# Git Commands Cheat Codes

## CHAPTER ONE : CREATING NEW REPO AND COMMTING CODE TO IT.

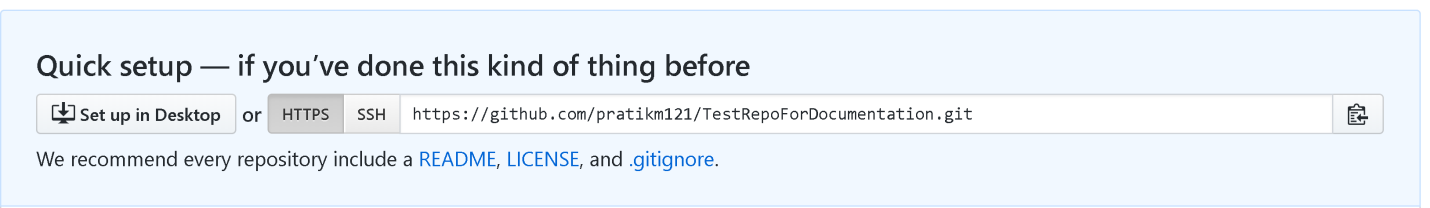
STEP 1 : Log into your GitHub account and create a new repository

<https://github.com/new>

Repo Name :- TestRepoForDocumentation

Description :- Repo created for documentation purpose

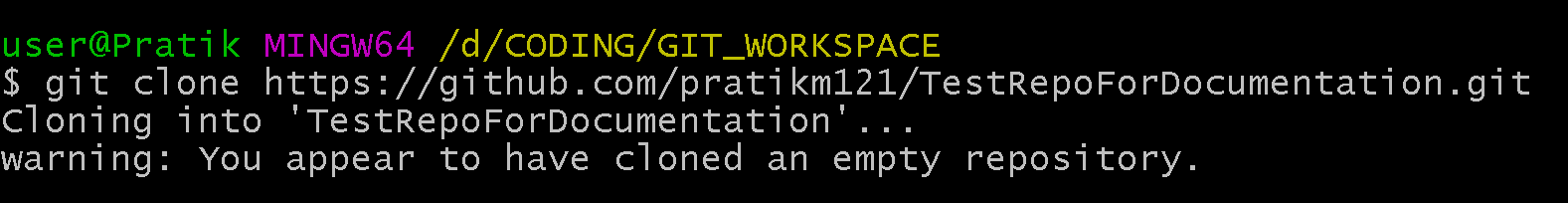
Then click Create repository.



STEP 2 : Get clone repository from GitHub

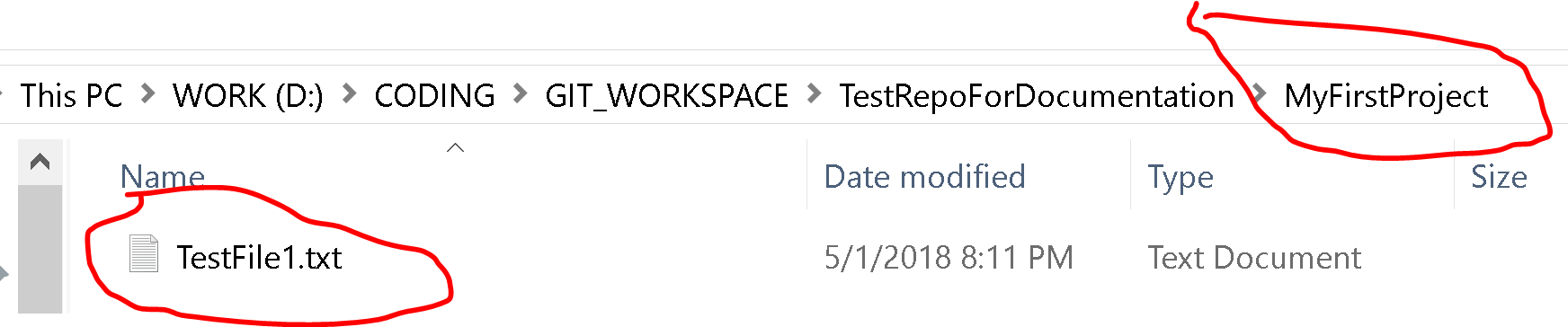
git clone <https://github.com/pratikm121/TestRepoForDocumentation.git>

1. Change to the directory where you to want to clone your repo and right click open Git Bash here.
2. Clone your repo into local machine



