

GIT: From Beginner To Fearless

GIT Tagging Activity: Using tags to mark releases and specific commits

Brian Gorman, Author/Instructor/Trainer

©2019 - MajorGuidanceSolutions



Introduction

Tags are a great way to place an important milestone on a specific commit in a repository. Often, a release is marked with a tag. Additionally, tags might denote a specific feature implementation or even something like a bug fix.

There are two types of tags: Lightweight and Annotated. The major difference is in how they are stored behind the scenes and what can be displayed from the tag details. Both types are bookmarks to a specific commit, but the annotated tag lists information about the commit and committer, while the lightweight tag is more of just a pointer to a specific commit.

In this activity, we'll take a look at working with tags in our repositories.

Let's get started!

GFBTF: Git Tagging Activity

Step 1: Make sure you have a working repository.

- a) Either clone a repo or get the latest on master for a repo. Ideally, the repo would have a few commits in it at least, as well as no tags.

```
[git clone
```

```
https://github.com/majorguidancesolutions/SimpleActivityRepo  
TaggingDemo]
```

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder  
$ git clone https://github.com/majorguidancesolutions/SimpleActivityRepo Tagging  
Demo  
Cloning into 'TaggingDemo'...  
remote: Counting objects: 58, done.  
remote: Compressing objects: 100% (39/39), done.  
remote: Total 58 (delta 28), reused 47 (delta 17), pack-reused 0  
unpacking objects: 100% (58/58), done.
```

```
[cd TaggingDemo]  
[git checkout master]  
[git fetch origin]  
[git pull origin master]  
[git checkout -b TaggingDemo]
```

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder  
$ cd TaggingDemo/
```

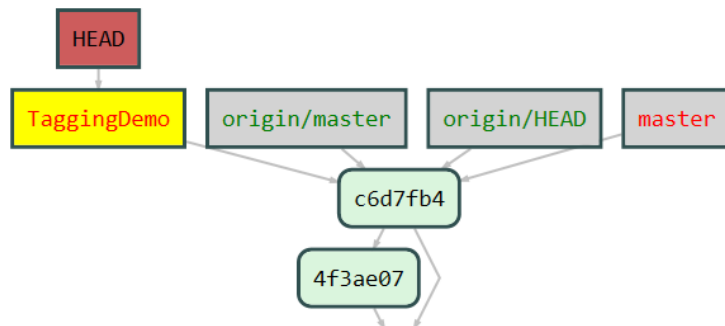
```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (master)  
$ git checkout master  
Already on 'master'  
Your branch is up-to-date with 'origin/master'.
```

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (master)  
$ git fetch origin
```

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (master)  
$ git pull origin master  
From https://github.com/majorguidancesolutions/SimpleActivityRepo  
* branch      master       -> FETCH_HEAD  
Already up-to-date.
```

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (master)  
$ git checkout -b TaggingDemo  
Switched to a new branch 'TaggingDemo'
```

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)  
$
```



Notes

Step 2: Commit and tag a few times, list tags

- a) Commit and create a lightweight tag, then list the tags on the repo

```
[code info.txt]
//make a change
[git commit -am "tagging demo commit 1"]
```

First we'll add a lightweight tag:

```
[git tag beginning-tag-demo] //creates a lightweight tag
[git tag] //lists tags
[git tag -l] //lists tags
[git tag --list] //lists tags
```

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git tag beginning-tag-demo

Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git tag
beginning-tag-demo

Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git tag -l
beginning-tag-demo

Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git tag --list
beginning-tag-demo
```

- b) Commit and create an annotated tag, then list the tags on the repo

```
[code info.txt]
//make a change
[git commit -am "tagging demo commit 2"]
[git tag v1.0.0.1]
```

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git commit -am "Tagging demo commit #2"
[TaggingDemo e161f83] Tagging demo commit #2
1 file changed, 1 insertion(+)

Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git tag -a v1.0.0.1
```

