

COURSES

Login

HIRE WITH US

How to install, configure and use GIT on ubuntu?

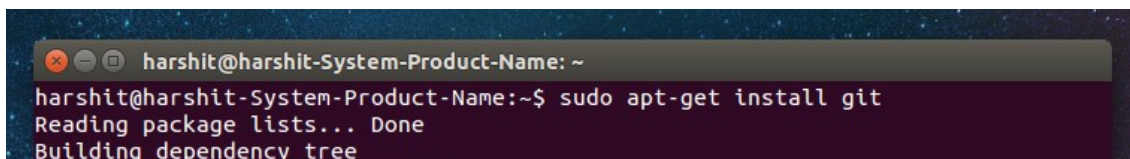
In the [previous article](#) we learnt about what is Git all about. Now we'll learn a more about git and its repository in more detail

Git is a **distributed revision control system** with an emphasis on speed, data integrity, and support for distributed, non-linear workflows. Every Git working directory is a full-fledged repository with complete history and full version-tracking capabilities, independent of network access or a central server.

GitHub is a **Web-based Git repository hosting service**, which offers all of the distributed revision control. Unlike Git, which is strictly a command-line tool, GitHub provides a Web-based graphical interface. GitHub is used for 'version control'. This means that GitHub is used for software development projects when more than a person(or a group) is working on it. What GitHub does is that it creates a cloud-based centralized repository for everyone working in the group and allows everyone working on the project to update the information. One can also use BitBucket, if not GitHub.

INSTALLING GIT:

Step 1: Open the Terminal and type `sudo apt-get install git`

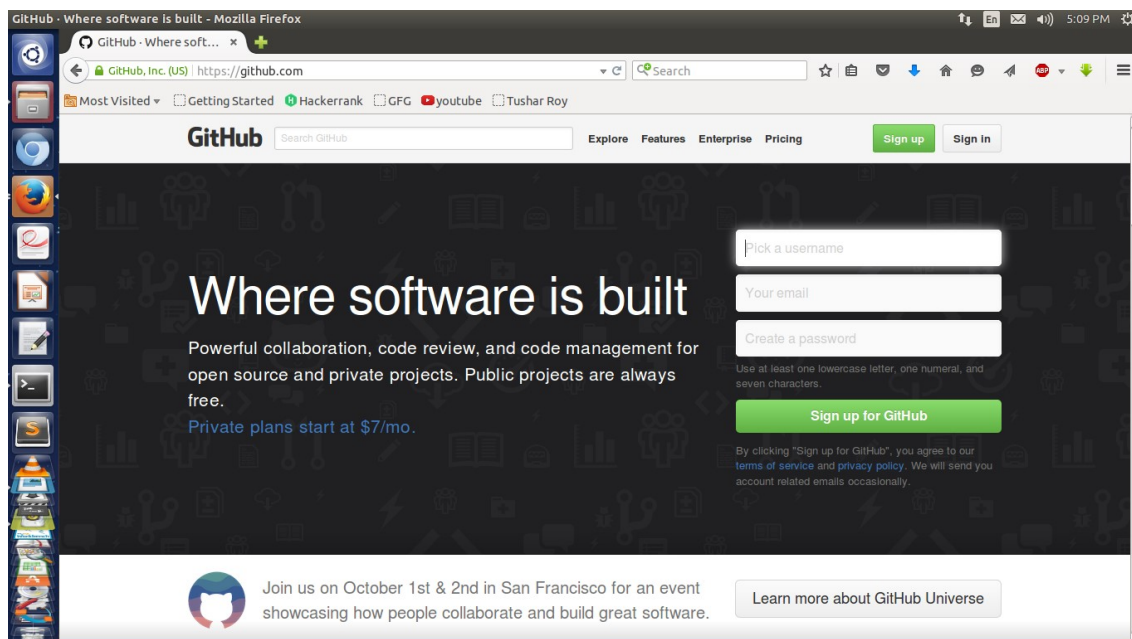


```
harshit@harshit-System-Product-Name: ~  
harshit@harshit-System-Product-Name:~$ sudo apt-get install git  
Reading package lists... Done  
Building dependency tree
```

```
Reading state information... Done
Suggested packages:
  git-daemon-run git-daemon-sysvinit git-doc git-el git-email git-gui gitk
  gitweb git-arch git-bzr git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git
0 upgraded, 1 newly installed, 0 to remove and 23 not upgraded.
Need to get 0 B/2,511 kB of archives.
After this operation, 20.3 MB of additional disk space will be used.
Selecting previously unselected package git.
(Reading database ... 274644 files and directories currently installed.)
Preparing to unpack .../git_1%3a1.9.1-1ubuntu0.1_i386.deb ...
Unpacking git (1:1.9.1-1ubuntu0.1) ...
Setting up git (1:1.9.1-1ubuntu0.1) ...
harshit@harshit-System-Product-Name:~$
```

