

What is Version Control -

Version control systems are integral to building software. They combine your repository of project files with a history of all your code changes, making it easy to edit and understand your code over time.

The main benefit of using a version control system is that it keeps your team's workflows organized as they work through various types of releases. With one in place, team members can easily research, track, and undo code. They can work on the same code simultaneously without code conflicts. And they track who made what changes when and why.

But before you implement one into your team's workflow, you need to figure out which version control system is right for you. While most options out there have similar benefits, their differences are important.

What is SVN -

Also known as Subversion, SVN represents the most popular centralized version control system on the market. With a centralized system, all files and historical data are stored on a central server. And developers commit their changes directly to that central server repository.

What is git -

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

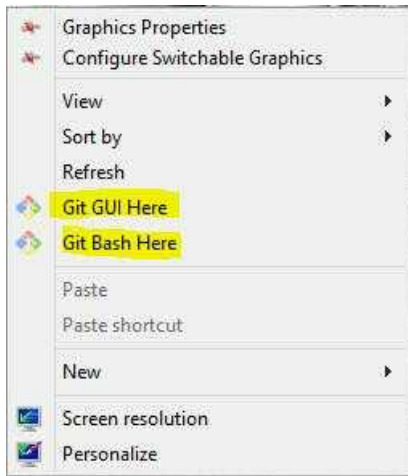
Git vs. Subversion: Comparison Chart

GIT VERSUS SUBVERSION		
Git	Subversion	
Git is a distributed version control system used for source code management.	Subversion (or SVN) is a centralized versioning and revision control system.	
It creates a local repository to store everything locally instead of using a centralized server.	It uses a centralized server to store changes in source code.	
Network access is not mandatory for Git operations.	Network is required for almost all of SVN operations.	
A subproject is called a Git "submodule".	A subproject is called an "SVN External".	
Git does not have a global revision number.	SVN does have a global revision number.	
Git contents are cryptographically check-summed using the SHA-1 hash algorithm.	SVN does not have hashed contents.	

Setup git and github -

1. Download git from <https://git-scm.com>

Right click to your mouse you will get below options - Select Git Bash Here



What is github -

GitHub is a web-based hosting service for version control using Git. It is mostly used for computer code. It offers all of the distributed version control and source code management functionality of Git as well as adding its own features

2. Create account in github.com and create a new repository.

Create a new repository

A repository contains all project files, including the revision history.

Owner: manavidcloud / Repository name *: gitproject01 ✓

Great repository names are short and memorable. Need inspiration? How about shiny-octo-umbrella?

Description (optional): gitproject01

☒ Public: Anyone can see this repository. You choose who can commit.

☐ Private: You choose who can see and commit to this repository.

☐ Initialize this repository with a README: This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None Add a license: None ⓘ

Create repository

```
Navid@mohamadali MINGW64 ~
```

```
$ mkdir -p gitproject/gitproject01; cd gitproject/gitproject01
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01
```

```
$ echo "My first gitproject01 for test" >> sample.txt
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01
```

```
$ ls -la
```

```
total 1
```

```
drwxr-xr-x 1 Navid 197121 0 Mar 12 00:47 ./
```

```
drwxr-xr-x 1 Navid 197121 0 Mar 12 00:46 ../
```

```
-rw-r--r-- 1 Navid 197121 31 Mar 12 00:47 sample.txt
```

```
=====
```

Setup git master

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01
```

```
$ git init
```

```
### The folder which have .git its a git project
```

```
Initialized empty Git repository in C:/Users/Navid/gitproject/gitproject01/.git/
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
```

```
$ ls -la
```

```
total 5
```

```
drwxr-xr-x 1 Navid 197121 0 Mar 12 00:47 ./
```

```
drwxr-xr-x 1 Navid 197121 0 Mar 12 00:46 ../
```

```
drwxr-xr-x 1 Navid 197121 0 Mar 12 00:47 .git/
```

```
-rw-r--r-- 1 Navid 197121 31 Mar 12 00:47 sample.txt
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
```

```
$ git add .
```

```
warning: LF will be replaced by CRLF in sample.txt.
```

```
The file will have its original line endings in your working directory
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
```

```
$ git commit -m "gitproject01 and we have create our first sample file" -a
```

```
*** Please tell me who you are.
```

```
Run
```

```
git config --global user.email "you@example.com"
```

```
git config --global user.name "Your Name"
```

```
to set your account's default identity.
```

```
Omit --global to set the identity only in this repository.
```

```
fatal: unable to auto-detect email address (got 'Navid@mohamadali.(none)')
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
```

```
$ git config --global user.email "mohamadabdulnavidcloud@gmail.com"
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
```

```
$ git config --global user.name "manavidcloud"
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
```

```
$ git config --list | grep -i user
```

```
user.email=mohamadabdulnavidcloud@gmail.com
```

```
user.name=manavidcloud
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
```

```
$ git commit -m "gitproject01 and we have create our first sample file" -a
```

```
[master (root-commit) 3ff66ee] gitproject01 and we have create our first sample file
```

```
1 file changed, 1 insertion(+)
```

```
create mode 100644 sample.txt
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
```

```
$ git push origin master
```

```
fatal: 'origin' does not appear to be a git repository
```

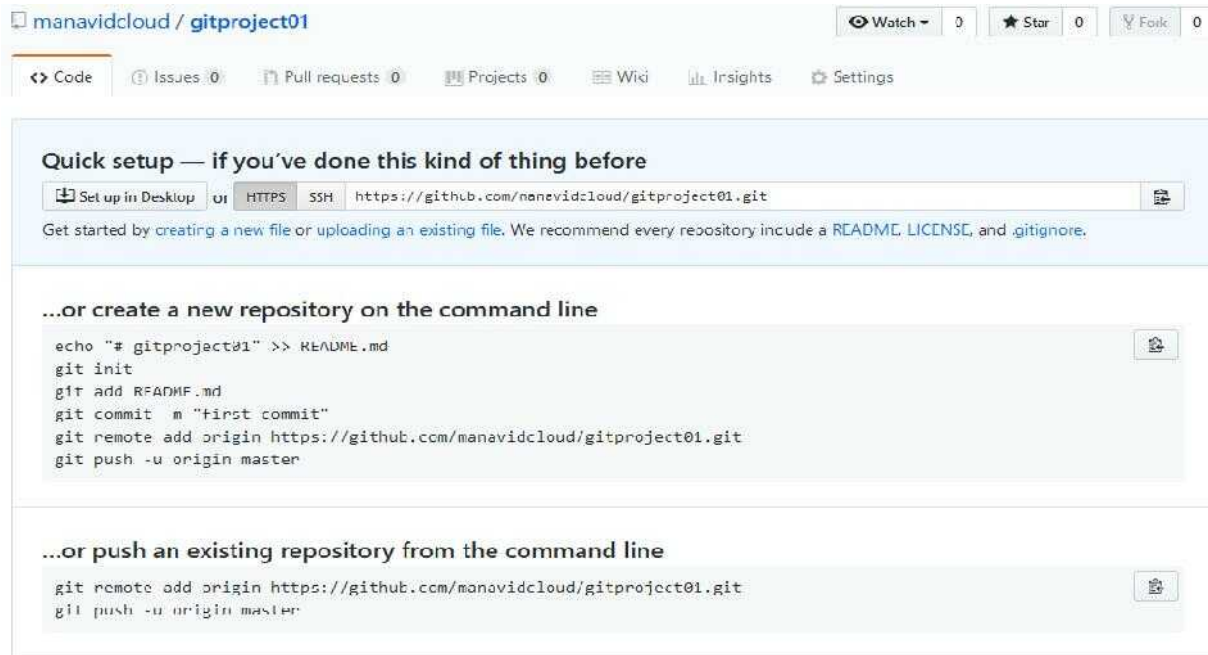
```
fatal: Could not read from remote repository.
```