TAG_EDITMSG — Visual Studio Code

```
Edit Selection View Go Debug Tasks Help
TAG_EDITMSG x
1 Version #1 released 2017.09.26 02.00.00.000
2 #
3 # Write a message for tag:
4 # v1.0.0.1
5 # Lines starting with '#' will be ignored.
6
```

```
[code info.txt]
//make another change
[git commit -am "tagging demo commit 3"]
[git tag -a -m "version 1.0.0.2 released 2017.09.26 02:05:00.000" v1.0.0.2]

Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git tag -a -m "version 1.0.0.2 released 2017.09.26 02:05:00.000" v1.0.0.2
```

```
[git tag]
```

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git tag
beginning-tag-demo
v1.0.0.1
v1.0.0.2
```

```
[git log --oneline]
```

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git log --oneline
90a9cab (HEAD -> TaggingDemo, tag: v1.0.0.2) Tagging demo commit #3
e161f83 (tag: v1.0.0.1) Tagging demo commit #2
54821bf (tag: beginning-tag-demo) Tagging demo commit #1
c6d7fb4 (origin/master, origin/HEAD, master) Merge pull request #7 from majorguidancesolutions/feature-picked-branch
4f3ae07 CherryPickingActivity - commit #2
```

Step 3: Show tag info

Once we have commits tagged, we can actually use the tag just the same as we would use a commit id. This means we can check them out, show them, diff them, etc.

- a) Show tag info, see the difference between lightweight and annotated tags:

```
[git show v1.0.0.2] //annotated
```

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git show v1.0.0.2
tag v1.0.0.2
Tagger: Brian L. Gorman <blgorman@gmail.com>
Date: Tue Sep 26 01:59:53 2017 -0500

version 1.0.0.2 released 2017.09.26 02:05:00.000

commit 90a9cab6fe8c7523ef719bcb52b01e35249a50a1 (HEAD -> TaggingDemo, tag: v1.0.2)
Author: Brian L. Gorman <blgorman@gmail.com>
Date: Tue Sep 26 01:57:39 2017 -0500

    Tagging demo commit #3

diff --git a/info.txt b/info.txt
index 67a6f7c..6a3b104 100644
--- a/info.txt
+++ b/info.txt
@@ -23,5 +23,6 @@ Squash and Merge commit #9
    cherry pick merge #1
    cherry pick merge #2
```

```
[git show beginning-tag-demo] //lightweight
```

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git show beginning-tag-demo
commit 54821bf864da49add8feb19a07f31df2e833539f (tag: beginning-tag-demo)
Author: Brian L. Gorman <blgorman@gmail.com>
Date: Tue Sep 26 01:50:35 2017 -0500

    Tagging demo commit #1

diff --git a/info.txt b/info.txt
index 0a7e1d4..abd6b6f 100644
--- a/info.txt
+++ b/info.txt
@@ -22,3 +22,5 @@ Squash and Merge commit #9
    cherry pick merge #1
    cherry pick merge #2
+
+Tagging Demo 1
```

b) Checkout a tag

```
[git checkout v1.0.0.1]
```

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git checkout v1.0.0.1
Note: checking out 'v1.0.0.1'.

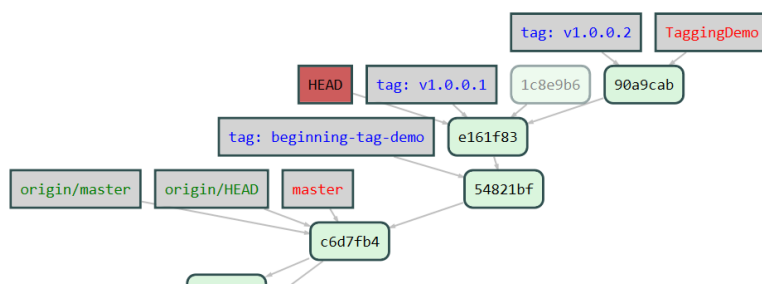
You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

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 e161f83... Tagging demo commit #2
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo ((v1.0.0.1))
```

The checkout happens in a detached head state. When that happens, we can look around and do stuff, but if we want to use it for a commit we need to then checkout a branch and commit on that branch.



Notice the head is pointing to `e161f83`, where tag for `v1.0.0.1` is also pointing.

