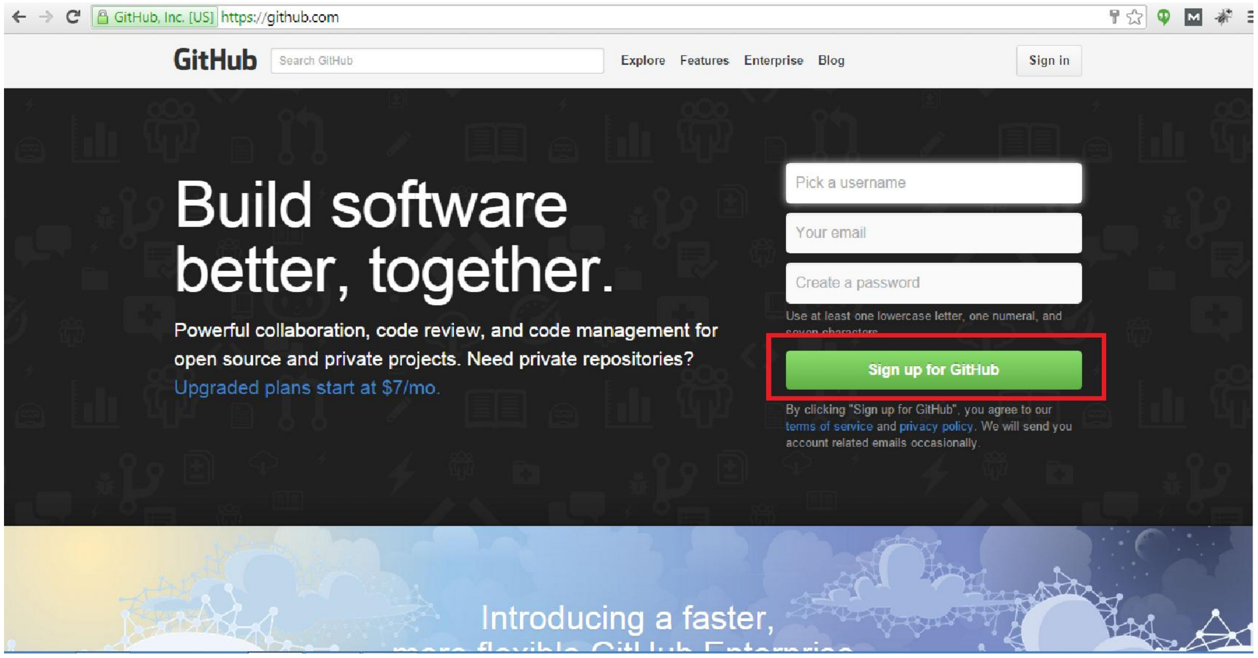
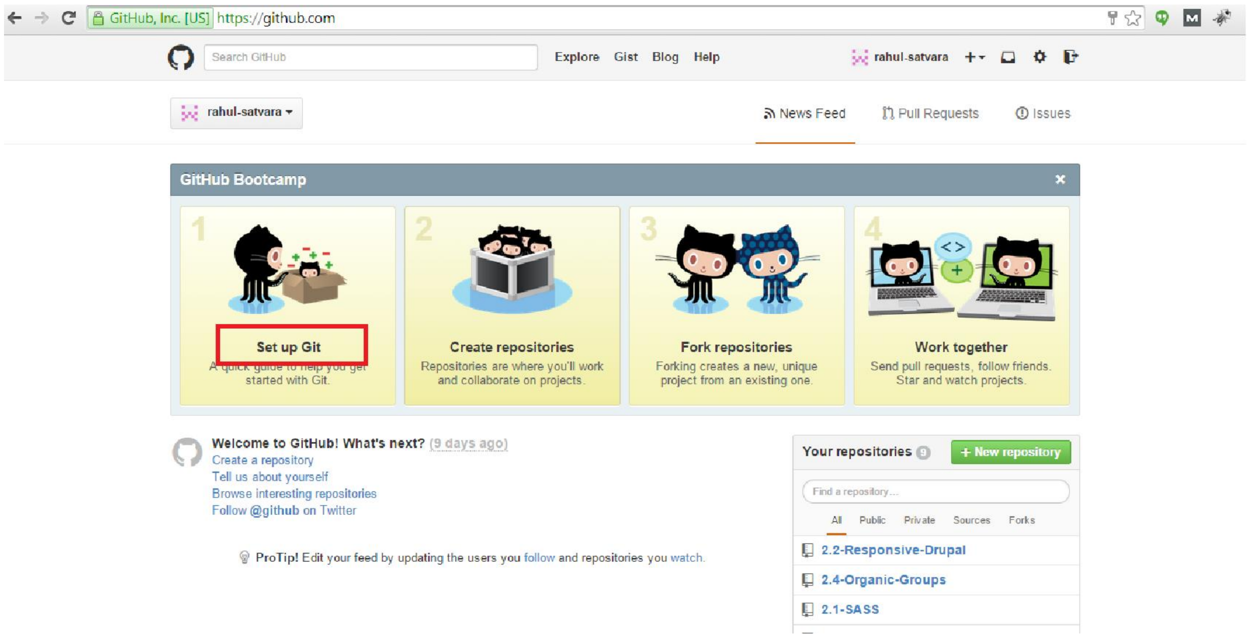
1. First create account in git go to https://github.com/ And Give username, email and password and click on signup.

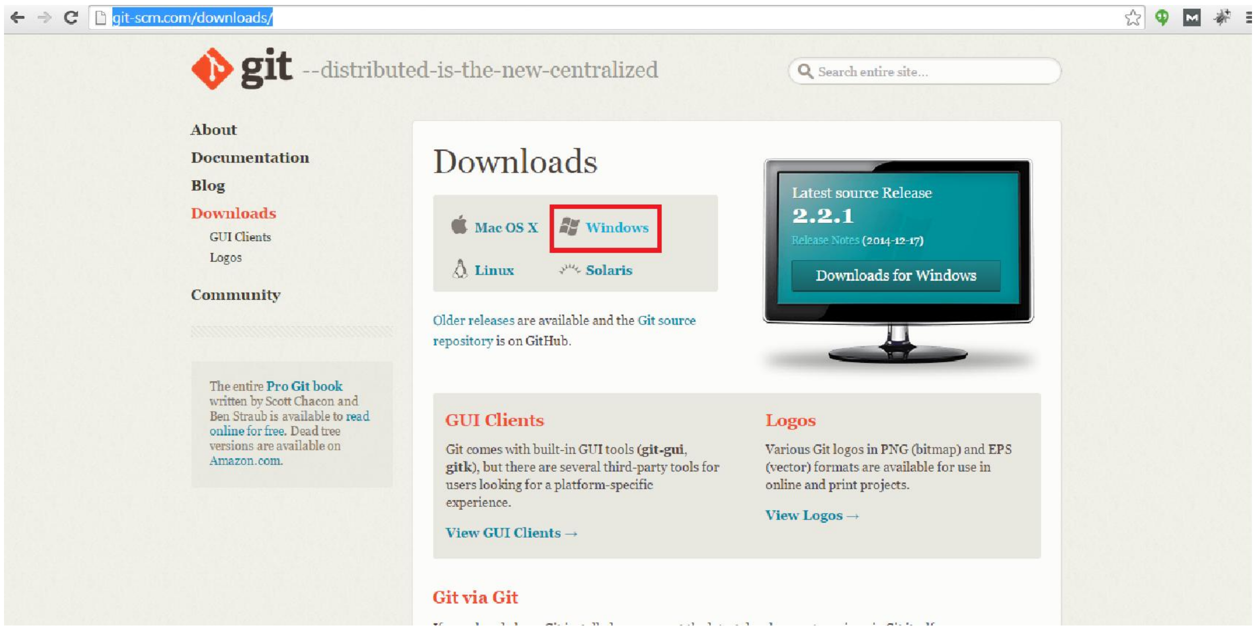

GitHub, Inc. [US] https://github.com
GitHub
Search GitHub
Explore
Features
Enterprise
Blog
Sign in
Build software better, together.
Powerful collaboration, code review, and code management for open source and private projects. Need private repositories?
Upgraded plans start at $7/mo.
Pick a username
Your email
Create a password
Use at least one lowercase letter, one numeral, and seven characters.
Sign up for GitHub
By clicking "Sign up for GitHub", you agree to our terms of service and privacy policy. We will send you account related emails occasionally.
Introducing a faster,

    a. Then sign in into your account.
    b. After sign-in you will get this screen.


GitHub, Inc. [US] https://github.com
Search GitHub
Explore
Gist
Blog
Help
rahul-satvara
News Feed
Pull Requests
Issues
GitHub Bootcamp
Set up Git
A quick guide to help you get started with Git.
Create repositories
Repositories are where you'll work and collaborate on projects.
Fork repositories
Forking creates a new, unique project from an existing one.
Work together
Send pull requests, follow friends. Star and watch projects.
Welcome to GitHub! What's next? (9 days ago)
Create a repository
Tell us about yourself
Browse interesting repositories
Follow @github on Twitter
ProTip! Edit your feed by updating the users you follow and repositories you watch.
Your repositories 9
+ New repository
Find a repository...
All
Public
Private
Sources
Forks
2.2-Responsive-Drupal
2.4-Organic-Groups
2.1-SASS

    c. Click on **Set up Git** it will show you some basic setup guide to install git.

2. Second step is to go to this site http://git-scm.com/downloads/ and download git terminal so you can run commands for git.
   You can see deferent version for deferent os. So you can download for mac, windowns or linux etc.
3. I'm working on windows so I downloaded setup for windows version.



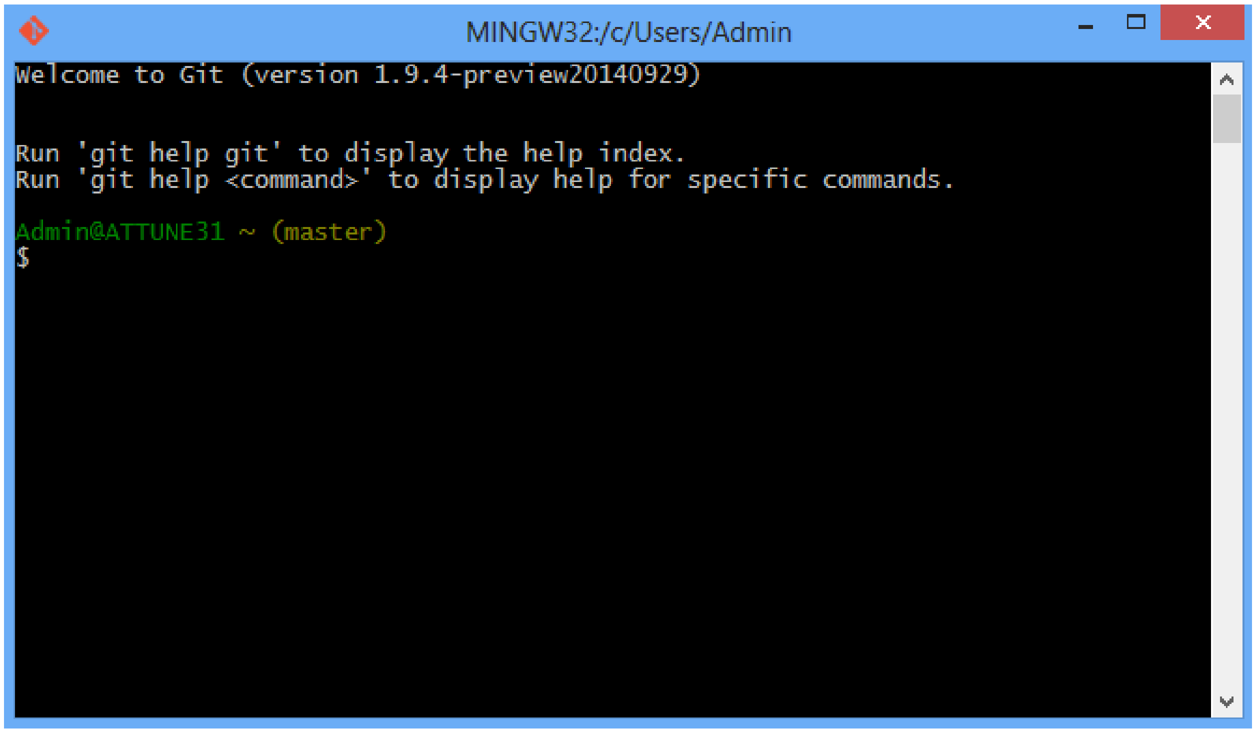
The entire Pro Git book written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.

Older releases are available and the Git source repository is on GitHub.

Git comes with built-in GUI tools (git-gui, gitk), but there are several third-party tools for users looking for a platform-specific experience.

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

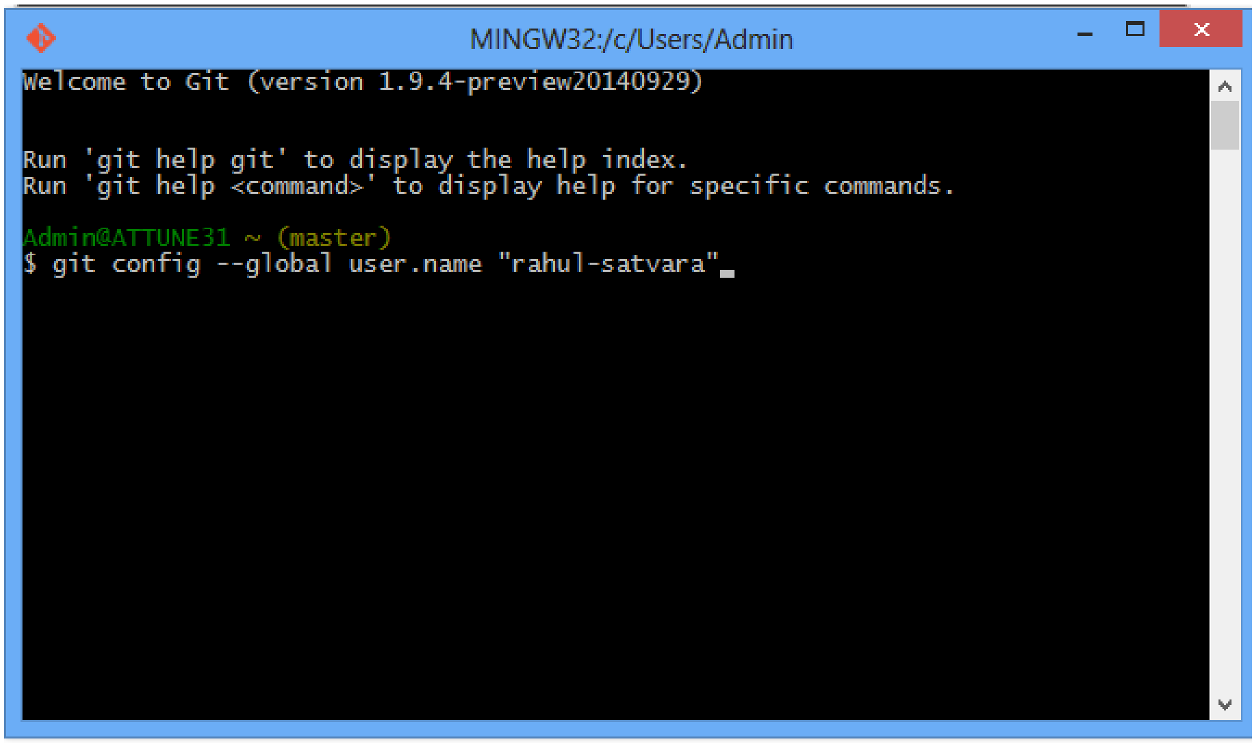
4. After downloading git software setup bobble click on it and install it.
5. After installation On your computer, open the **Git Shell** application.



```
Welcome to Git (version 1.9.4-preview20140929)


Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

Admin@ATTUNE31 ~ (master)
$
```

6. Next Setting up git(you can see steps on https://help.github.com/articles/set-up-git/).
    a. Tell Git your *name* so your commits will be properly labeled. Type everything after the `$` here:

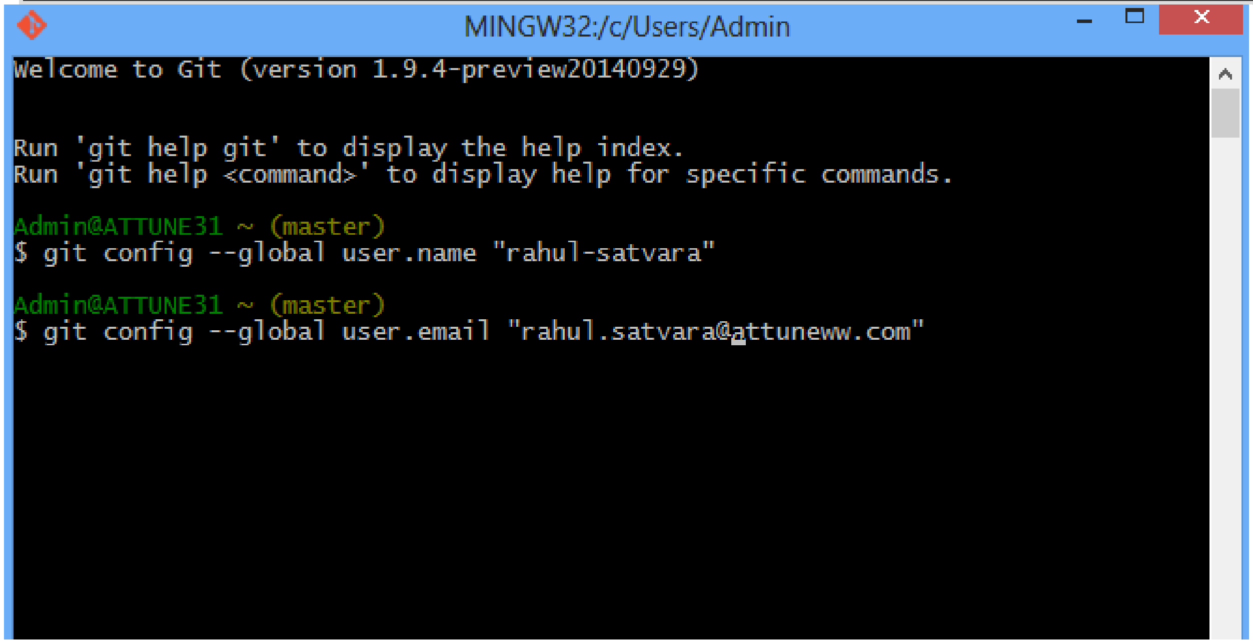
```
git config --global user.name "YOUR NAME"
```



```
Welcome to Git (version 1.9.4-preview20140929)


Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

Admin@ATTUNE31 ~ (master)
$ git config --global user.name "rahul-satvara"
```

b. Tell Git the *email address* that will be associated with your Git commits. The email you specify should be the same one found in your email settings. To keep your email address hidden, see "Keeping your email address private".