This will make a local copy of your remote repository

1. Create your project and add files to them and make changes



1. Once all the changes are made we will stage the changes to pushed.First change the directory to the project folder under the repo

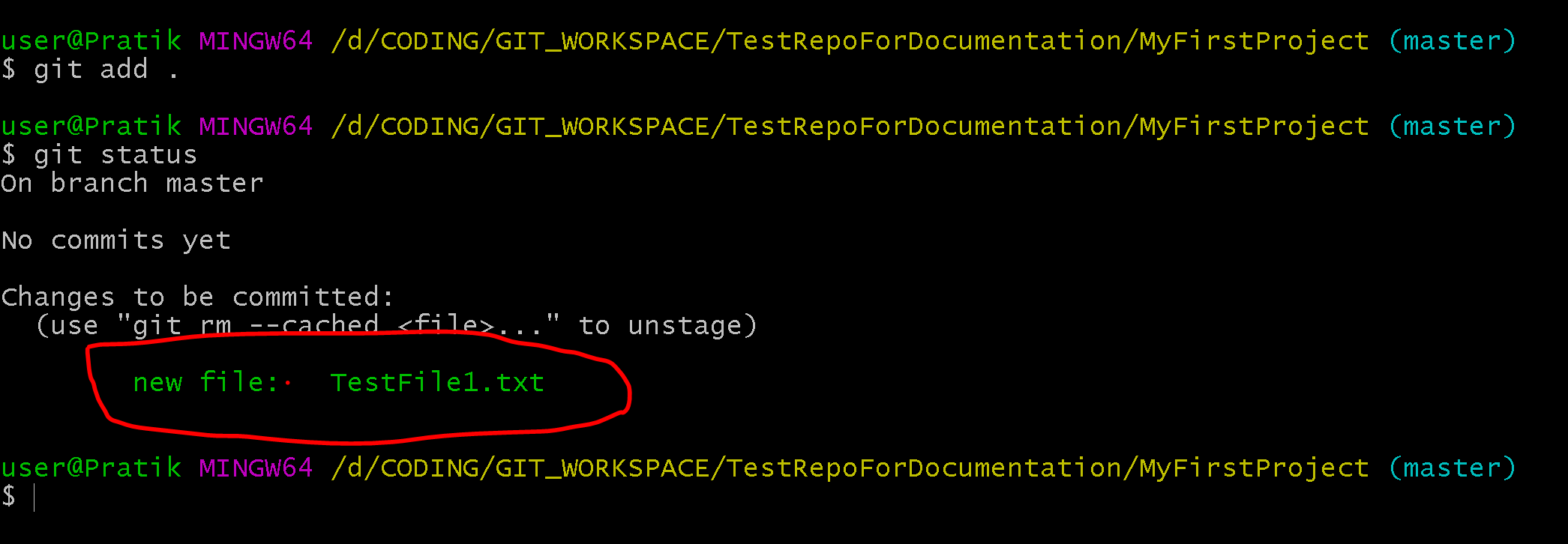


1. Add all the files to the staging area

**git add .**

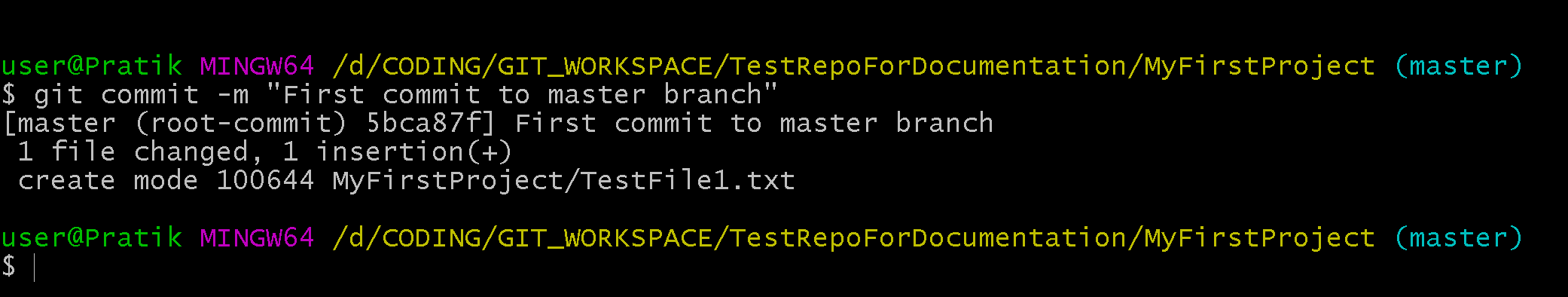
1. Check the status of the added files (optional)