```
[git checkout TaggingDemo]
```

Step 4: Use expressions/wildcards to list specific tags

a) Get all the tags with v1 in the tag

```
[git tag -l "v1.*"]
```

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git tag -l "v1.*"
v1.0.0.1
v1.0.0.2
```

Step 5: Create tags on previous commits

Need to have some commits in the history. If not enough, create two or three commits so that a couple of them don't have tags. Find a commit without a tag on it

a) Create a lightweight tag on a previous commit

```
[git log --oneline]
```

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git log --oneline
90a9cab (HEAD -> TaggingDemo, tag: v1.0.0.2) Tagging demo commit #3
e161f83 (tag: v1.0.0.1) Tagging demo commit #2
54821bf (tag: beginning-tag-demo) Tagging demo commit #1
c6d7fb4 (origin/master, origin/HEAD, master) Merge pull request #7 from majorguidancesolutions/feature-picked-branch
4f3ae07 CherryPickingActivity - commit #2
59a4cde Update readme.txt
3fd06fc Squash and merge feature2 (#6)
fa75127 Merge pull request #5 from majorguidancesolutions/SquashAndMergeFeature
3741e4f A single new commit on feature
728f97e Merge branch 'master' into SquashAndMergeFeature
5449a4a Squash and merge feature (#4)
729bc24 Squash And Merge Commit#4
c2d0c0d Squash And Merge Commit#3
a0b8ada Squash And Merge Commit#2
7ab5f41 Squash And Merge Commit#1
1a3444a Merge pull request #3 from majorguidancesolutions/GitAmendDemo
0cac76b changed info.txt and added readme.txt
```

Here there are plenty of candidates. I'm going to put a lightweight tag on 728f97e for squash and merge completed

```
[git tag squash-and-merge-completed 728f97e]
```

```
[git tag]
```

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git tag squash-and-merge-completed 728f97e

Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git tag
beginning-tag-demo
squash-and-merge-completed
v1.0.0.1
v1.0.0.2
```

```
[git log --oneline]
```

```
3741e4f Squash and merge feature2 (#6)
fa75127 Merge pull request #5 from majorguidancesolutions/SquashAndMergeFeature
3741e4f A single new commit on feature
728f97e (tag: squash-and-merge-completed) Merge branch 'master' into SquashAndMergeFeature
5449a4a Squash and merge feature (#4)
729bc24 Squash And Merge Commit#4
```

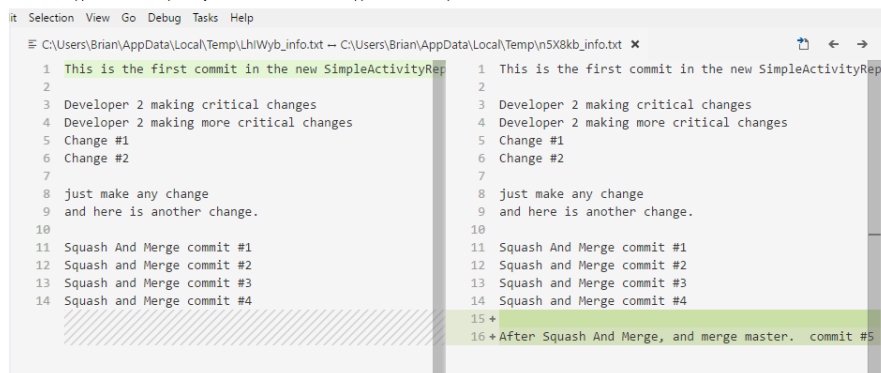
```
[git show squash-and-merge-completed]
```

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git show squash-and-merge-completed
commit 728f97ed574a90737bd6b6a4ed51ca3d3b995e6b (tag: squash-and-merge-completed)
Merge: 729bc24 5449a4a
Author: Brian L. Gorman <blgorman@gmail.com>
Date: Mon Sep 25 00:42:51 2017 -0500

