Git Tag

Git has the ability to tag specific points in history as being important. Typically people use this functionality to mark release points (v1.0, and so on). In this section, you’ll learn how to list the available tags, how to create new tags, and what the different types of tags are.

# Creating Tags:

Git supports two types of tags: 1. Lightweight 2. Annotated.

**Annotated :** Annotated tags, however, are stored as full objects in the Git database. They’re checksummed; contain the tagger name, email, and date; have a tagging message.

Creating an annotated tag in Git is simple. The easiest way is to specify **-a** when you run the tag command

$ git tag -a v1.0 -m "my release version 1.0"



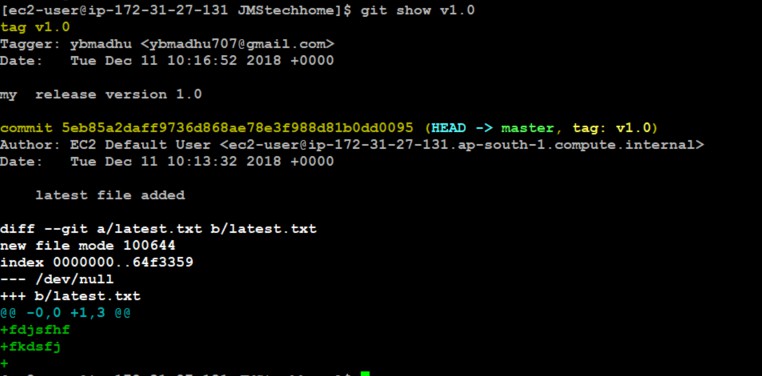
Check the list of tags use

$ git log

you can see the tag data along with the commit that was tagged by using the **git show**

command:

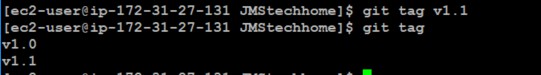
$ git show v1.0



lightweight: A lightweight tag is very much like a branch that doesn’t change — it’s just a pointer to a specific commit.

This is basically the commit checksum stored in a file — no other information is kept. To create a lightweight tag, don’t supply any of the -a, -s, or -m options, just provide a tag name.

$ git tag v1.1

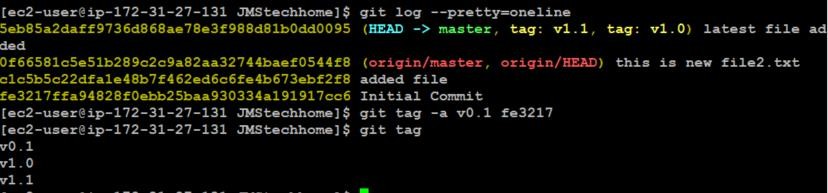


# Tagging Later:

Take hash code using git log git tag v0.0.1 3f42f2d

You can also tag commits after you’ve moved past them. Suppose your commit history looks like this:

$ git log --pretty=oneline



# Sharing Tags

By default, the git push command doesn’t transfer tags to remote servers. You will have to explicitly push tags to a shared server after you have created them. This process is just like sharing remote branches — you can run like below.

