Git (Written by Rahul Gupta)

Saturday, March 28, 2020 4:02 PM

Introduction

Some Basic Points on GIT

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

- Git is Version Control System
- Snapshot in Git is called commit
- GIT is Opensource Distributed revision control system
- Each changes is committed with a log message
- Files can be rolled back to earlier version

VCS /GIT

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

Few things you should know before starting of git.
We will discuss each one of them when we are going to to do lab sections.

GIT server : Where is git going to be installed

- Cloud based or in-house setup

GIT Client :

- User Workstation

GIT repository:

- Directory where you initialized the git metadata only that data can be recorded.

GIT Stage:

- The object on which you want to enable versioning, You need to add the data in reference of git know as trace or in stage state.
- Those object which are not in stage state will not come under commit(non stage or non trace file)

Branch

- Equivalent to partition
- First commit belongs to master

GIT commit

- Last restoration point of your changes.

##Up next we are going to perform Lab1 where we see how git works.

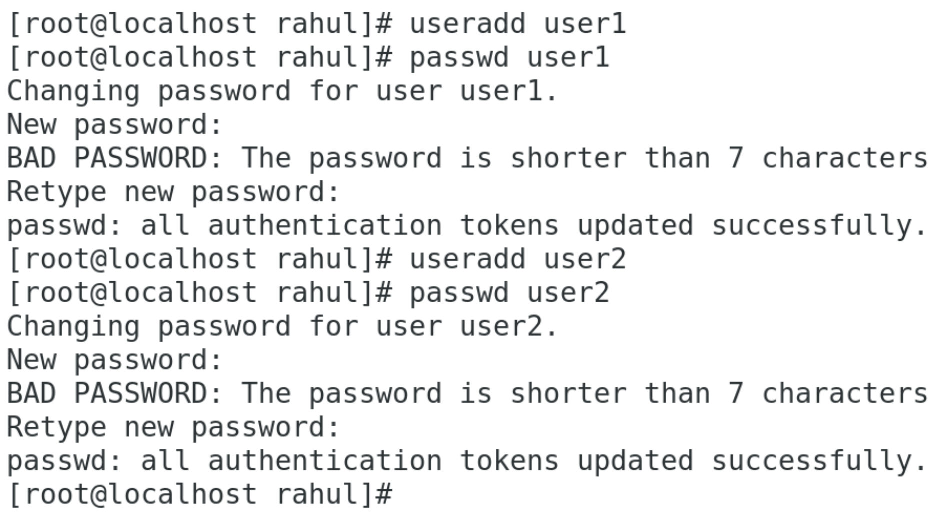
GIT LAB 1 (CREATION OF USER'S AND ENABLING GIT)

Wednesday, March 4, 2020 7:07 AM

Lab Practice 1

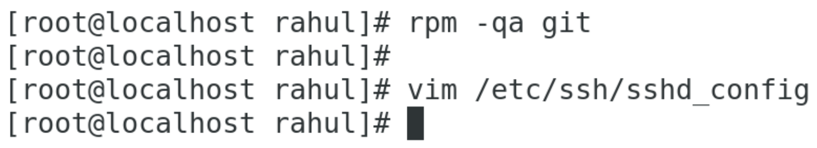
GIT LAB 1 (CREATION OF USER'S AND ENABLING GIT)

Step 1 : Create two user for git (user1 and user2)

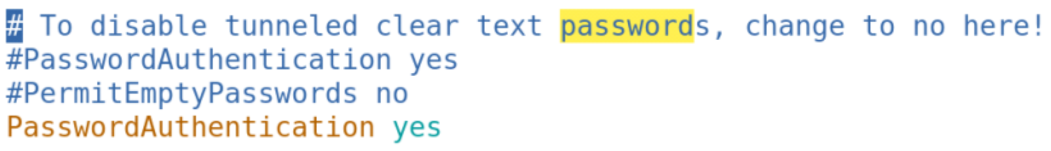


```
[root@localhost rahul]# useradd user1
[root@localhost rahul]# passwd user1
Changing password for user user1.
New password:
BAD PASSWORD: The password is shorter than 7 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost rahul]# useradd user2
[root@localhost rahul]# passwd user2
Changing password for user user2.
New password:
BAD PASSWORD: The password is shorter than 7 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost rahul]#
```

- Enabling ssh access using password based

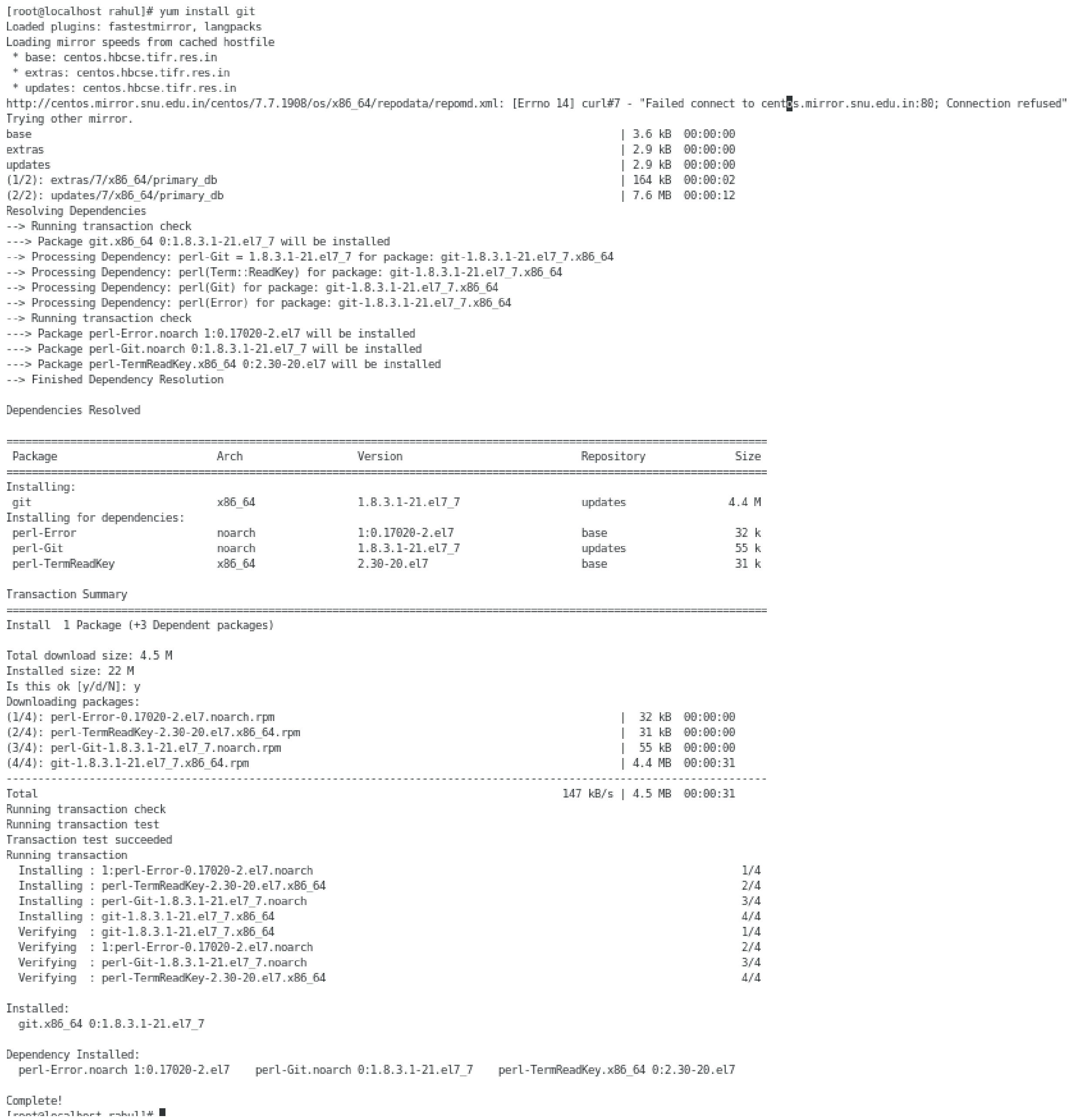


```
[root@localhost rahul]# rpm -qa git
[root@localhost rahul]#
[root@localhost rahul]# vim /etc/ssh/sshd_config
[root@localhost rahul]#
```



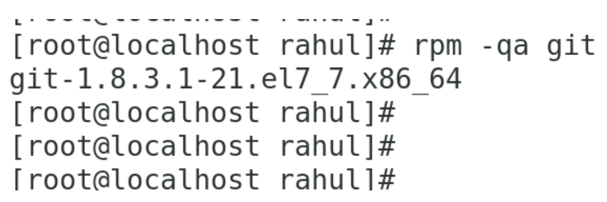
```
# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no
PasswordAuthentication yes
```

Step 2: Install GIT from root user



```
[root@localhost rahul]# yum install git
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: centos.hbcse.tifr.res.in
 * extras: centos.hbcse.tifr.res.in
 * updates: centos.hbcse.tifr.res.in
http://centos.mirror.snu.edu.in/centos/7.7.1908/os/x86_64/repodata/repomd.xml: [Errno 14] curl#7 - "Failed connect to centos.mirror.snu.edu.in:80; Connection refused"
Trying other mirror.
base                                                                                            | 3.6 kB  00:00:00
extras                                                                                          | 2.9 kB  00:00:00
updates                                                                                         | 2.9 kB  00:00:00
(1/2): extras/7/x86_64/primary_db                                                               | 164 kB  00:00:02
(2/2): updates/7/x86_64/primary_db                                                              | 7.6 MB  00:00:12
Resolving Dependencies
--> Running transaction check
---> Package git.x86_64 0:1.8.3.1-21.el7_7 will be installed
--> Processing Dependency: perl-Git = 1.8.3.1-21.el7_7 for package: git-1.8.3.1-21.el7_7.x86_64
--> Processing Dependency: perl(Term::ReadKey) for package: git-1.8.3.1-21.el7_7.x86_64
--> Processing Dependency: perl(Git) for package: git-1.8.3.1-21.el7_7.x86_64
--> Processing Dependency: perl(Error) for package: git-1.8.3.1-21.el7_7.x86_64
--> Running transaction check
---> Package perl-Error.noarch 1:0.17020-2.el7 will be installed
---> Package perl-Git.noarch 0:1.8.3.1-21.el7_7 will be installed
---> Package perl-TermReadKey.x86_64 0:2.30-20.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=======================================================================================================================
 Package                          Arch                   Version                                  Repository       Size
=======================================================================================================================
Installing:
 git                              x86_64                 1.8.3.1-21.el7_7                         updates         4.4 M
Installing for dependencies:
 perl-Error                       noarch                 1:0.17020-2.el7                          base             32 k
 perl-Git                         noarch                 1.8.3.1-21.el7_7                         updates          55 k
 perl-TermReadKey                 x86_64                 2.30-20.el7                              base             31 k

Transaction Summary
=======================================================================================================================
Install  1 Package (+3 Dependent packages)

Total download size: 4.5 M
Installed size: 22 M
Is this ok [y/d/N]: y
Downloading packages:
(1/4): perl-Error-0.17020-2.el7.noarch.rpm                                                      |  32 kB  00:00:00
(2/4): perl-TermReadKey-2.30-20.el7.x86_64.rpm                                                  |  31 kB  00:00:00
(3/4): perl-Git-1.8.3.1-21.el7_7.noarch.rpm                                                     |  55 kB  00:00:00
(4/4): git-1.8.3.1-21.el7_7.x86_64.rpm                                                          | 4.4 MB  00:00:31
-----------------------------------------------------------------------------------------------------------------------
Total                                                                                  147 kB/s | 4.5 MB  00:00:31
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : 1:perl-Error-0.17020-2.el7.noarch                                                                   1/4
  Installing : perl-TermReadKey-2.30-20.el7.x86_64                                                                 2/4
  Installing : perl-Git-1.8.3.1-21.el7_7.noarch                                                                    3/4
  Installing : git-1.8.3.1-21.el7_7.x86_64                                                                         4/4
  Verifying  : git-1.8.3.1-21.el7_7.x86_64                                                                         1/4
  Verifying  : 1:perl-Error-0.17020-2.el7.noarch                                                                   2/4
  Verifying  : perl-Git-1.8.3.1-21.el7_7.noarch                                                                    3/4
  Verifying  : perl-TermReadKey-2.30-20.el7.x86_64                                                                 4/4

Installed:
  git.x86_64 0:1.8.3.1-21.el7_7

Dependency Installed:
  perl-Error.noarch 1:0.17020-2.el7    perl-Git.noarch 0:1.8.3.1-21.el7_7    perl-TermReadKey.x86_64 0:2.30-20.el7

Complete!
[root@localhost rahul]#
```