Step 2: Goto www.github.com and sign into your account. If you're a new user, you can simply sign-up. (You can also use www.bitbucket.org as an alternative, but we will use github here).

You'll have a username from here. Let us say that it's [your_username](#)



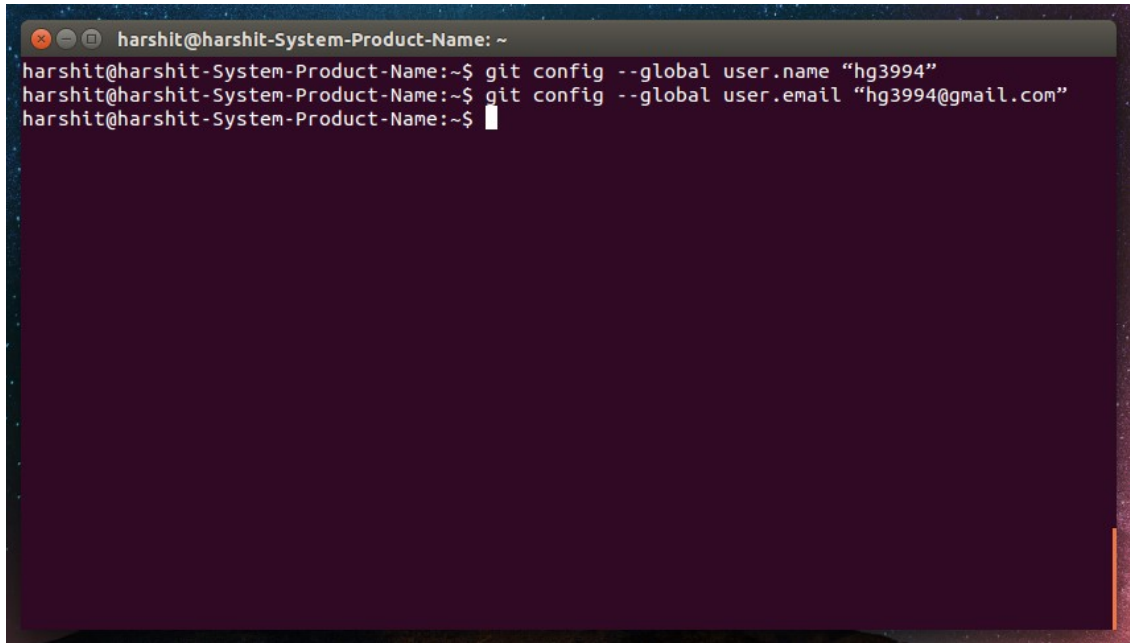
CONFIGURING GIT:

Step 1: Go back to the terminal and type this to configure git

`git config --global user.name "your_username"`

Step 2: Now type this to link your email too.