**git status**



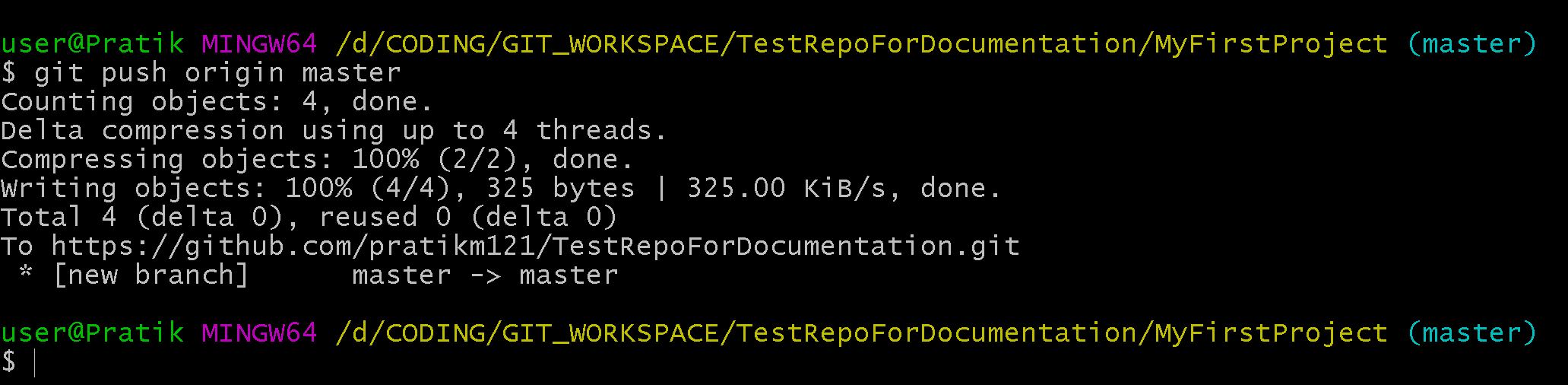
1. Then commit the changes with a message .

**git commit -m "First commit to master branch"**



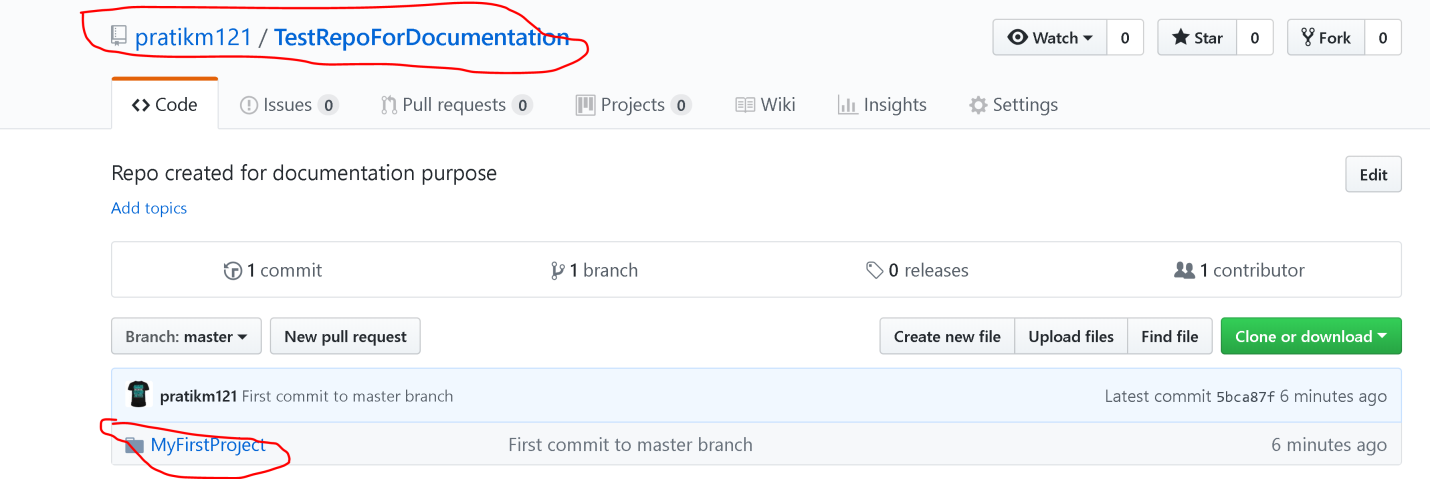
1. Now the changes are commited to local repo in your machine. Next we need to push the changes to your remote repo in GitHub.com

**git push origin master**



This will sync your remote repo with your local repo.

