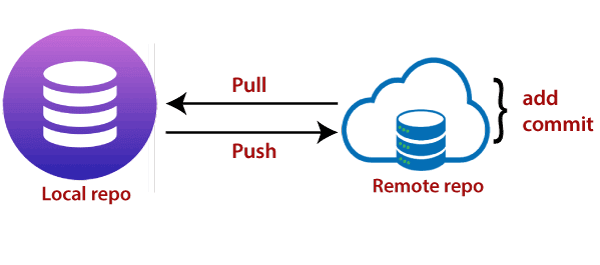
Git Push

The push term refers to upload local repository content to a remote repository. Pushing is an act of transfer commits from your local repository to a remote repository. Pushing is capable of overwriting changes; caution should be taken when pushing.



Moreover, we can say the push updates the remote refs with local refs. Every time you push into the repository, it is updated with some interesting changes that you made. If we do not specify the location of a repository, then it will push to default location at **origin master**.

The "git push" command is used to push into the repository. The push command can be considered as a tool to transfer commits between local and remote repositories. The basic syntax is given below:

1. $ git push <option> [<Remote URL><branch name><refspec>...]

Push command supports many additional options. Some options are as follows under push tags.

Git Push Tags

**<repository>:** The repository is the destination of a push operation. It can be either a URL or the name of a remote repository.

**<refspec>:** It specifies the destination ref to update source object.

**--all:** The word "all" stands for all branches. It pushes all branches.

**--prune:** It removes the remote branches that do not have a local counterpart. Means, if you have a remote branch say demo, if this branch does not exist locally, then it will be removed.

**--mirror:** It is used to mirror the repository to the remote. Updated or Newly created local refs will be pushed to the remote end. It can be force updated on the remote end. The deleted refs will be removed from the remote end.

**--dry-run:** Dry run tests the commands. It does all this except originally update the repository.

**--tags:** It pushes all local tags.

**--delete:** It deletes the specified branch.

**-u:** It creates an upstream tracking connection. It is very useful if you are going to push the branch for the first time.

Git Push Origin Master

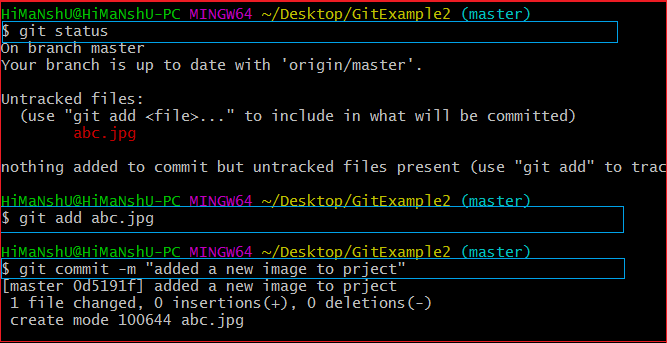
Git push origin master is a special command-line utility that specifies the remote branch and directory. When you have multiple branches and directory, then this command assists you in determining your main branch and repository.

Generally, the term **origin stands** for the remote repository, and master is considered as the main branch. So, the entire statement "**git push origin master**" pushed the local content on the master branch of the remote location.

**Syntax:**

1. $ git push origin master

Let's understand this statement with an example.

Let's make a new commit to my existing repository, say **GitExample2**. I have added an image to my local repository named **abc.jpg** and committed the changes. Consider the below image: 

In the above output, I have attached a picture to my local repository. The git status command is used to check the status of the repository. The git status command will be performed as follows:

1. $ git status

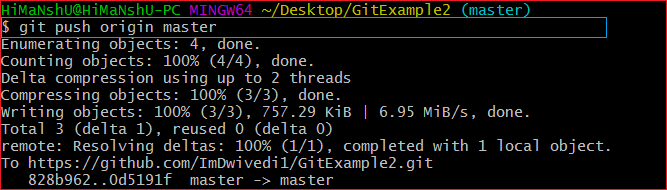
It shows the status of the untracked image **abc.jpg**. Now, add the image and commit the changes as:

1. $ git add abc.jpg
2. $git commit -m "added a new image to project."