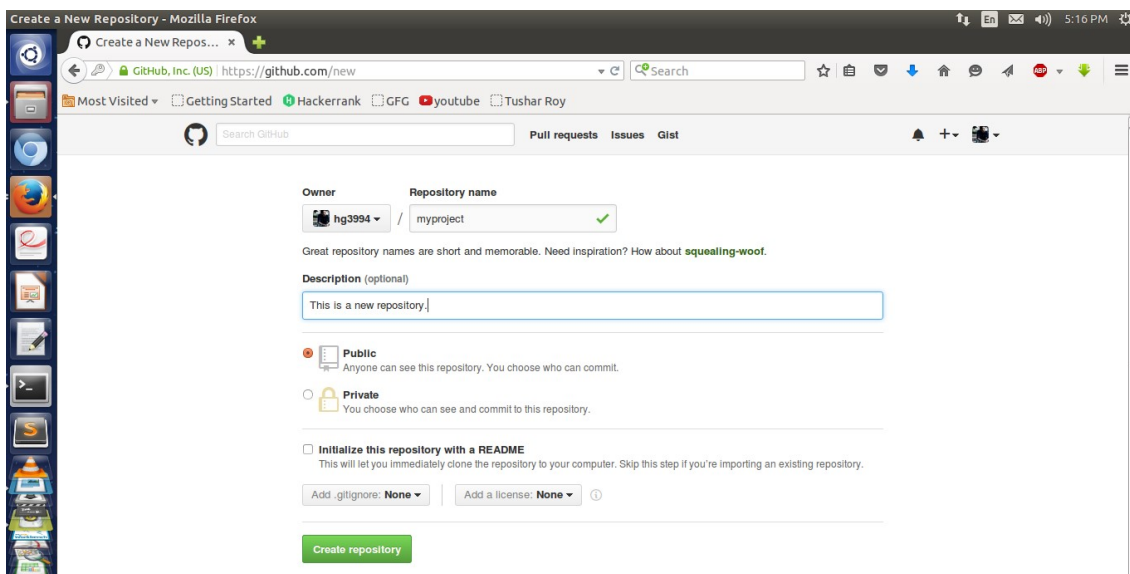
`git config --global user.email "your_emailid"`

A terminal window with a dark purple background and a starry space theme. The prompt is 'harshit@harshit-System-Product-Name: ~'. Two commands are entered: 'git config --global user.name "hg3994"' and 'git config --global user.email "hg3994@gmail.com"'. The cursor is at the end of the second command.

```
harshit@harshit-System-Product-Name: ~  
harshit@harshit-System-Product-Name:~$ git config --global user.name "hg3994"  
harshit@harshit-System-Product-Name:~$ git config --global user.email "hg3994@gmail.com"  
harshit@harshit-System-Product-Name:~$
```

USING GIT:

Step 1: Go to your github account and create a repository with a name(lets say name of your project). We are creating a repository with the name myproject





Step 2: Make a folder with the name of your project and change your current directory to that directory.

```
mkdir myproject
```

```
cd myproject
```

```
harshit@harshit-System-Product-Name: ~/myproject
harshit@harshit-System-Product-Name:~$ git config --global user.name "hg3994"
harshit@harshit-System-Product-Name:~$ git config --global user.email "hg3994@gmail.com"
harshit@harshit-System-Product-Name:~$ 
harshit@harshit-System-Product-Name:~$ mkdir myproject
harshit@harshit-System-Product-Name:~$ cd myproject/
harshit@harshit-System-Product-Name:~/myproject$
```

