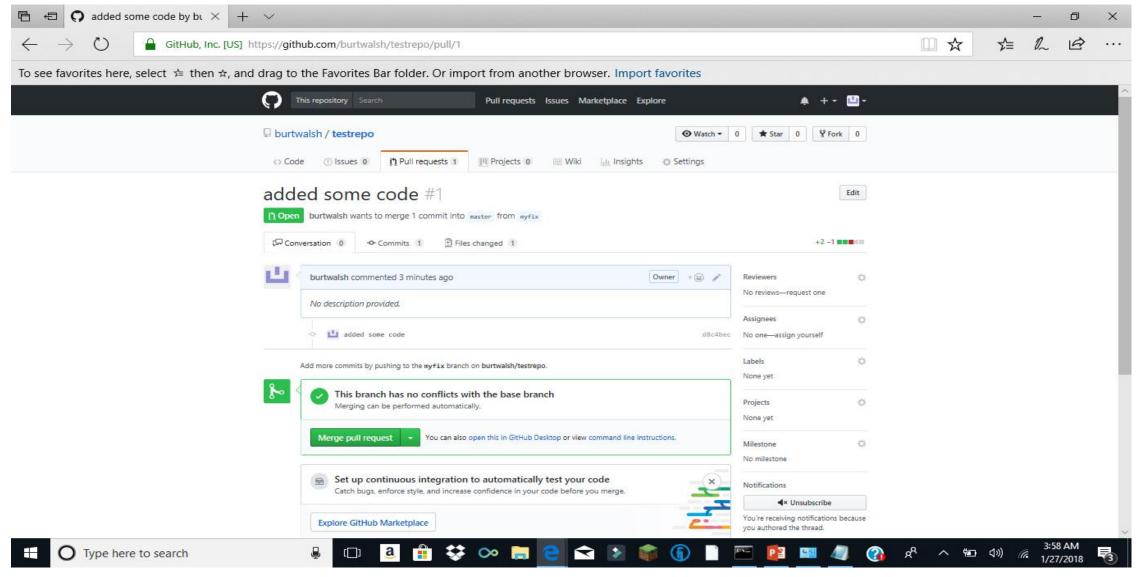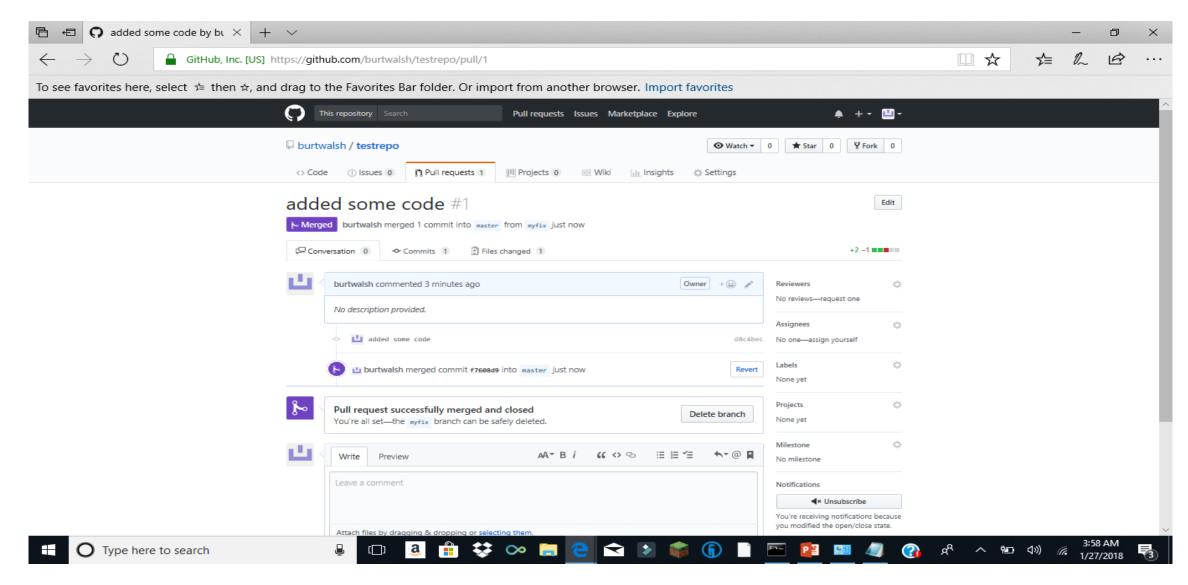
At the Remote Click Pull Requests

Review Code

Merge Pull Request

Merged State

Fetch all Changes to the Local Repo

- When working with the remote you need to (git fetch) to ensure all changes that have occurred at the remote are brought down to the local repo

- MANY PEOPLE WILL BE UPDATING THE REMOTE FROM THEIR LOCAL REPOS

- After the FETCH you have a copy of all the code, branches and associated commits locally (You are in sync!)

- (git checkout origin/master)

- View the updated and merged One.txt file

```
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>
C:\remotework\testrepo>git fetch
remote: Counting objects: 1, done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (1/1), done.
From https://github.com/burtwalsh/testrepo
   77e4f21..f7608d9  master     -> origin/master

C:\remotework\testrepo>git status
On branch myfix
nothing to commit, working tree clean

C:\remotework\testrepo>notepad One.txt

C:\remotework\testrepo>git checkout origin/master
Note: checking out 'origin/master'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

  git checkout -b <new-branch-name>

HEAD is now at f7608d9 Merge pull request #1 from burtwalsh/myfix

C:\remotework\testrepo>notepad One.txt

C:\remotework\testrepo>
```

One - Notepad

File Edit Format View Help

One
I added this Locally GOING TO PUSH it to the remote in a new branch

4:00 AM
1/27/2018

Why use GIT

- Git was created by **Linus Torvalds in 2005**
- **Linus created Linux (from Unix)**
- **Need to support distributed updates and collaboration of big projects**
- **Hard to keep deltas consistent so based upon diffs**