- ➢ Check if git is installed from user1 and from root

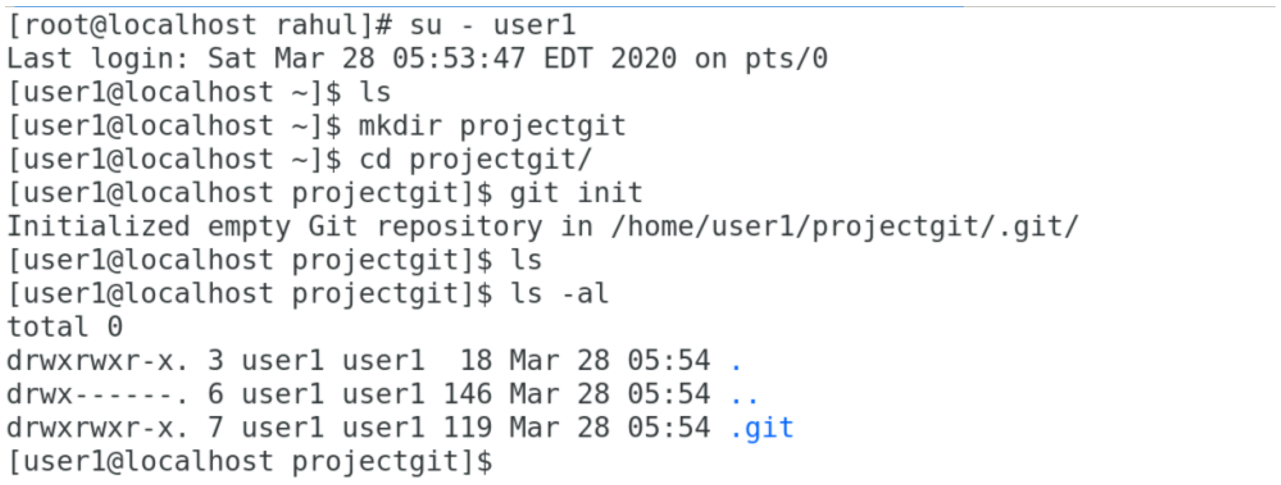


```
[root@localhost rahul]# rpm -qa git
git-1.8.3.1-21.el7_7.x86_64
[root@localhost rahul]#
[root@localhost rahul]#
[root@localhost rahul]#
```

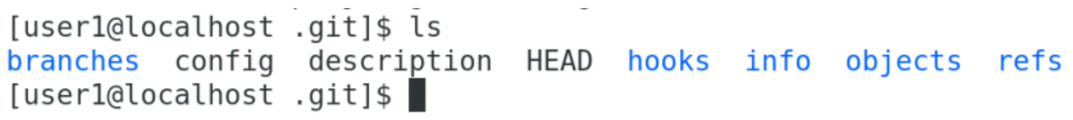
Step3:

From user1 >

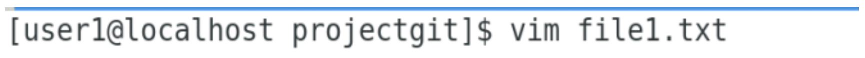
Initializing git on directory "projectgit"

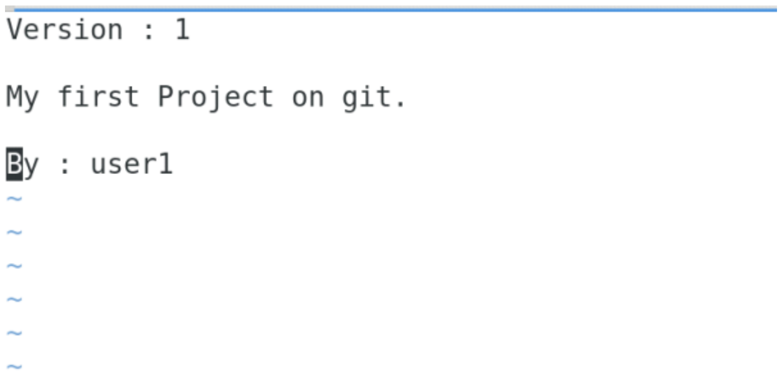

```
[root@localhost rahul]# su - user1
Last login: Sat Mar 28 05:53:47 EDT 2020 on pts/0
[user1@localhost ~]$ ls
[user1@localhost ~]$ mkdir projectgit
[user1@localhost ~]$ cd projectgit/
[user1@localhost projectgit]$ git init
Initialized empty Git repository in /home/user1/projectgit/.git/
[user1@localhost projectgit]$ ls
[user1@localhost projectgit]$ ls -al
total 0
drwxrwxr-x. 3 user1 user1  18 Mar 28 05:54 .
drwx------. 6 user1 user1 146 Mar 28 05:54 ..
drwxrwxr-x. 7 user1 user1 119 Mar 28 05:54 .git
[user1@localhost projectgit]$
```

- Checking what's inside git:

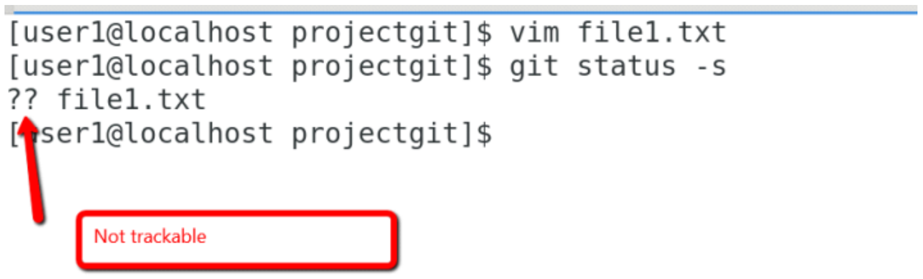

```
[user1@localhost .git]$ ls
branches  config  description  HEAD  hooks  info  objects  refs
[user1@localhost .git]$ 
```

- Creating test file


```
[user1@localhost projectgit]$ vim file1.txt
```


```
Version : 1

My first Project on git.

By : user1
~
~
~
~
~
~
```

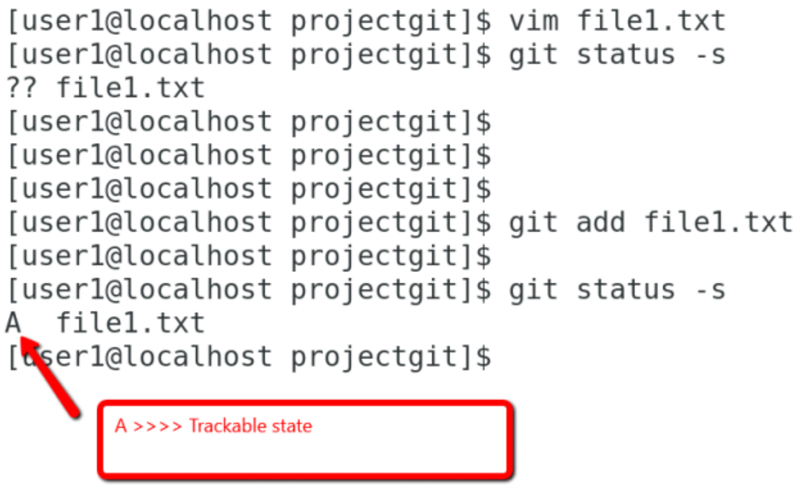
Step 4: checking via git command status of this newly created file:



```
[user1@localhost projectgit]$ vim file1.txt
[user1@localhost projectgit]$ git status -s
?? file1.txt
[user1@localhost projectgit]$
```

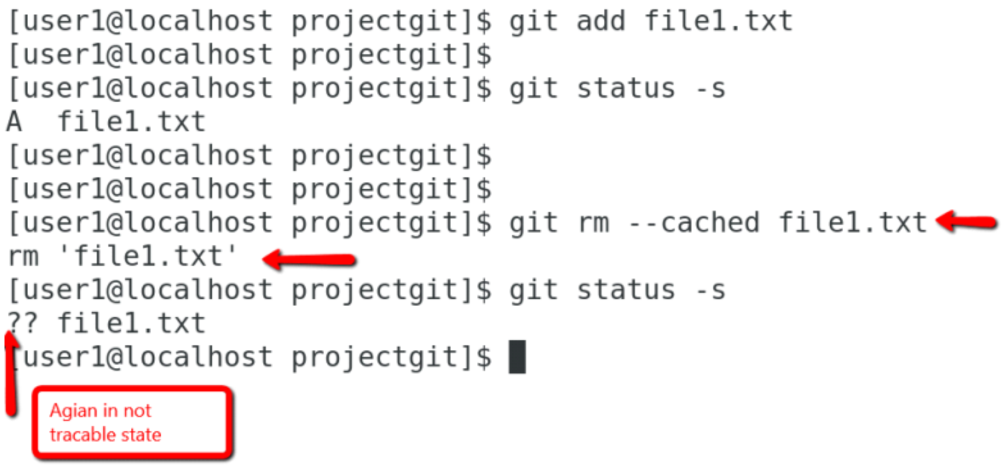
Currently file is not in trackable state:

To bring it to trace state:



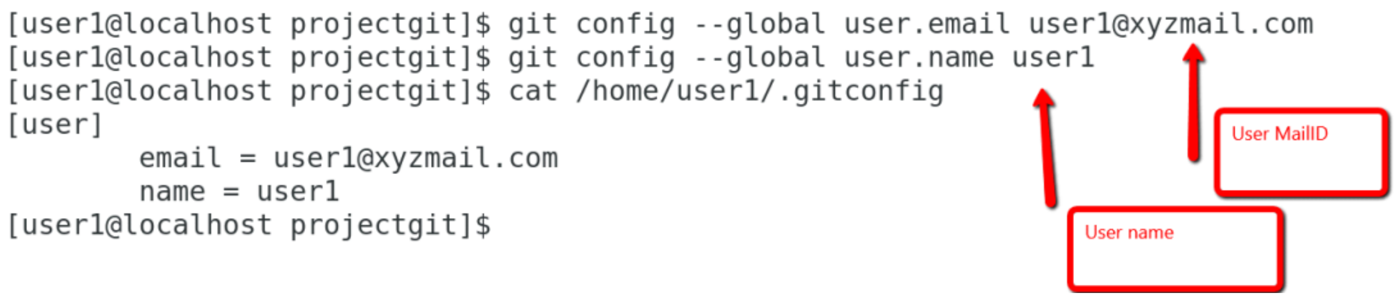
```
[user1@localhost projectgit]$ vim file1.txt
[user1@localhost projectgit]$ git status -s
?? file1.txt
[user1@localhost projectgit]$
[user1@localhost projectgit]$
[user1@localhost projectgit]$
[user1@localhost projectgit]$ git add file1.txt
[user1@localhost projectgit]$
[user1@localhost projectgit]$ git status -s
A  file1.txt
[user1@localhost projectgit]$
```

To remove it from tracking:



```
[user1@localhost projectgit]$ git add file1.txt
[user1@localhost projectgit]$
[user1@localhost projectgit]$ git status -s
A  file1.txt
[user1@localhost projectgit]$
[user1@localhost projectgit]$
[user1@localhost projectgit]$ git rm --cached file1.txt
rm 'file1.txt'
[user1@localhost projectgit]$ git status -s
?? file1.txt
[user1@localhost projectgit]$
```

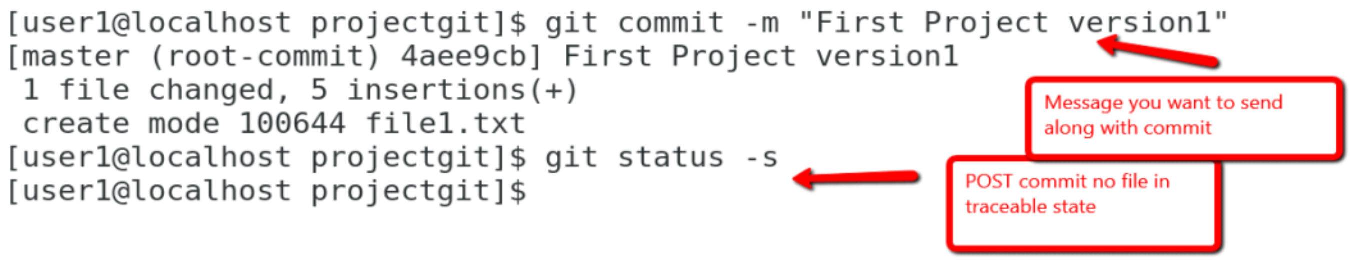
Step 5: To configure user 2 to use git . Need to run below mentioned commands .
Post running command you can see in .gitconfig file value is saved



```
[user1@localhost projectgit]$ git config --global user.email user1@xyzmail.com
[user1@localhost projectgit]$ git config --global user.name user1
[user1@localhost projectgit]$ cat /home/user1/.gitconfig
[user]
        email = user1@xyzmail.com
        name = user1
[user1@localhost projectgit]$
```

Step 6:
Need to commit the changes :
Commit means you are ready to push the changes to git server.
(commit doesn't means that your changes has been pushed to server)

