**Git clone** :: clones the repo to the working directory.(Need not to be .git repo) **.GIT**

**Git fetch:::: to the local repo (.Git is must) Copies the contents to the local created repo(.git)**

**Git remote ::**

**If you want to see the original existence of your cloned repository,** use the git remote command. It can be used as:

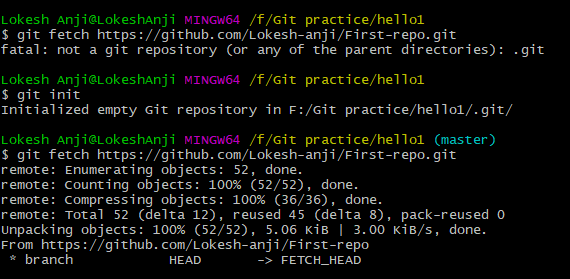
Syntax:

1. $ git remote

Output:

Git Remote

The given command is providing the remote name as the origin. Origin is the default name for the remote server, which is given by Git.



**Git status**:: It shows what is the change made in the file and that are to be add and yet to commit to the local repo ,,so that from there it can be pushed to Central repo..

$ **git diff** :::: It shows the diff in the contents of a file from last commit to the current.

**diff --git a/info.txt b/info.txt**

**index b944917..6c58dc4 100644**

**--- a/info.txt**

**+++ b/info.txt**

@@ -1,2 +1,4 @@

My name is Lokesh

-i'm from Bellary

\ No newline at end of file

+i'm from Bellary

+

+I'm currently working in Capgemini"

\ No newline at end of file

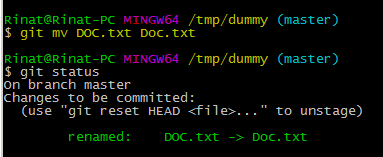
Lokesh Anji@LokeshAnji MINGW64 ~/test (master)

--------------------------------------------------------

**git config --global user.username Lokesh-anji**

<https://dev.to/dhruv/essential-git-commands-every-developer-should-know-2fl>

**Git mv ::** move or rename the files and add changes to the staging area9



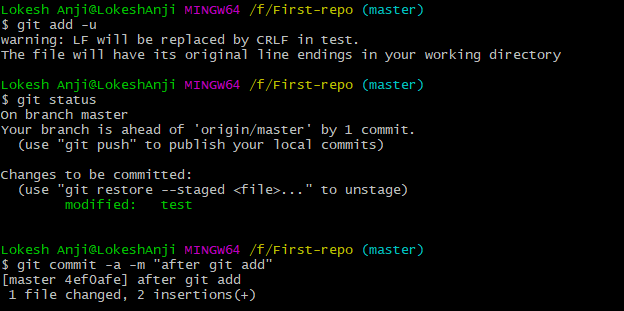
**Git rm::** delete the files and add changes to the staging area **(rm –rf + git add)**

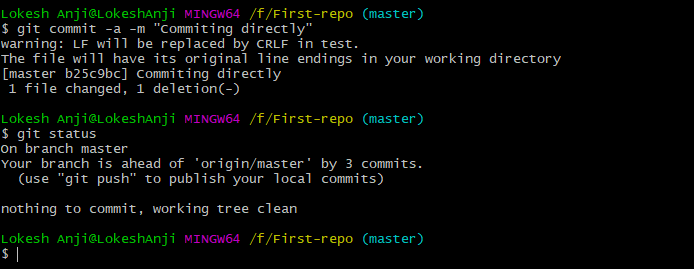
**Git checkout –b**:: to create branch and to jump and on to it

Git config ::



The illustration shows that “git add” is the only operation that can cause the index to diverge from the local repository. The only reason to use “git add” is so that “git commit” will see your changes. The -a option to “git commit” causes “git commit” to run “git add -u” first – in which case you never need to run "git add -u” explicitly – in which case the index stays in sync with the repository head. This is how the trick in “git without the index” works: if you always use commit via “git commit -a”, you can ignore the index





**Note**:: Git will not show or identify if the newly created folder/file doesn’t contain files or data in it.

**Note**:: Auto merge will happen when two developers working on diff files.

Fast-forwarding

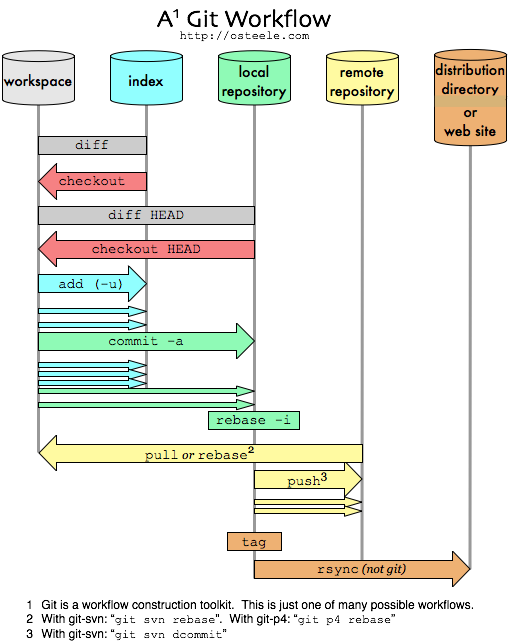
**Note**:: whenever we create a branch from the current branch ,whatever the contents present at this branch will copy accordingly.