Step 3: Now we want to initiate Git for this folder

```
git init
```

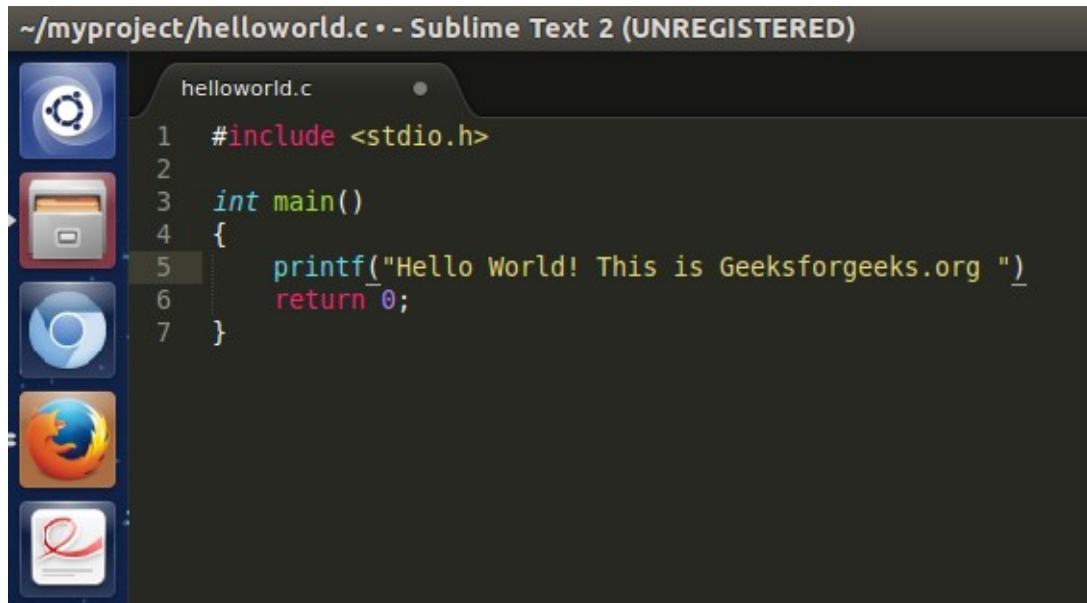
```
harshit@harshit-System-Product-Name: ~/myproject
harshit@harshit-System-Product-Name:~/myproject$ git init
Initialized empty Git repository in /home/harshit/myproject/.git/
harshit@harshit-System-Product-Name:~/myproject$
```

Step 4: Now we will set up the remote, which tells git where the repository is located.

```
git remote add origin https://github.com/your_username/myproject.git
```

```
harshit@harshit-System-Product-Name: ~/myproject
harshit@harshit-System-Product-Name:~/myproject$ git remote add origin https://github.com/hg3994/myproject.git
harshit@harshit-System-Product-Name:~/myproject$
```

We have now configured and installed git and, created and configured a repository. Lets say we have a simple file in the myproject folder helloworld.c and we want it to share it with a friend who is working on the same project.



```
~/myproject/helloworld.c • - Sublime Text 2 (UNREGISTERED)

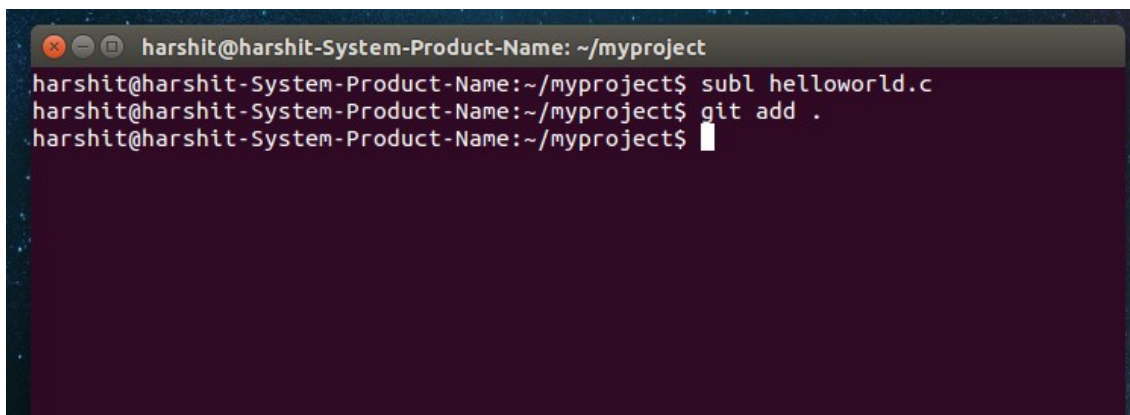
helloworld.c
1  #include <stdio.h>
2
3  int main()
4  {
5      printf("Hello World! This is Geeksforgeeks.org ")
6      return 0;
7  }
```

Step 5: To add this file we will type

`git add helloworld.c`

Or if we have a lot of files to be transferred from the folder to our git account, then we can use the command.