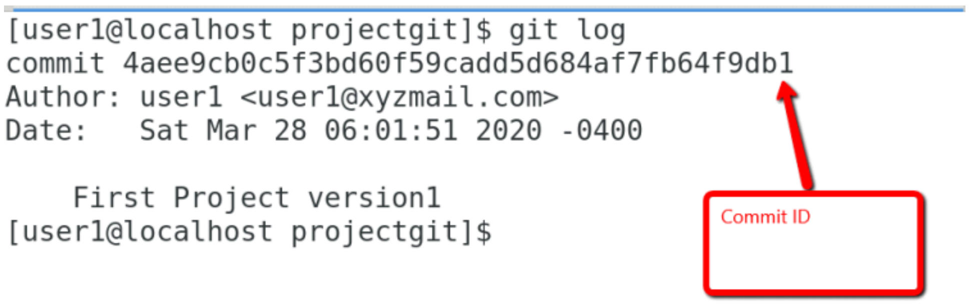
- Commit is local restoration Point of your changes.



```
[user1@localhost projectgit]$ git commit -m "First Project version1"
[master (root-commit) 4aee9cb] First Project version1
 1 file changed, 5 insertions(+)
 create mode 100644 file1.txt
[user1@localhost projectgit]$ git status -s
[user1@localhost projectgit]$
```

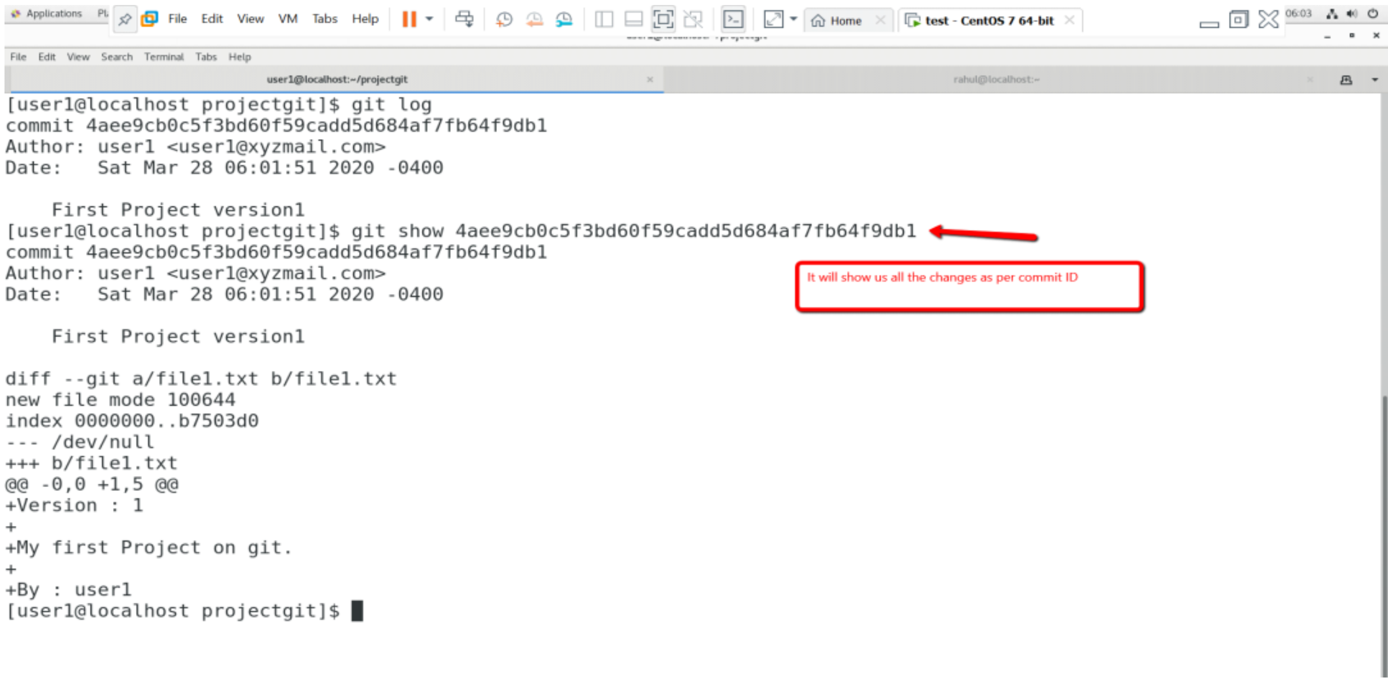
Message you want to send along with commit
POST commit no file in traceable state

- Git logs to check your commit



```
[user1@localhost projectgit]$ git log
commit 4aee9cb0c5f3bd60f59cadd5d684af7fb64f9db1
Author: user1 <user1@xyzmail.com>
Date:   Sat Mar 28 06:01:51 2020 -0400

    First Project version1
[user1@localhost projectgit]$
```

- Git show to show us all the changes:


```
[user1@localhost projectgit]$ git log
commit 4aee9cb0c5f3bd60f59cadd5d684af7fb64f9db1
Author: user1 <user1@xyzmail.com>
Date:   Sat Mar 28 06:01:51 2020 -0400

    First Project version1
[user1@localhost projectgit]$ git show 4aee9cb0c5f3bd60f59cadd5d684af7fb64f9db1
commit 4aee9cb0c5f3bd60f59cadd5d684af7fb64f9db1
Author: user1 <user1@xyzmail.com>
Date:   Sat Mar 28 06:01:51 2020 -0400

    First Project version1

diff --git a/file1.txt b/file1.txt
new file mode 100644
index 0000000..b7503d0
--- /dev/null
+++ b/file1.txt
@@ -0,0 +1,5 @@
+Version : 1
+
+My first Project on git.
+
+By : user1
[user1@localhost projectgit]$ 
```
It will show us all the changes as per commit ID

##Up next we are going to see how GIT server works. Till now we have see git client.

GIT server : Setting up Github

Saturday, March 28, 2020 6:08 PM

So far we have created user : user1.
User 1 initiated the Git services. He created his file and commit that file.
Now post commit we need to push this file to git server.

Git server can be of two type :

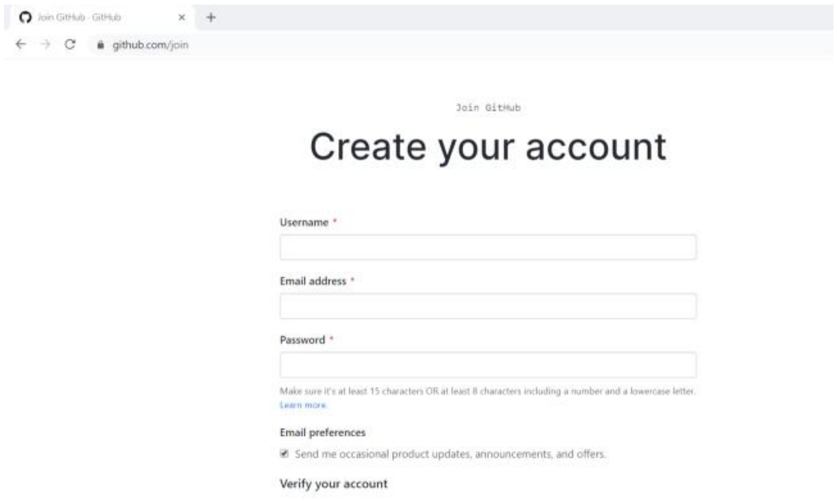
1. Cloud based (Gitlab,Github etc)
2. In house.

We are going to first discuss cloud based and later we will discuss about inhouse.

For this session we are going to use Github.

To join GitHub we need to have a account. No worries we can create account for free just follow below mentioned steps.

Step1: open Github join page:



Join GitHub - GitHub

github.com/join

Join GitHub

Create your account

Username *

Email address *

Password *

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. Learn more.

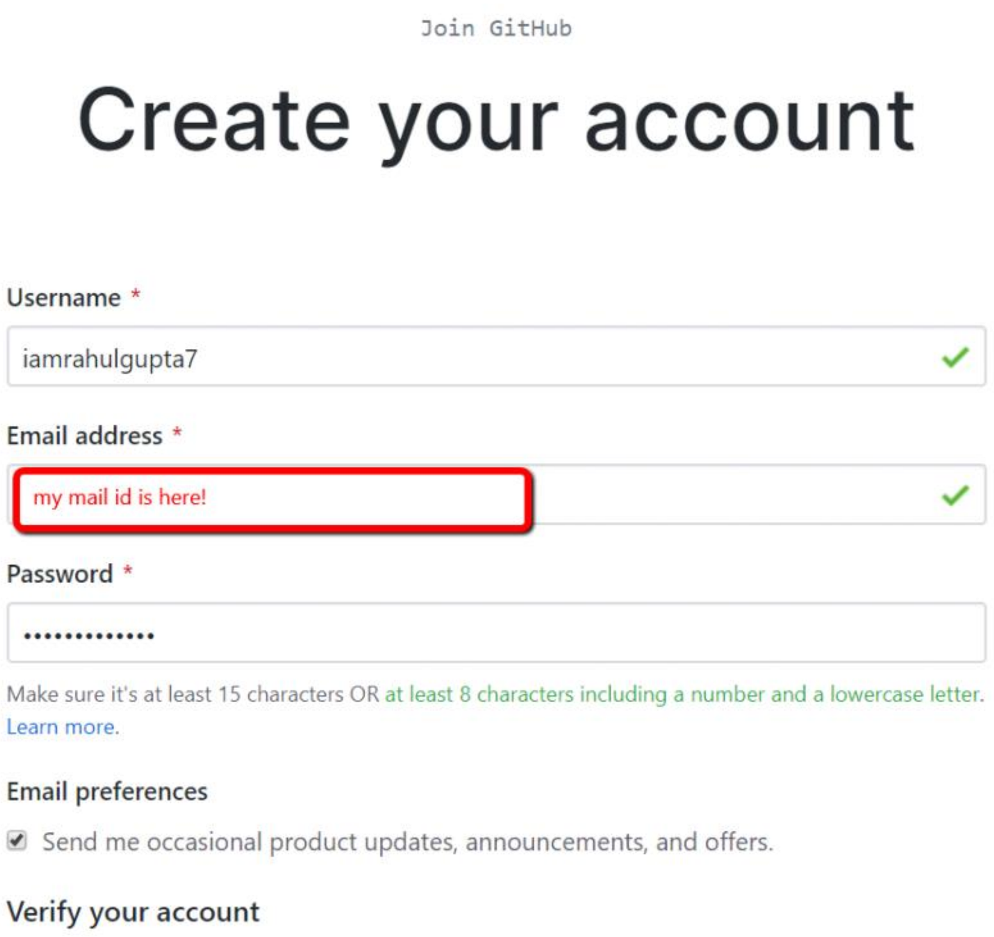
Email preferences

[✓] Send me occasional product updates, announcements, and offers.

Verify your account

- ➢ Enter your details.

I am using my mail ID to create a account


Join GitHub

Create your account

Username *

iamrahulgupta7

Email address *

my mail id is here!

Password *

••••••••••••••

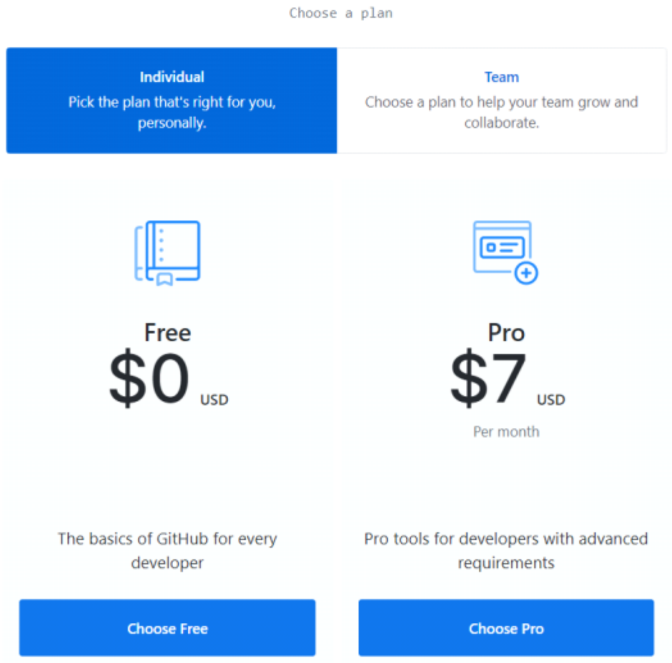
Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. Learn more.

Email preferences

☑ Send me occasional product updates, announcements, and offers.

Verify your account

- Choose a plan.