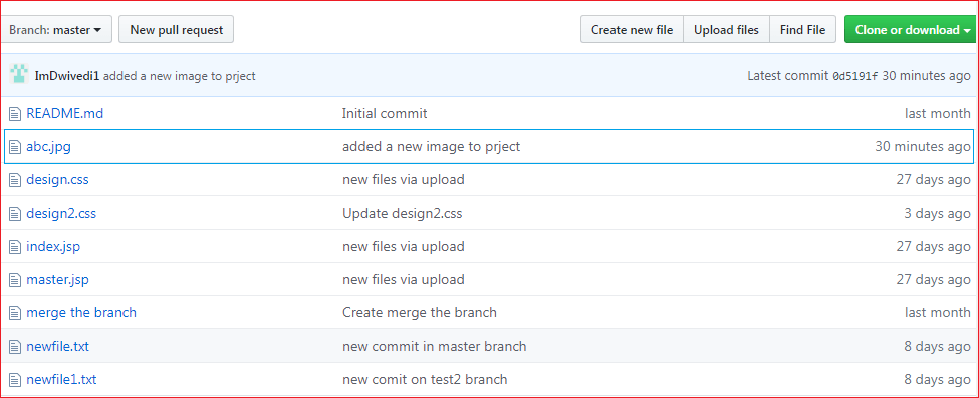
The image is wholly tracked in the local repository. Now, we can push it to origin master as:

1. $ git push origin master

**Output:**



The file **abc.jpg** is successfully pushed to the origin master. We can track it on the remote location. I have pushed these changes to my GitHub account. I can track it there in my repository. Consider the below image:



In the above output, the pushed file abc.jpg is uploaded on my GitHub account's master branch repository.

Git Force Push

The git force push allows you to push local repository to remote without dealing with conflicts. It is used as follows:

1. $ git push <remote><branch> -f

Or

1. $ git push <remote><branch> -force

The -f version is used as an abbreviation of force. The remote can be any remote location like GitHub, Subversion, or any other git service, and the branch is a particular branch name. For example, we can use git push origin master -f.

We can also omit the branch in this command. The command will be executed as:

1. $git push <remote> -f

We can omit both the remote and branch. When the remote and the branch both are omitted, the default behavior is determined by **push.default** setting of git config. The command will be executed as:

1. $ git push -f

How to Safe Force Push Repository:

There are several consequences of force pushing a repository like it may replace the work you want to keep. Force pushing with a lease option is capable of making fail to push if there are new commits on the remote that you didn't expect. If we say in terms of git, then we can say it will make it fail if remote contains untracked commit. It can be executed as:

1. $git push <remote><branch> --force-with-lease

Git push -v/--verbose

The -v stands for verbosely. It runs command verbosely. It pushed the repository and gave a detailed explanation about objects. Suppose we have added a **newfile2.txt** in our local repository and commit it. Now, when we push it on remote, it will give more description than the default git push. Syntax of push verbosely is given below:

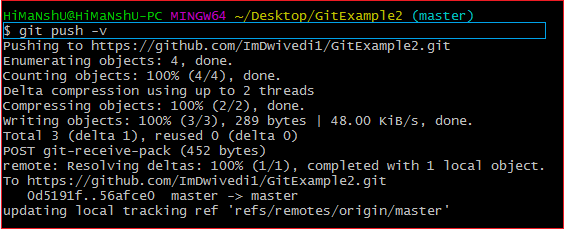
**Syntax:**

1. $ git push -v

Or

1. $ git push --verbose

Consider the below output:



If we compare the above output with the default git option, we can see that git verbose gives descriptive output.

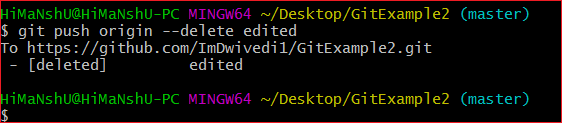
Delete a Remote Branch

We can delete a remote branch using git push. It allows removing a remote branch from the command line. To delete a remote branch, perform below command:

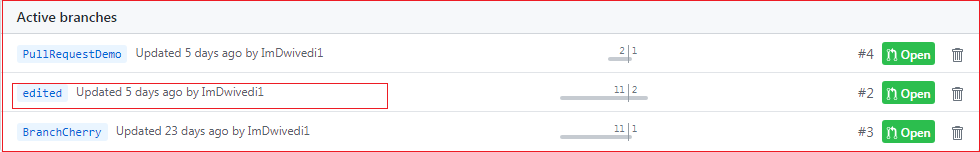
**Syntax:**

1. $ git push origin -delete edited

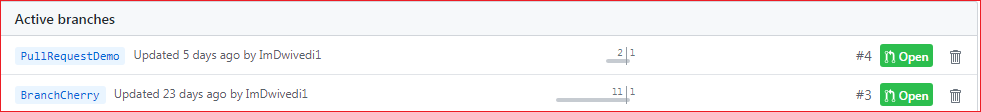
**Output:**



In the above output, the git push origin command is used with -delete option to delete a remote branch. I have deleted my remote branch **edited** from the repository. Consider the below image:



It is a list of active branches of my remote repository before the operating command.



The above image displays the list of active branches after deleting command. Here, you can see that the branch **edited** has removed from the repository.