GIT TAGS

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).

Creating Tags:

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

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

---> based on the tag will pull the code through jenkins and will deploy particular version of tagging

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

git remote
git remote add <originname> master
git push test1 master

git tag
git tag v0.1 → lightweight
$git tag -a v1.0 -m "my release version 1.0" → annotated
git push test1 v1.0 → check the releases in github
git push test1 --tags → push all tags at a time

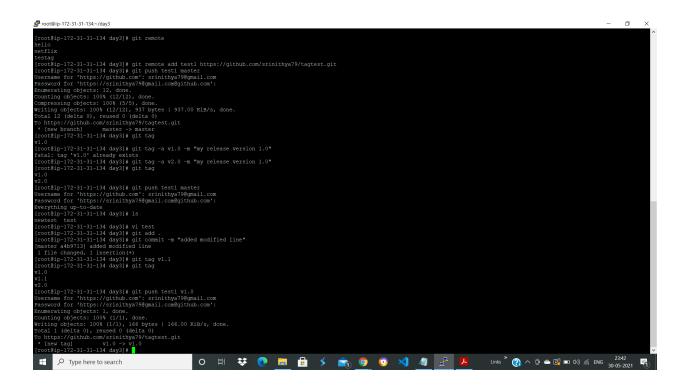**git tag -d v0.1 → delete the tag**
git tag -d $(git tag -l) → deletes all the tag
git tag -l