```
Please make sure you have the correct access rights  
and the repository exists.
```

```
=====
```

Setup git origin

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git remote -v
```

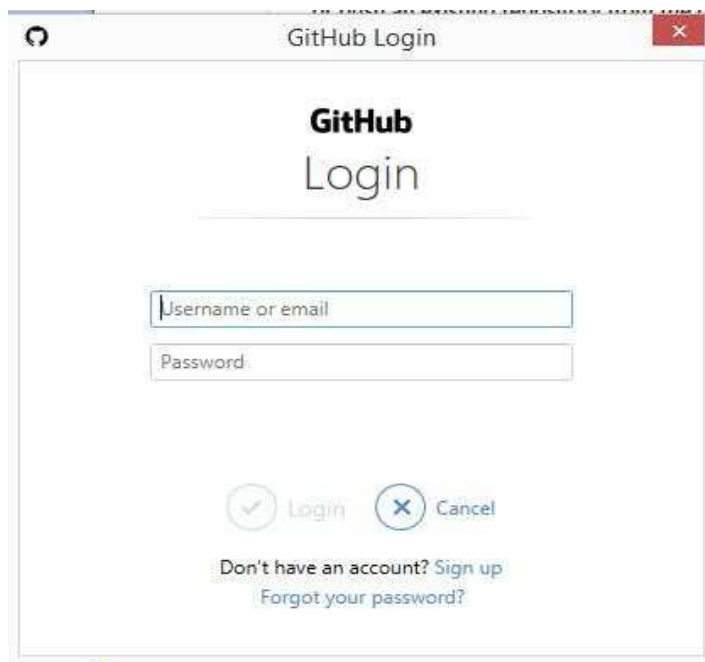


```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git remote add origin https://github.com/manavidcloud/gitproject01.git
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git remote -v
origin https://github.com/manavidcloud/gitproject01.git (fetch)
origin https://github.com/manavidcloud/gitproject01.git (push)
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git push origin master
```

Popup will come enter username and password -

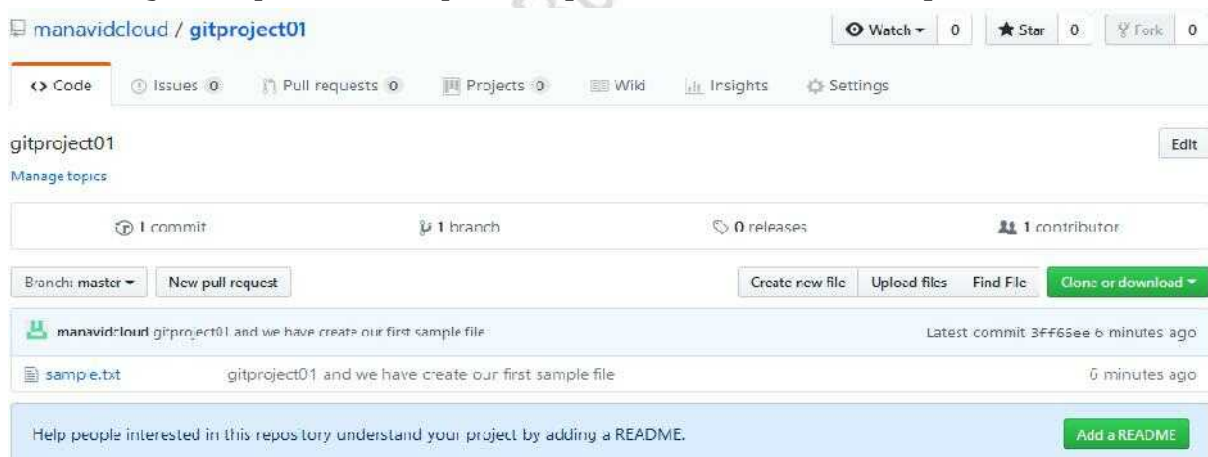


If credentials accepted then you will get below output -

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 275 bytes | 137.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/manavidcloud/gitproject01.git
 * [new branch]      master -> master
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$
```

Check in github you can see your **sample.txt** in under the repo -



Note: git config --global / git remote add origin its only run at first time.

Let s see for second file you only have to run below commands -

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ echo "Second file to test gitproject01" >> secondfile.txt
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git add secondfile.txt
```

warning: LF will be replaced by CRLF in secondfile.txt.
The file will have its original line endings in your working directory

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git commit -m "gitproject01 test02" secondfile.txt
warning: LF will be replaced by CRLF in secondfile.txt.
The file will have its original line endings in your working directory
[master 5204bf2] gitproject01 test02
 1 file changed, 1 insertion(+)
 create mode 100644 secondfile.txt
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git push origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 321 bytes | 160.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/manavidcloud/gitproject01.git
 3ff66ee..5204bf2  master -> master
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$
```

What is LF and CRLF

In Unix systems the end of a line is represented with a line feed (LF). In windows a line is represented with a carriage return (CR) and a line feed (LF) thus (CRLF). when you get code from git that was uploaded from a unix system they will only have an LF.

Disable warning: LF will be replaced by CRLF -

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git config --global core.autocrlf false
```

Note: On next commit you should not get any message.

Check git version

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git version
git version 2.21.0.windows.1
```

Update git version

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git update-git-for-windows
Git for Windows 2.21.0.windows.1 (64bit)
Up to date
```

Remove commit using HEAD from local repo

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git log
commit efdc4790f9c39f1964b286d25a89a8d5c1868c66 (HEAD -> master)
```


Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sat Mar 23 02:13:22 2019 +0530

sample02

commit 24cf3a4d251786e93bbe8ba4fd76a55dfbf40502
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sat Mar 23 02:10:07 2019 +0530

gitproject01 and we have create our first sample file

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git reset --hard HEAD^
HEAD is now at 24cf3a4 gitproject01 and we have create our first sample file
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git log
commit 24cf3a4d251786e93bbe8ba4fd76a55dfbf40502 (HEAD -> master)
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sat Mar 23 02:10:07 2019 +0530
```

gitproject01 and we have create our first sample file

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$
```

Note: Single ^ means only one if you add ^^ it will delete as per LIFO as per commit.

To push above changes at remote

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git push -f origin master
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/manavidcloud/project01.git
+ 979cf02...24cf3a4 master -> master (forced update)
```

Note: It will not allow with "git push origin master" hence use -f option.