Choose a plan

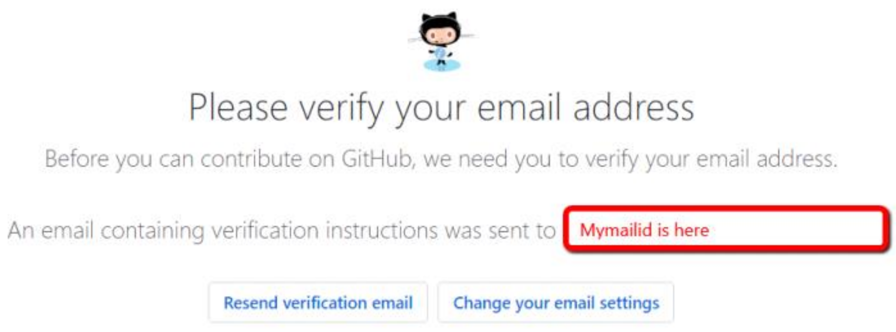
Individual — Pick the plan that's right for you, personally.

Team — Choose a plan to help your team grow and collaborate.

Free — $0 USD — The basics of GitHub for every developer — Choose Free

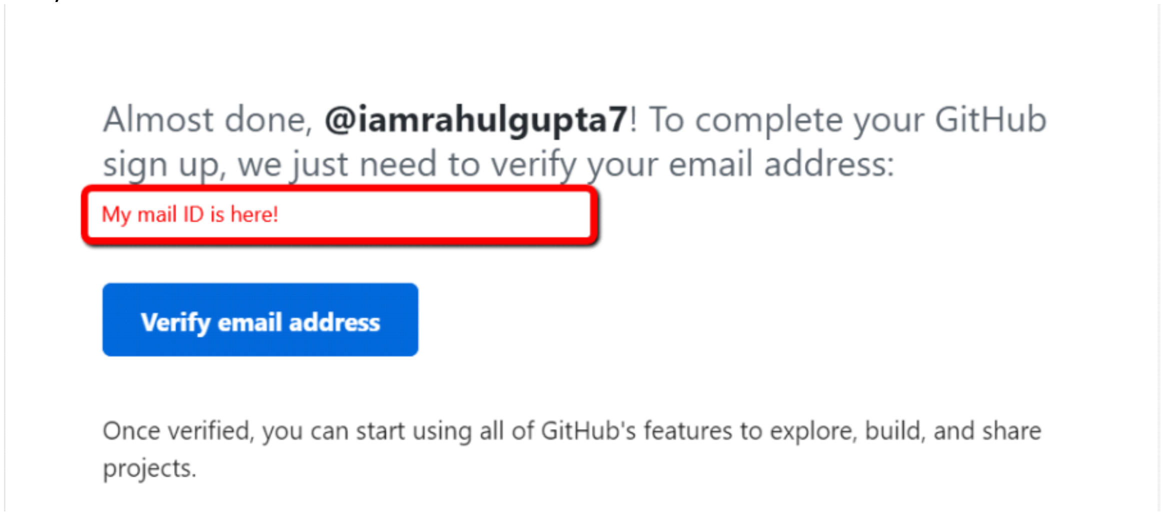
Pro — $7 USD Per month — Pro tools for developers with advanced requirements — Choose Pro

- Mail will be sent for verification


Please verify your email address

Before you can contribute on GitHub, we need you to verify your email address.

An email containing verification instructions was sent to Mymailid is here

- In my mailbox verification mail:
- 

Almost done, **@iamrahulgupta7**! To complete your GitHub sign up, we just need to verify your email address:

My mail ID is here!

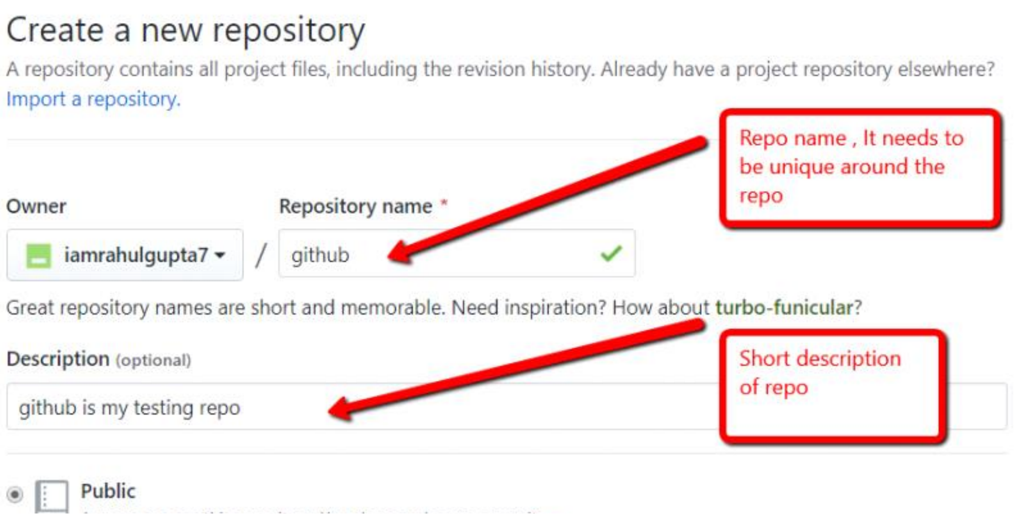
Once verified, you can start using all of GitHub's features to explore, build, and share projects.

Post verification , your account will be active and you can use github

Step2:
Creating new repo in github.

- Login into github
- Create Repo

I have create a new repo name : github ,


Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

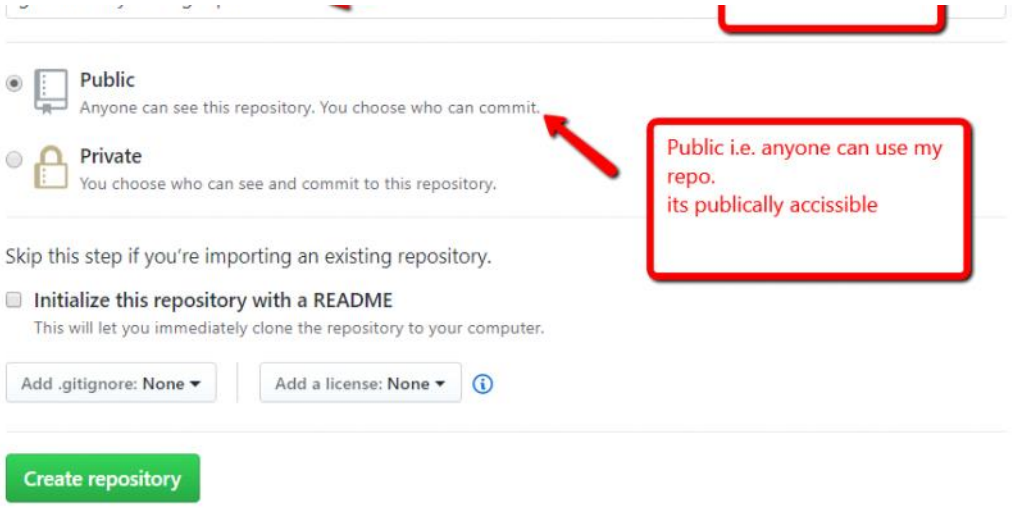
Owner: iamrahulgupta7 / Repository name *: github

Great repository names are short and memorable. Need inspiration? How about turbo-funicular?

Description (optional): github is my testing repo

Public

Repo name , It needs to be unique around the repo
Short description of repo



Public

Anyone can see this repository. You choose who can commit.

Private

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

Initialize this repository with a README

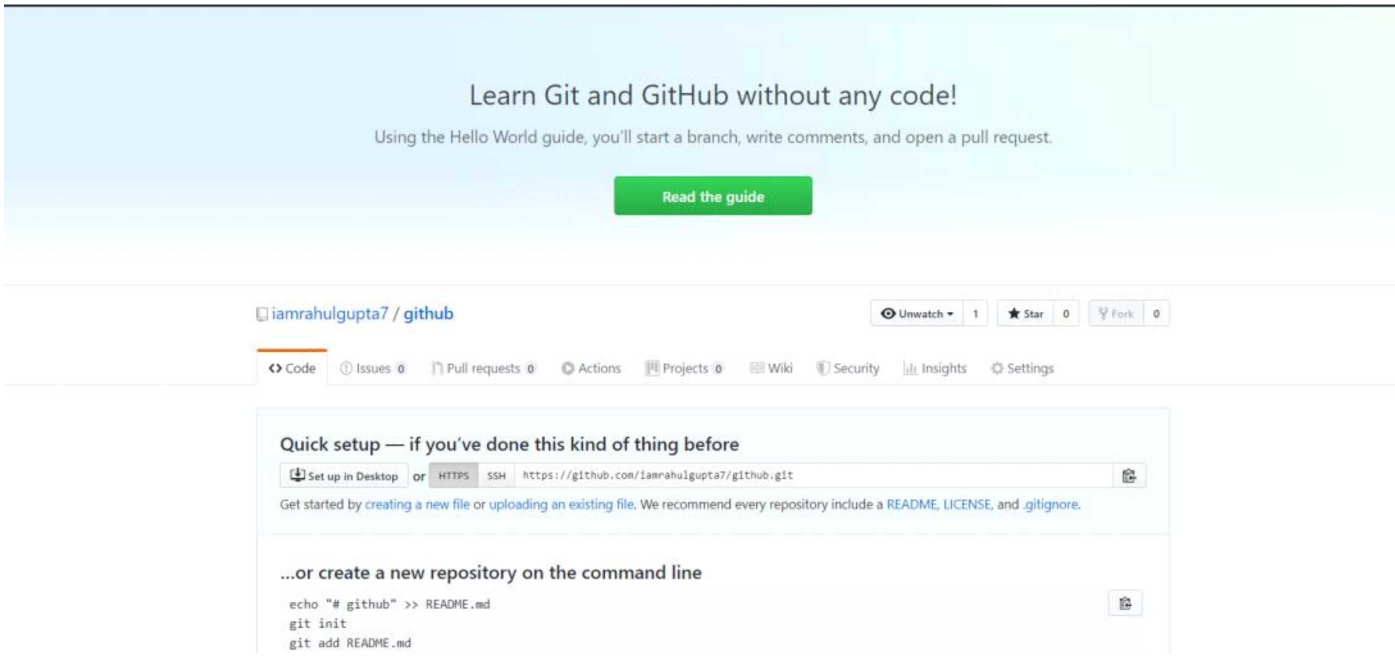
This will let you immediately clone the repository to your computer.

Add .gitignore: None ▾ | Add a license: None ▾

Create repository

Public i.e. anyone can use my repo.
its publically accissible

Repo is created



Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

Read the guide

iamrahulgupta7 / github

Unwatch ▾ 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Actions | Projects 0 | Wiki | Security | Insights | Settings