    Merge branch 'master' into SquashAndMergeFeature
```

```
[git difftool squash-and-merge-completed 3741e4f]
```

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git difftool squash-and-merge-completed 3741e4f
```



b) Create an annotated tag on a previous commit

```
[git log --oneline]
```

```
e161f85 (tag: v1.0.0.1) Tagging demo commit #2
54821bf (tag: beginning-tag-demo) Tagging demo commit #1
c6d7fb4 (origin/master, origin/HEAD, master) Merge pull request #7 from majorguidancesolutions/feature-picked-branch
4f3ae07 CherryPickingActivity - commit #2
59a4cde Update readme.txt
3fd06fc Squash and merge feature2 (#6)
fa75127 Merge pull request #5 from majorguidancesolutions/SquashAndMergeFeature
3741e4f A single new commit on feature
728f97e (tag: squash-and-merge-completed) Merge branch 'master' into SquashAndMergeFeature
5449a4a Squash and merge feature (#4)
729bc24 Squash And Merge Commit#4
```

How about fa75127 this time.

```
[git tag -a -m "Code Review Completed" code-review-completed fa75127]
```

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git tag -a -m "Code Review Completed" code-review-completed fa75127
```

```
[git tag]
```

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git tag
beginning-tag-demo
code-review-completed
squash-and-merge-completed
v1.0.0.1
v1.0.0.2
```



```
[git show code-review-completed]
```

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git show code-review-completed
tag code-review-completed
Tagger: Brian L. Gorman <blgorman@gmail.com>
Date: Tue Sep 26 02:25:48 2017 -0500

Code Review Completed

commit fa751272965cd550ac657226e70a47a180cda3b0 (tag: code-review-completed)
Merge: 5449a4a 3741e4f
Author: majorguidancesolutions <brian@majorguidancesolutions.com>
Date: Mon Sep 25 00:57:05 2017 -0500

Merge pull request #5 from majorguidancesolutions/SquashAndMergeFeature

Squash and merge feature
```

Step 6: Delete a tag locally

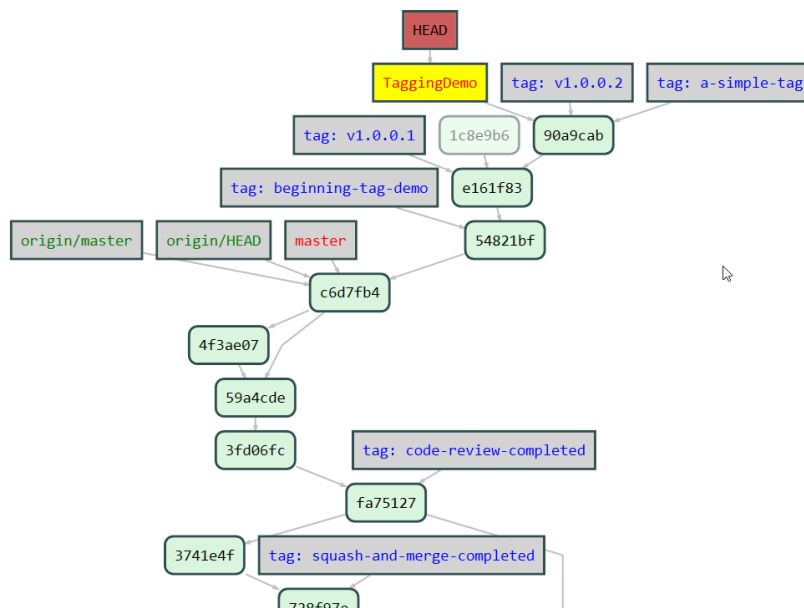
- a) Create a simple tag then delete it

```
[git tag a-simple-tag]
```

```
[git tag]
```

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git tag a-simple-tag

Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git tag
a-simple-tag
beginning-tag-demo
code-review-completed
squash-and-merge-completed
v1.0.0.1
v1.0.0.2
```



```
[git log --oneline]
```