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git pull origin master
From https://github.com/manavidcloud/project01
* branch      master      -> FETCH_HEAD
Already up to date.
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ ls -la
total 9
drwxr-xr-x 1 Ali_baba 197121  0 Mar 24 03:31 ./
drwxr-xr-x 1 Ali_baba 197121  0 Mar 24 01:17 ../
drwxr-xr-x 1 Ali_baba 197121  0 Mar 24 03:32 .git/
-rw-r--r-- 1 Ali_baba 197121 31 Mar 24 03:31 sample.txt
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git log
commit 24cf3a4d251786e93bbe8ba4fd76a55dfbf40502 (HEAD -> master, origin/master)
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sat Mar 23 02:10:07 2019 +0530
```

gitproject01 and we have create our first sample file

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$
```

Merge conflicts

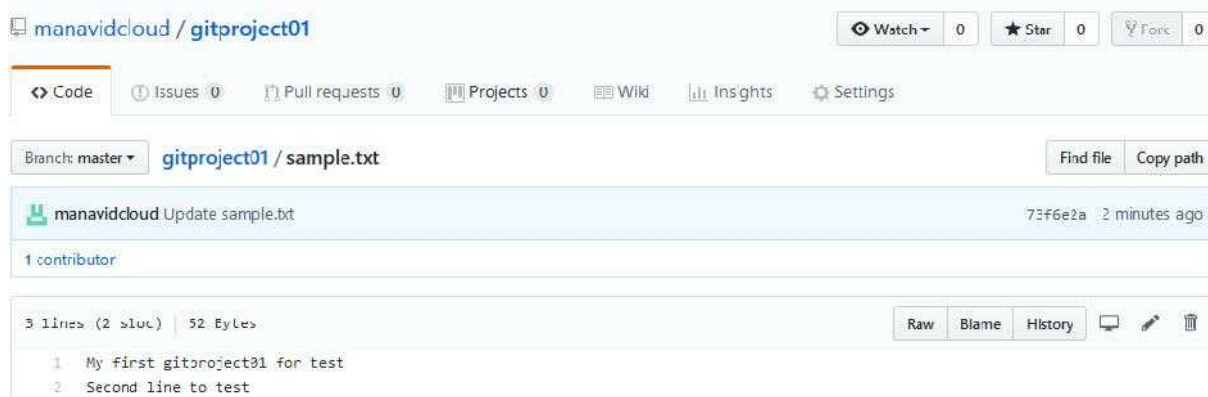
If two developers updating the same code sometime it become merge conflicts.

1. Got to git bash and see content of sample.txt file

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ cat sample.txt
My first gitproject01 for test
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
```

2. Go to github and update sample.txt and commit the changes -



3. Add content of sample.txt from git bash -

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ echo "Update second line from git bash" >> sample.txt

Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ cat sample.txt
My first gitproject01 for test
Update second line from git bash

Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$

Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git add sample.txt

Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git commit -m "Updating sample.txt file from gitbash to test merge" sample.txt
[master 9f85a8c] Updating sample.txt file from gitbash to test merge
1 file changed, 1 insertion(+)
```

Below is best example of merge conflicts as one developer updated the sample.txt from github and second developer from bashgit.

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git push origin master
To https://github.com/manavidcloud/gitproject01.git
! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'https://github.com/manavidcloud/gitproject01.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.

Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$
```

In this case developer who wanted to update the record from sample.txt file, he have to first pull the latest one.

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git pull origin master
From https://github.com/manavidcloud/gitproject01
* branch      master      -> FETCH_HEAD
Auto-merging sample.txt
CONFLICT (content): Merge conflict in sample.txt
Automatic merge failed; fix conflicts and then commit the result.
```

To see the git commit status

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master|MERGING)
$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)
```

```
Unmerged paths:
  (use "git add <file>..." to mark resolution)
```

```
      both modified:   sample.txt
```

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

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master|MERGING)
$
```

Now check sample.txt file you will know what is the update from other developer in the file.

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master|MERGING)
$ cat sample.txt
My first gitproject01 for test
<<<<<<< HEAD
Update second line from git bash          ---> This line is update by you
=====
Second line to test
>>>>>>> 73f6e2ae4d99e4298410e478750161021800da94  --> Its update by another developer
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master|MERGING)
$
```

Remove all special character line from file -

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master|MERGING)
$ vi sample.txt
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master|MERGING)
$ cat sample.txt
My first gitproject01 for test
Update second line from git bash
Second line to test
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master|MERGING)
$ git add sample.txt
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master|MERGING)
$ git commit -m "Merge conflits issue fixed" sample.txt
fatal: cannot do a partial commit during a merge.
```

Generally after git commit -m "" <filename> merching issue should be resolve but if you get error then use -i option before <filename> to fix it.

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master|MERGING)
$ git status
On branch master
All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)
```

```
Changes to be committed:
```

modified: sample.txt

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master|MERGING)
$ git commit -m "Merge conflits issue fixed" -i sample.txt
[master 46f7a00] Merge conflits issue fixed
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git push origin master
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 741 bytes | 67.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/manavidcloud/gitproject01.git
73f6e2a..46f7a00 master -> master
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$
```

manavidcloud / gitproject01

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

gitproject01 Edit

Manage topics

7 commits 1 branch 0 releases 1 contributor

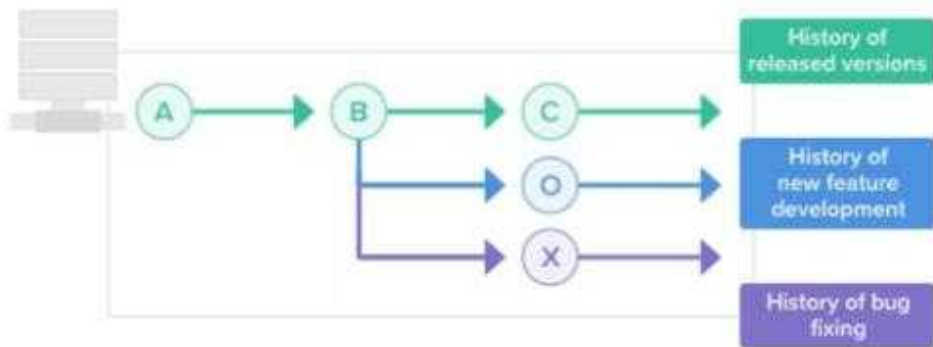
Branch: master New pull request Create new file Upload files Find File Clone or download

manavidCloud Updated	Latest commit c4ae652 15 hours ago
gt.docx	Updated 15 hours ago
sample.txt	Merge conflits issue fixed 15 hours ago
secondfile.txt	gitproject01 test02 2 days ago

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

Git Branch

A Git branch is essentially an independent line of development. You can take advantage of branching when working on new features or bug fixes because it isolates your work from that of other team members.



```

Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ ls -ltra
total 222
drwxr-xr-x 1 Navid 197121      0 Mar 12 00:46 ./
-rw-r--r-- 1 Navid 197121    33 Mar 12 01:24 secondfile.txt
-rw-r--r-- 1 Navid 197121    88 Mar 13 00:19 sample.txt
drwxr-xr-x 1 Navid 197121      0 Mar 13 00:19 ./
-rw-r--r-- 1 Navid 197121 214443 Mar 13 00:37 git.docx
drwxr-xr-x 1 Navid 197121      0 Mar 13 00:38 .git/
  
```

To list how many branch you have in your project -

```

Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git branch --list      # List local branch
* master                 #Current branch in which you are working ""
  
```

```

Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$
  
```

```

Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git branch -a          # List local and remote branch
* master
  remotes/origin/master
  
```

```

Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git branch fix01       # Create new branch
  
```

```

Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git branch --list
  fix01
* master
  
```

```

Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git branch -a
  fix01
* master
  remotes/origin/master
  
```

```

Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git checkout fix01     # To switch from one branch to another
Switched to branch 'fix01'
  
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (fix01)
$ ls -ltra
total 222
drwxr-xr-x 1 Navid 197121      0 Mar 12 00:46 ../
-rw-r--r-- 1 Navid 197121     33 Mar 12 01:24 secondfile.txt
-rw-r--r-- 1 Navid 197121     88 Mar 13 00:19 sample.txt
drwxr-xr-x 1 Navid 197121      0 Mar 13 00:19 ./
-rw-r--r-- 1 Navid 197121 214443 Mar 13 00:37 git.docx
drwxr-xr-x 1 Navid 197121      0 Mar 13 15:42 .git/
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (fix01)
$ git branch --list
* fix01
  master
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (fix01)
$ git branch -a
* fix01
  master
  remotes/origin/master
```