Refresh your remote repo page and check if the change reflects or not.



## CHAPTER TWO : CREATING NEW BRANCH AND COMMTING CODE TO IT.

STEP 1 : Creating a branch and moving the code to branch

a. Create a new branch name R\_XXX\_Manvi

git branch R\_XXX\_Manvi

b. Switch to the new branch from master branch

This tutorial provides a list of the most common GIT commands, a short description for them and example usage. For a detailed description of all the GIT commands please check this page.

git config

Sets configuration values for your user name, email, gpg key, preferred diff algorithm, file formats and more. Examples:

git config --global user.name "My Name"

git config --global user.email "user@domain.com"

========================================================================

git init

Initializes a git repository – creates the initial .git directory in a new or in an existing project. Example:

git init

Initialized empty Git repository in /home/username/GIT/.git/

========================================================================

git clone

Creates a GIT repository copy from a remote source. Also adds the original location as a remote so you can fetch from it again and push to it if you have permissions. Example:

git clone git@github.com:user/test.git

========================================================================

git add

Adds files changes in your working directory to your index. Example:

git add .

========================================================================

git rm

Removes files from your index and your working directory so they will not be tracked. Example:

git rm filename

========================================================================

git commit

Takes all of the changes written in the index, creates a new commit object pointing to it and sets the branch to point to that new commit. Examples:

git commit -m ‘committing added changes’

git commit -a -m ‘committing all changes, equals to git add and git commit’

========================================================================

git status

Shows you the status of files in the index versus the working directory. It will list out files that are untracked (only in your working directory), modified (tracked but not yet updated in your index), and staged (added to your index and ready for committing). Example:

git status

# On branch master #

# Initial commit #

# Untracked files: #

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

README

========================================================================

git branch

Lists existing branches, including remote branches if ‘-a’ is provided. Creates a new branch if a branch name is provided. Example:

git branch -a \* master remotes/origin/master

========================================================================

git merge

Merges one or more branches into your current branch and automatically creates a new commit if there are no conflicts. Example:

git merge newbranchversion

========================================================================

git reset

Resets your index and working directory to the state of your last commit. Example:

git reset --hard HEAD

========================================================================

git tag

Tags a specific commit with a simple, human readable handle that never moves. Example:

git tag -a v1.0 -m 'this is version 1.0 tag'

========================================================================

git pull

Fetches the files from the remote repository and merges it with your local one. Example:

git pull origin

========================================================================

git push

Pushes all the modified local objects to the remote repository and advances its branches. Example:

git push origin master

========================================================================

git remote

Shows all the remote versions of your repository. Example:

git remote origin

========================================================================

git log

Shows a listing of commits on a branch including the corresponding details. Example:

git log commit

84f241e8a0d768fb37ff7ad40e294b61a99a0abe Author: User <user@domain.com> Date: Mon May 3 09:24:05 2010 +0300 first commit

========================================================================

git diff

Generates patch files or statistics of differences between paths or files in your git repository, or your index or your working directory. Example:

git diff

========================================================================

git archive

Creates a tar or zip file including the contents of a single tree from your repository. Example:

1

git archive --format=zip master^ README >file.zip

========================================================================

git gc

Garbage collector for your repository. Optimizes your repository. Should be run occasionally. Example:

git gc

Counting objects: 7, done.

Delta compression using up to 2 threads.

Compressing objects: 100% (5/5), done.

Writing objects: 100% (7/7), done.

Total 7 (delta 1), reused 0 (delta 0)

========================================================================

git fsck

Does an integrity check of the Git file system, identifying corrupted objects. Example:

git fsck

========================================================================

git prune

Removes objects that are no longer pointed to by any object in any reachable branch. Example:

git prune

## Delete a remote branch

$ git push origin --delete <branch> # Git version 1.7.0 or newer

$ git push origin :<branch> # Git versions older than 1.7.0

## Delete a local branch

$ git branch --delete <branch>

$ git branch -d <branch> # Shorter version

$ git branch -D <branch> # Force delete un-merged branches