$ git push origin <tagname>

$ git push origin v1.0



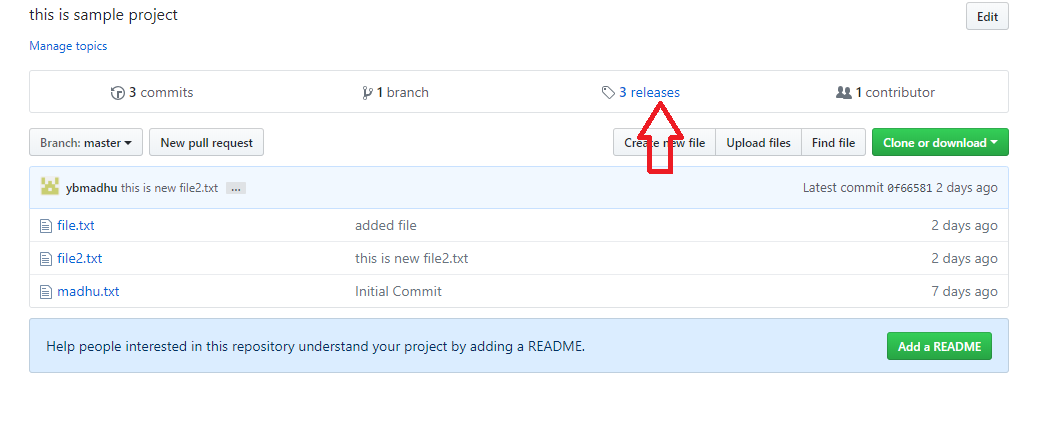
If you have a lot of tags that you want to push up at once, you can also use the **--tags** option to the **git push** command. This will transfer all of your tags to the remote servers that are not already there.

$ git push origin --tags

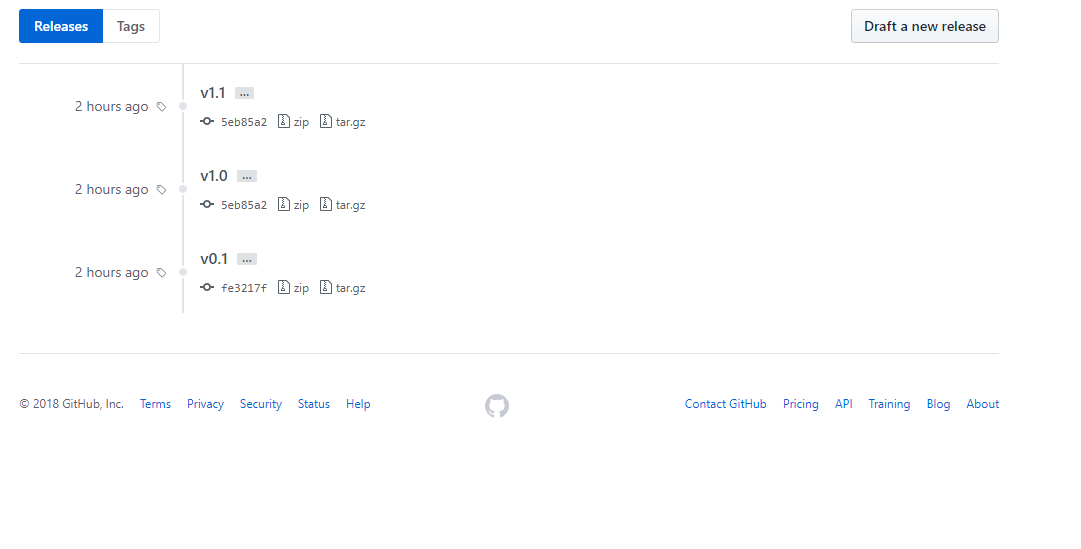


Now, when someone else clones or pulls from your repository, they will get all your tags as well.

Check in git hub our tags are pushed or not.



Click the release link it will open a tag list in github.



# Deleting Tags:

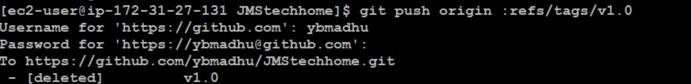
To delete a tag on your local repository, you can use **git tag -d <tagname>**. For example, we could remove our lightweight tag above as follows.

$ git tag –d v0.1



Note that this does not remove the tag from any remote servers. In order to update any remotes, you must use **git push <remote> :refs/tags/<tagname>**

$ git push origin :refs/tags/v1.0



# Delete all tags at time:

* #Delete local tags.
* git tag -d $(git tag -l)
* #Fetch remote tags.
* git fetch
* #Delete remote tags.
* x # Pushing once should be faster than multiple times
* #Delete local tags.
* git tag -d $(git tag -l)

# Remote deletes

* git ls-remote --tags origin | awk '/^(.\*)(\s+)(.\*[a-z0-9])$/ {print ":" $2}' | xargs git push origin