Create new file in fix01 branch and push it on github -

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (fix01)
$ echo "this is bugfix file for test our environment" >> debugfix01
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (fix01)
$ ls -ltra
total 223
drwxr-xr-x 1 Navid 197121      0 Mar 12 00:46 ../
-rw-r--r-- 1 Navid 197121     33 Mar 12 01:24 secondfile.txt
-rw-r--r-- 1 Navid 197121     88 Mar 13 00:19 sample.txt
-rw-r--r-- 1 Navid 197121 214443 Mar 13 00:37 git.docx
drwxr-xr-x 1 Navid 197121      0 Mar 13 15:42 .git/
drwxr-xr-x 1 Navid 197121      0 Mar 13 16:40 ./
-rw-r--r-- 1 Navid 197121     45 Mar 13 16:40 debugfix01
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (fix01)
$ git add debugfix01
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (fix01)
$ git commit -m "debugfile" debugfix01
warning: LF will be replaced by CRLF in debugfix01.
The file will have its original line endings in your working directory
[fix01 86d0248] debugfile
 1 file changed, 1 insertion(+)
 create mode 100644 debugfix01
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (fix01)
$ git push origin fix01 # After origin mention the branch name
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 390 bytes | 130.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/manavidcloud/gitproject01.git
   c4ae652..86d0248  fix01 -> fix01
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (fix01)
$
```

Check in git hub refresh the page you will get master and fix01 branch -

Before -

manavidcloud / gitproject01

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

gitproject01 Edit

Manage topics

7 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

Switch branches/tags

Find or create a branch...

Branches Tags

✓ master

Latest commit c4ae652 16 hours ago

Updated 16 hours ago

Merge conflicts issue fixed 16 hours ago

gitproject01 test02 2 days ago

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

After -

manavidcloud / gitproject01

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

gitproject01 Edit

Manage topics

7 commits 2 branches 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

Switch branches/tags

Find or create a branch...

Branches Tags

fix01

✓ master

Latest commit c4ac652 16 hours ago

Updated 16 hours ago

Merge conflicts issue fixed 16 hours ago

gitproject01 test02 2 days ago

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

In master branch we have only one file so fix01 should have two in it.

manavidcloud / gitproject01

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

gitproject01 Edit

Manage topics

7 commits 2 branches 0 releases 1 contributor

Branch: master New pull request

Create new file Upload files Find File Clone or download

manavidcloud Updated Latest commit c4ae652 16 hours ago

git.docx	Updated	16 hours ago
sample.txt	Merge conflicts issue fixed	16 hours ago
secondfile.txt	gitproject01 test02	2 days ago

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

fix01 branch -

manavidcloud / gitproject01

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

gitproject01 Edit

Manage topics

0 commits 2 branches 0 releases 1 contributor

Your recently pushed branches:

fix01 (1 minute ago) Compare & pull request

Branch: fix01 New pull request

Create new file Upload files Find File Clone or download

This branch is 1 commit ahead of master. Pull request Compare

manavidcloud debugfile Latest commit 36d6248 2 minutes ago

debugfix01	debugfile	2 minutes ago
git.docx	Updated	16 hours ago
sample.txt	Merge conflicts issue fixed	16 hours ago
secondfile.txt	gitproject01 test02	2 days ago

If we consider debugfix01 is our fix for the issue which is tested by QA team and confirmed to us then we have to merge master branch with fix01 branch by using below steps -

Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (fix01)

```
$ git checkout master # Changed the branch
Switched to branch 'master'
```

Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)

```
$ ls -ltra
total 222
drwxr-xr-x 1 Navid 197121      0 Mar 12 00:46 ../
-rw-r--r-- 1 Navid 197121     33 Mar 12 01:24 secondfile.txt
-rw-r--r-- 1 Navid 197121     88 Mar 13 00:19 sample.txt
-rw-r--r-- 1 Navid 197121 214443 Mar 13 00:37 git.docx
drwxr-xr-x 1 Navid 197121      0 Mar 13 17:14 ./
drwxr-xr-x 1 Navid 197121      0 Mar 13 17:14 .git/
```

Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)

```
$ git merge fix01 # Merge Master with fix01
Updating c4ae652..86d0248
```



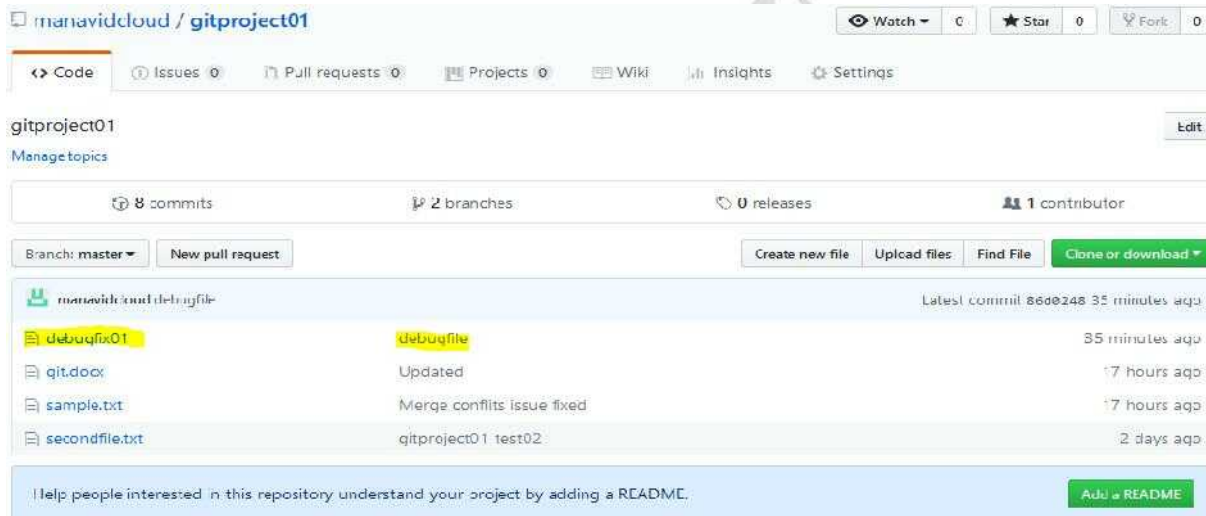
```
Fast-forward
 debugfix01 | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 debugfix01
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ ls -ltra
total 223
drwxr-xr-x 1 Navid 197121      0 Mar 12 00:46 ./
-rw-r--r-- 1 Navid 197121     33 Mar 12 01:24 secondfile.txt
-rw-r--r-- 1 Navid 197121     88 Mar 13 00:19 sample.txt
-rw-r--r-- 1 Navid 197121 214443 Mar 13 00:37 git.docx
-rw-r--r-- 1 Navid 197121     46 Mar 13 17:15 debugfix01 #fix01 file is visible
drwxr-xr-x 1 Navid 197121      0 Mar 13 17:15 ./
drwxr-xr-x 1 Navid 197121      0 Mar 13 17:15 .git/
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git add .
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git commit -m "Merge file to test branch" debugfix01
On branch master # Ignore it
nothing to commit, working tree clean # Ignore it
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git push origin master
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/manavidcloud/gitproject01.git
c4ae652..86d0248 master -> master
```



Now delete the fix01 branch locally -

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git branch --list
fix01
* master
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git branch -d fix01
Deleted branch fix01 (was 86d0248).
```