git show v1.0 → to get the details

git tag v0.1 35f984 → take first 6 number of the commit

git commit -m "modified" -a → all at a commit

1. What is git tags?
2. When we will use git tags?
3. How many types of git tags ?
4. What is lightweight git tag? How to create light weight git tag?
5. What is annotated git tag ? How to create annotated git tag?
6. What is difference between lightweight tag and annotated tag?
7. How to check all all commits in local?
8. How to see particular tag information?
9. How to create a tag for older commits ?
10. How to check a particular tag?
11. How to check git log with pretty mode?
12. How to list out all tags?
13. How to get all tags to local from remote repository?
14. How to push tags to remote repository?
15. How to delete a tag in local? how to delete all tags at a time in local?
16. How to delete a tag in remote repository? how to delete all tags at a time in remote repository?

```
[root@ip-172-31-31-134 day3]# git tag
[root@ip-172-31-31-134 day3]# git tag -a v1.0 -m "my release version 1.0"
fatal: too many params
[root@ip-172-31-31-134 day3]# git tag
[root@ip-172-31-31-134 day3]# git tag -a v1.0 -m "my release version 1.0"
[root@ip-172-31-31-134 day3]# git tag
v1.0
[root@ip-172-31-31-134 day3]# git show v1.0
tag v1.0
Tagger: Srinithya <srinithya79@gmail.com>
Date:   Sun May 30 17:40:07 2021 +0000

my release version 1.0

commit aa60688e949229a484d38c2bda06ab184e446bf9 (HEAD -> master, tag: v1.0, dev)
Author: Srinithya <srinithya79@gmail.com>
Date:   Mon May 24 17:41:58 2021 +0000

    Fourth commit

diff --git a/newtest b/newtest
new file mode 100644
index 0000000..ae69c92
--- /dev/null
+++ b/newtest
@@ -0,0 +1 @@
+Fourth Line
[root@ip-172-31-31-134 day3]#
```

```
[root@ip-172-31-31-134 day3]# git remote
hello
netflix
testag
[root@ip-172-31-31-134 day3]# git remote add test1 https://github.com/srinithya79/tagtest.git
[root@ip-172-31-31-134 day3]# git push test1 master
Username for 'https://github.com': srinithya79@gmail.com
Password for 'https://srinithya79@gmail.com@github.com':
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (12/12), 937 bytes | 937.00 KiB/s, done.
Total 12 (delta 0), reused 0 (delta 0)
To https://github.com/srinithya79/tagtest.git
 * [new branch]      master -> master
[root@ip-172-31-31-134 day3]# git tag
v1.0
[root@ip-172-31-31-134 day3]# git tag -a v1.0 -m "my release version 1.0"
fatal: tag 'v1.0' already exists
[root@ip-172-31-31-134 day3]# git tag -a v2.0 -m "my release version 1.0"
[root@ip-172-31-31-134 day3]# git tag
v1.0
v2.0
[root@ip-172-31-31-134 day3]# git push test1 master
Username for 'https://github.com': srinithya79@gmail.com
Password for 'https://srinithya79@gmail.com@github.com':
Everything up-to-date
[root@ip-172-31-31-134 day3]# ls
newtest  test
[root@ip-172-31-31-134 day3]# vi test
[root@ip-172-31-31-134 day3]# git add .
[root@ip-172-31-31-134 day3]# git commit -m "added modified line"
[master a4b9713] added modified line
 1 file changed, 1 insertion(+)
[root@ip-172-31-31-134 day3]# git tag v1.1
[root@ip-172-31-31-134 day3]# git tag
v1.0
v1.1
v2.0
[root@ip-172-31-31-134 day3]# git push test1 v1.0
Username for 'https://github.com': srinithya79@gmail.com
Password for 'https://srinithya79@gmail.com@github.com':
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 166 bytes | 166.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0)
To https://github.com/srinithya79/tagtest.git
 * [new tag]         v1.0 -> v1.0
[root@ip-172-31-31-134 day3]#
```

```
Date:   Mon May 24 17:30:20 2021 +0000

    First commit
[root@ip-172-31-31-134 day3]# git tag v0.1 35f984
[root@ip-172-31-31-134 day3]# git tag
v0.1
v1.0
v1.1
v2.0
[root@ip-172-31-31-134 day3]# git log
commit a4b97136e9227c505bb94a3177cbdce836902b4c (HEAD -> master, tag: v1.1)
Author: Srinithya <srinithya79@gmail.com>
Date:   Sun May 30 18:07:31 2021 +0000

    added modified line

commit aa60688e949229a484d38c2bda06ab184e446bf9 (tag: v2.0, tag: v1.0, test1/master, dev)
Author: Srinithya <srinithya79@gmail.com>
Date:   Mon May 24 17:41:58 2021 +0000

    Fourth commit

commit 35f984febfe5ca10ab40c8e8be6d21d4f1afb7ce (tag: v0.1)
Author: Srinithya <srinithya79@gmail.com>
Date:   Mon May 24 17:49:37 2021 +0000

    Third commit

commit 6e80292b719c0608487e0e9f062f8a29808b2590
Author: Srinithya <srinithya79@gmail.com>
Date:   Mon May 24 17:35:45 2021 +0000

    Second commit

commit b5562519ab38903bff24e2e0c7d79fc6cd05c8f3
Author: root <root@ip-172-31-31-134.us-east-2.compute.internal>
Date:   Mon May 24 17:30:20 2021 +0000

    First commit
[root@ip-172-31-31-134 day3]# git push test1 v0.1
Username for 'https://github.com': srinithya79@gmail.com
Password for 'https://srinithya79@gmail.com@github.com':
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/srinithya79/tagtest.git
 * [new tag]         v0.1 -> v0.1
[root@ip-172-31-31-134 day3]# git push test1 v1.1
Username for 'https://github.com': srinithya79@gmail.com
Password for 'https://srinithya79@gmail.com@github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
```

```
aa60688 (tag: v2.0, tag: v1.0, test1/master, dev) Fourth commit
35f984f (tag: v0.1) Third commit
6e80292 Second commit
b556251 First commit
[root@ip-172-31-31-134 day3]# git commit -m "modified" -a
On branch master
nothing to commit, working tree clean
[root@ip-172-31-31-134 day3]# git tag
v0.1
v1.0
v1.1
v2.0
[root@ip-172-31-31-134 day3]# git push test1 tags
error: src refspec tags does not match any
error: failed to push some refs to 'https://github.com/srinithya79/tagtest.git'
[root@ip-172-31-31-134 day3]# git push test1 --tags
Username for 'https://github.com': srinithya79@gmail.com
Password for 'https://srinithya79@gmail.com@github.com':
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 166 bytes | 166.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0)
To https://github.com/srinithya79/tagtest.git
 * [new tag]         v2.0 -> v2.0
[root@ip-172-31-31-134 day3]# git log
commit a4b9713e9227c505bb94a3177cbdce836902b4c (HEAD -> master, tag: v1.1)
Author: Srinithya <srinithya79@gmail.com>
Date:   Sun May 30 18:07:31 2021 +0000

    added modified line

commit aa60688e949229a484d38c2bda06ab184e446bf9 (tag: v2.0, tag: v1.0, test1/master, dev)
Author: Srinithya <srinithya79@gmail.com>
Date:   Mon May 24 17:41:58 2021 +0000

    Fourth commit

commit 35f984febfe5ca10ab40c8e8be6d21d4f1afb7ce (tag: v0.1)
Author: Srinithya <srinithya79@gmail.com>
Date:   Mon May 24 17:49:37 2021 +0000

    Third commit

commit 6e80292b719c0608487e0e9f062f8a29808b2590
Author: Srinithya <srinithya79@gmail.com>
Date:   Mon May 24 17:35:45 2021 +0000

    Second commit

commit b5562519ab38903bff24e2e0c7d79fc6cd05c8f3
```

```
Date:   Mon May 24 17:30:20 2021 +0000

    First commit
[root@ip-172-31-31-134 day3]# git tag -d v0.1
Deleted tag 'v0.1' (was 35f984f)
[root@ip-172-31-31-134 day3]# git tag
v1.0
v1.1
v2.0
[root@ip-172-31-31-134 day3]# git push test1 :refs/tags/v0.1
Username for 'https://github.com': srinithya79@gmail.com
Password for 'https://srinithya79@gmail.com@github.com':
To https://github.com/srinithya79/tagtest.git
 - [deleted]         v0.1
[root@ip-172-31-31-134 day3]# git tag -l
v1.0
v1.1
v2.0
[root@ip-172-31-31-134 day3]# git tag -d $(git tag -l)
Deleted tag 'v1.0' (was 888b78c)
Deleted tag 'v1.1' (was a4b9713)
Deleted tag 'v2.0' (was 11235cd)
[root@ip-172-31-31-134 day3]# git fetch
[root@ip-172-31-31-134 day3]# git tag
[root@ip-172-31-31-134 day3]# git log
commit a4b9713e9227c505bb94a3177cbdce836902b4c (HEAD -> master)
Author: Srinithya <srinithya79@gmail.com>
Date:   Sun May 30 18:07:31 2021 +0000

    added modified line

commit aa60688e949229a484d38c2bda06ab184e446bf9 (test1/master, dev)
Author: Srinithya <srinithya79@gmail.com>
Date:   Mon May 24 17:41:58 2021 +0000

    Fourth commit

commit 35f984febfe5ca10ab40c8e8be6d21d4f1afb7ce
Author: Srinithya <srinithya79@gmail.com>
Date:   Mon May 24 17:49:37 2021 +0000

    Third commit

commit 6e80292b719c0608487e0e9f062f8a29808b2590
Author: Srinithya <srinithya79@gmail.com>
Date:   Mon May 24 17:35:45 2021 +0000

    Second commit

commit b5562519ab38903bff24e2e0c7d79fc6cd05c8f3
```