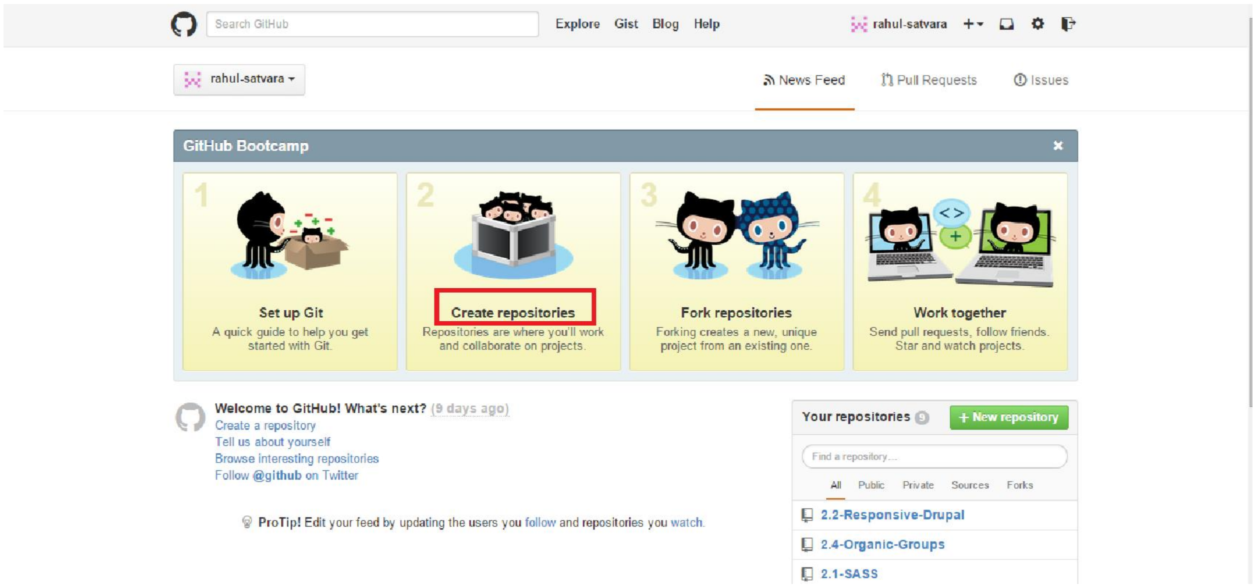
```
git config --global user.email "YOUR EMAIL ADDRESS"
```



```
Welcome to Git (version 1.9.4-preview20140929)


Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

Admin@ATTUNE31 ~ (master)
$ git config --global user.name "rahul-satvara"

Admin@ATTUNE31 ~ (master)
$ git config --global user.email "rahul.satvara@attuneww.com"
```

c. Now our git is set up.

7. After setup git we are going to create repository.
   Repository is where your all work collaborates on projects.
   I think I've repository as a sort of folder that is uploaded onto the github website and it contains all the code projects.



1 Set up Git
A quick guide to help you get started with Git.

2 Create repositories
Repositories are where you'll work and collaborate on projects.

3 Fork repositories
Forking creates a new, unique project from an existing one.

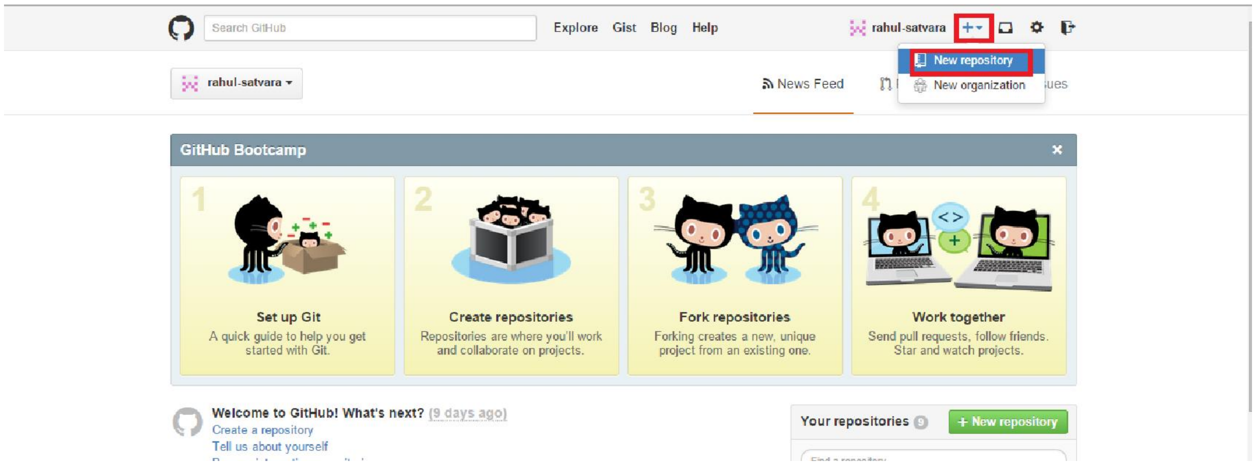
4 Work together
Send pull requests, follow friends. Star and watch projects.

Welcome to GitHub! What's next? (9 days ago)
Create a repository
Tell us about yourself
Browse interesting repositories
Follow @github on Twitter

ProTip! Edit your feed by updating the users you follow and repositories you watch.

2.2-Responsive-Drupal
2.4-Organic-Groups
2.1-SASS

a. In the upper-right corner of any page, click, and then click **new repository**.



1 Set up Git
A quick guide to help you get started with Git.

2 Create repositories
Repositories are where you'll work and collaborate on projects.

3 Fork repositories
Forking creates a new, unique project from an existing one.

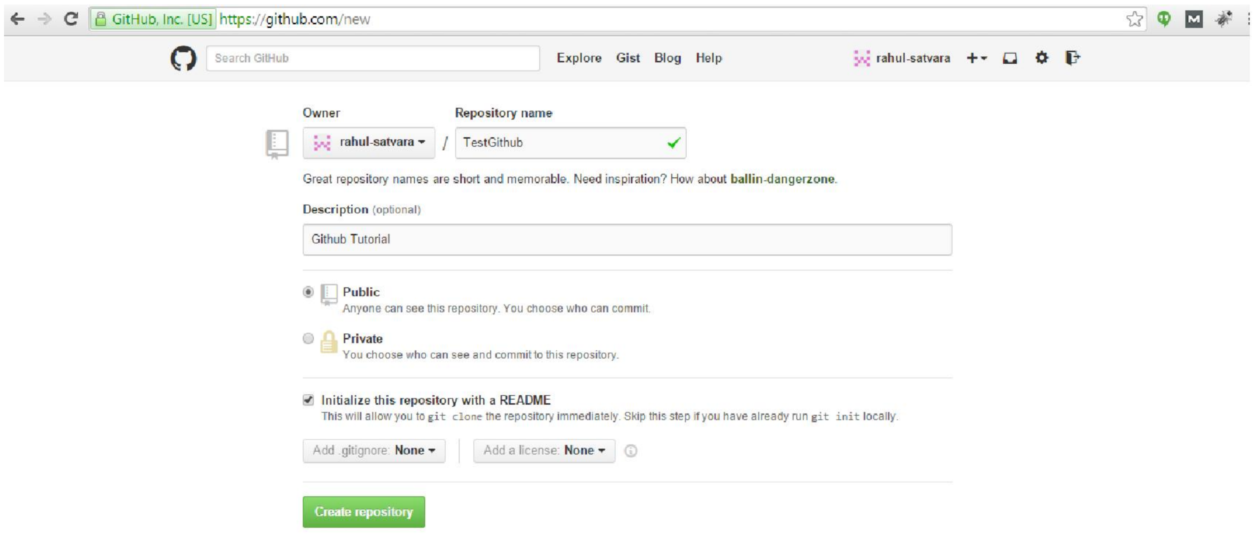
4 Work together
Send pull requests, follow friends. Star and watch projects.

Welcome to GitHub! What's next? (9 days ago)
Create a repository
Tell us about yourself

b. Enter any name you want to give to your repository I give it **TestGithub**. Also give some description to your Repository.

c. You can make repository as **Private** but for that you have to upgrade account. Here we select **Public.**

d. **Initialize this repository with a README.**, README is basically a text file about page.



Owner: rahul-satvara / Repository name: TestGithub

Great repository names are short and memorable. Need inspiration? How about ballin-dangerzone.

Description (optional): Github Tutorial

(●) Public
Anyone can see this repository. You choose who can commit.

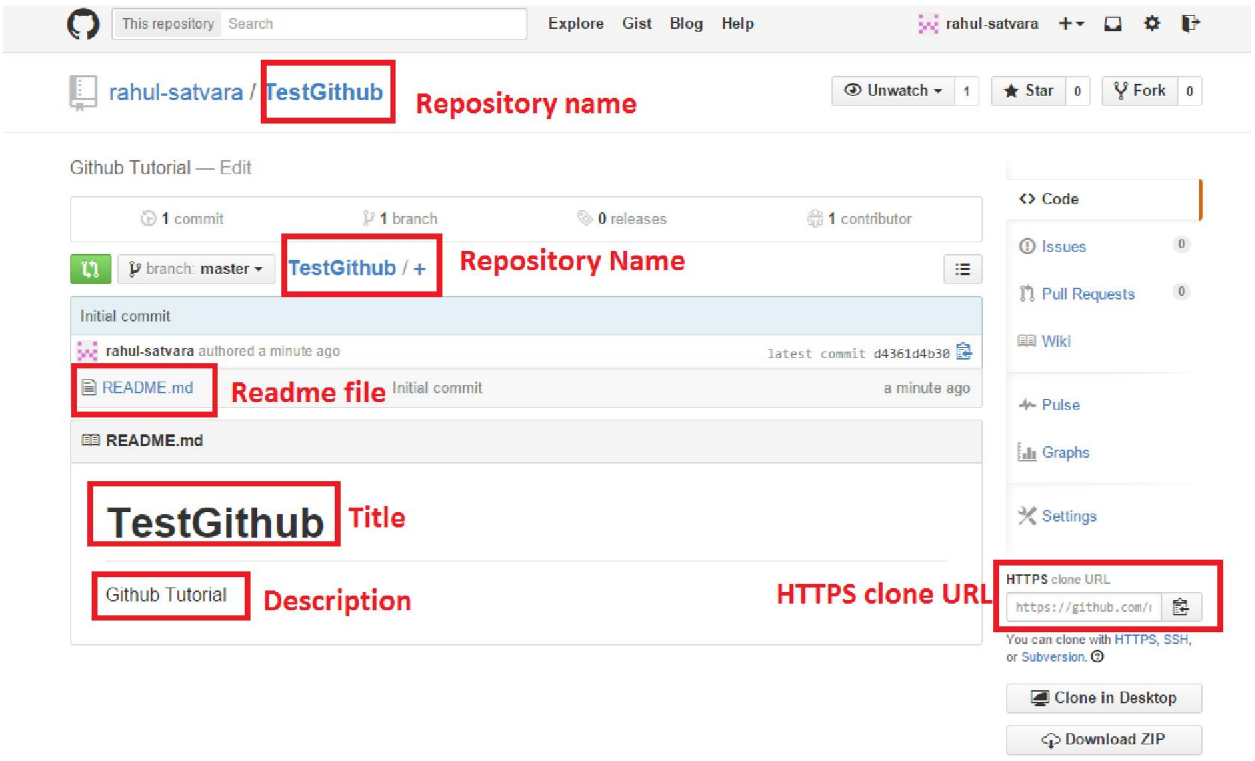
( ) Private
You choose who can see and commit to this repository.

[✓] Initialize this repository with a README
This will allow you to `git clone` the repository immediately. Skip this step if you have already run `git init` locally.

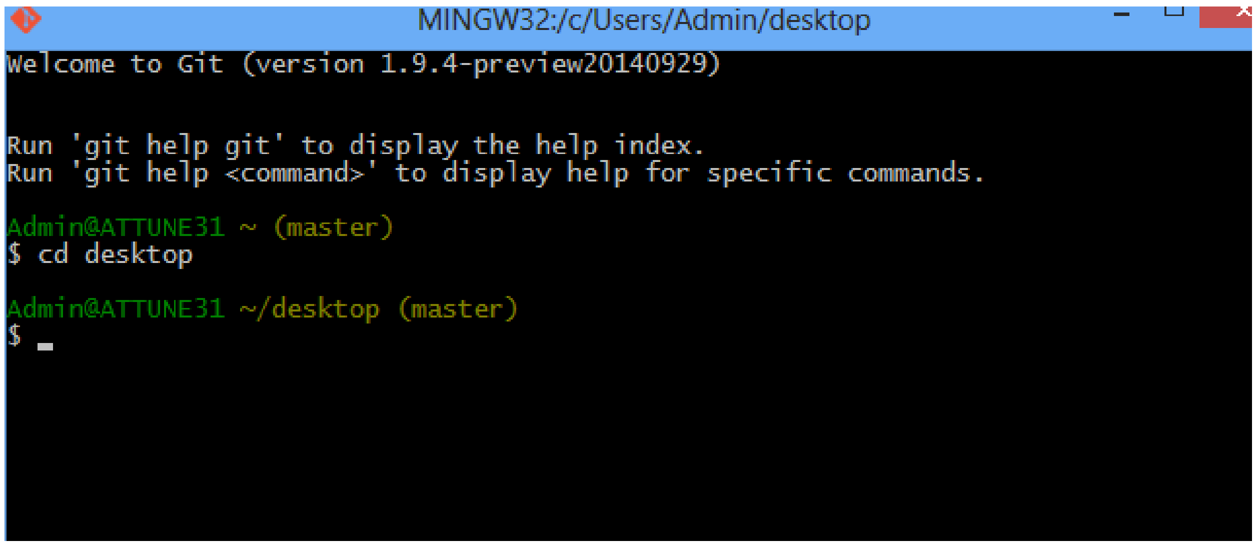
Add .gitignore: None | Add a license: None

e. Click on **Create Repository.**

f. After creating repository we have
    i. TestGithub Repo.
    ii. Title
    iii. README file.
    iv. Description
    v. HTTPS clone URL (right hand side) to clone repository.