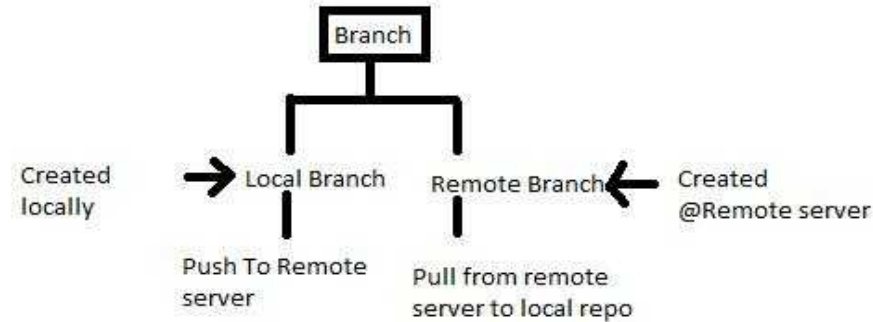
```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git branch --list
* master
```

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$
```

Above command only remove branch from locally. You can still find fix01 in your github
To delete branch from github (remote server) -

```
Navid@mohamadali MINGW64 ~/gitproject/gitproject01 (master)
$ git push origin -d fix01
To https://github.com/manavidcloud/gitproject01.git
- [deleted]          fix01
```

Now check in github your fix01 should be deleted



Note: I am using new local system and repo. Please don't compare it with above examples.

Git Tags

Note: Head always shows the latest commit

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git log
commit 49af660974bf50bc93ca02ac98f3cd660963a3b2 (HEAD -> master, origin/master)
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sun Mar 24 03:41:03 2019 +0530
```

update

```
commit 24cf3a4d251786e93bbe8ba4fd76a55dfbf40502
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sat Mar 23 02:10:07 2019 +0530
```

gitproject01 and we have create our first sample file

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ ls -la
total 912
drwxr-xr-x 1 Ali_baba 197121 0 Mar 24 03:40 ./
drwxr-xr-x 1 Ali_baba 197121 0 Mar 24 01:17 ../
drwxr-xr-x 1 Ali_baba 197121 0 Mar 24 03:41 .git/
-rw-r--r-- 1 Ali_baba 197121 10942 Mar 24 03:40 commands.xlsx
-rw-r--r-- 1 Ali_baba 197121 911039 Mar 24 03:40 git.docx
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ echo "Line 01 for tag example" >> test01
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git add .
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git commit -m "Line1" .
[master 0899200] Line1
1 file changed, 1 insertion(+)
create mode 100644 test01
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ echo "Line 02 for tag example" >> test01
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git add .
```

```

Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git commit -m "Line12" .
[master b0a771d] Line12
1 file changed, 1 insertion(+)

Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ echo "Line 03 for tag example" >> test01

Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git add .

Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git commit -m "Line123_final" .
[master 1fb58e9] Line123_final
1 file changed, 1 insertion(+)

```

```

Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git log
commit 1fb58e9cf2e308ec93e740763b964867b010d8af (HEAD -> master)
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sun Mar 24 04:01:48 2019 +0530

    Line123_final

commit b0a771dcf03f09a8174e880ba3b61f0e05a18baa
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sun Mar 24 04:01:06 2019 +0530

    Line12

commit 0899200a33ed1f3e9e21d109ef0da190d659f97b
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sun Mar 24 04:00:08 2019 +0530

    Line1

commit 49af660974bf50bc93ca02ac98f3cd660963a3b2 (origin/master)
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sun Mar 24 03:41:03 2019 +0530

```

This are committed but not push to origin. We can evaluate it as its showing (HEAD -> master)

```

    update

commit 24cf3a4d251786e93bbe8ba4fd76a55dfbf40502
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sat Mar 23 02:10:07 2019 +0530

    gitproject01 and we have create our first sample file

```

committed and pushed on origin.
(origin/master)

```

Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git push origin master
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (9/9), 809 bytes | 269.00 KiB/s, done.
Total 9 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/manavidcloud/project01.git
49af660..1fb58e9 master -> master

```

```

Ali_baba@alibaba MINGW64 /gitproject/project01 (master)

```

```
$ cat test01
Line 01 for tag example
Line 02 for tag example
Line 03 for tag example
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git log
commit 1fb58e9cf2e308ec93e740763b964867b010d8af (HEAD -> master, origin/master)
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sun Mar 24 04:01:48 2019 +0530

    Line123_final

commit b0a771dcf03f09a8174e880ba3b61f0e05a18baa
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sun Mar 24 04:01:06 2019 +0530

    Line12

commit 0899200a33ed1f3e9e21d109ef0da190d659f97b
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sun Mar 24 04:00:08 2019 +0530

    Line1

commit 49af660974bf50bc93ca02ac98f3cd660963a3b2
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sun Mar 24 03:41:03 2019 +0530

    update

commit 24cf3a4d251786e93bbe8ba4fd76a55dfbf40502
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sat Mar 23 02:10:07 2019 +0530

    gitproject01 and we have create our first sample file

Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$
```

All commits are pushed at origin.

Validated as its showing
(HEAD -> master, origin/master)

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"
```

To list the tag

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git tag

To add the tag (in below example it will tag to latest commit)
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git tag -a v1.0 -m "Release for project01 latest changes"
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git tag
v1.0
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git log
commit 1fb58e9cf2e308ec93e740763b964867b010d8af (HEAD -> master, tag: v1.0, origin/master)
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
```

Annotated tag example

i.e nothing but the tag with details.

Date: Sun Mar 24 04:01:48 2019 +0530

Line123_final

```
commit b0a771dcf03f09a8174e880ba3b61f0e05a18baa
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sun Mar 24 04:01:06 2019 +0530
```

Line12

```
commit 0899200a33ed1f3e9e21d109ef0da190d659f97b
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sun Mar 24 04:00:08 2019 +0530
```

Line1

```
commit 49af660974bf50bc93ca02ac98f3cd660963a3b2
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sun Mar 24 03:41:03 2019 +0530
```

update

```
commit 24cf3a4d251786e93bbe8ba4fd76a55dfbf40502
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sat Mar 23 02:10:07 2019 +0530
```

gitproject01 and we have create our first sample file

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$
```

To check tag details (commit id, tagger, Author, Date)

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git tag
v1.0
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git show v1.0
tag v1.0
Tagger: manavidcloud <mohamadabdulnavidcloud@gmail.com> #who committed
Date: Sun Mar 24 04:30:08 2019 +0530 #when its created
```

Release for project01 latest changes **#Comment for latest message**

```
commit 1fb58e9cf2e308ec93e740763b964867b010d8af (HEAD -> master, tag: v1.0, origin/master)
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sun Mar 24 04:01:48 2019 +0530
```

Line123_final

```
diff --git a/test01 b/test01
index 52ea676..7f1d7ed 100644
--- a/test01
+++ b/test01
@@ -1,2 +1,3 @@
 Line 01 for tag example
 Line 02 for tag example
+Line 03 for tag example
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git push --tags
```

```
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 185 bytes | 185.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0)
To https://github.com/manavidcloud/project01.git
 * [new tag]         v1.0 -> v1.0
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$
```

Go to github and your repository. Tags are comes in "releases"

Before -

manavidcloud / project01

Watch

0

Star

0

Fork

0

Code

Issues0

Pull requests0

Projects0

Wiki

Insights

Settings

project01

Edit

Manage topics

5 commits

1 branch

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find File

Clone or download

manavidcloud Line123_final

Latest commit 1fb58e9 32 minutes ago

commands.xlsx	update	an hour ago
git.docx	update	an hour ago
test01	Line123_final	32 minutes ago

After -

manavidcloud / project01

Watch

0

Star

0

Fork

0

Code

Issues0

Pull requests0

Projects0

Wiki

Insights

Settings

project01

Edit

Manage topics

5 commits

1 branch

1 release

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find File

Clone or download

manavidcloud Line123_final

Latest commit 1fb58e9 an hour ago

commands.xlsx	update	an hour ago
git.docx	update	an hour ago
test01	Line123_final	an hour ago



Lightweight tag

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ echo "Example for light weight tag" >> test02
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git add .
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git commit -m "light01" .
[master e5669fb] light01
1 file changed, 1 insertion(+)
create mode 100644 test02
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git tag v1.1
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git tag
v1.0
v1.1
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git push origin v1.1
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
```



```
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 293 bytes | 58.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/manavidcloud/project01.git
 * [new tag]          v1.1 -> v1.1
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$
```

Note: Using tag command only push the tags not the file.