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH https://github.com/iamrahulgupta7/github.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# github" >> README.md
git init
git add README.md
```

Git server setup has been completed successfully.

To access the repo from command line we have two method: https and SSH

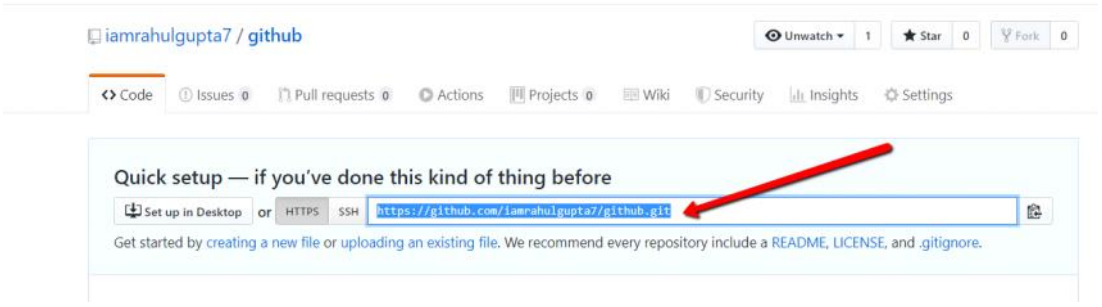
##Up next For user1 we will use https and for user2 we will use SSH for grabbing all the content.

Git via HTTP

Saturday, March 28, 2020 11:23 PM

For user1 we are going to use http to access repo. Once we are able to access git server we will push our file : 'file1.txt' which we have created to git server.

Step1: Checking git we can see there are no current file present.
Copy http link present on git server.

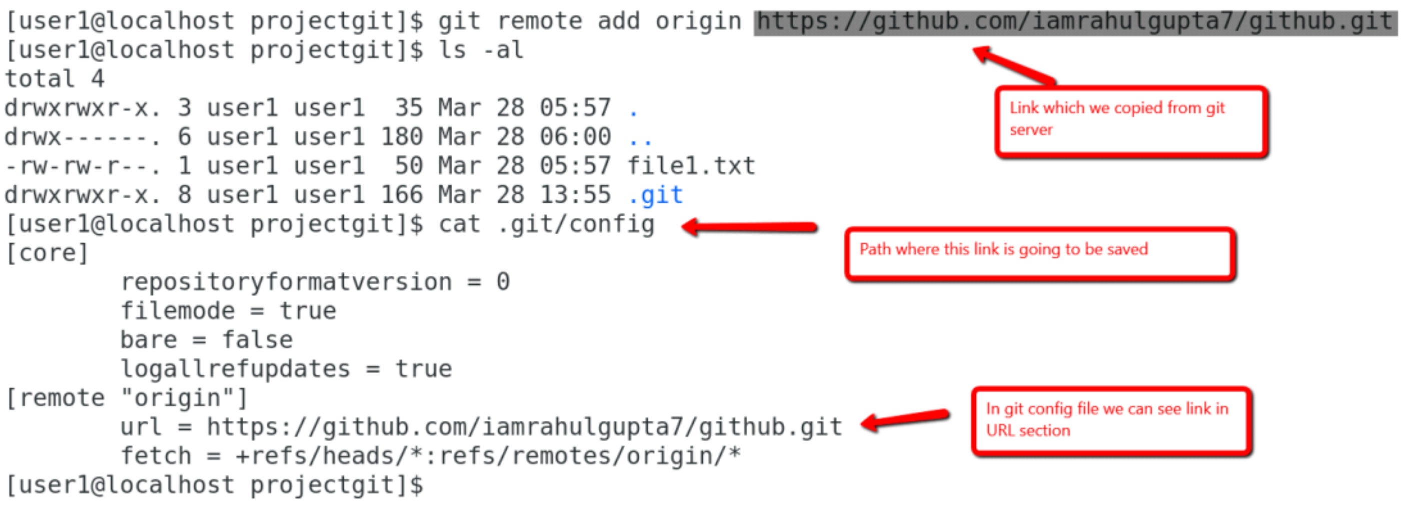


Quick setup — if you've done this kind of thing before

https://github.com/iamrahulgupta7/github.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

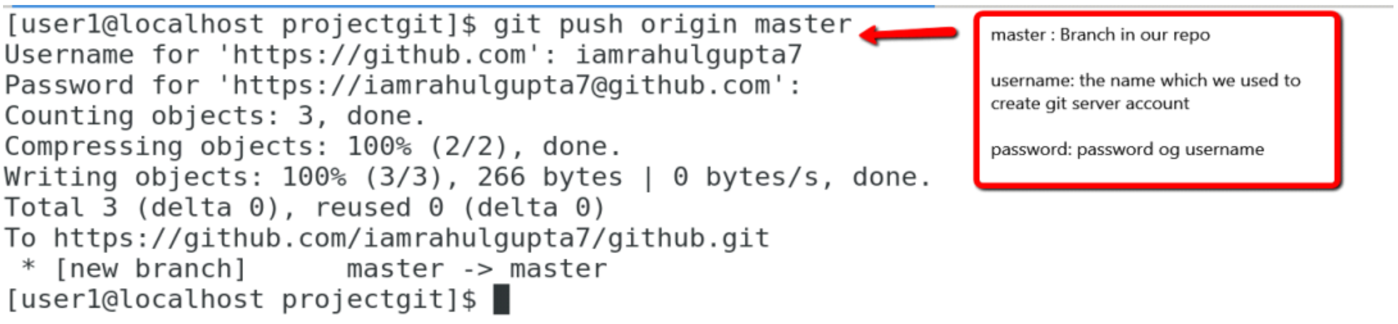
Step2 : from user1 :



```
[user1@localhost projectgit]$ git remote add origin https://github.com/iamrahulgupta7/github.git
[user1@localhost projectgit]$ ls -al
total 4
drwxrwxr-x. 3 user1 user1  35 Mar 28 05:57 .
drwx------. 6 user1 user1 180 Mar 28 06:00 ..
-rw-rw-r--. 1 user1 user1  50 Mar 28 05:57 file1.txt
drwxrwxr-x. 8 user1 user1 166 Mar 28 13:55 .git
[user1@localhost projectgit]$ cat .git/config
[core]
        repositoryformatversion = 0
        filemode = true
        bare = false
        logallrefupdates = true
[remote "origin"]
        url = https://github.com/iamrahulgupta7/github.git
        fetch = +refs/heads/*:refs/remotes/origin/*
[user1@localhost projectgit]$
```

Link which we copied from git server
Path where this link is going to be saved
In git config file we can see link in URL section

Step3: Now as we have mentioned which Git server our user need to connect.
Lets get connected to this Git server and push out file on git server



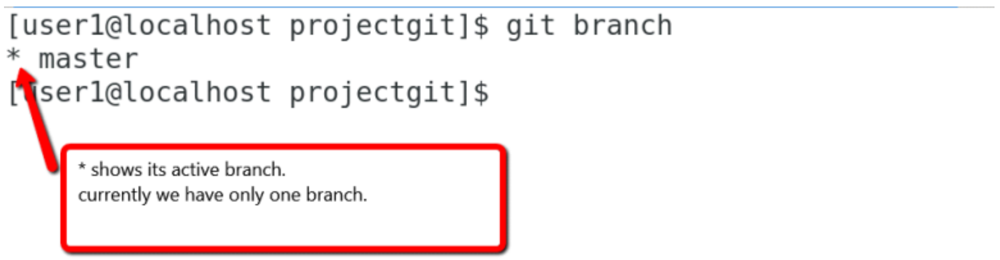
```
[user1@localhost projectgit]$ git push origin master
Username for 'https://github.com': iamrahulgupta7
Password for 'https://iamrahulgupta7@github.com':
Counting objects: 3, done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 266 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/iamrahulgupta7/github.git
 * [new branch]      master -> master
[user1@localhost projectgit]$
```

master : Branch in our repo

username: the name which we used to create git server account

password: password og username

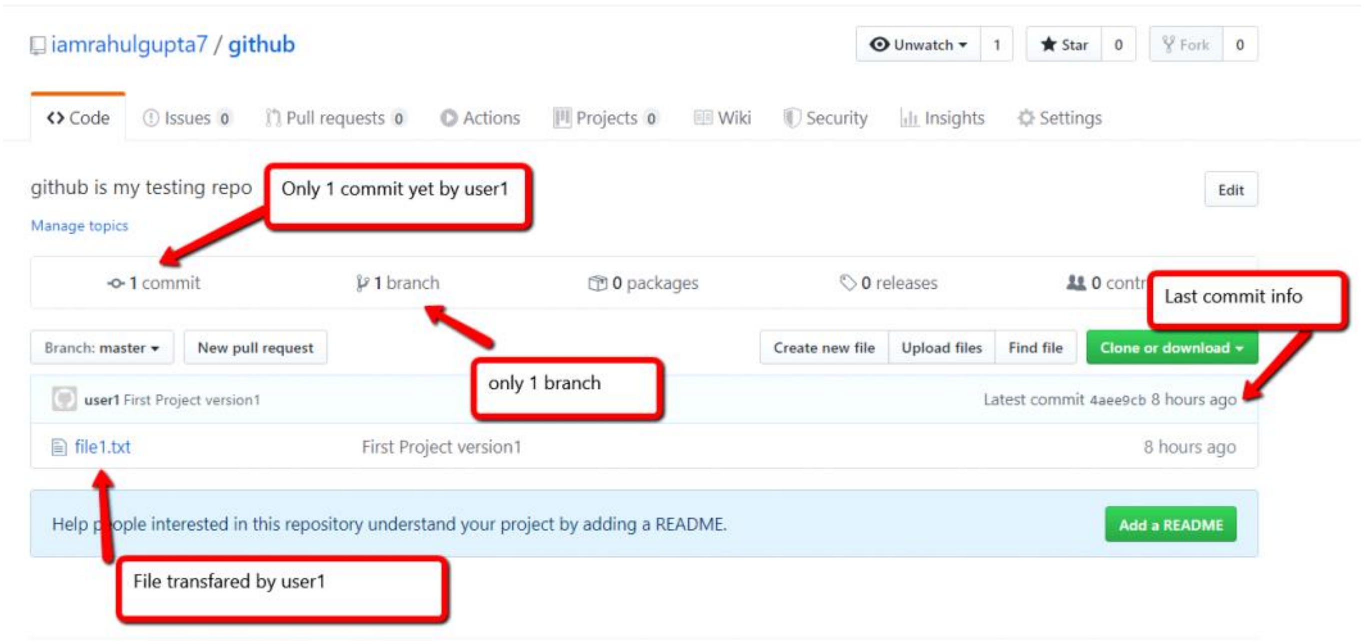
Step4: We can check our branch



```
[user1@localhost projectgit]$ git branch
* master
[user1@localhost projectgit]$
```

* shows its active branch.
currently we have only one branch.

Step5: On Git server now we can out file is showing.



github is my testing repo

Only 1 commit yet by user1

Last commit info

only 1 branch

user1 First Project version1 — Latest commit 4aee9cb 8 hours ago

file1.txt — First Project version1 — 8 hours ago

Help people interested in this repository understand your project by adding a README.

File transfared by user1

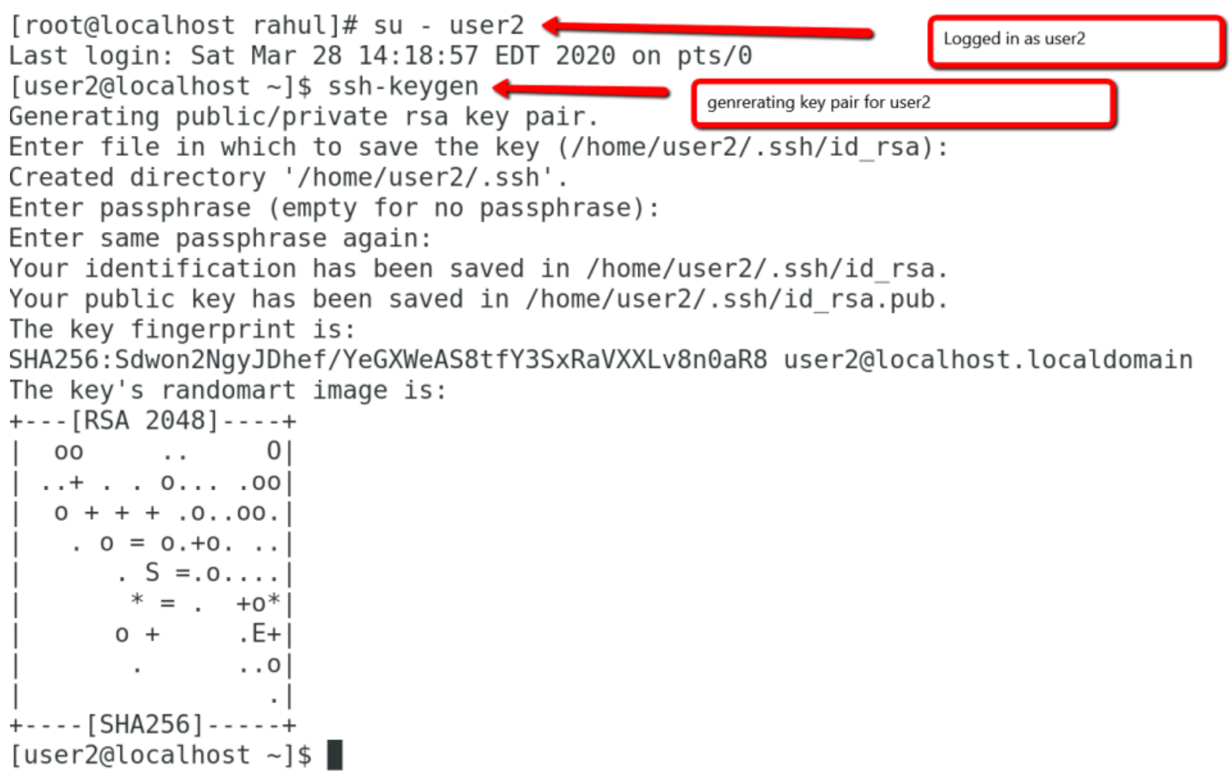
##Up next user2 we will use SSH to access git server same as user1 can do.

Git via SSH

Saturday, March 28, 2020 11:46 PM

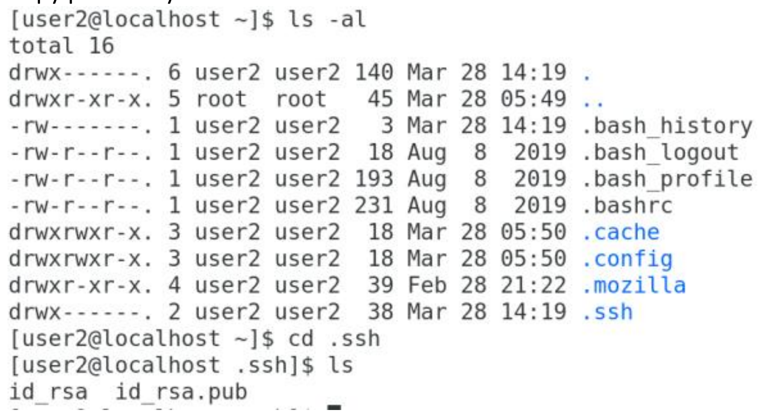
For User 2 we will access GIT server and pull the repo already created by user1.Post that we will create File2.txt and push it to Git server.

Step1: For accessing git server via SSH we need ssh keypair.
To generate key pair:



```
[root@localhost rahul]# su - user2
Last login: Sat Mar 28 14:18:57 EDT 2020 on pts/0
[user2@localhost ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user2/.ssh/id_rsa):
Created directory '/home/user2/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user2/.ssh/id_rsa.
Your public key has been saved in /home/user2/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:Sdwon2NgyJDhef/YeGXWeAS8tfY3SxRaVXXLv8n0aR8 user2@localhost.localdomain
The key's randomart image is:
+---[RSA 2048]----+
|  oo      ..     O|
| ..+ . . o... .oo|
|  o + + + .o..oo.|
|   . o = o.+o. ..|
|      . S =.o....|
|       * = .  +o*|
|      o +     .E+|
|       .      ..o|
|                .|
+----[SHA256]-----+
[user2@localhost ~]$
```

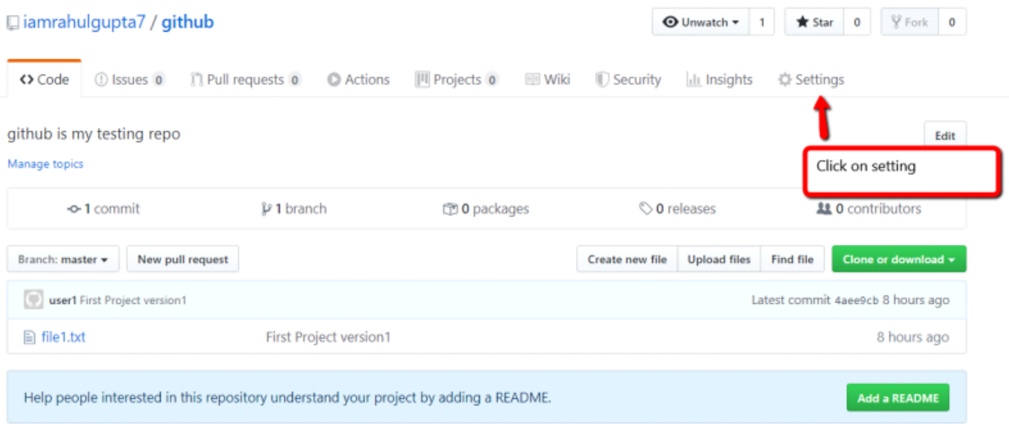
- Copy public key



```
[user2@localhost ~]$ ls -al
total 16
drwx------. 6 user2 user2 140 Mar 28 14:19 .
drwxr-xr-x. 5 root  root   45 Mar 28 05:49 ..
-rw-------. 1 user2 user2   3 Mar 28 14:19 .bash_history
-rw-r--r--. 1 user2 user2  18 Aug  8  2019 .bash_logout
-rw-r--r--. 1 user2 user2 193 Aug  8  2019 .bash_profile
-rw-r--r--. 1 user2 user2 231 Aug  8  2019 .bashrc
drwxrwxr-x. 3 user2 user2  18 Mar 28 05:50 .cache
drwxrwxr-x. 3 user2 user2  18 Mar 28 05:50 .config
drwxr-xr-x. 4 user2 user2  39 Feb 28 21:22 .mozilla
drwx------. 2 user2 user2  38 Mar 28 14:19 .ssh
[user2@localhost ~]$ cd .ssh
[user2@localhost .ssh]$ ls
id_rsa  id_rsa.pub
```

```
[user2@localhost .ssh]$ ls
id_rsa  id_rsa.pub
[user2@localhost .ssh]$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCmm2H/niPk1pUTFSO6kDjMaTKWqG4FcSmUAmpT45hqc
fKKJt1gBG1SKf0VW3ILkYj+Cci8/BKHh4yNE5dcNnoLokfpZY4p+f/O3fmtJjoLB1fmrklcgPAoFkM9L2
BN7ZeW5/A7IOiaXDH2ok7vF5T1SX1s2RA6PekHRt98dfjPjN+26Bfo1Elro6Wwr4v7DhFYNP4w9pSxuk8
j+YeUyU+KtFgcFCxLY8Lw5P user2@localhost.localdomain
[user2@localhost .ssh]$
```

Step2: Copy public key and we need paste it in git server > settings> deploy key > add new key

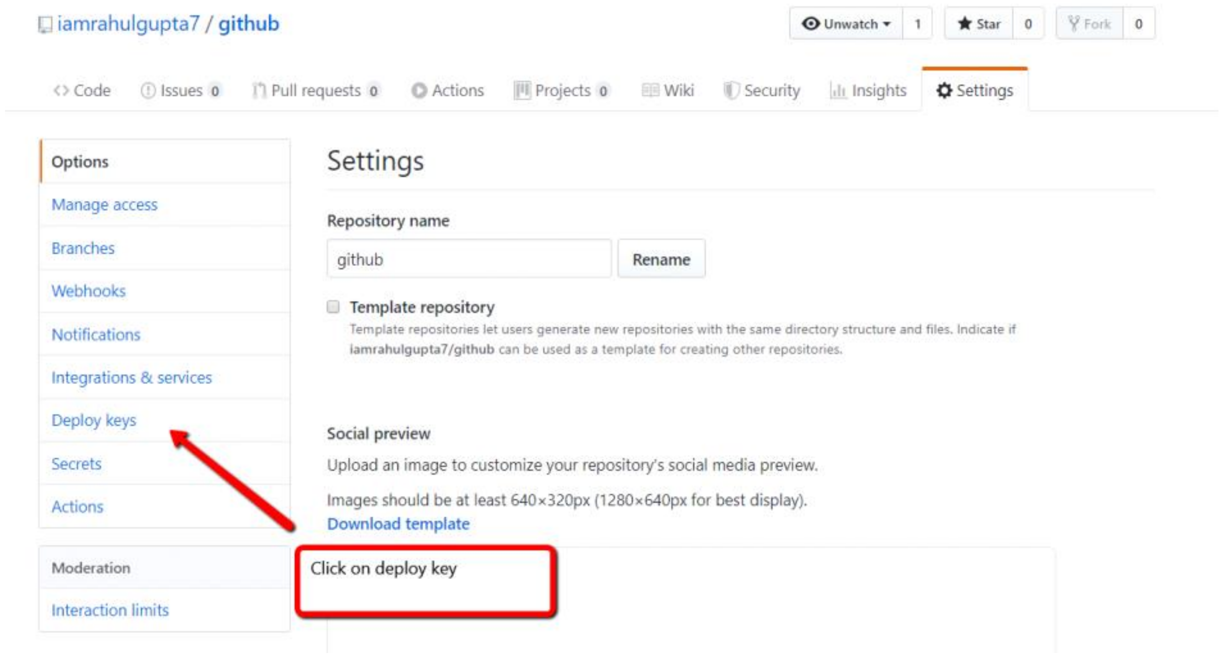


github is my testing repo

Click on setting

user1 First Project version1		Latest commit 4aee9cb 8 hours ago
file1.txt	First Project version1	8 hours ago

Help people interested in this repository understand your project by adding a README.



Settings

Repository name

github

Template repository

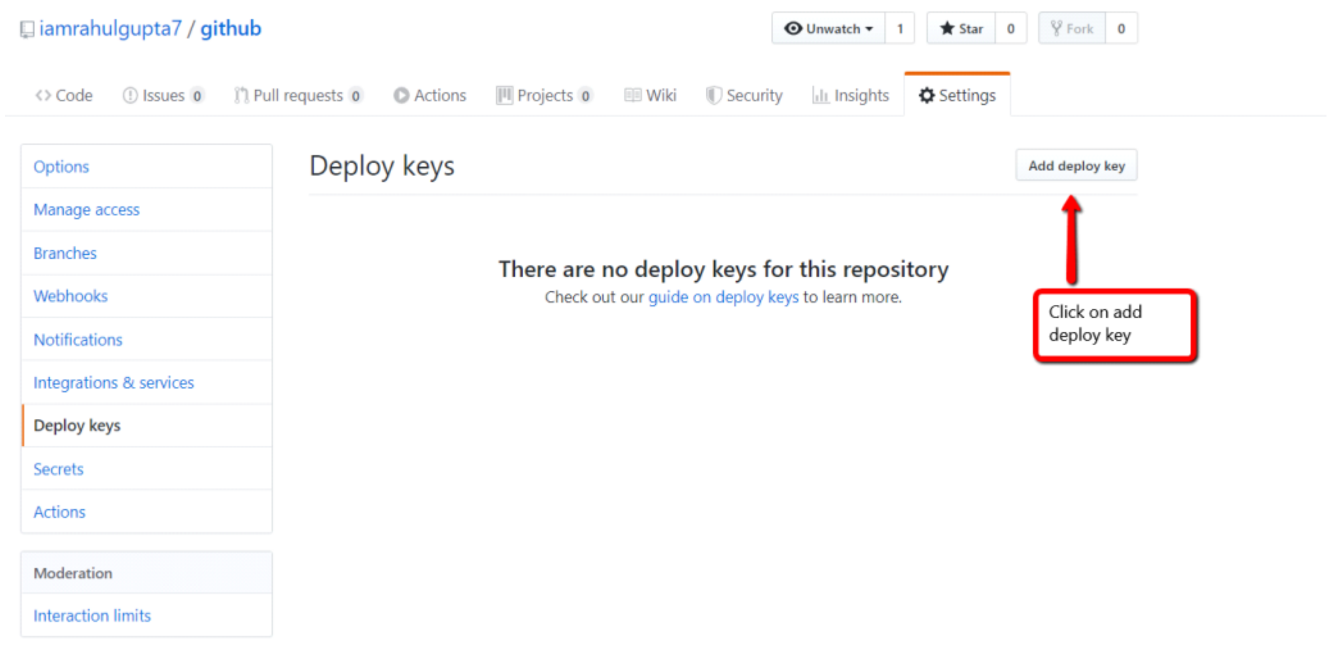
Template repositories let users generate new repositories with the same directory structure and files. Indicate if iamrahulgupta7/github can be used as a template for creating other repositories.

Social preview

Upload an image to customize your repository's social media preview.

Images should be at least 640×320px (1280×640px for best display).

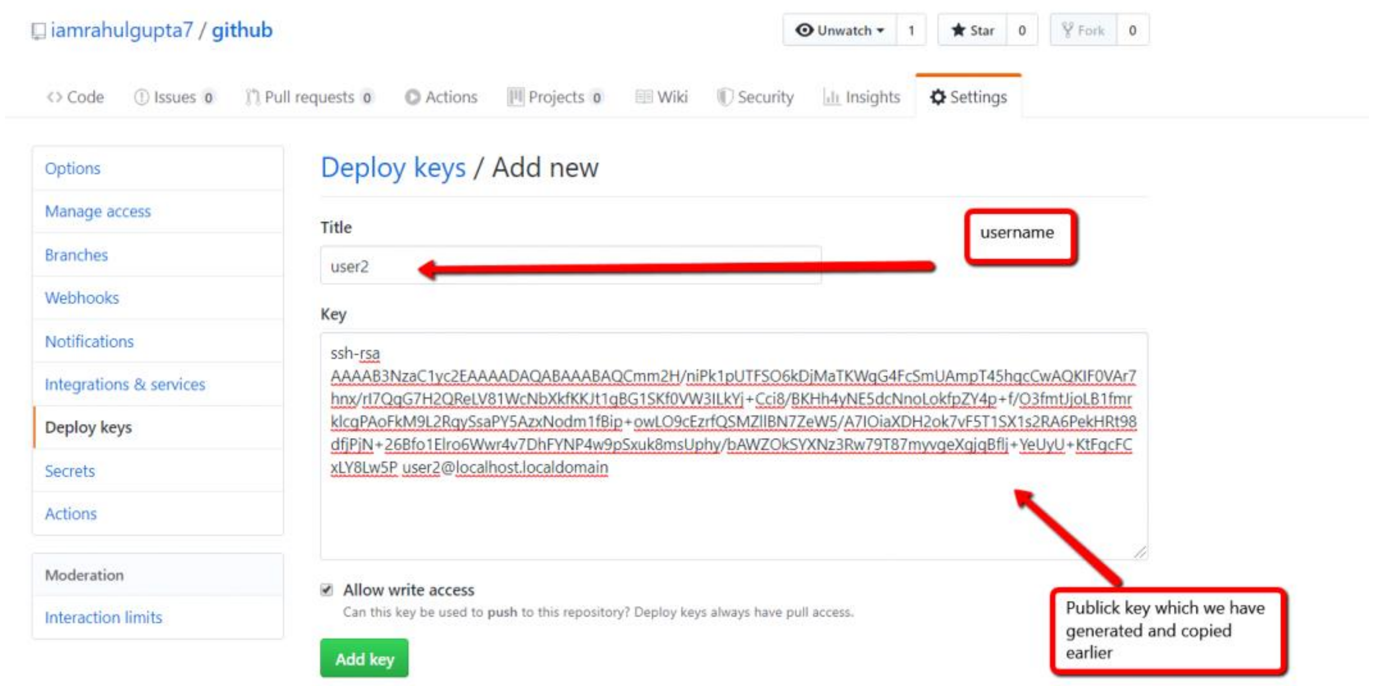
Click on deploy key



Deploy keys

There are no deploy keys for this repository

Check out our guide on deploy keys to learn more.



Deploy keys / Add new

Title: user2

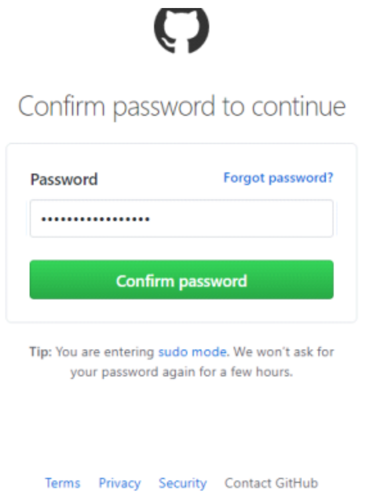
Key:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCmm2H/niPk1pUTFSO6kDjMaTKWqG4FcSmUAmpT45hqcCwAQKIF0VAr7
hnx/rI7QqG7H2QReLV81WcNbXkfKKJt1qBG1SKf0VW3ILkYj+Cci8/BKHh4yNE5dcNnoLokfpZY4p+f/O3fmtJjoLB1fmr
klcgPAoFkM9L2RqySsaPY5AzxNodm1fBip+owLO9cEzrfQSMZlIBN7ZeW5/A7IOiaXDH2ok7vF5T1SX1s2RA6PekHRt98
dfjPjN+26Bfo1Elro6Wwr4v7DhFYNP4w9pSxuk8msUphy/bAWZOkSYXNz3Rw79T87myvgeXqjqBflj+YeUyU+KtFgcFC
xLY8Lw5P user2@localhost.localdomain
```