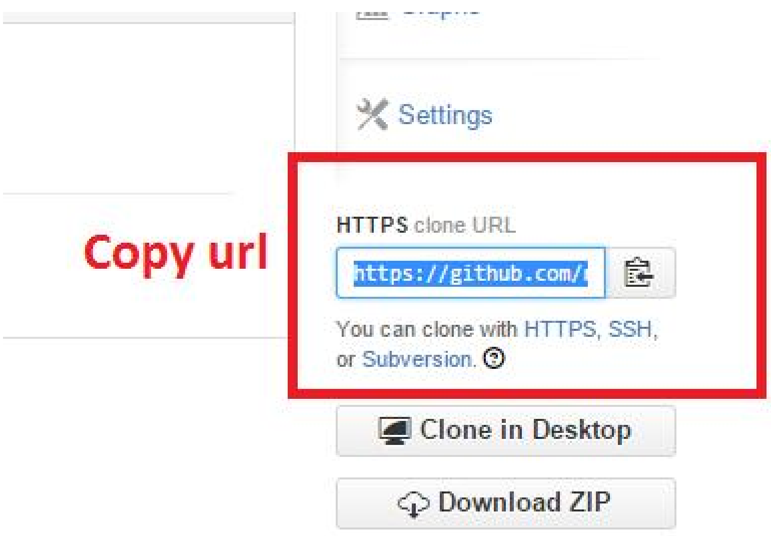


8. After done above steps let’s go to git Bash.
9. Go to desktop directory by typing cd desktop.

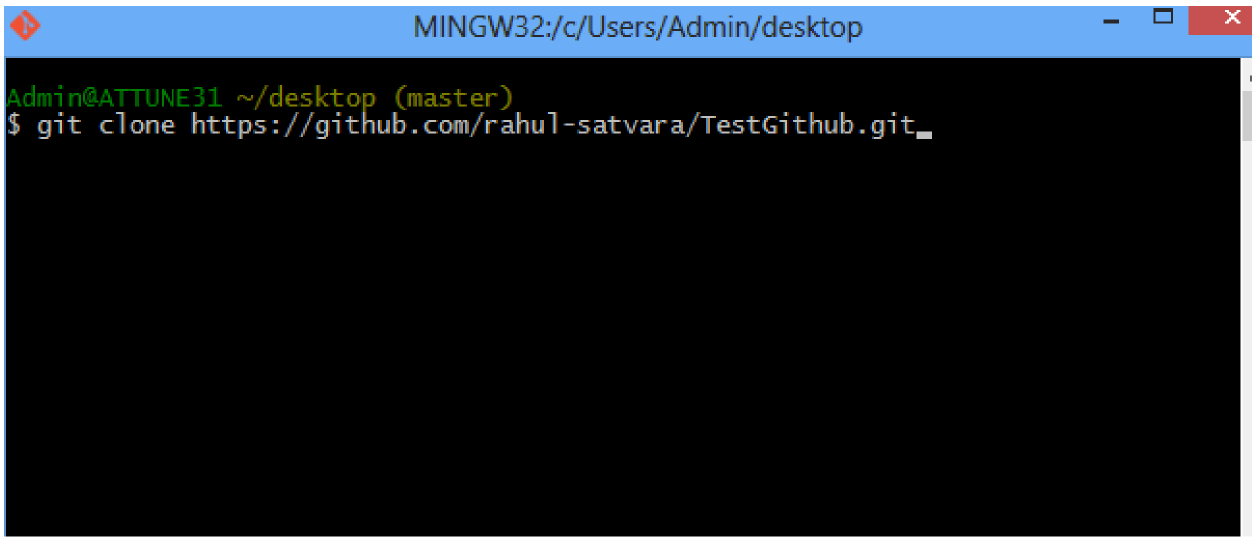


```
Welcome to Git (version 1.9.4-preview20140929)


Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

Admin@ATTUNE31 ~ (master)
$ cd desktop

Admin@ATTUNE31 ~/desktop (master)
$ _
```

10.Now I want to clone repo that I create in github site.

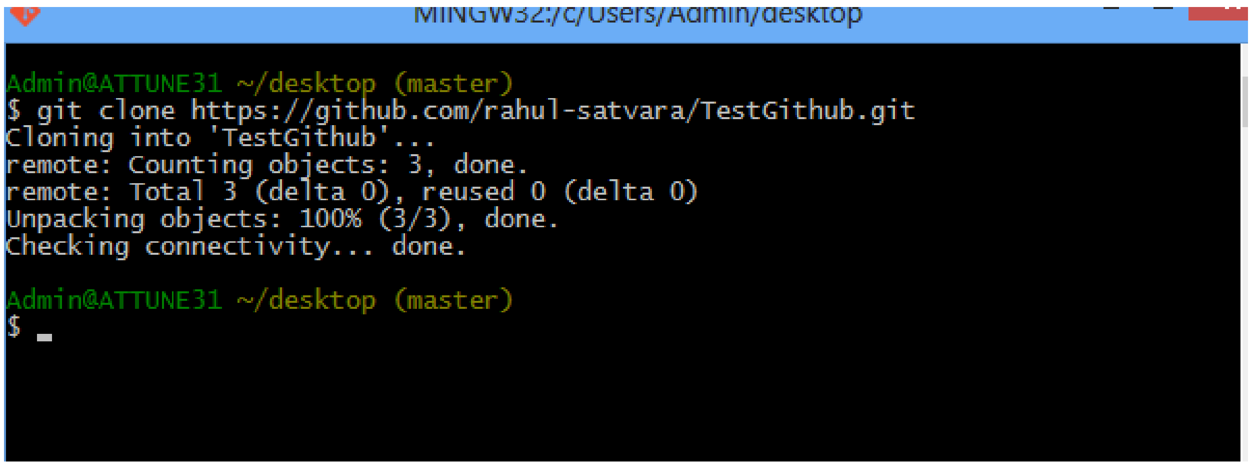


Paste in gitbash.



```
Admin@ATTUNE31 ~/desktop (master)
$ git clone https://github.com/rahul-satvara/TestGithub.git
```

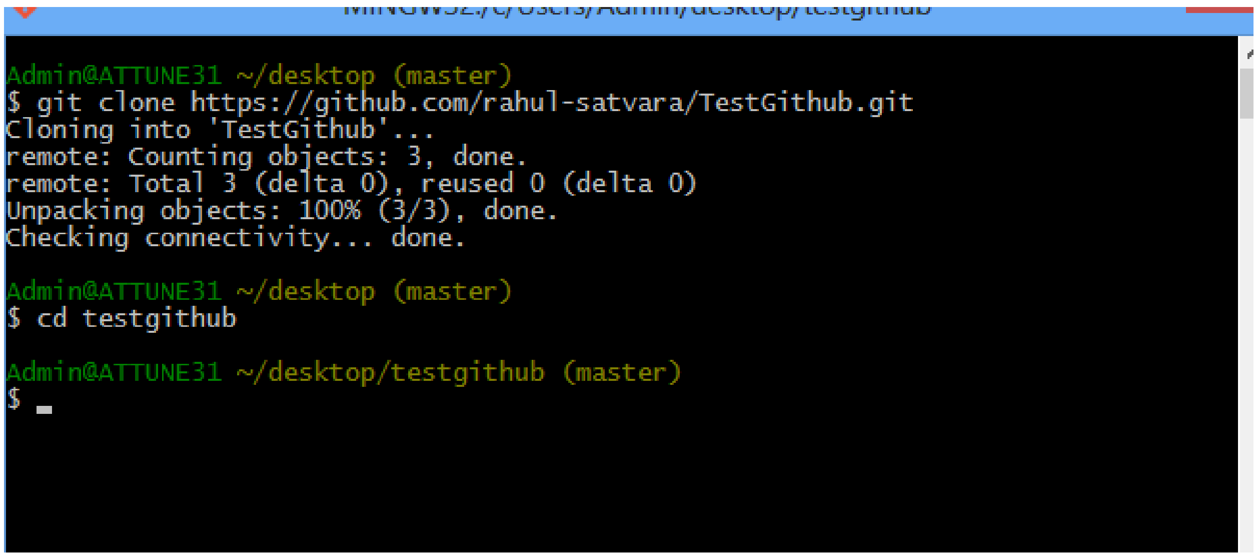
11.After running above command repository (**TestGithub**) is clone in our desktop with all resources. But we have only one file readme.



```
Admin@ATTUNE31 ~/desktop (master)
$ git clone https://github.com/rahul-satvara/TestGithub.git
Cloning into 'TestGithub'...
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
Checking connectivity... done.

Admin@ATTUNE31 ~/desktop (master)
$ _
```

12.Now go into TestGithub Repo Folder.

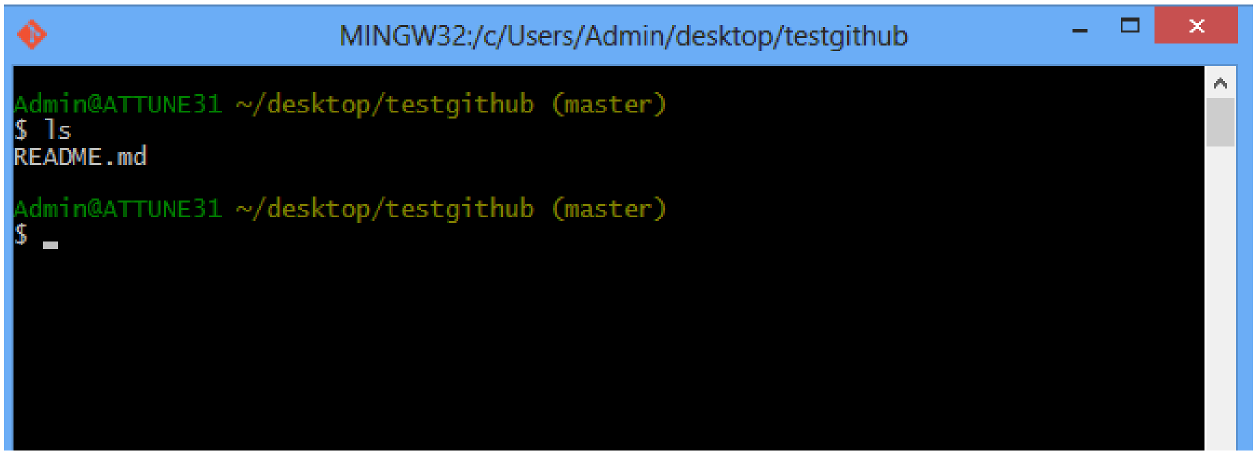
Cd TestGithub



```
Admin@ATTUNE31 ~/desktop (master)
$ git clone https://github.com/rahul-satvara/TestGithub.git
Cloning into 'TestGithub'...
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
Checking connectivity... done.

Admin@ATTUNE31 ~/desktop (master)
$ cd testgithub

Admin@ATTUNE31 ~/desktop/testgithub (master)
$ _
```

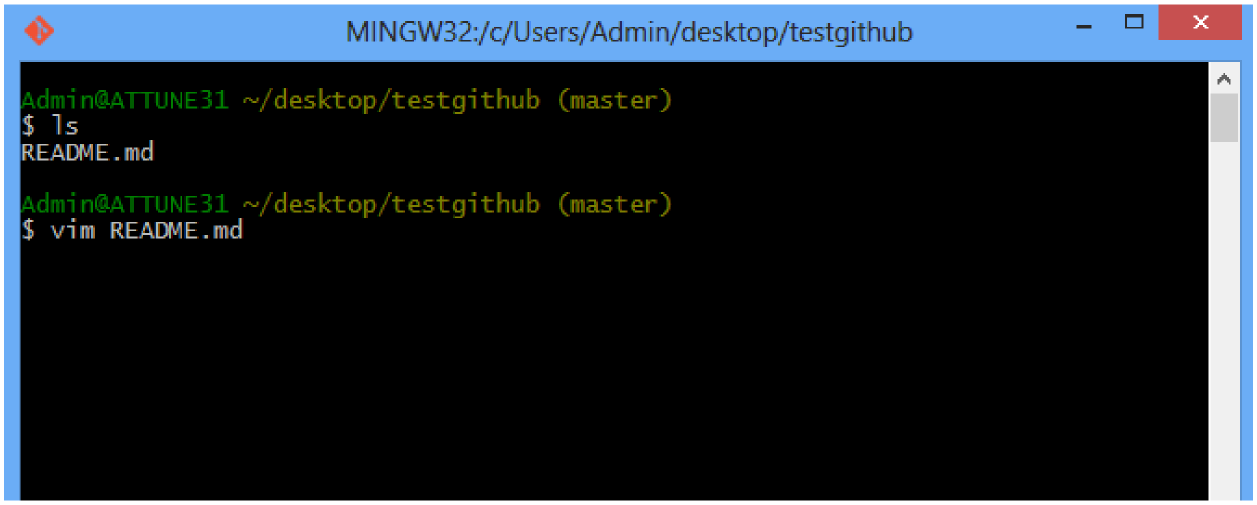
13.So now we are in the testgithub folder. It also a master branch to our current working directory.
All implementation done here we can create other branches and that emerge into master branch for distributed environment. Where more than one person working on same project. All branchs are merging into master that is final master project.

14.Type ls command. So you can see there is one Readme.md file.



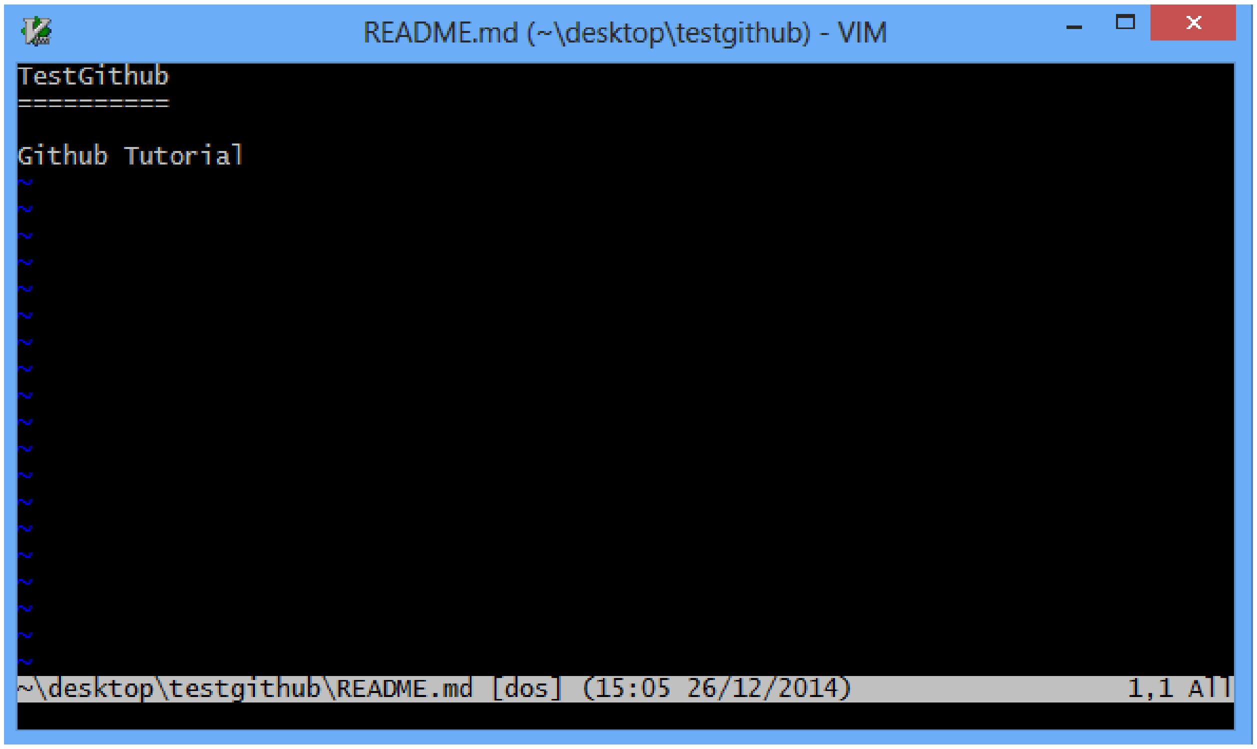
```
Admin@ATTUNE31 ~/desktop/testgithub (master)
$ ls
README.md

Admin@ATTUNE31 ~/desktop/testgithub (master)
$ _
```

15.Now select editor to edit file.