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git log --oneline
90a9cab (HEAD -> TaggingDemo, tag: v1.0.0.2, tag: a-simple-tag) Tagging demo com
mit #3
e161f83 (tag: v1.0.0.1) Tagging demo commit #2
54821bf (tag: beginning-tag-demo) Tagging demo commit #1
c6d7fb4 (origin/master, origin/HEAD, master) Merge pull request #7 from majorgui
dancesolutions/feature-picked-branch
```

Looks like we have two tags on commit 90a9cab now...

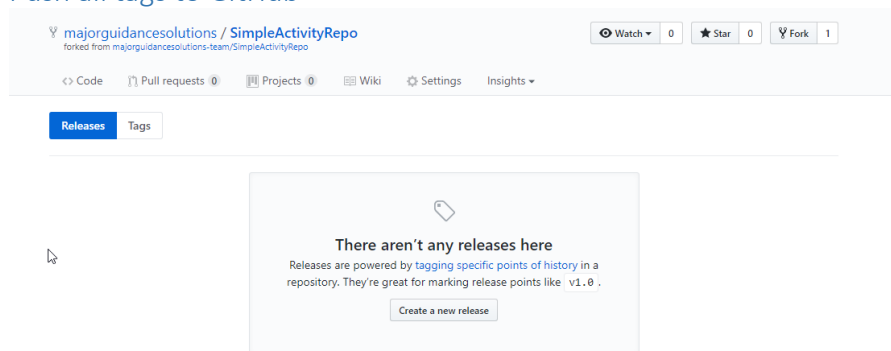
```
[git tag -d a-simple-tag]
```

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git tag -d a-simple-tag
Deleted tag 'a-simple-tag' (was 90a9cab)
```

Step 7: Working with Tags at GitHub

We need to be able to push our tags, as well as delete tags that are pushed.

a) Push all tags to GitHub



Right now, there are 0 releases at GitHub, and 0 tags. Note that we can create a tag right at GitHub with the button above, “Create a new release” If we did this, we could then get our local repository up to date with tags from REMOTE with

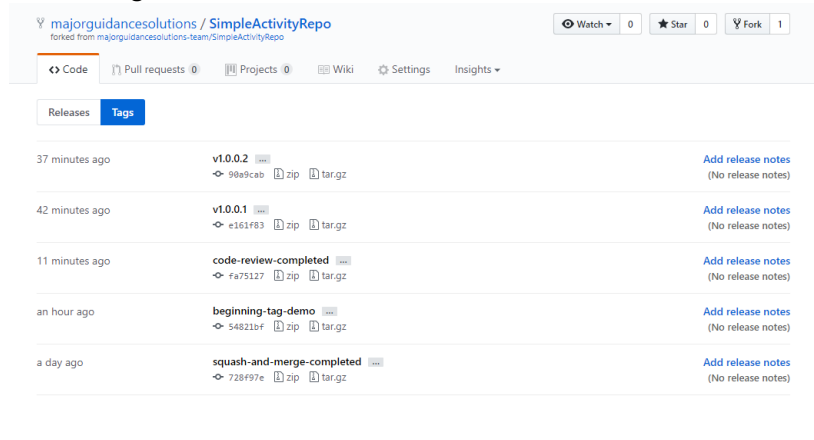
```
[git fetch --tags]
```

However, we aren’t going to worry about that. Instead, let’s push our tags

```
[git push --tags]
```

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git push --tags
Counting objects: 12, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (12/12), done.
Writing objects: 100% (12/12), 1.26 KiB | 0 bytes/s, done.
Total 12 (delta 6), reused 0 (delta 0)
remote: Resolving deltas: 100% (6/6), completed with 2 local objects.
To https://github.com/majorguidancesolutions/SimpleActivityRepo
 * [new tag]          beginning-tag-demo -> beginning-tag-demo
 * [new tag]          code-review-completed -> code-review-completed
 * [new tag]          squash-and-merge-completed -> squash-and-merge-completed
 * [new tag]          v1.0.0.1 -> v1.0.0.1
 * [new tag]          v1.0.0.2 -> v1.0.0.2
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$
```

And looking at GitHub:



majorguidancesolutions / SimpleActivityRepo

forked from majorguidancesolutions-team/SimpleActivityRepo

Code Pull requests 0 Projects 0 Wiki Settings Insights

Releases Tags

37 minutes ago	v1.0.0.2	98a9cab	zip	tar.gz	Add release notes (No release notes)
42 minutes ago	v1.0.0.1	e161f83	zip	tar.gz	Add release notes (No release notes)
11 minutes ago	code-review-completed	f875127	zip	tar.gz	Add release notes (No release notes)
an hour ago	beginning-tag-demo	54821bf	zip	tar.gz	Add release notes (No release notes)
a day ago	squash-and-merge-completed	728f97e	zip	tar.gz	Add release notes (No release notes)

The cool thing to note is that you can get a download of the repo at any of the release points.

b) Set config to always push tags [Optional]

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git config --global push.followTags true
```