github.com/srinithya79/tagtest/tags

Search or jump to...   Pull requests   Issues   Marketplace   Explore

srinithya79 / tagtest

Unwatch ▼ 1   ☆ Star 0   Fork 0

<> Code   ⊙ Issues   ⊔ Pull requests   ⊙ Actions   ⊡ Projects   ⊞ Wiki   ⊙ Security   Insights   ⚙ Settings

Releases   Tags

🏷

There aren't any releases here

Releases are powered by tagging specific points of history in a repository.
They're great for marking release points like `v1.0`.

Create a new release

© 2021 GitHub, Inc.   Terms   Privacy   Security   Status   Docs   Contact GitHub   Pricing   API   Training   Blog   About

Type here to search

Links   ENG   23:40 30-05-2021

🏷 Tags

v1.0 ⋯
🕐 30 minutes ago   ⟜ aa60688   🗎 zip   🗎 tar.gz   ⋯

© 2021 GitHub, Inc.   Terms   Privacy   Security   Status   Docs   Contact GitHub   Pricing   API   Training   Blog   About

Type here to search

Links   ENG   23:40 30-05-2021

GIT STASH

git stash temporarily (stashes) changes you've made to your working copy so you can work on something else, and then come back and re-apply them later on. Stashing is handy if you need to quickly switch context and work on something else, but you're mid-way through a code change and aren't quite ready to commit.

Basic Commands:
git stash :stash the uncommitted changes .
git stash list: show all the stashes.
git stash apply: apply the top stash to the current branch
git stash apply n: apply nth no stash
git stash pop n: apply the stash and remove from stash list
git stash clear : remove the entire stash
git stash drop n: remove particular stash