Scenario

Committed one code and releases couple of code with committed, tag, pushed. Now observed there is issue in code v1.0. Now how to tag first code which don't have tag.

example:

```
a -- committed but not tagged
b -- v1.0 } Committed, pushed, tagged
c -- v2.0 }
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git log
commit e5669f8e5338e8fb1ee580fc5c8e56b79b49804a (HEAD -> master, tag: v1.1)
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
```

Date: Mon Mar 25 07:09:42 2019 +0530

light01

commit b5eaa3c27e9ae1e80e44e256e64d927684f8389a (origin/master)
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Mon Mar 25 01:24:35 2019 +0530

new

commit 6c288ad2ec4be8c71200e4093731e9f3276e9d79
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Mon Mar 25 01:00:26 2019 +0530

update

commit f189e49d22491829a5c71b4dbc6d959e6fc84463
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sun Mar 24 04:59:19 2019 +0530

Latest

commit 1fb58e9cf2e308ec93e740763b964867b010d8af (tag: v1.0) #Found bug in this code
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sun Mar 24 04:01:48 2019 +0530

Line123_final

commit b0a771dcf03f09a8174e880ba3b61f0e05a18baa # Have to tag it so we can use for deployment
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sun Mar 24 04:01:06 2019 +0530

Line12

commit 0899200a33ed1f3e9e21d109ef0da190d659f97b
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sun Mar 24 04:00:08 2019 +0530

Line1

commit 49af660974bf50bc93ca02ac98f3cd660963a3b2
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sun Mar 24 03:41:03 2019 +0530

update

commit 24cf3a4d251786e93bbe8ba4fd76a55dfbf40502
:

Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
\$ git tag v0.1 b0a771dcf # git tag <version> <commit id>

Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
\$ git tag
v0.1
v1.0
v1.1

Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
\$ git log
commit e5669fbe5338e8fb1ee580fc5c8e56b79b49804a (HEAD -> master, tag: v1.1)
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Mon Mar 25 07:09:42 2019 +0530

light01

```
commit b5eaa3c27e9ae1e80e44e256e64d927684f8389a (origin/master)
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Mon Mar 25 01:24:35 2019 +0530
```

new

```
commit 6c288ad2ec4be8c71200e4093731e9f3276e9d79
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Mon Mar 25 01:00:26 2019 +0530
```

update

```
commit f189e49d22491829a5c71b4dbc6d959e6fc84463
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sun Mar 24 04:59:19 2019 +0530
```

Latest

```
commit 1fb58e9cf2e308ec93e740763b964867b010d8af (tag: v1.0)
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sun Mar 24 04:01:48 2019 +0530
```

Line123_final

```
commit b0a771dcf03f09a8174e880ba3b61f0e05a18baa (tag: v0.1) #Now our working code have tag
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sun Mar 24 04:01:06 2019 +0530
```

Line12

```
commit 0899200a33ed1f3e9e21d109ef0da190d659f97b
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sun Mar 24 04:00:08 2019 +0530
```

Line1

```
commit 49af660974bf50bc93ca02ac98f3cd660963a3b2
Author: manavidcloud <mohamadabdulnavidcloud@gmail.com>
Date: Sun Mar 24 03:41:03 2019 +0530
```

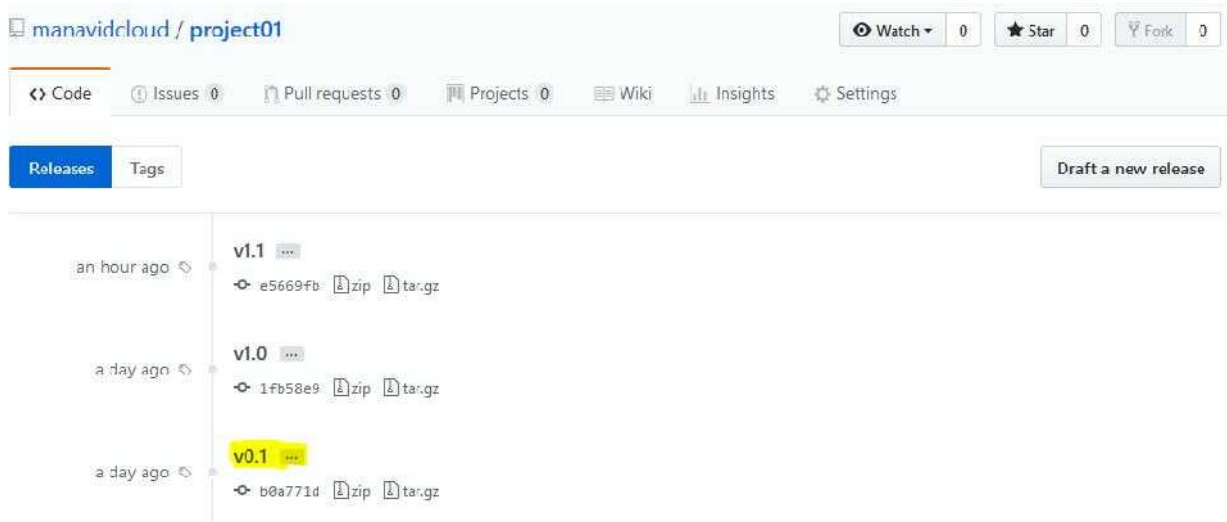
update

```
commit 24cf3a4d251786e93bbe8ba4fd76a55dfbf40502
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git push origin v0.1
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/manavidcloud/project01.git
* [new tag] v0.1 -> v0.1
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$
```