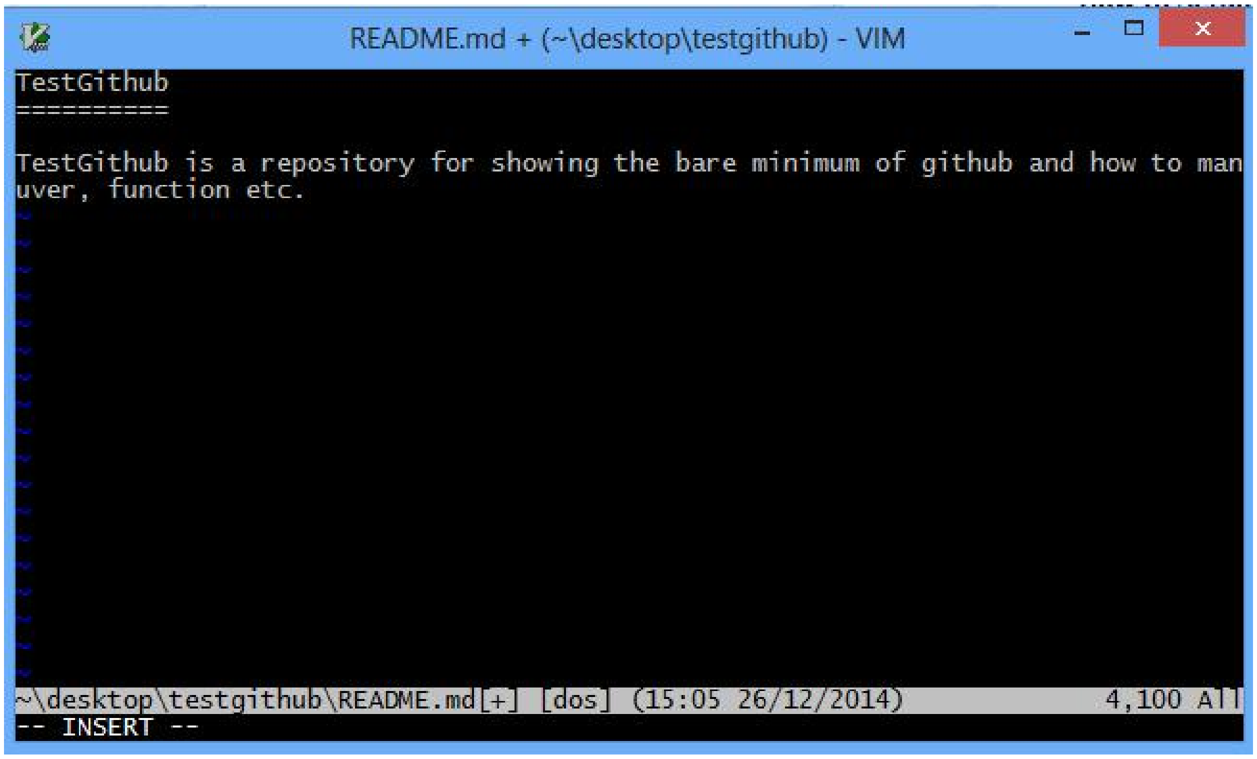
```
Admin@ATTUNE31 ~/desktop/testgithub (master)
$ ls
README.md

Admin@ATTUNE31 ~/desktop/testgithub (master)
$ vim README.md
```

16.Now we are in vim editor so we can edit out file here. You can see there is a title and description showing in editor.



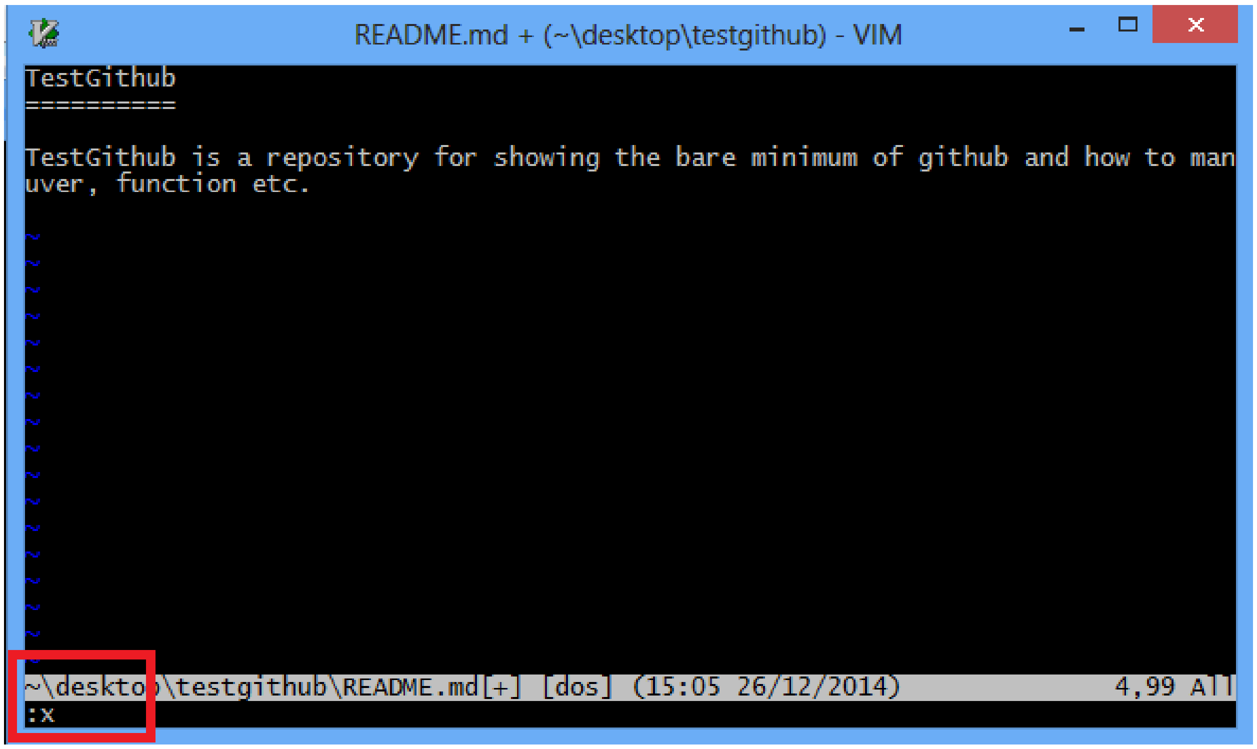
```
README.md (~\desktop\testgithub) - VIM
TestGithub
==========

Github Tutorial
~
~
~\desktop\testgithub\README.md [dos] (15:05 26/12/2014)                1,1 All
```

17.So now I'm modifying description.



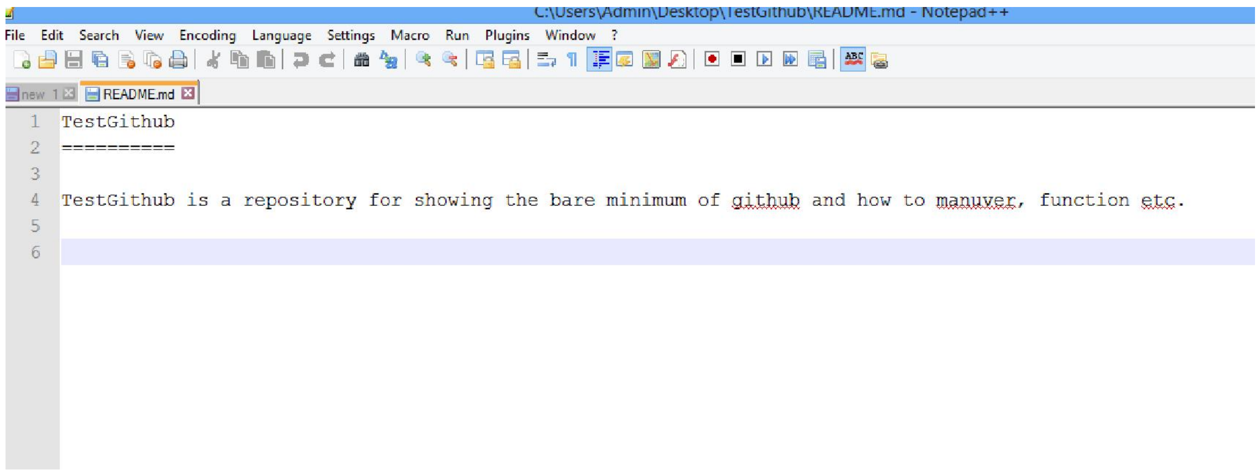
```
README.md + (~\desktop\testgithub) - VIM
TestGithub
==========

TestGithub is a repository for showing the bare minimum of github and how to man
uver, function etc.
~
~
~\desktop\testgithub\README.md[+] [dos] (15:05 26/12/2014)            4,100 All
-- INSERT --
```

18. Now press Esc button to get out from insert mode. If you want edit then press Insert button.
19. Now type :X to save and exit from editor.



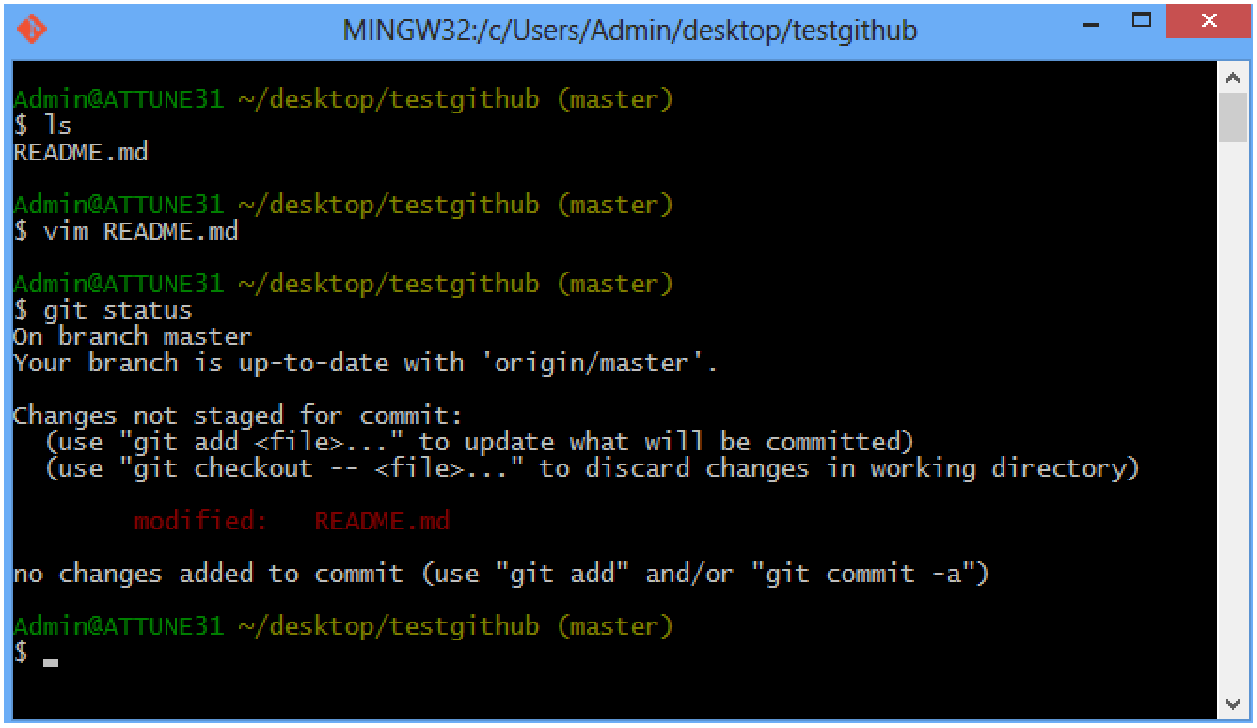
```
TestGithub
==========

TestGithub is a repository for showing the bare minimum of github and how to man
uver, function etc.

~\desktop\testgithub\README.md[+] [dos] (15:05 26/12/2014)                4,99 All
:x
```

20. Now our README file is modified with description.


C:\Users\Admin\Desktop\TestGithub\README.md - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
new 1
README.md

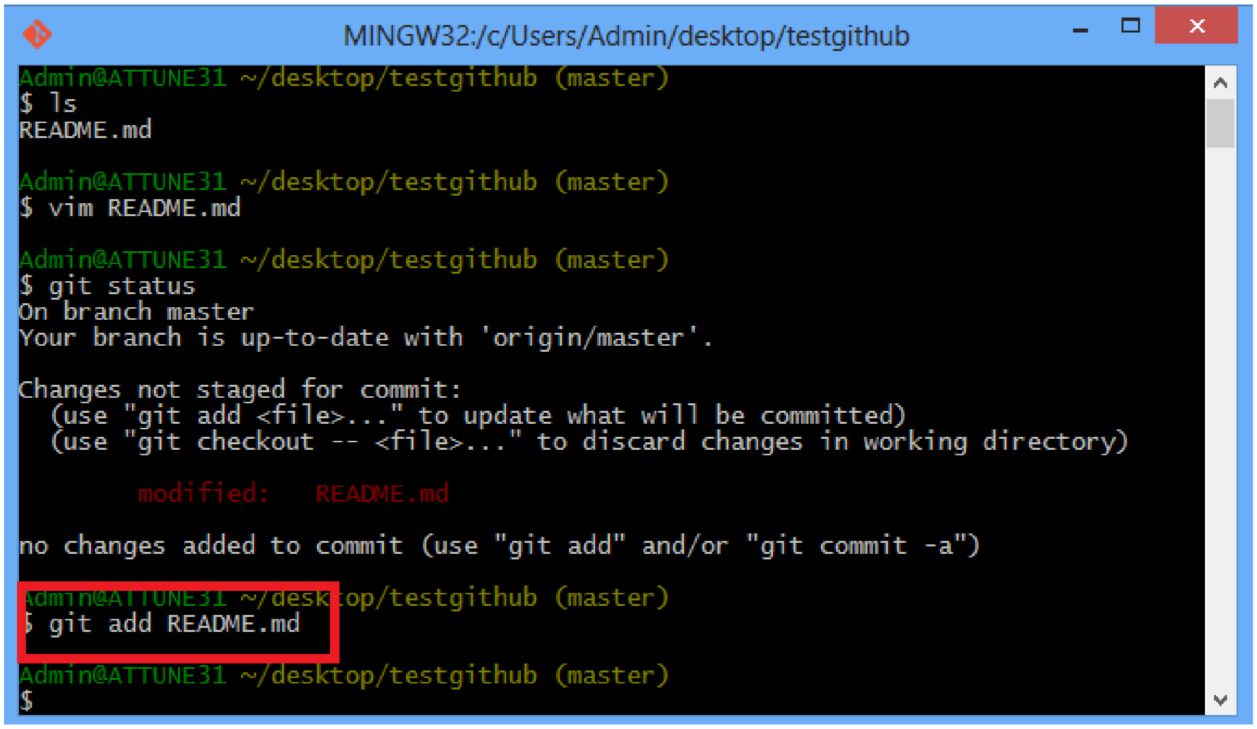
```
TestGithub
==========

TestGithub is a repository for showing the bare minimum of github and how to manuver, function etc.
```

21. Now run command **git status** to know that what thing is modified in our repository.



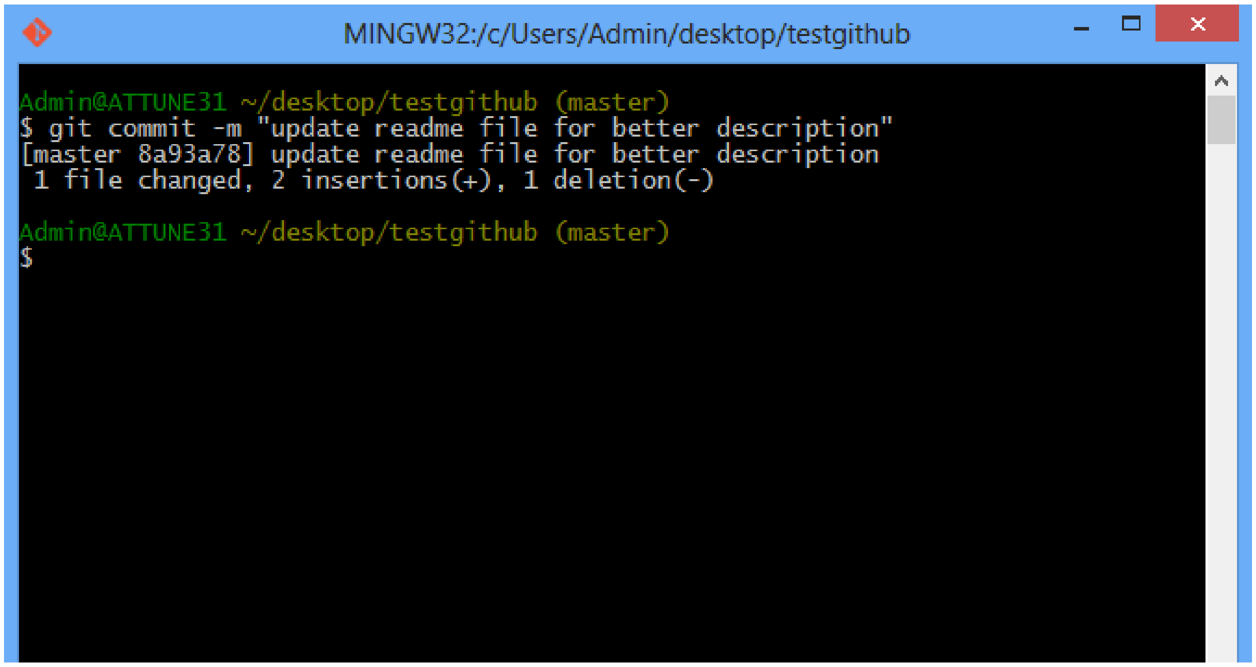
```
Admin@ATTUNE31 ~/desktop/testgithub (master)
$ ls
README.md

Admin@ATTUNE31 ~/desktop/testgithub (master)
$ vim README.md

Admin@ATTUNE31 ~/desktop/testgithub (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   README.md

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

Admin@ATTUNE31 ~/desktop/testgithub (master)
$ _
```

22. After modifying file I want to update my repository online with this new readme because I did change and I want to update description I want to say that this file has a new description.
23. Now I need to add changes right. So you can use git head to add new files into staging area think about it this way right so you are on your local directory.
    Staging area is just a place that where it's not exactly on the report on the web sites yet by in between your local an online
24. So for that run command git add. To push all changes on to website.



```
Admin@ATTUNE31 ~/desktop/testgithub (master)
$ ls
README.md

Admin@ATTUNE31 ~/desktop/testgithub (master)
$ vim README.md

Admin@ATTUNE31 ~/desktop/testgithub (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   README.md

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

Admin@ATTUNE31 ~/desktop/testgithub (master)
$ git add README.md

Admin@ATTUNE31 ~/desktop/testgithub (master)
$
```