[✓] Allow write access

Can this key be used to **push** to this repository? Deploy keys always have pull access.

Enter username and password of Git server

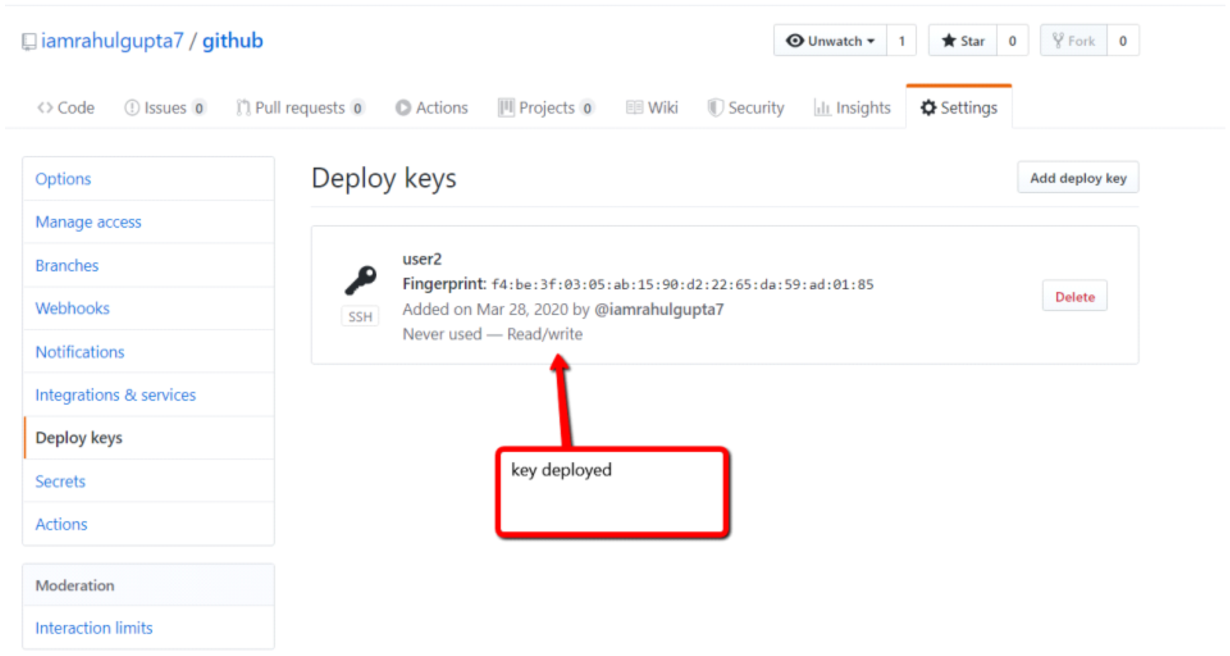

Confirm password to continue

Password

Forgot password?

••••••••••••••••••

Tip: You are entering sudo mode. We won't ask for your password again for a few hours.

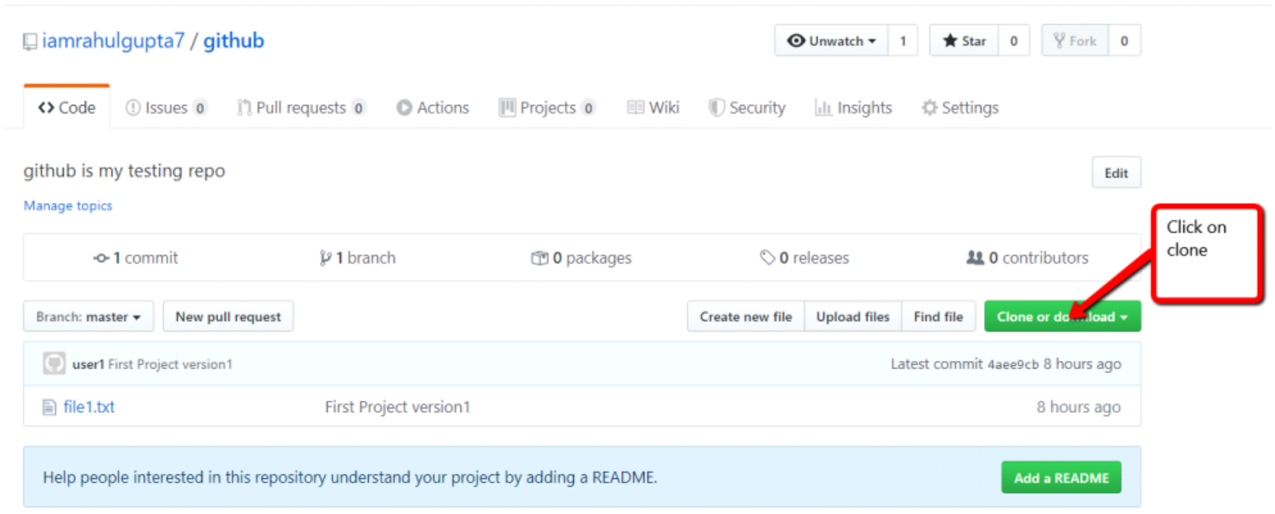

iamrahulgupta7 / github

Deploy keys

user2
Fingerprint: f4:be:3f:03:05:ab:15:90:d2:22:65:da:59:ad:01:85
Added on Mar 28, 2020 by @iamrahulgupta7
Never used — Read/write

key deployed

Step 3: For user 2 now we need ssh access link


iamrahulgupta7 / github

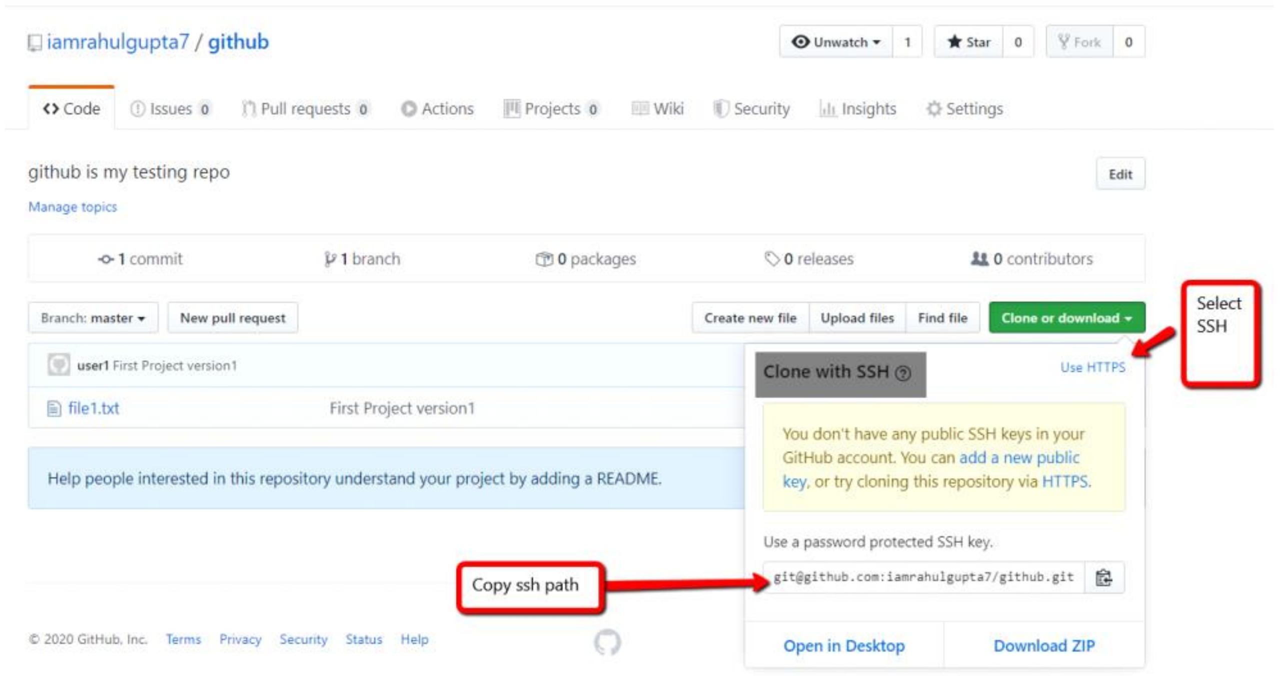
github is my testing repo

Click on clone

user1 First Project version1 — Latest commit 4aee9cb 8 hours ago

file1.txt — First Project version1 — 8 hours ago

Help people interested in this repository understand your project by adding a README.


iamrahulgupta7 / github
Unwatch 1
Star 0
Fork 0
Code
Issues 0
Pull requests 0
Actions
Projects 0
Wiki
Security
Insights
Settings
github is my testing repo
Edit
Manage topics
1 commit
1 branch
0 packages
0 releases
0 contributors
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
user1 First Project version1
file1.txt
First Project version1
Help people interested in this repository understand your project by adding a README.
Clone with SSH
Use HTTPS
Select SSH
You don't have any public SSH keys in your GitHub account. You can add a new public key, or try cloning this repository via HTTPS.
Use a password protected SSH key.
git@github.com:iamrahulgupta7/github.git
Copy ssh path
Open in Desktop
Download ZIP
© 2020 GitHub, Inc. Terms Privacy Security Status Help

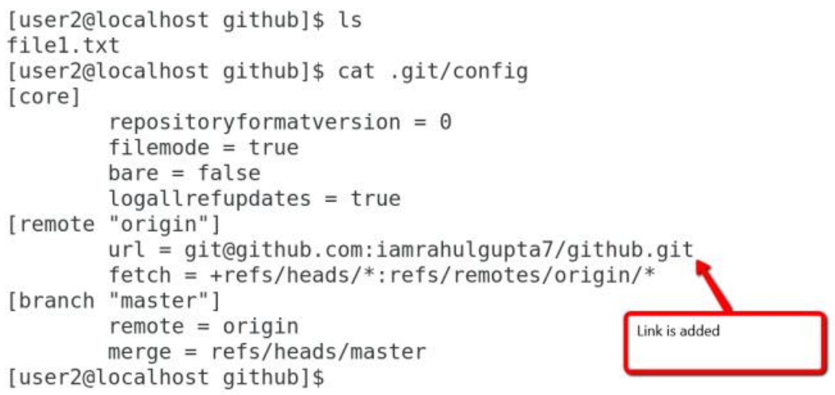
Step 4: From user2, run below command

```
[user2@localhost ~]$ git clone git@github.com:iamrahulgupta7/github.git
Cloning into 'github'...
The authenticity of host 'github.com (13.234.210.38)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXUpJWGl7E1IGOCspRomTxdCARLviKw6E5SY8.
RSA key fingerprint is MD5:16:27:ac:a5:76:28:2d:36:63:1b:56:4d:eb:df:a6:48.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,13.234.210.38' (RSA) to the list of known hosts.
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
[user2@localhost ~]$ 
```

host address we have copied from GIT server

- Now we have clone the repo 'github' to user2. now we can note it doesn't ask for username password which it asked In case of user1 because we are using key based ssh access not username password based.

```
[user2@localhost ~]$ ls
github
[user2@localhost ~]$ 
```