`git add .`

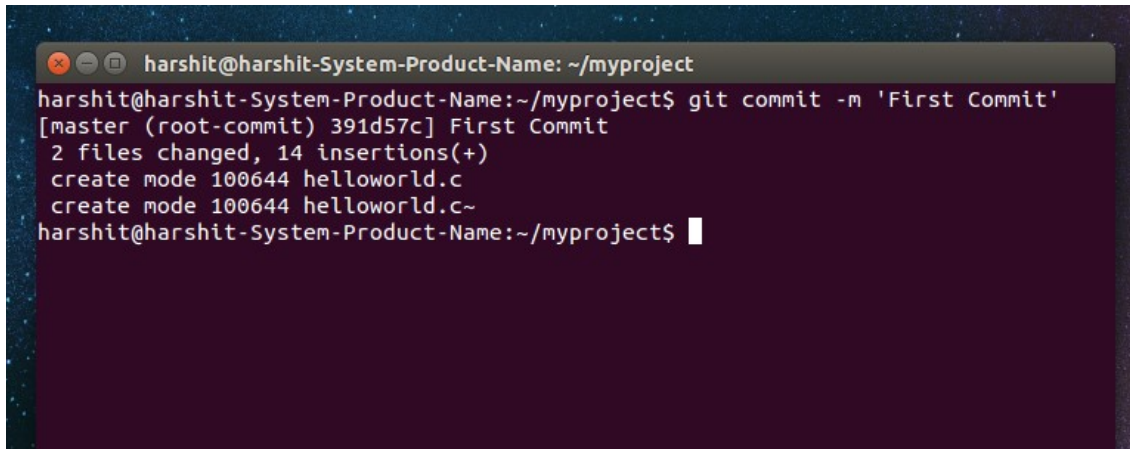


```
harshit@harshit-System-Product-Name: ~/myproject
harshit@harshit-System-Product-Name:~/myproject$ subl helloworld.c
harshit@harshit-System-Product-Name:~/myproject$ git add .
harshit@harshit-System-Product-Name:~/myproject$
```

This would transfer the file(s) in the list which we will later commit.

Step 6: Next, when we are finished adding the files, then we will have to commit adding.

`git commit -m 'your_message'`



```
harshit@harshit-System-Product-Name: ~/myproject
harshit@harshit-System-Product-Name:~/myproject$ git commit -m 'First Commit'
[master (root-commit) 391d57c] First Commit
2 files changed, 14 insertions(+)
create mode 100644 helloworld.c
create mode 100644 helloworld.c~
harshit@harshit-System-Product-Name:~/myproject$
```

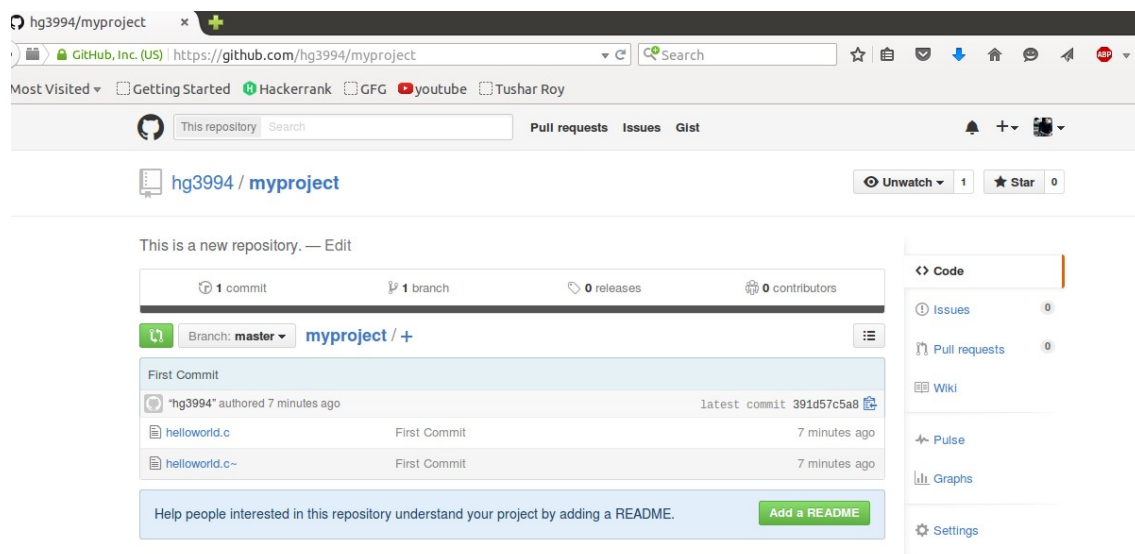
Step 7: Next, we need to push the commit that we just made on to the repository at github