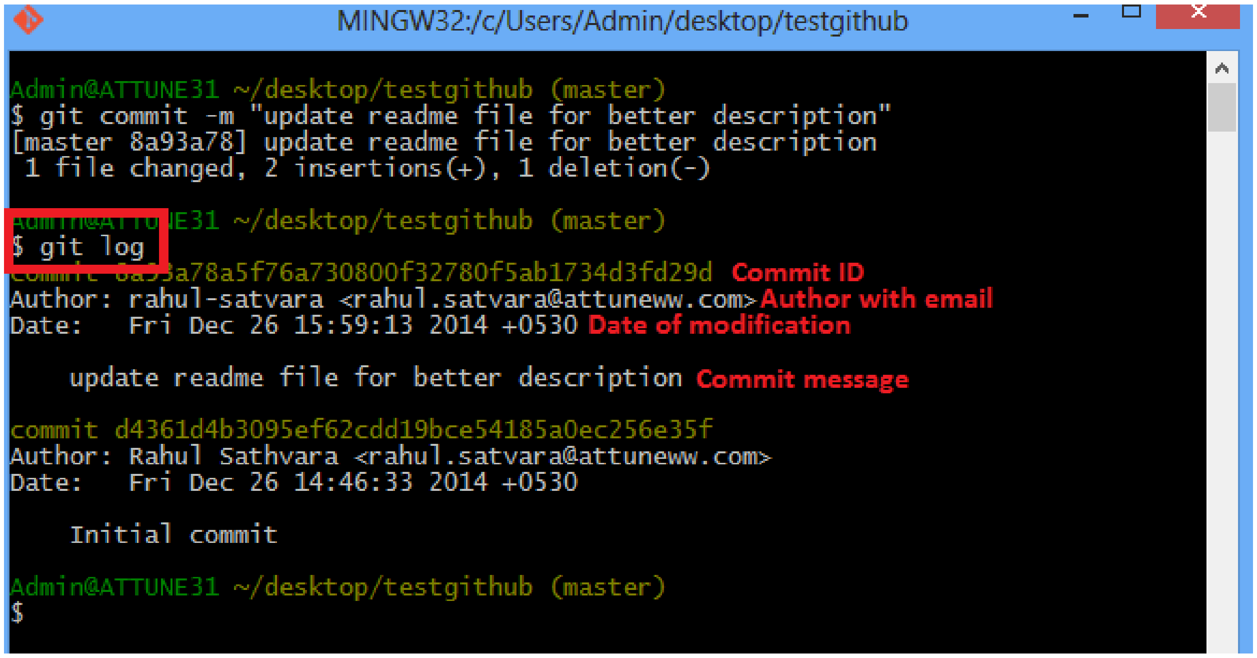
25.Now commit changes we done. Commit is a documenting the change and your finalizing staging area to be ready to push onto the rebook now. Just type **commit –m "Your Message"** message is for documenting changes to know which person done did change and what type change done.



```
Admin@ATTUNE31 ~/desktop/testgithub (master)
$ git commit -m "update readme file for better description"
[master 8a93a78] update readme file for better description
 1 file changed, 2 insertions(+), 1 deletion(-)

Admin@ATTUNE31 ~/desktop/testgithub (master)
$
```

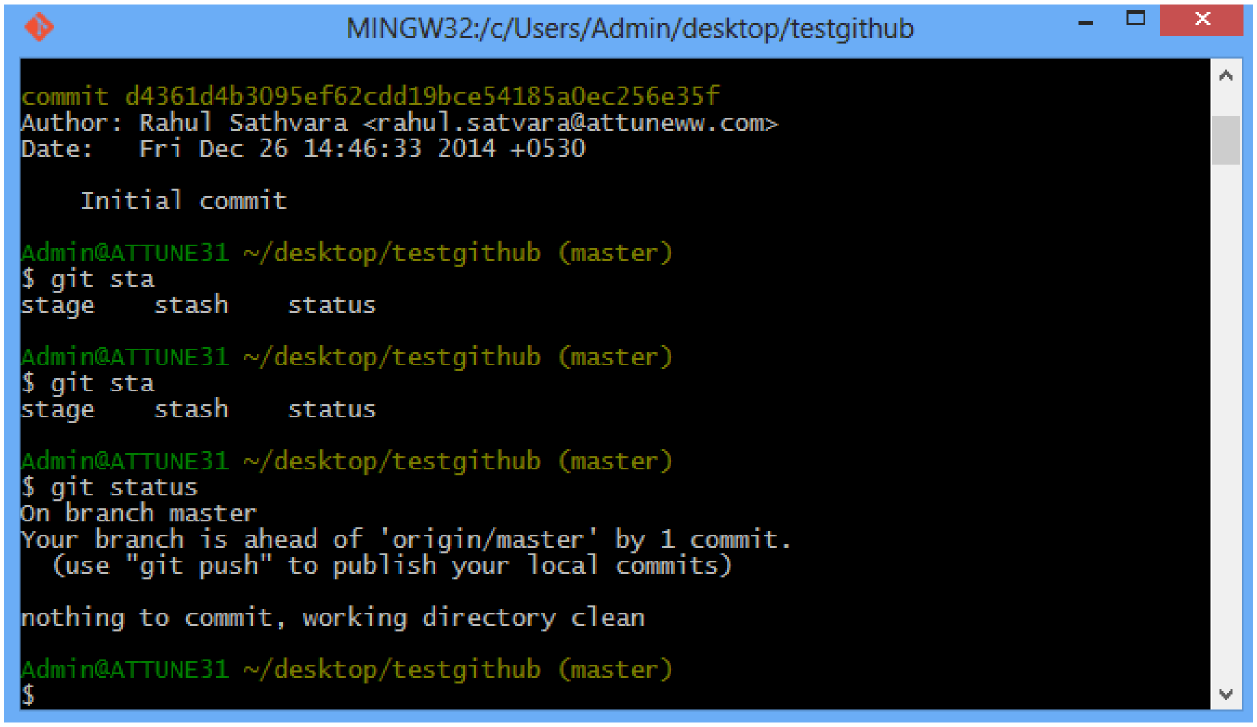
26.Now check log to know what happen.

Type: **git log**



```
Admin@ATTUNE31 ~/desktop/testgithub (master)
$ git commit -m "update readme file for better description"
[master 8a93a78] update readme file for better description
 1 file changed, 2 insertions(+), 1 deletion(-)

Admin@ATTUNE31 ~/desktop/testgithub (master)
$ git log
commit 8a93a78a5f76a730800f32780f5ab1734d3fd29d
Author: rahul-satvara <rahul.satvara@attuneww.com>
Date:   Fri Dec 26 15:59:13 2014 +0530

    update readme file for better description

commit d4361d4b3095ef62cdd19bce54185a0ec256e35f
Author: Rahul Sathvara <rahul.satvara@attuneww.com>
Date:   Fri Dec 26 14:46:33 2014 +0530

    Initial commit

Admin@ATTUNE31 ~/desktop/testgithub (master)
$
```

27.Now check status it is good to check status.

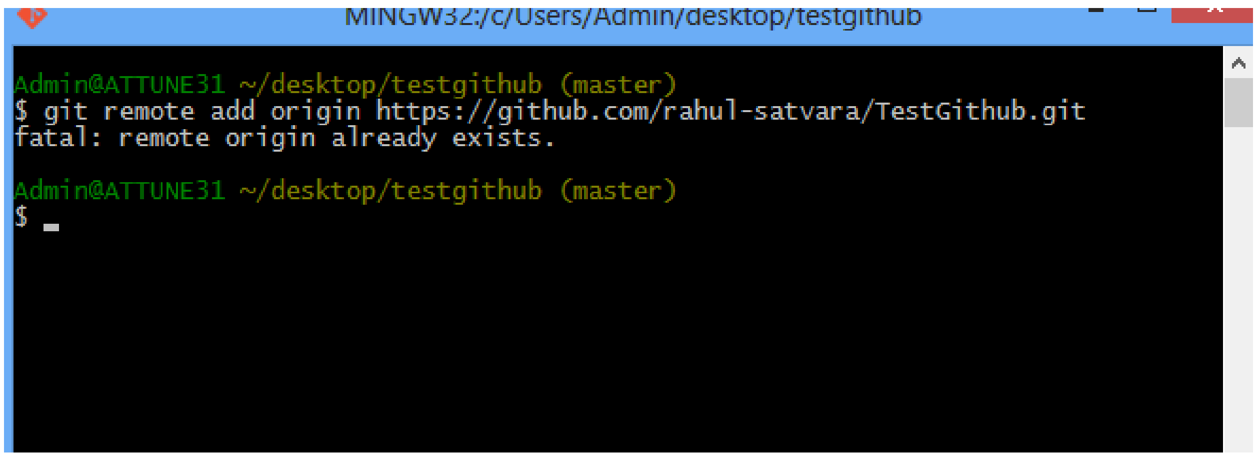


```
commit d4361d4b3095ef62cdd19bce54185a0ec256e35f
Author: Rahul Sathvara <rahul.satvara@attuneww.com>
Date:   Fri Dec 26 14:46:33 2014 +0530

    Initial commit

Admin@ATTUNE31 ~/desktop/testgithub (master)
$ git sta
stage    stash    status

Admin@ATTUNE31 ~/desktop/testgithub (master)
$ git sta
stage    stash    status

Admin@ATTUNE31 ~/desktop/testgithub (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working directory clean

Admin@ATTUNE31 ~/desktop/testgithub (master)
$
```

28.Your branch is ahead of ‘origin/master’ origin is out current working
directory and master is remote directory.

29.So we are in local origin to let’s move to remote origin.
30.Just type :-

git remote add origin https://github.com/rahul-satvara/TestGithub.git



```
Admin@ATTUNE31 ~/desktop/testgithub (master)
$ git remote add origin https://github.com/rahul-satvara/TestGithub.git
fatal: remote origin already exists.

Admin@ATTUNE31 ~/desktop/testgithub (master)
$
```

We get error remote origin already exist because whenever you clone a repository it automatically create origin because of course you are cloning your are downloading it. It is going to create origin to wish you are currently at.

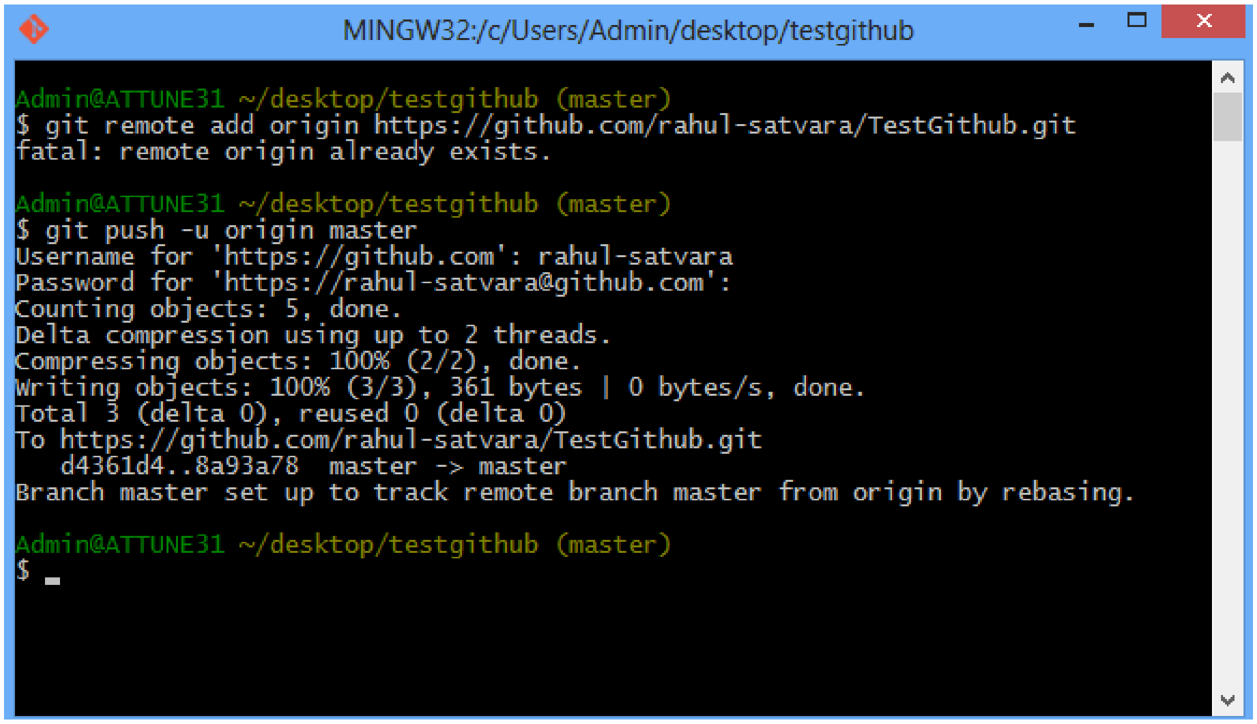
But if you are creating let’s say new completely new repository and let’s say I did not clone then I would have to get repo and clone.

This is necessary step if you are not cloning.

31.So after add origin we push change into master branch that is on remote location.

Git push –u origin master

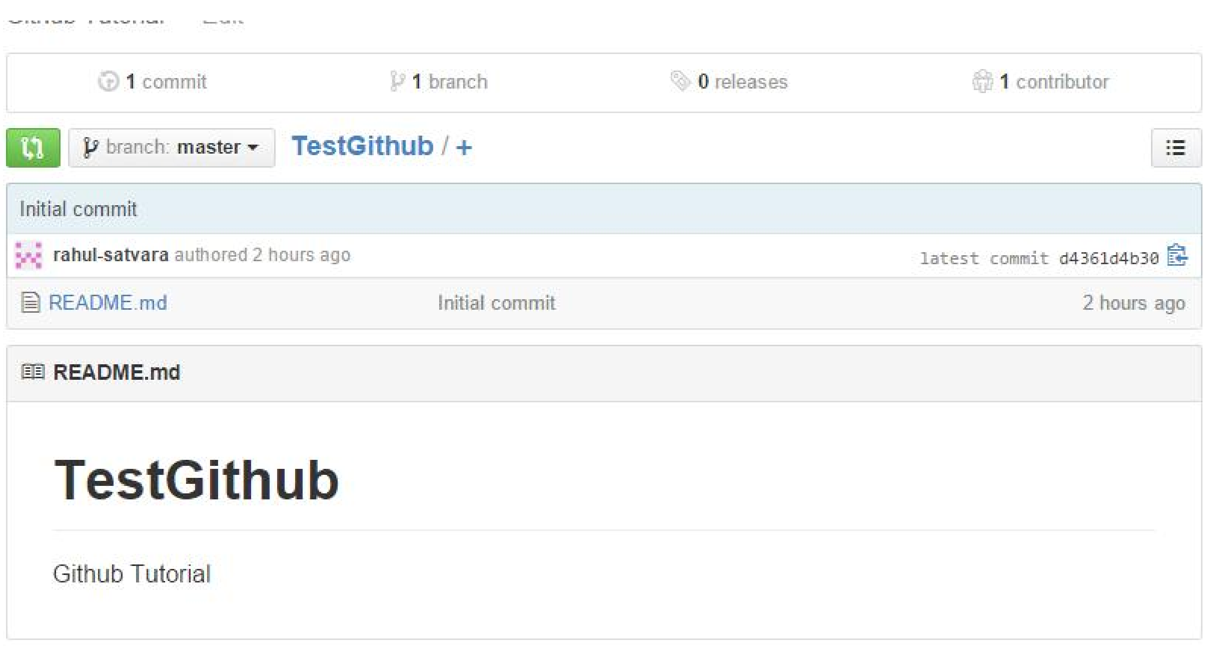
32.It will ask you username and Password to access remote repository.



```
Admin@ATTUNE31 ~/desktop/testgithub (master)
$ git remote add origin https://github.com/rahul-satvara/TestGithub.git
fatal: remote origin already exists.

Admin@ATTUNE31 ~/desktop/testgithub (master)
$ git push -u origin master
Username for 'https://github.com': rahul-satvara
Password for 'https://rahul-satvara@github.com':
Counting objects: 5, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 361 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/rahul-satvara/TestGithub.git
   d4361d4..8a93a78  master -> master
Branch master set up to track remote branch master from origin by rebasing.

Admin@ATTUNE31 ~/desktop/testgithub (master)
$ _
```

33.Now go to your browser refresh page.

