**Git Command steps followed to push new changes.**

Go to Git after installing and configuring in your system you will have a command prompt window opened follow the steps below to push a change into the remote repository.

**Step 1:** Go to working directory using cd command

**syntax: cd C:/Suriya/newprct**

**Note: Check if it shows (Master) - meaning the directory/folder was configured as a repo.**

**Step 2:** Clone remote repository using Git clone command

**syntax: git clone https://github.com/sselva62/Remote-test.git**

**Note: The remote repo link can be HTTPS/SSH Link** - might require user passcode

**Step 3:** Use ls to find which directory you are at.

**syntax: ls**

**response:**

Remote-test/

**Step 4:** Use cd to go to the directory you want to work on.

**syntax: cd C:/Suriya/newprct/Remote-test**

Add any file to the working directory from your file manager/ make changes to existing file.

In this example, I have copied “**test.txt”** file from documents folder to the local cloned repo under **Remote-test** manually.

**Step 5:** Check the status of the modified/added file in the working directory.

**Syntax:** git status

**Normal response will be as follows:**

On branch master

Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

**Response if changes are made in the working directory:**

On branch master

Your branch is up to date with 'origin/master'.

Untracked files:

(use "git add <file>..." to include in what will be committed)

test.txt

nothing added to commit but untracked files present (use "git add" to track)

**Step 6:** Add file changed/created/added to the repo directory using Git add command.

**Syntax: git add test.txt (or) git add -A (**to add all changed files**).**

**Step 7:** Check if file is added by using git statuscommand.

**Syntax: git status**

**Response**

On branch master

Your branch is up to date with 'origin/master'.

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

new file: test.txt

**Step 7:**  Commit changes to your cloned repo using git commit command. (It’s like saving the changes for your cloned repo only).

**Syntax: git commit -m “Type any commit msg example: Test v2”**

**Response:** [master 485436f] A test for commit

1 file changed, 1 insertion(+)

create mode 100644 test.txt

**Step 8:** Push the changes to the master branch of the remote repo in this case the git hub repository.

**Syntax: git push -u origin master**

**Note: It will ask for password to login to the repo your id and private password.**

**Response after providing password:**

Enumerating objects: 4, done.

Counting objects: 100% (4/4), done.

Delta compression using up to 8 threads

Compressing objects: 100% (2/2), done.

Writing objects: 100% (3/3), 325 bytes | 325.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

To https://github.com/sselva62/Remote-test.git

9b8cc9d..485436f master -> master

Branch 'master' set up to track remote branch 'master' from 'origin'.

**Step 9:** check if the changes reflected in the remote repo if not shown, refresh to see if the changes are reflecting in there.

Most of the steps followed here is the same as working with Linux commands like to create a new directory (Folder) we use mkdir/md command

Example:

**mkdir test directory.**

**Note:** This command should be given from the working directory/ from the place in which you want to create the directory.