**Note**:: Whenever will pull code from github and assume their it has 3 branches ,and when we do clone,fetch or pull ,by default GIT will create a Master branch and it’ll put on it and the other branches won’t be their. **((This applies only when pulling the code for the first time from Github))**

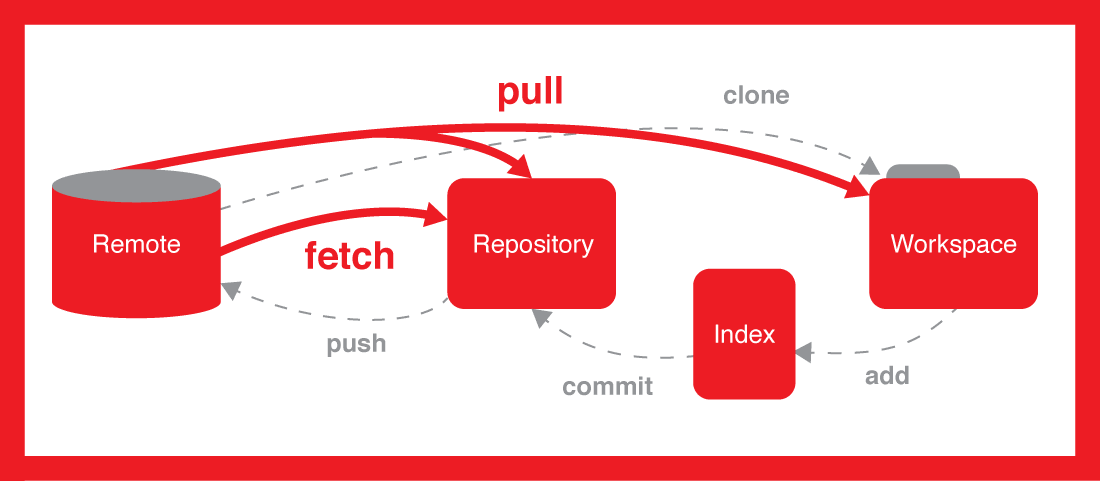
**Note::** When we create a file on one branch and if checkout to the other branch without adding to the staging area/committing to the local repo ,then that file will be available to all other branches in that local repo.

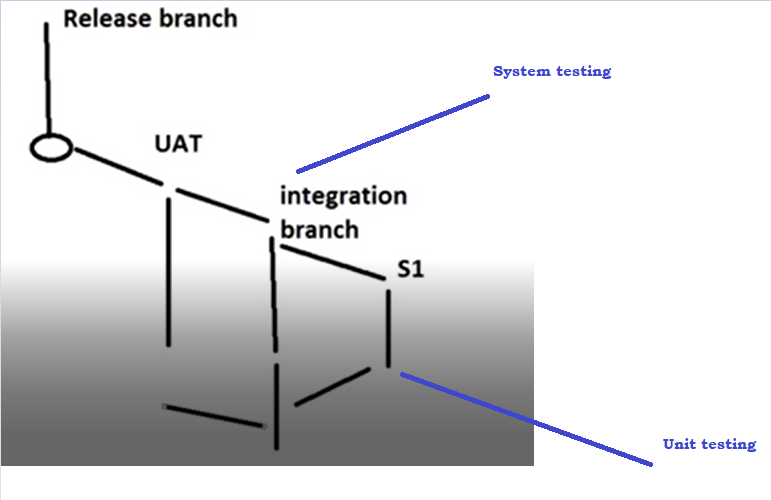
**Tag** :: giving some label to the commit

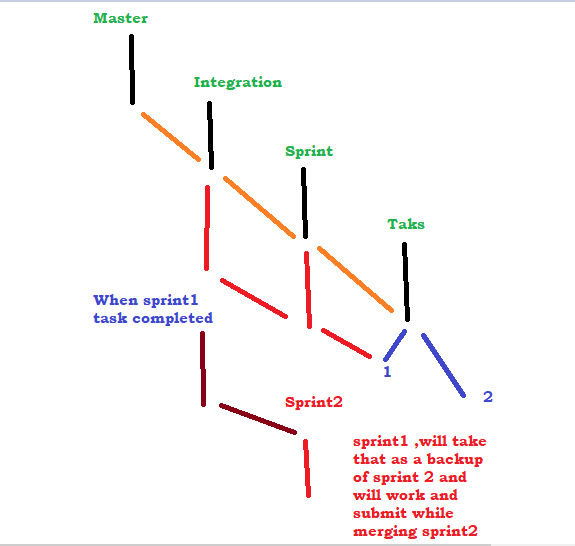
**.Git🡪 config file (origin)**



[enter image description here](https://i.stack.imgur.com/zUInQ.png)

[](https://i.stack.imgur.com/nWYnQ.png)





Blob—file

Tree---folder