Before Refresh:-



Initial commit

rahul-satvara authored 2 hours ago — latest commit d4361d4b30

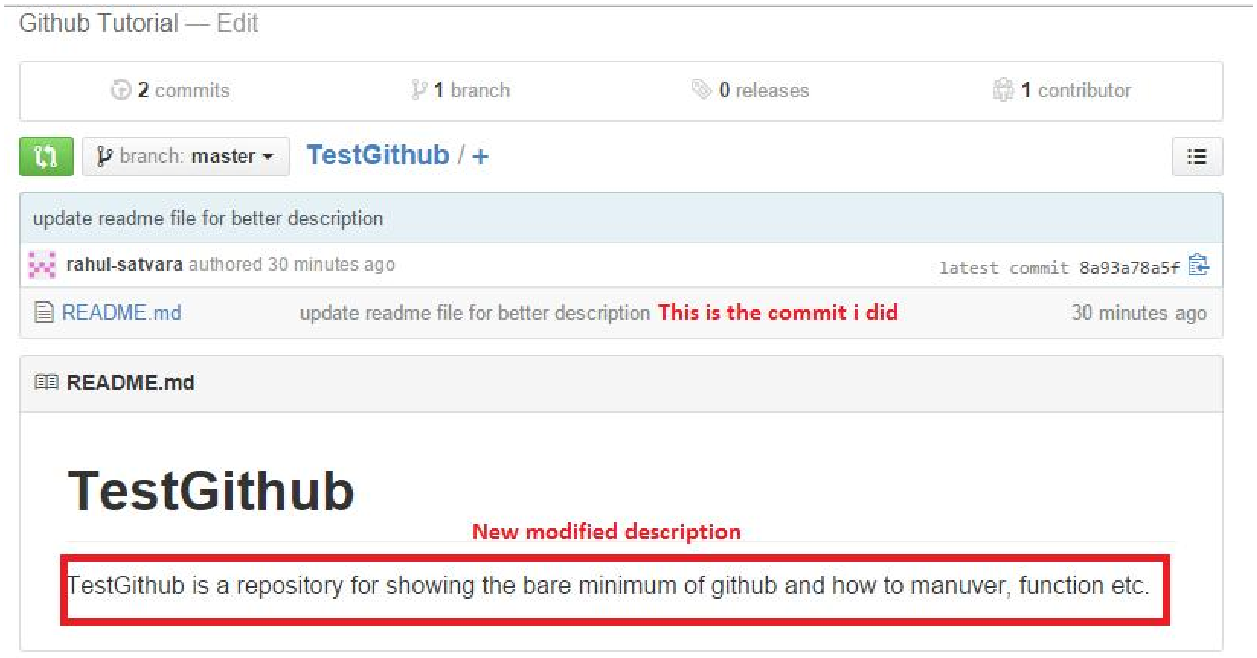
README.md — Initial commit — 2 hours ago

README.md

TestGithub

Github Tutorial

After Refresh:-


Github Tutorial — Edit

2 commits | 1 branch | 0 releases | 1 contributor

branch: master | TestGithub / +

update readme file for better description

rahul-satvara authored 30 minutes ago — latest commit 8a93a78a5f

README.md — update readme file for better description **This is the commit i did** — 30 minutes ago

README.md

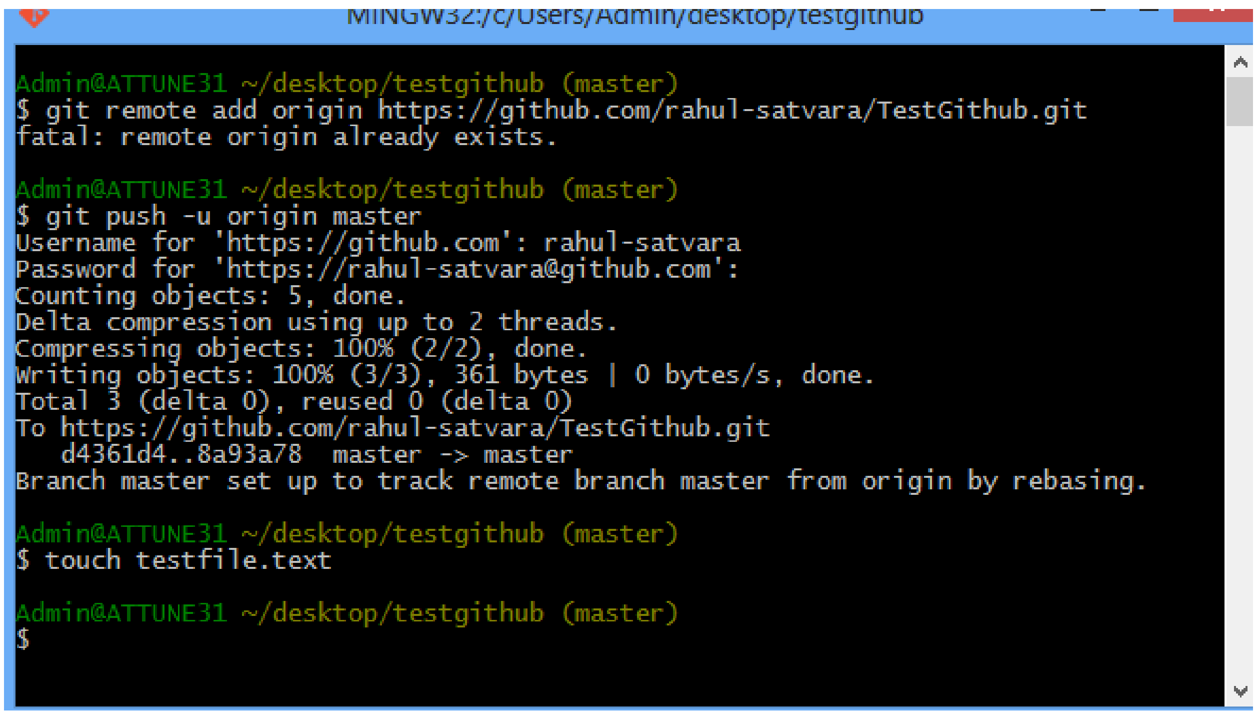
TestGithub

New modified description

TestGithub is a repository for showing the bare minimum of github and how to manuver, function etc.

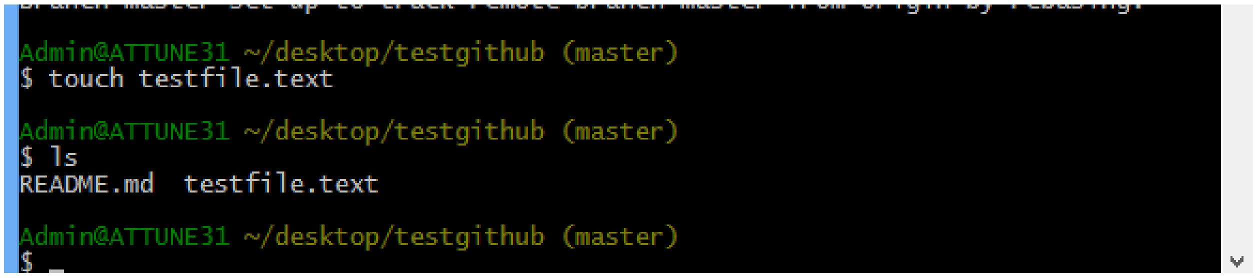
34.You can see new changes are pushed in our remote repository.

35.Now we will make a new file. You can use this commend **touch testfile.text to** create new file on local.


MINGW32:/c/Users/Admin/desktop/testgithub

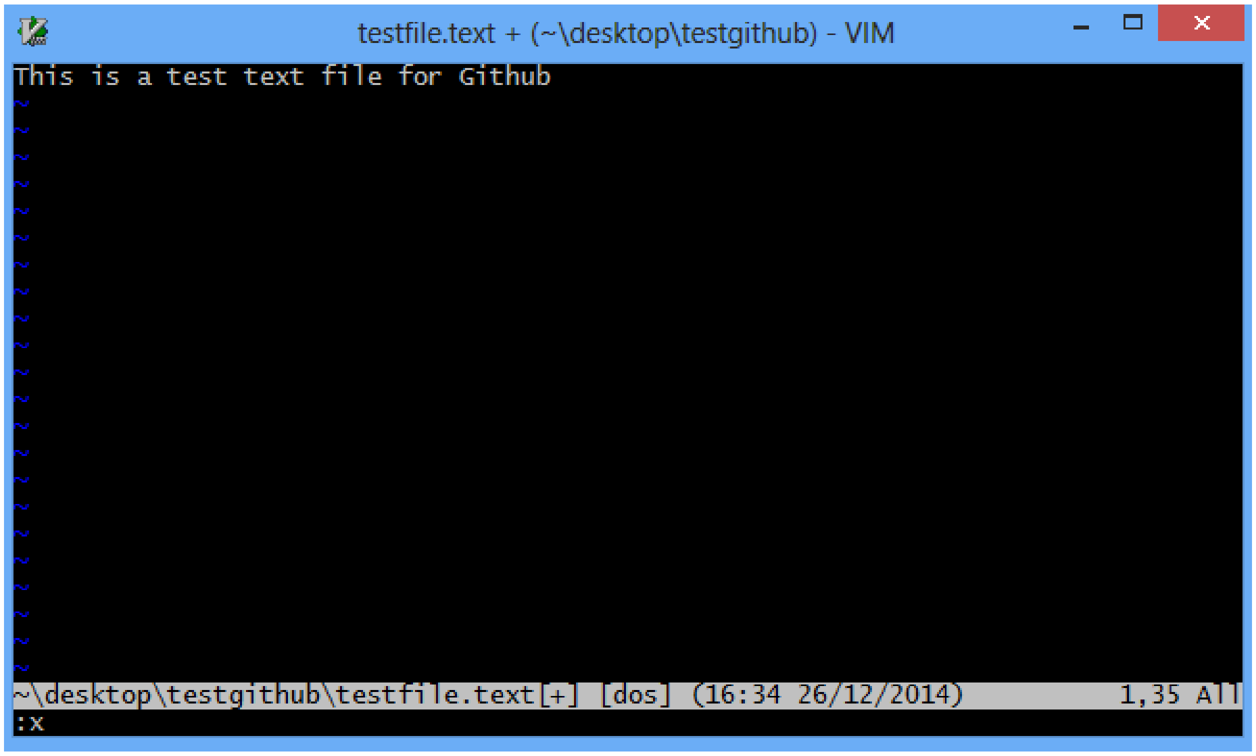
```
Admin@ATTUNE31 ~/desktop/testgithub (master)
$ git remote add origin https://github.com/rahul-satvara/TestGithub.git
fatal: remote origin already exists.

Admin@ATTUNE31 ~/desktop/testgithub (master)
$ git push -u origin master
Username for 'https://github.com': rahul-satvara
Password for 'https://rahul-satvara@github.com':
Counting objects: 5, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 361 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/rahul-satvara/TestGithub.git
   d4361d4..8a93a78  master -> master
Branch master set up to track remote branch master from origin by rebasing.

Admin@ATTUNE31 ~/desktop/testgithub (master)
$ touch testfile.text

Admin@ATTUNE31 ~/desktop/testgithub (master)
$
```

36.Now use **ls** to see new created file.



```
Admin@ATTUNE31 ~/desktop/testgithub (master)
$ touch testfile.text

Admin@ATTUNE31 ~/desktop/testgithub (master)
$ ls
README.md  testfile.text

Admin@ATTUNE31 ~/desktop/testgithub (master)
$ _
```

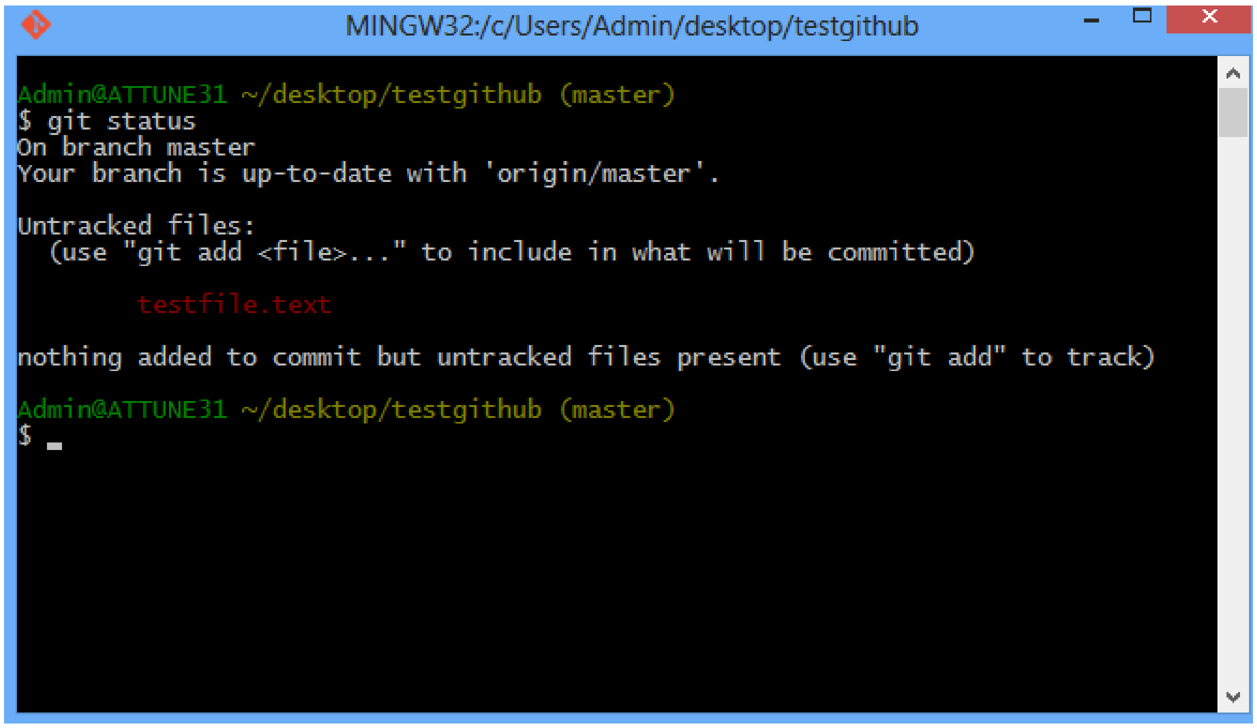
37. This file not created in remote repo. For that we have to add, commit and push to upload on remote repo.
38. First we add some text in testfile.
    a. Type vim testfile.text
    b. Type some text.
    c. Press Esc and
    d. Type :x to save and exit.



testfile.text + (~\desktop\testgithub) - VIM

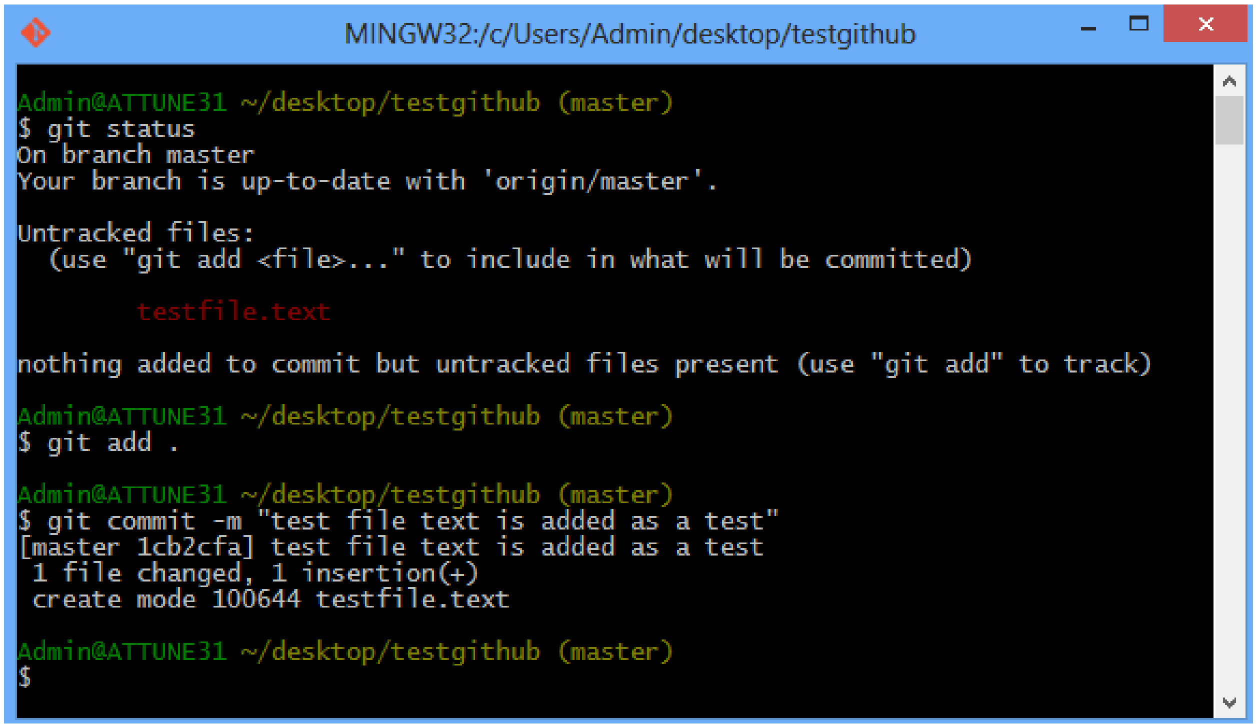
```
This is a test text file for Github
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~\desktop\testgithub\testfile.text[+]  [dos]  (16:34 26/12/2014)          1,35 All
:x
```