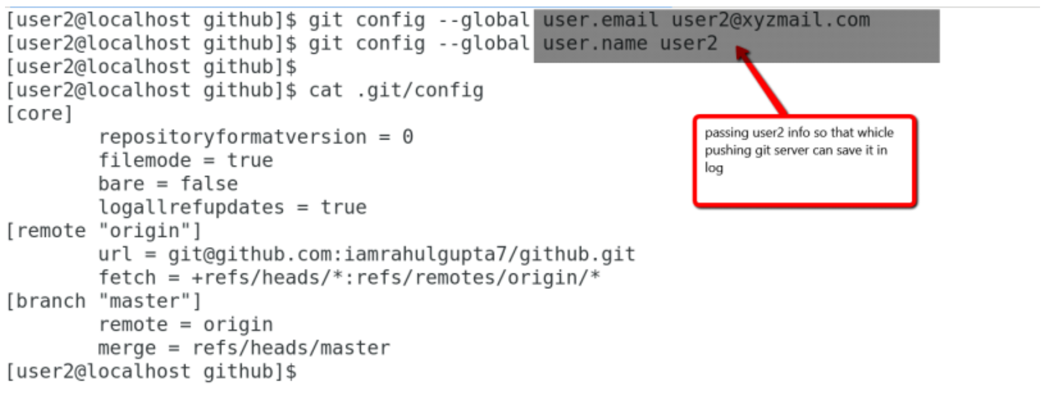


```
[user2@localhost github]$ ls
file1.txt
[user2@localhost github]$ cat .git/config
[core]
        repositoryformatversion = 0
        filemode = true
        bare = false
        logallrefupdates = true
[remote "origin"]
        url = git@github.com:iamrahulgupta7/github.git
        fetch = +refs/heads/*:refs/remotes/origin/*
[branch "master"]
        remote = origin
        merge = refs/heads/master
[user2@localhost github]$
```

Step 5:

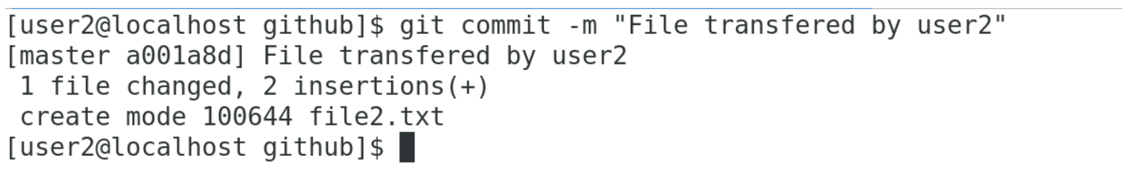
- Creating new txt file file2.txt from user2
- Add this file in staging/tracking
- Check status of this file

```
[user2@localhost github]$ ls
file1.txt
[user2@localhost github]$ vim file2.txt
[user2@localhost github]$ ls
file1.txt  file2.txt
[user2@localhost github]$ git add .
[user2@localhost github]$ git status -s
A  file2.txt
[user2@localhost github]$
```


passing user2 info so that whicle pushing git server can save it in log

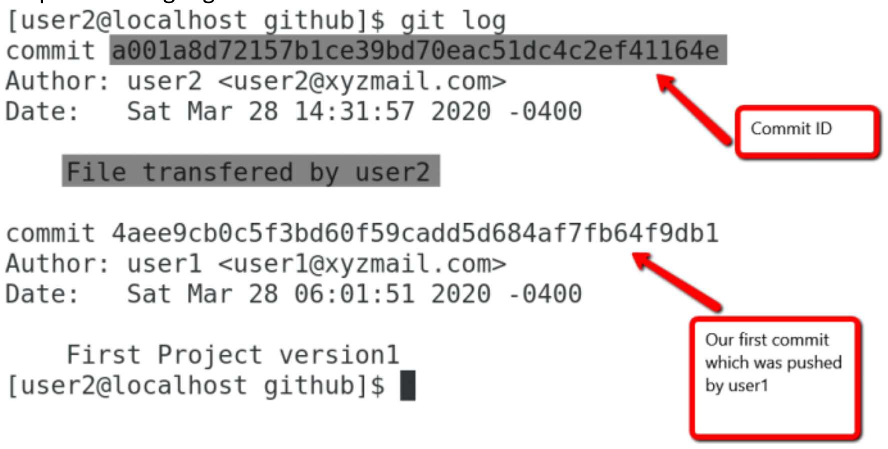
```
[user2@localhost github]$ git config --global user.email user2@xyzmail.com
[user2@localhost github]$ git config --global user.name user2
[user2@localhost github]$
[user2@localhost github]$ cat .git/config
[core]
        repositoryformatversion = 0
        filemode = true
        bare = false
        logallrefupdates = true
[remote "origin"]
        url = git@github.com:iamrahulgupta7/github.git
        fetch = +refs/heads/*:refs/remotes/origin/*
[branch "master"]
        remote = origin
        merge = refs/heads/master
[user2@localhost github]$
```

Step 6: Commit the changes



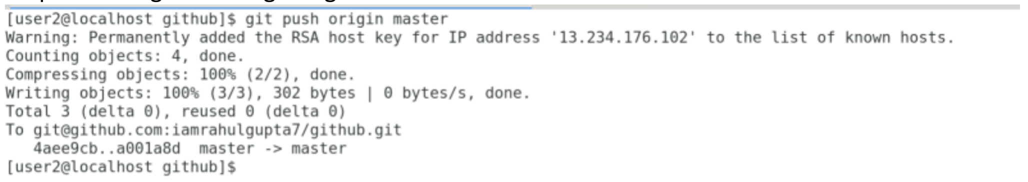
```
[user2@localhost github]$ git commit -m "File transfered by user2"
[master a001a8d] File transfered by user2
 1 file changed, 2 insertions(+)
 create mode 100644 file2.txt
[user2@localhost github]$ 
```

Step8: Checking logs of commit



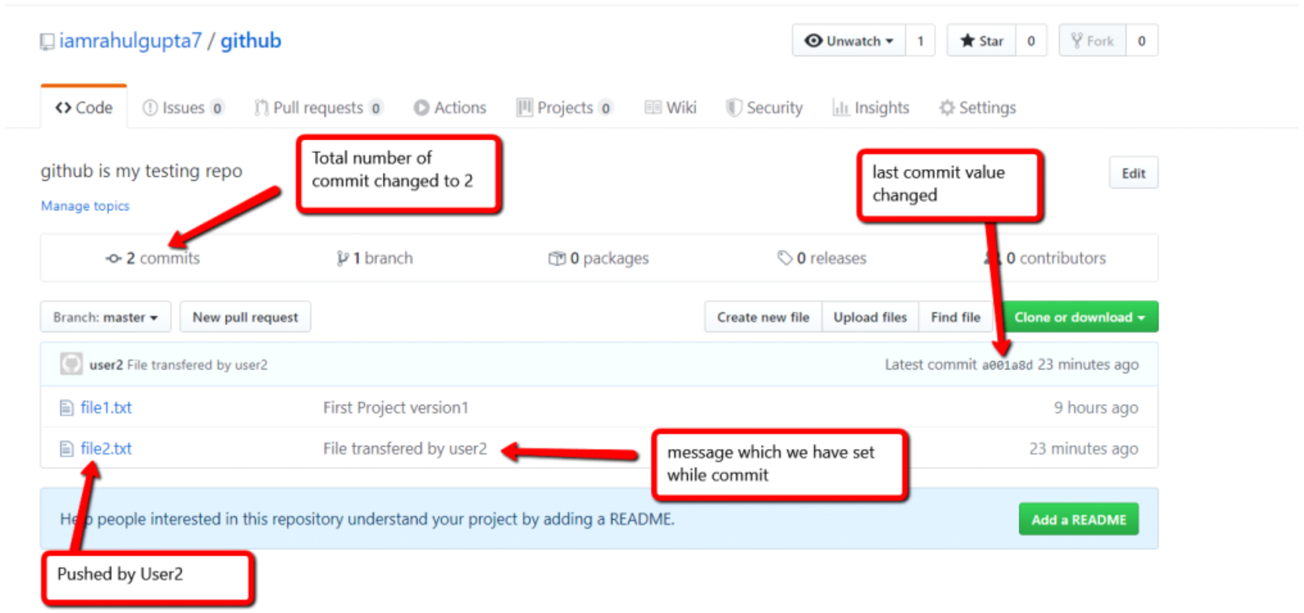
```
[user2@localhost github]$ git log
commit a001a8d72157b1ce39bd70eac51dc4c2ef41164e
Author: user2 <user2@xyzmail.com>
Date:   Sat Mar 28 14:31:57 2020 -0400

    File transfered by user2

commit 4aee9cb0c5f3bd60f59cadd5d684af7fb64f9db1
Author: user1 <user1@xyzmail.com>
Date:   Sat Mar 28 06:01:51 2020 -0400

    First Project version1
[user2@localhost github]$ 
```

Step9: Pushing the change to git server



```
[user2@localhost github]$ git push origin master
Warning: Permanently added the RSA host key for IP address '13.234.176.102' to the list of known hosts.
Counting objects: 4, done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 302 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To git@github.com:iamrahulgupta7/github.git
   4aee9cb..a001a8d  master -> master
[user2@localhost github]$
```

Step 10: checking on GIT server if changes has been pushed



iamrahulgupta7 / github

github is my testing repo

2 commits · 1 branch · 0 packages · 0 releases · 0 contributors

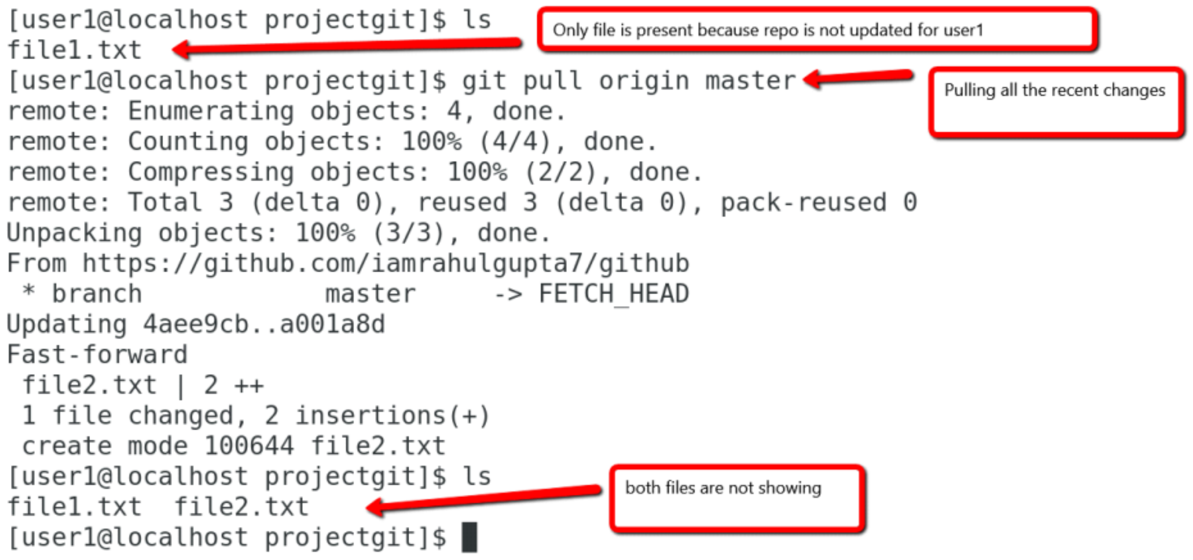
Branch: master

user2 File transfered by user2 — Latest commit a001a8d 23 minutes ago

```
file1.txt     First Project version1      9 hours ago
file2.txt     File transfered by user2    23 minutes ago
```

Help people interested in this repository understand your project by adding a README.

Step 11: For user 1> checking all the update in repo



```
[user1@localhost projectgit]$ ls
file1.txt
[user1@localhost projectgit]$ git pull origin master
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/iamrahulgupta7/github
 * branch            master     -> FETCH_HEAD
Updating 4aee9cb..a001a8d
Fast-forward
 file2.txt | 2 ++
 1 file changed, 2 insertions(+)
 create mode 100644 file2.txt
[user1@localhost projectgit]$ ls
file1.txt  file2.txt
[user1@localhost projectgit]$ 
```

Only file is present because repo is not updated for user1
Pulling all the recent changes
both files are not showing

##Up next we will check push-pull error