List log in pretty mode -

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git log --pretty=oneline
e5669fbe5338e8fb1ee580fc5c8e56b79b49804a (HEAD -> master, tag: v1.1) light01
b5eaa3c27e9ae1e80e44e256e64d927684f8389a (origin/master) new
6c288ad2ec4be8c71200e4093731e9f3276e9d79 update
f189e49d22491829a5c71b4dbc6d959e6fc84463 Latest
1fb58e9cf2e308ec93e740763b964867b010d8af (tag: v1.0) Line123_final
b0a771dcf03f09a8174e880ba3b61f0e05a18baa (tag: v0.1) Line12
0899200a33ed1f3e9e21d109ef0da190d659f97b Line1
49af660974bf50bc93ca02ac98f3cd660963a3b2 update
24cf3a4d251786e93bbe8ba4fd76a55dfbf40502 gitproject01 and we have create our first sample file
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$
```

To push all tags

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git log --pretty=oneline
e5669fbe5338e8fb1ee580fc5c8e56b79b49804a (HEAD -> master, tag: v1.1) light01
b5eaa3c27e9ae1e80e44e256e64d927684f8389a (origin/master) new
6c288ad2ec4be8c71200e4093731e9f3276e9d79 update
f189e49d22491829a5c71b4dbc6d959e6fc84463 Latest
1fb58e9cf2e308ec93e740763b964867b010d8af (tag: v1.0) Line123_final
b0a771dcf03f09a8174e880ba3b61f0e05a18baa (tag: v0.1) Line12
0899200a33ed1f3e9e21d109ef0da190d659f97b Line1
49af660974bf50bc93ca02ac98f3cd660963a3b2 update
24cf3a4d251786e93bbe8ba4fd76a55dfbf40502 gitproject01 and we have create our first sample file
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git tag v0.0 0899200a33
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git log --pretty=oneline
e5669fbe5338e8fb1ee580fc5c8e56b79b49804a (HEAD -> master, tag: v1.1) light01
b5eaa3c27e9ae1e80e44e256e64d927684f8389a (origin/master) new
6c288ad2ec4be8c71200e4093731e9f3276e9d79 update
f189e49d22491829a5c71b4dbc6d959e6fc84463 Latest
1fb58e9cf2e308ec93e740763b964867b010d8af (tag: v1.0) Line123_final
b0a771dcf03f09a8174e880ba3b61f0e05a18baa (tag: v0.1) Line12
0899200a33ed1f3e9e21d109ef0da190d659f97b (tag: v0.0) Line1
49af660974bf50bc93ca02ac98f3cd660963a3b2 update
24cf3a4d251786e93bbe8ba4fd76a55dfbf40502 gitproject01 and we have create our first sample file
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
```

```
$ git push origin --tags
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/manavidcloud/project01.git
* [new tag]          v0.0 -> v0.0
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$
```

Delete tag locally

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git log --pretty=oneline
e5669fbe5338e8fb1ee580fc5c8e56b79b49804a (HEAD -> master, tag: v1.1) light01
b5eaa3c27e9ae1e80e44e256e64d927684f8389a (origin/master) new
6c288ad2ec4be8c71200e4093731e9f3276e9d79 update
f189e49d22491829a5c71b4dbc6d959e6fc84463 Latest
1fb58e9cf2e308ec93e740763b964867b010d8af (tag: v1.0) Line123_final
b0a771dcf03f09a8174e880ba3b61f0e05a18baa (tag: v0.1) Line12
0899200a33ed1f3e9e21d109ef0da190d659f97b (tag: v0.0) Line1
49af660974bf50bc93ca02ac98f3cd660963a3b2 update
24cf3a4d251786e93bbe8ba4fd76a55dfbf40502 gitproject01 and we have create our first sample file
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git tag
v0.0
v0.1
v1.0
v1.1
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git tag -d v0.0
Deleted tag 'v0.0' (was 0899200)
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git tag
v0.1
v1.0
v1.1
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git log --pretty=oneline
e5669fbe5338e8fb1ee580fc5c8e56b79b49804a (HEAD -> master, tag: v1.1) light01
b5eaa3c27e9ae1e80e44e256e64d927684f8389a (origin/master) new
6c288ad2ec4be8c71200e4093731e9f3276e9d79 update
f189e49d22491829a5c71b4dbc6d959e6fc84463 Latest
1fb58e9cf2e308ec93e740763b964867b010d8af (tag: v1.0) Line123_final
b0a771dcf03f09a8174e880ba3b61f0e05a18baa (tag: v0.1) Line12
0899200a33ed1f3e9e21d109ef0da190d659f97b Line1
49af660974bf50bc93ca02ac98f3cd660963a3b2 update
24cf3a4d251786e93bbe8ba4fd76a55dfbf40502 gitproject01 and we have create our first sample file
```

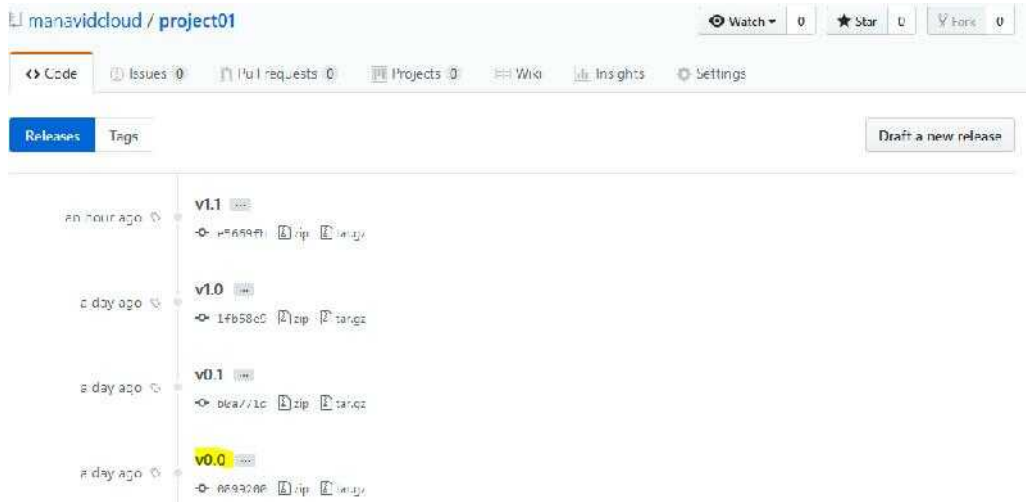
```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$
```

Delete tag from origin

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git push origin :refs/tags/v0.0
To https://github.com/manavidcloud/project01.git
- [deleted]          v0.0
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$
```

Before -



After -



Fetch code/tags from remote which removed in local

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git tag -l
v0.1
v1.0
v1.1
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git tag -d $(git tag -l)
Deleted tag 'v0.1' (was b0a771d)
Deleted tag 'v1.0' (was 62fdb37)
Deleted tag 'v1.1' (was e5669fb)
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git tag -l
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git fetch
From https://github.com/manavidcloud/project01
* [new tag]          v0.1      -> v0.1
* [new tag]          v1.0      -> v1.0
* [new tag]          v1.1      -> v1.1
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git tag -l
v0.1
v1.0
v1.1
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
```

\$

Delete remote tags and recover it

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git tag -l
v0.1
v1.0
v1.1

Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git push origin --delete $(git tag -l) # can use -d instead of --delete
To https://github.com/manavidcloud/project01.git
- [deleted]          v0.1
- [deleted]          v1.0
- [deleted]          v1.1

Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
```

Check your tags from remote should be deleted.

Now recover it with the help of local available tags.