39. Let’s check a status.



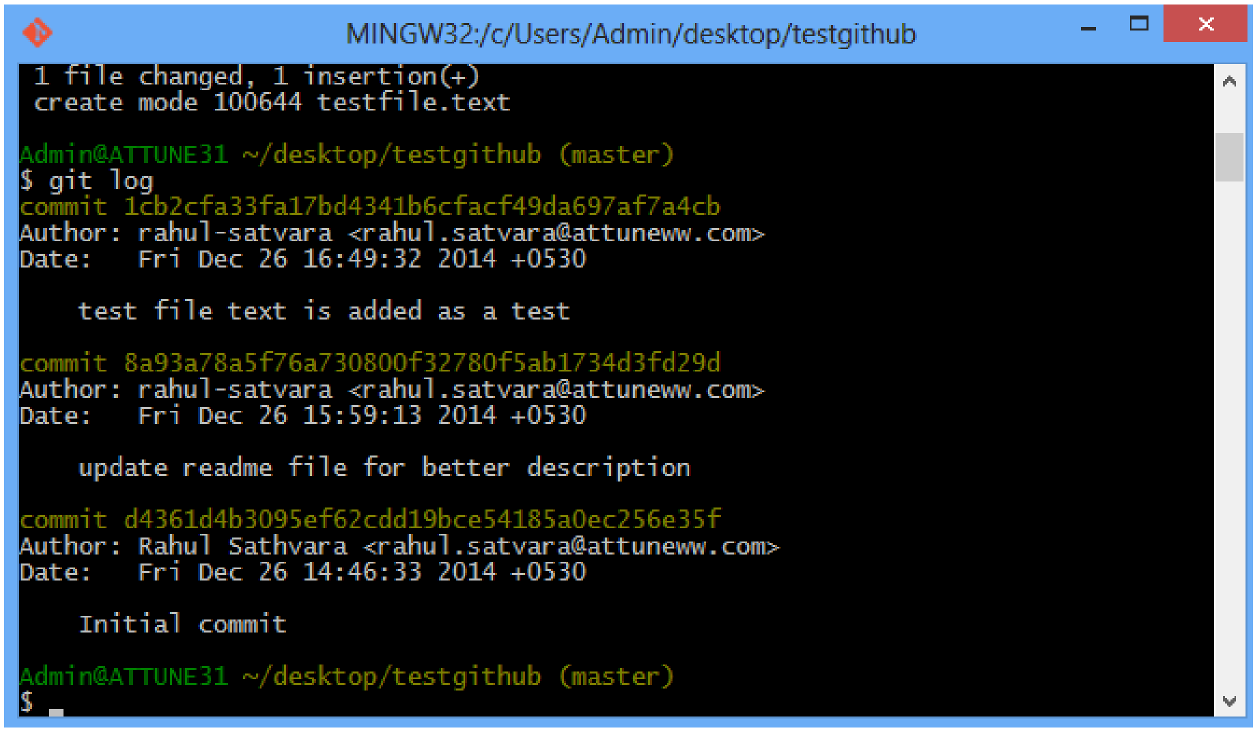
```
Admin@ATTUNE31 ~/desktop/testgithub (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.

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

        testfile.text

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

Admin@ATTUNE31 ~/desktop/testgithub (master)
$ _
```

40. You can see that text is added in file and you can see that this file is not tracked yet.
41. So we need to add and commit.



```
Admin@ATTUNE31 ~/desktop/testgithub (master)
$ git status
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)

        testfile.text

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

Admin@ATTUNE31 ~/desktop/testgithub (master)
$ git add .

Admin@ATTUNE31 ~/desktop/testgithub (master)
$ git commit -m "test file text is added as a test"
[master 1cb2cfa] test file text is added as a test
 1 file changed, 1 insertion(+)
 create mode 100644 testfile.text

Admin@ATTUNE31 ~/desktop/testgithub (master)
$
```

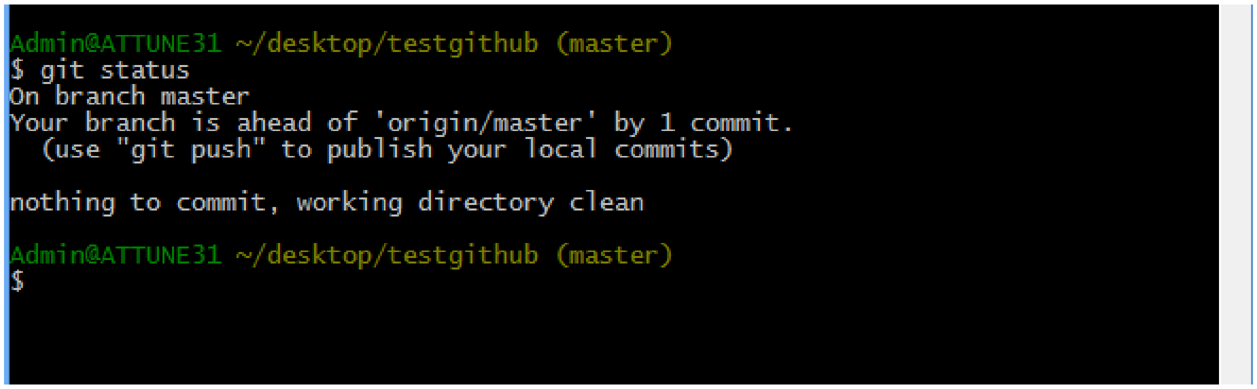
42.Let’s check log.



```
 1 file changed, 1 insertion(+)
 create mode 100644 testfile.text

Admin@ATTUNE31 ~/desktop/testgithub (master)
$ git log
commit 1cb2cfa33fa17bd4341b6cfacf49da697af7a4cb
Author: rahul-satvara <rahul.satvara@attuneww.com>
Date:   Fri Dec 26 16:49:32 2014 +0530

    test file text is added as a test

commit 8a93a78a5f76a730800f32780f5ab1734d3fd29d
Author: rahul-satvara <rahul.satvara@attuneww.com>
Date:   Fri Dec 26 15:59:13 2014 +0530

    update readme file for better description

commit d4361d4b3095ef62cdd19bce54185a0ec256e35f
Author: Rahul Sathvara <rahul.satvara@attuneww.com>
Date:   Fri Dec 26 14:46:33 2014 +0530

    Initial commit

Admin@ATTUNE31 ~/desktop/testgithub (master)
$ _
```

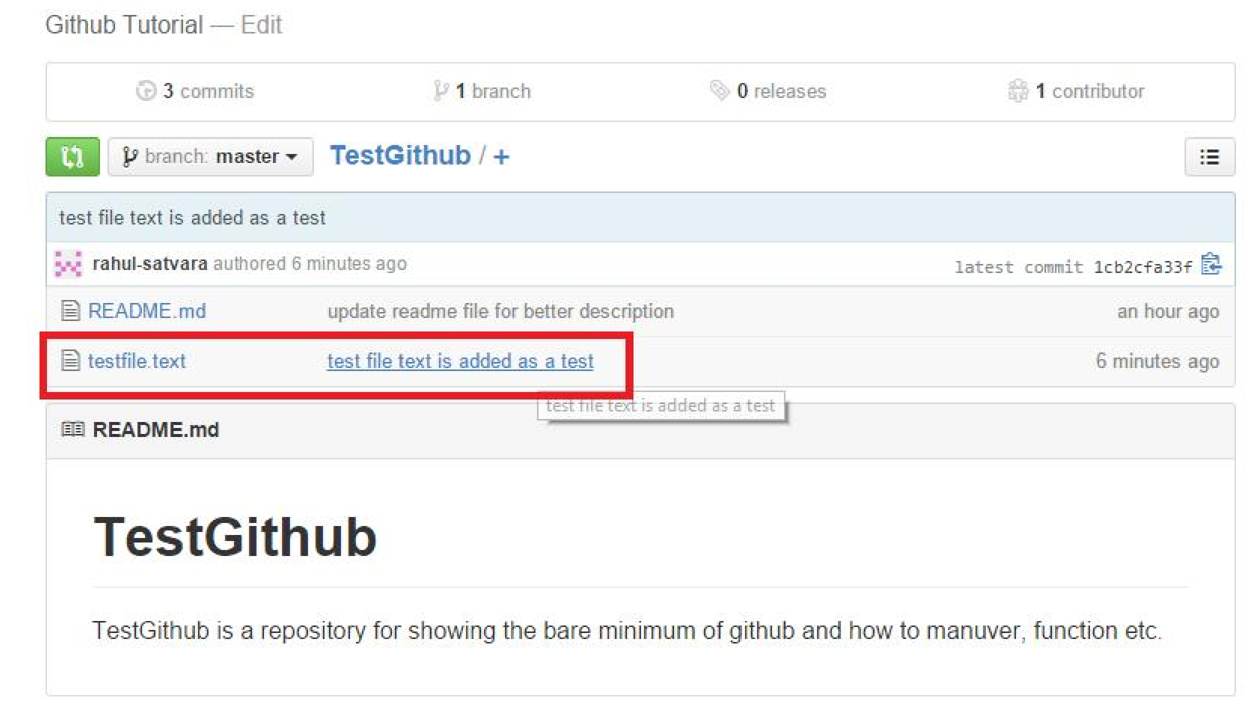
Here you can check all commit and commit id and Commit messages also.

43.Now check status.



```
Admin@ATTUNE31 ~/desktop/testgithub (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working directory clean

Admin@ATTUNE31 ~/desktop/testgithub (master)
$
```

44.Nothing will be added in remote repo until run push.
45.So run push command.
46.It will ask you username and Password just enter it.
47.Go to your browser and refresh page.


Github Tutorial — Edit

3 commits | 1 branch | 0 releases | 1 contributor

branch: master TestGithub / +

test file text is added as a test

rahul-satvara authored 6 minutes ago latest commit 1cb2cfa33f

README.md	update readme file for better description	an hour ago
testfile.text	test file text is added as a test	6 minutes ago

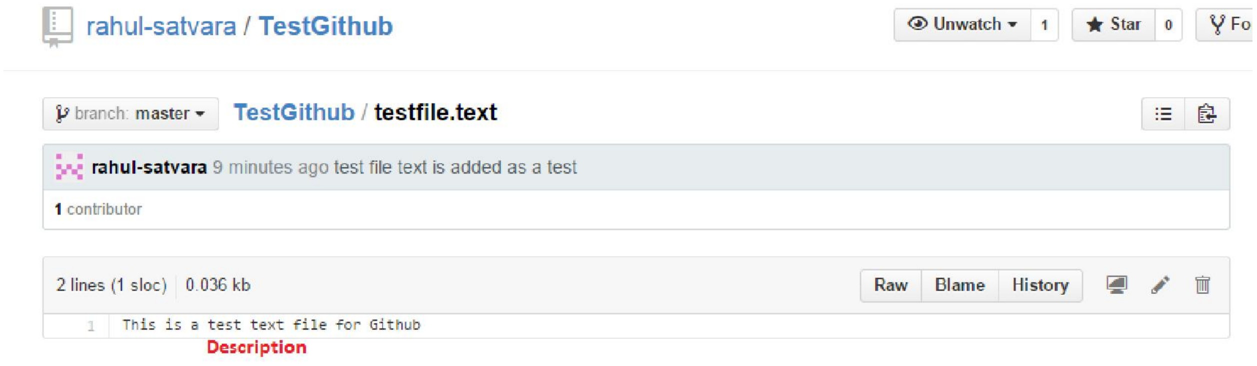
README.md

TestGithub

TestGithub is a repository for showing the bare minimum of github and how to manuver, function etc.

48.So you can see new pushed file testfile.text with commit message.

49.Open testfile.text


rahul-satvara / TestGithub

branch: master TestGithub / testfile.text

rahul-satvara 9 minutes ago test file text is added as a test

1 contributor

2 lines (1 sloc) 0.036 kb

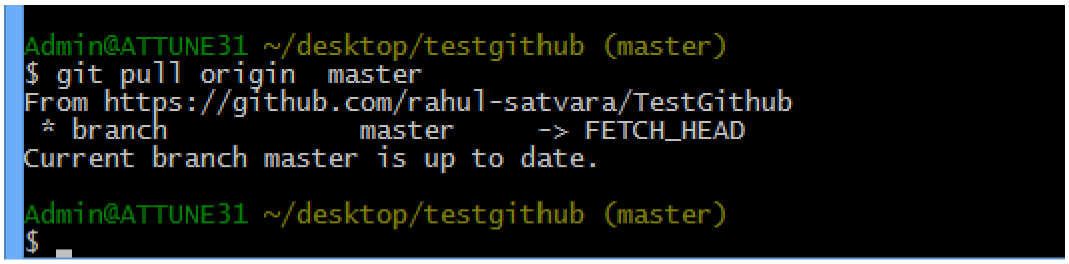
```
This is a test text file for Github
```

Here you can see description of testfile.

50.Now we will pull master repository to origin.

51.Just run **git pull origin master** this will pull all the content to current Repository and it will serve as Let's say that another friend was working on a different file now let's say he was working on like **a.text** now you can do get pull origin master to basically pull that in update on your local drive.

52.I just run git pull origin master.

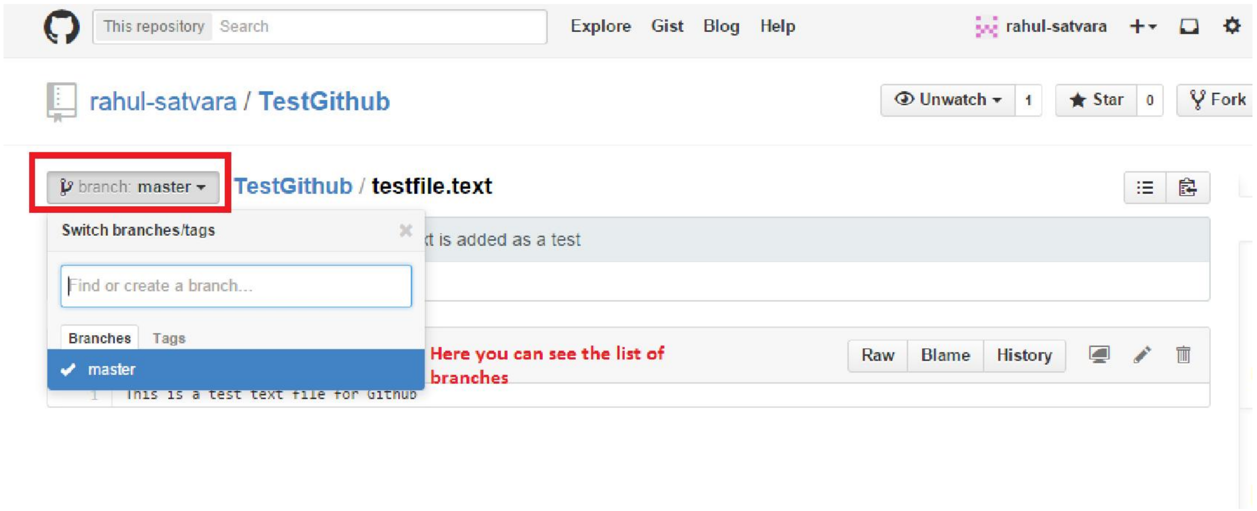


```
Admin@ATTUNE31 ~/desktop/testgithub (master)
$ git pull origin  master
From https://github.com/rahul-satvara/TestGithub
 * branch            master     -> FETCH_HEAD
Current branch master is up to date.

Admin@ATTUNE31 ~/desktop/testgithub (master)
$ 
```

53.So now you can see that Current branch master is up to date because we have everything clone on the repository.

54.So that is how the pushing pulling and commit work on git hub recall using the shall.

55. One Last thing is the braches.

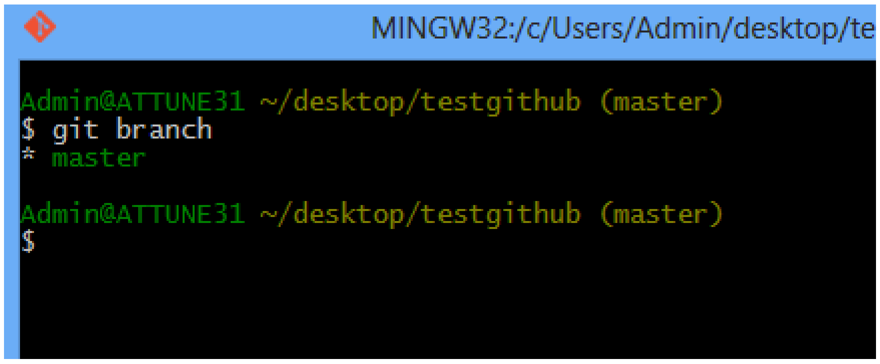


Here you can create a new branches and also see the list of branches.