`git push origin master`

It would automatically ask you for your username and password for github. After entering the details, go to github and refresh. The files would get added there.

Username for 'https://github.com': your_username

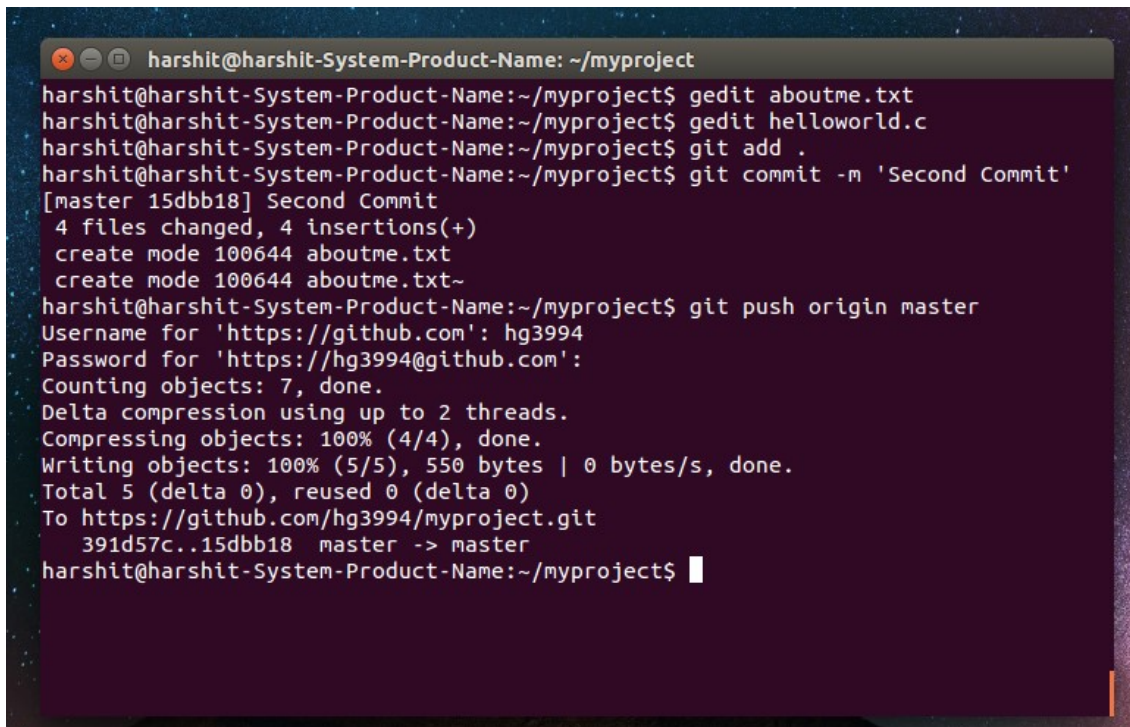
Password for 'https://your_username@github.com': *****