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git tag -l
v0.1
v1.0
v1.1

Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git push origin --tags
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 446 bytes | 74.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/manavidcloud/project01.git
* [new tag]          v0.1 -> v0.1
* [new tag]          v1.0 -> v1.0
* [new tag]          v1.1 -> v1.1

Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$
```

Scenario

If you delete local tags and try to push tags -d it will shows an error. How we can delete local and remote tags.

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git push origin --delete $(git tag -l)
fatal: --delete doesn't make sense without any refs

Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git push origin --tags
Everything up-to-date

Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git fetch
From https://github.com/manavidcloud/project01
* [new tag]          v0.1 -> v0.1
* [new tag]          v1.0 -> v1.0
* [new tag]          v1.1 -> v1.1

Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git push origin --delete $(git tag -l)
To https://github.com/manavidcloud/project01.git
- [deleted]          v0.1
```



```
- [deleted]          v1.0
- [deleted]          v1.1
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git tag -l
v0.1
v1.0
v1.1
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git tag -d $(git tag -l)
Deleted tag 'v0.1' (was b0a771d)
Deleted tag 'v1.0' (was 62fdb37)
Deleted tag 'v1.1' (was e5669fb)
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git tag -l
```

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$
```

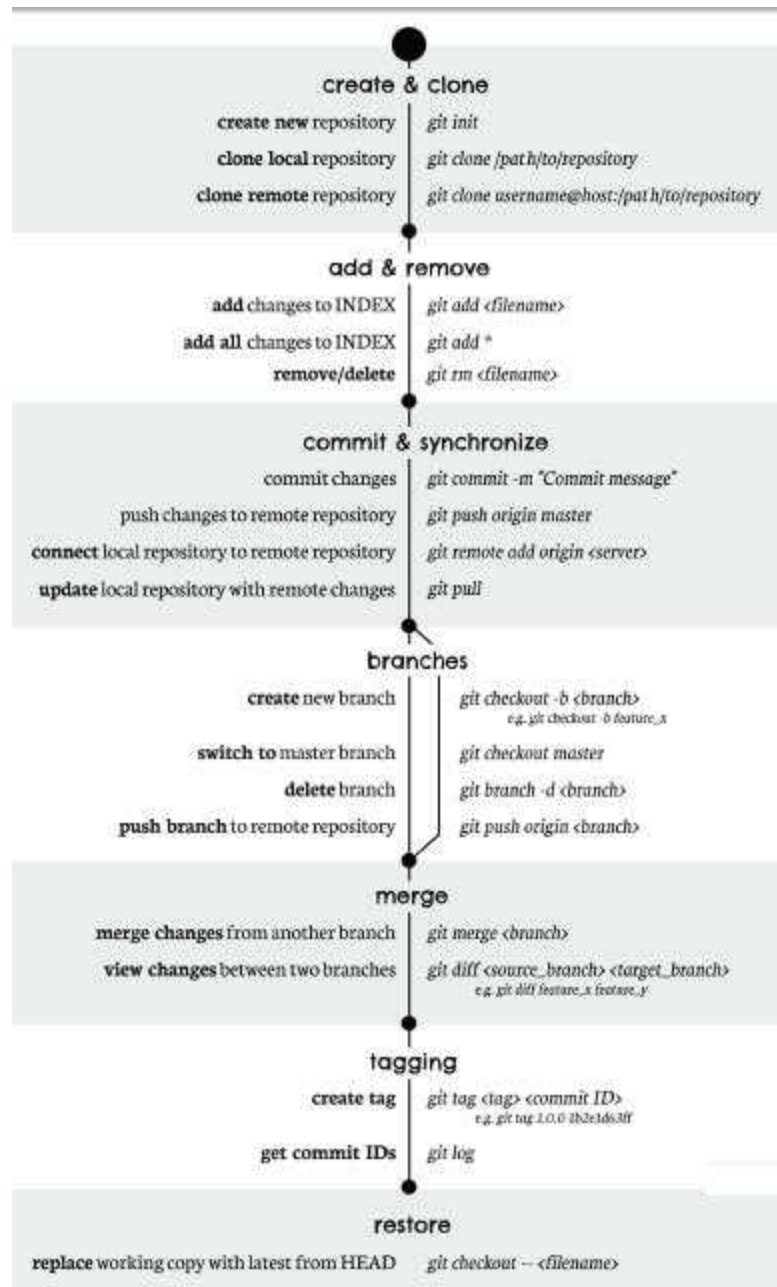
List remote tags with refs

```
Ali_baba@alibaba MINGW64 /gitproject/project01 (master)
$ git ls-remote --tags origin
6c288ad2ec4be8c71200e4093731e9f3276e9d79      refs/tags/v9.0
```

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

Remote deletes

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



Troubleshooting:

Git: How to avoid typing username & password when using HTTPS Authentication

```
ali_baba@alibaba MINGW64 /testenv (master)
$ git config --list | grep -i store
```

Run below command to solve the issue -

```
ali_baba@alibaba MINGW64 /testenv (master)
$ git config credential.helper store
```

```
ali_baba@alibaba MINGW64 /testenv (master)
$ git config --list | grep -i store
credential.helper=store
```

```
ali_baba@alibaba MINGW64 /testenv (master)
$
```

=====

Please make sure you have the correct access rights and asking username/password everytime.

```
ali_baba@alibaba MINGW64 /testenv (master)
$ git remote show origin
fatal: Could not read from remote repository.
```

Please make sure you have the correct access rights
and the repository exists.

```
ali_baba@alibaba MINGW64 /testenv (master)
```

Solution:

1. It was because of no SSH key on the machine. Check the SSH key locally:

```
ali_baba@alibaba MINGW64 /testenv (master)
$ cat ~/.ssh/id_rsa.pub
cat: /c/Users/ali_baba/.ssh/id_rsa.pub: No such file or directory
```

```
ali_baba@alibaba MINGW64 /testenv (master)
$
```

2. Generate the key from user home direcoty -

```
ali_baba@alibaba MINGW64 /testenv (master)
$ cd ~
```

```
ali_baba@alibaba MINGW64 ~
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/ali_baba/.ssh/id_rsa):
Created directory '/c/Users/ali_baba/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/ali_baba/.ssh/id_rsa.
Your public key has been saved in /c/Users/ali_baba/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:tlyJm0wR20FyDa71MLBkfFXZunABhgLWAJ1LV7HK34 ali_baba@alibaba
The key's randomart image is:
+---[RSA 3072]-----+
|++=O=Bo*+.+.|
|.O.O*+. =B.|
|. O*+.O=O.|
|...O.+ O..|
```

```

| .o. S o |
| ..E= = |
| . * |
+----[SHA256]-----+

```

ali_baba@alibaba MINGW64 ~

```
$ cat ~/.ssh/id_rsa.pub
ssh-rsa
```

```

AAAAAB3NzaC1yc2EAAAADAQABAAQGC6w1NRboY4yUSMcd705oVKnV0dAwB8fSGYSL9NPdxeiB26QonawMaZYotk6ag/qXCBVw+UoOZ
pfPdpBpAcB3lC309sccwQC9RRFhBhfYt98lfM4UK7ekChCtGIA486bvugSwc348aIA4EvuIZi4WJYOznacZXKQwbVTBay8vMAS7vtt8Q
t8c7z0R1Ykj4lVH2YCBazr1fTjr4xxppknrfCdp4bbjNElSwjCAY8vua16uwvxfz8gS7cQ+QS4wTyKPF39YmNuez8ZFGqHt56pvpExS7
OGDQFJm+NU+TBFL3t7/DYX8+

```

ali_baba@alibaba MINGW64 ~

3. Add the generated key in <https://github.com/settings/keys>

4. Check again it ust fixed the issue.

ali_baba@alibaba MINGW64 /testenv (master)

```

$ git remote show origin
* remote origin
  Fetch URL: https://github.com/manavidcloud/DevOps.git
  Push URL: https://github.com/manavidcloud/DevOps.git
  HEAD branch: master
  Remote branch:
    master tracked
  Local ref configured for 'git push':
    master pushes to master (up to date)

```

ali_baba@alibaba MINGW64 /testenv (master)

```
$
```

Reference(s) -

<https://git-scm.com/>

<https://github.com/>*

<http://google.com/>*

Pending:

TortoiseGit

<https://help.github.com/en/articles/enabling-branch-restrictions>

gitlab

Why do I need to do `--set-upstream` all the time?

Set default branch for user in git ?

Thank You

Navid..Bus Naam Hi Kaafi Hai