And also check in which branch you are working on.

56.Let's check list of branches using command.

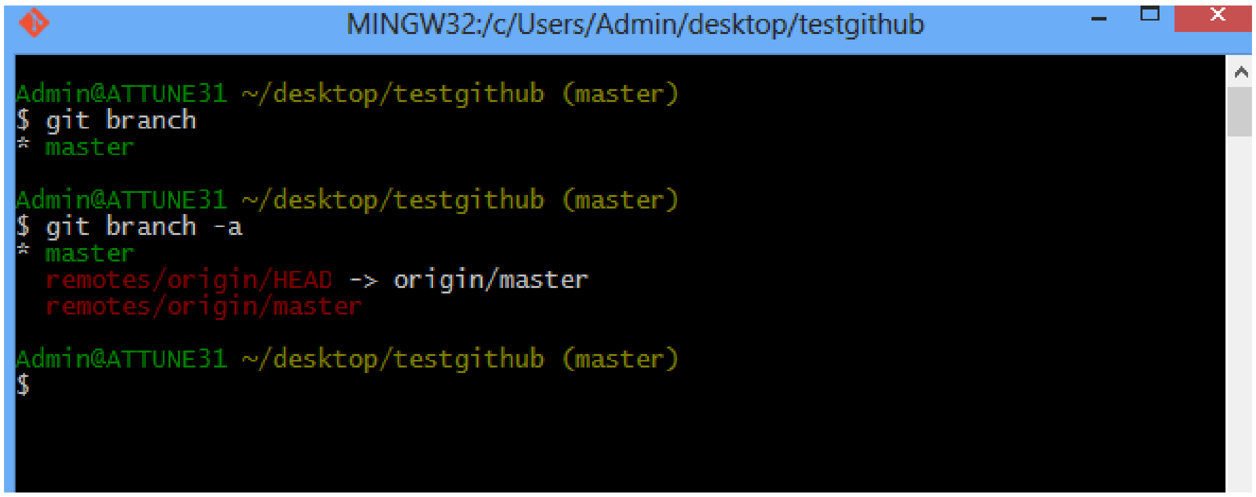
```
git branch
```



```
Admin@ATTUNE31 ~/desktop/testgithub (master)
$ git branch
* master

Admin@ATTUNE31 ~/desktop/testgithub (master)
$
```

Currently we have only one branch.

57. If you want to see all branches (including remote-tracking branches), use the *-a* for the `git branch` command.



```
Admin@ATTUNE31 ~/desktop/testgithub (master)
$ git branch
* master

Admin@ATTUNE31 ~/desktop/testgithub (master)
$ git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/master

Admin@ATTUNE31 ~/desktop/testgithub (master)
$
```

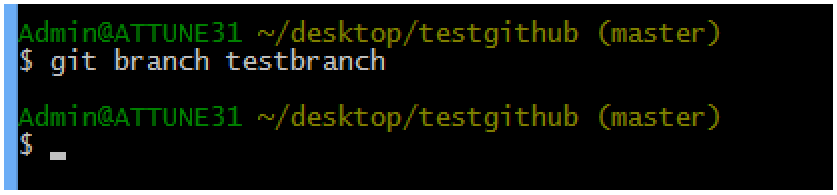
The *-v* option lists more information about the branches.

In order to list branches in a remote repository use the `git branch -r` command as demonstrated in the following example.

```
# lists branches in the remote repositories
git branch -r
```

58. You can create a new branch via the `git branch [newname]` command. This command allows specifying the starting

point (commit id, tag, remote or local branch). If not specified the commit to which the HEAD reference points is used to create the branch.



```
Admin@ATTUNE31 ~/desktop/testgithub (master)
$ git branch testbranch

Admin@ATTUNE31 ~/desktop/testgithub (master)
$ _
```

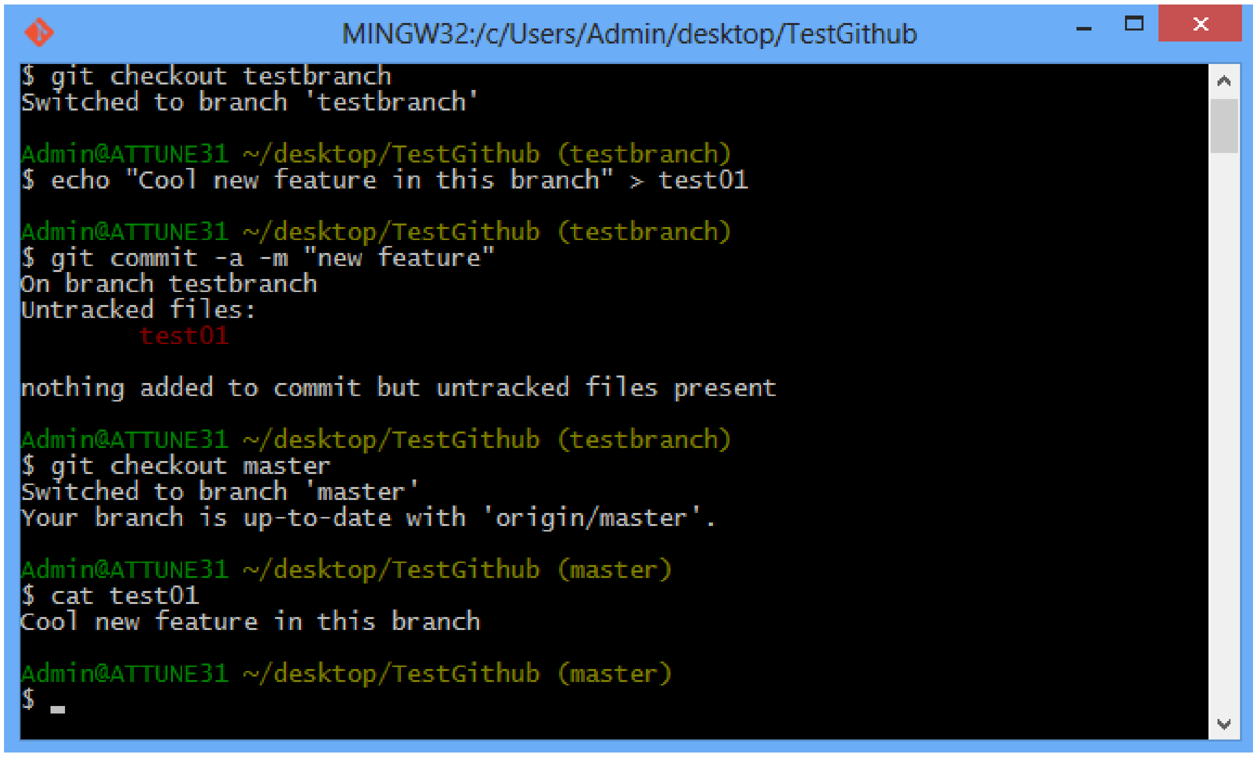
Checkout branch

59.Checkout branch:-

To start working in a branch you have to *checkout* the branch. If you *checkout* a branch, the HEAD pointer moves to the last commit in this branch and the files in the working tree are set to the state of this commit.

The following commands demonstrate how you switch to the branch called *testing*, perform some changes in this branch and switch back to the branch called *master*.

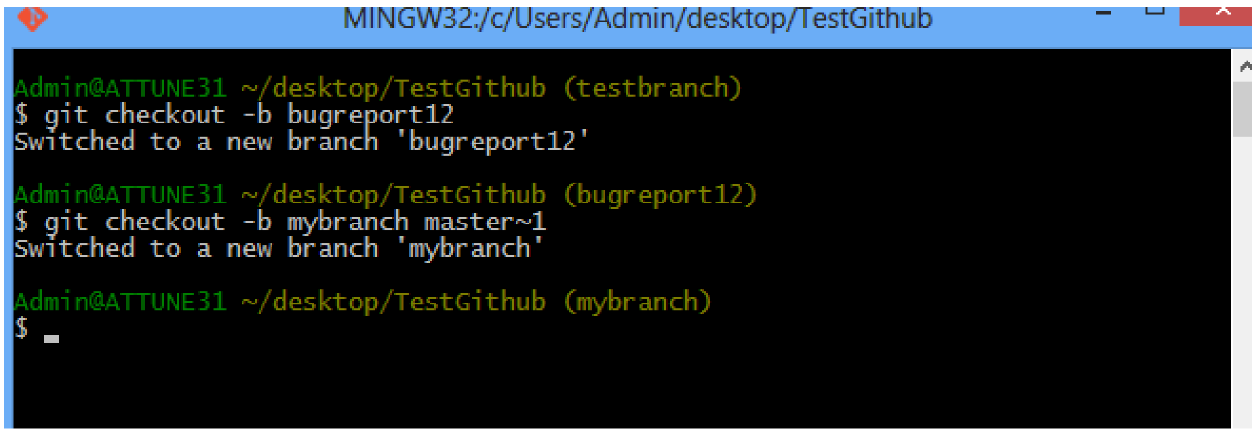
```
# switch to your new branch
git checkout testbranch

# does some changes
echo "Cool new feature in this branch" > test01
git commit -a -m "new feature"

# switch to the master branch
git checkout master

# check that the content of
# the test01 file is the old one
cat test01
```



```
$ git checkout testbranch
Switched to branch 'testbranch'

Admin@ATTUNE31 ~/desktop/TestGithub (testbranch)
$ echo "Cool new feature in this branch" > test01

Admin@ATTUNE31 ~/desktop/TestGithub (testbranch)
$ git commit -a -m "new feature"
On branch testbranch
Untracked files:
        test01

nothing added to commit but untracked files present

Admin@ATTUNE31 ~/desktop/TestGithub (testbranch)
$ git checkout master
Switched to branch 'master'
Your branch is up-to-date with 'origin/master'.

Admin@ATTUNE31 ~/desktop/TestGithub (master)
$ cat test01
Cool new feature in this branch

Admin@ATTUNE31 ~/desktop/TestGithub (master)
$
```

60. To create a branch and to switch to it at the same time you can use the `git checkout` command with the *-b* parameter.

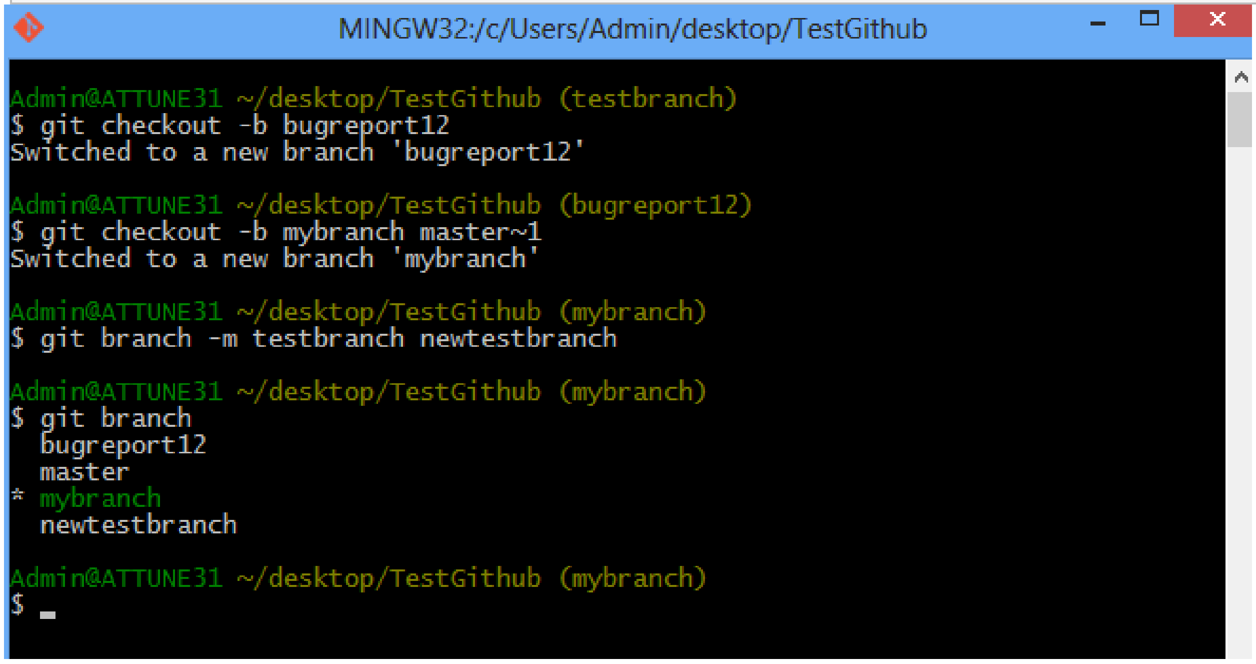
```
# create branch and switch to it
git checkout -b bugreport12

# creates a new branch based on the master branch
# Without the last commit
git checkout -b mybranch master~1
```



```
Admin@ATTUNE31 ~/desktop/TestGithub (testbranch)
$ git checkout -b bugreport12
Switched to a new branch 'bugreport12'

Admin@ATTUNE31 ~/desktop/TestGithub (bugreport12)
$ git checkout -b mybranch master~1
Switched to a new branch 'mybranch'

Admin@ATTUNE31 ~/desktop/TestGithub (mybranch)
$
```

61.Renaming a branch can be done with the following command.

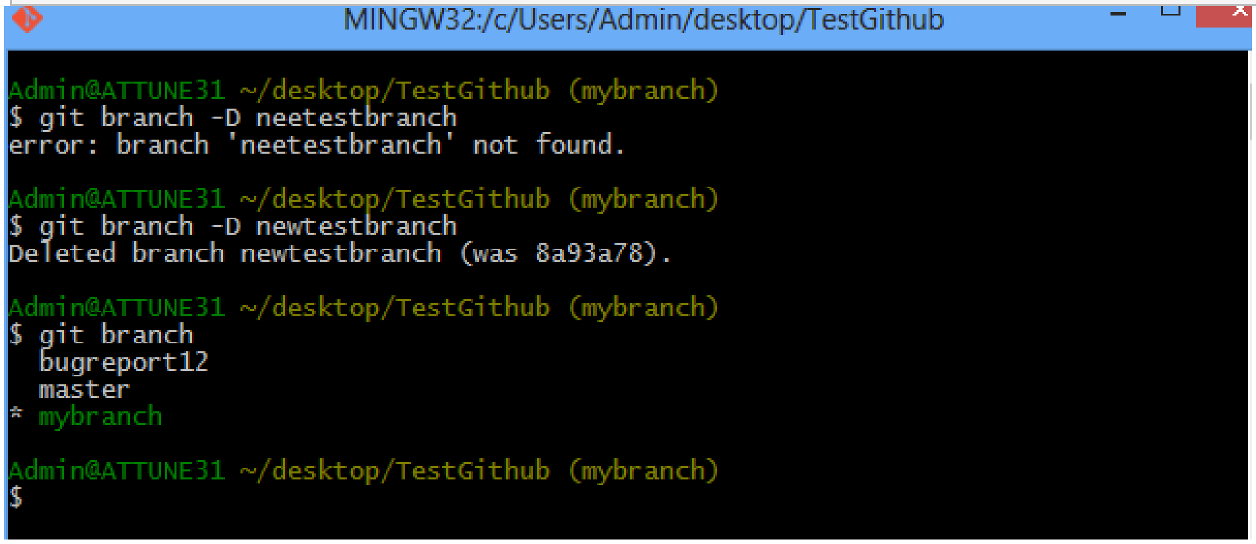
```
62. # rename branch
63. git branch -m [old_name] [new_name]
```



```
Admin@ATTUNE31 ~/desktop/TestGithub (testbranch)
$ git checkout -b bugreport12
Switched to a new branch 'bugreport12'

Admin@ATTUNE31 ~/desktop/TestGithub (bugreport12)
$ git checkout -b mybranch master~1
Switched to a new branch 'mybranch'

Admin@ATTUNE31 ~/desktop/TestGithub (mybranch)
$ git branch -m testbranch newtestbranch

Admin@ATTUNE31 ~/desktop/TestGithub (mybranch)
$ git branch
  bugreport12
  master
* mybranch
  newtestbranch

Admin@ATTUNE31 ~/desktop/TestGithub (mybranch)
$
```

64. To delete a branch which is not needed anymore, you can use the following command. You may get an error message that there are uncommitted changes if you did the previous examples step by step. Use force delete (uppercase *-D*) to delete it anyway.

```
# delete branch testing
git branch -d testing
# force delete testing
git branch -D testing
# check if branch has been deleted
git branch
```



MINGW32:/c/Users/Admin/desktop/TestGithub

```
Admin@ATTUNE31 ~/desktop/TestGithub (mybranch)
$ git branch -D neetestbranch
error: branch 'neetestbranch' not found.

Admin@ATTUNE31 ~/desktop/TestGithub (mybranch)
$ git branch -D newtestbranch
Deleted branch newtestbranch (was 8a93a78).

Admin@ATTUNE31 ~/desktop/TestGithub (mybranch)
$ git branch
  bugreport12
  master
* mybranch

Admin@ATTUNE31 ~/desktop/TestGithub (mybranch)
$
```