c) Delete a tag from remote

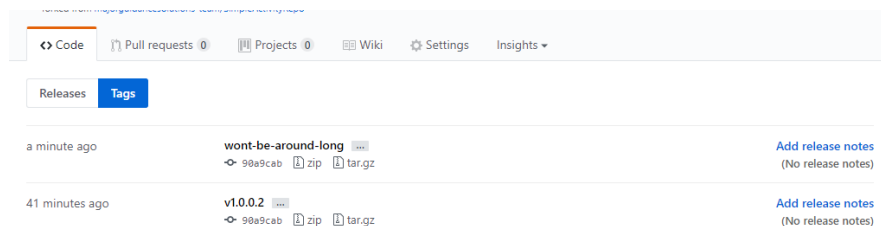
```
[git tag -a -m "This Tag is going to go away" wont-be-around-long]
```

```
[git push --tags]
```

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git tag -a -m "This Tag is going to go away" wont-be-around-long

Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git push --tags
Counting objects: 1, done.
Writing objects: 100% (1/1), 187 bytes | 0 bytes/s, done.
Total 1 (delta 0), reused 0 (delta 0)
To https://github.com/majorguidancesolutions/simpleActivityRepo
 * [new tag]          wont-be-around-long -> wont-be-around-long

Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$
```



majorguidancesolutions / SimpleActivityRepo

forked from majorguidancesolutions-team/SimpleActivityRepo

Code Pull requests 0 Projects 0 Wiki Settings Insights

Releases Tags

a minute ago	wont-be-around-long	98a9cab	zip	tar.gz	Add release notes (No release notes)
41 minutes ago	v1.0.0.2	98a9cab	zip	tar.gz	Add release notes (No release notes)

```
[git push origin :wont-be-around-long]
```

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git push origin :wont-be-around-long
To https://github.com/majorguidancesolutions/simpleActivityRepo
- [deleted]          wont-be-around-long

Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$
```

Code Pull requests 0 Projects 0 Wiki Settings Insights		
Releases Tags		
43 minutes ago	v1.0.0.2 <div> 98a9cab zip tar.gz </div>	Add release notes (No release notes)
an hour ago	v1.0.0.1 <div> e161f83 zip tar.gz </div>	Add release notes (No release notes)

Even though we removed from origin, we still need to delete locally

```
[git tag]
```

```
[git tag -d wont-be-around-long]
```

```
Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git tag
beginning-tag-demo
code-review-completed
squash-and-merge-completed
v1.0.0.1
v1.0.0.2
wont-be-around-long

Brian@SENTINEL MINGW64 /g/Data/GFBTF/DemoFolder/TaggingDemo (TaggingDemo)
$ git tag -d wont-be-around-long
Deleted tag 'wont-be-around-long' (was 8ed11c7)
```

This concludes our tagging activity.

Closing Thoughts

In this activity, we took a look at creating tags on our repository. There are two different types of tags, annotated and lightweight. Both can be useful, but if we want to tag a major release for public knowledge we should use the more verbose annotated tag. The lightweight tag is great for private use or simple pointers to commits along the way.

Just like a commit id in GIT, a tag can be interacted with to get information about the commit, differences between commits, and even checked out to a branch for further development.

Adding and getting tags from the public repo requires using push and pull with the --tags flag. Deleting from a public repository is much like deleting a branch from a public repository, by pushing with a : (colon) in front of the tag name.

The really cool thing at GitHub is that tags allow us to be download the repo as it was at the state of that tag directly for release/deploy.

Take a few minutes to make some notes about the various commands we've learned about in this activity, and practice using them.

Notes