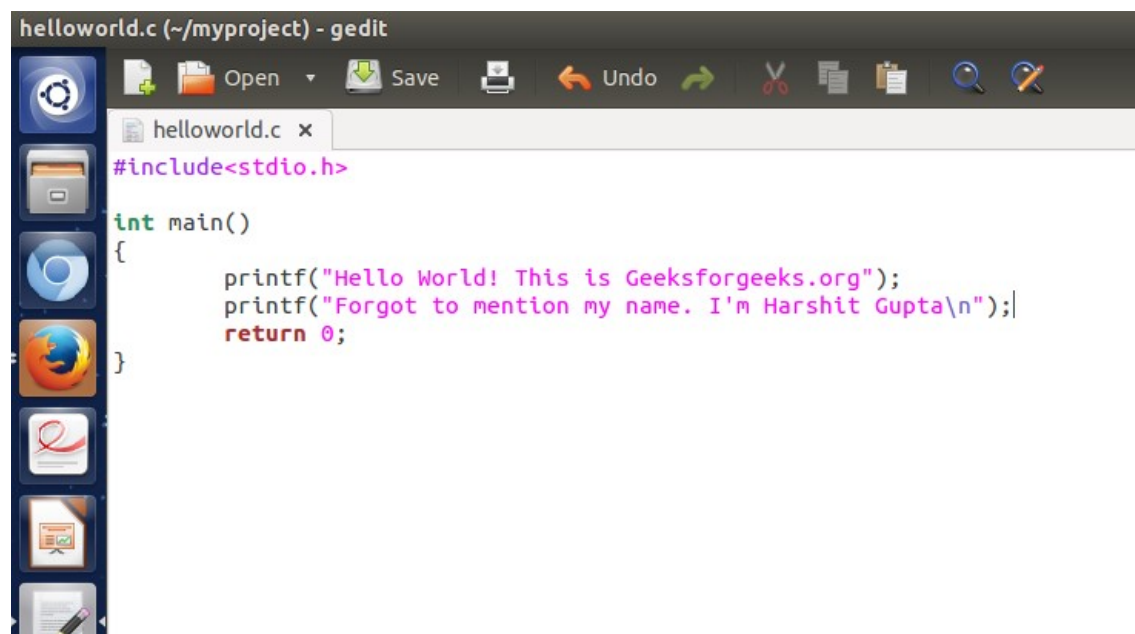
Step 8: We have successfully transferred a file on your github account. Now lets add one more file aboutme.txt and edit our file helloworld.c . Following the same procedure we will first add the files, commit and then push them to the github account.

git add .

git commit -m 'your_message' git push origin master

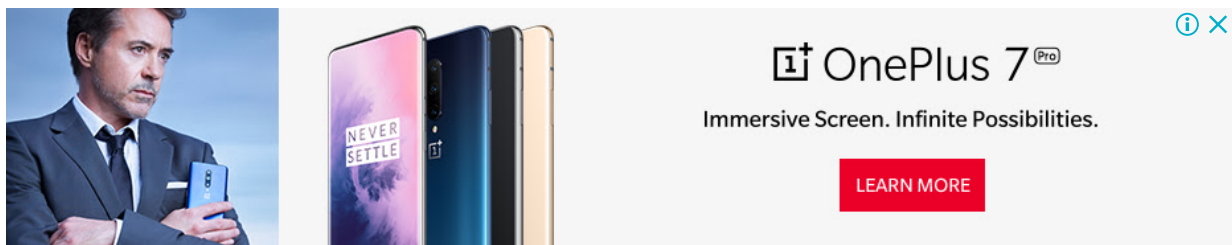
A terminal window with a dark purple background and white text. The window title is 'harshit@harshit-System-Product-Name: ~/myproject'. The user enters several git commands: 'gedit aboutme.txt', 'gedit helloworld.c', 'git add .', 'git commit -m 'Second Commit'', and 'git push origin master'. The output shows the creation of two files, a commit message 'Second Commit', and the successful push to the 'master' branch on GitHub. The push output includes details about object compression and writing.

```
harshit@harshit-System-Product-Name: ~/myproject
harshit@harshit-System-Product-Name:~/myproject$ gedit aboutme.txt
harshit@harshit-System-Product-Name:~/myproject$ gedit helloworld.c
harshit@harshit-System-Product-Name:~/myproject$ git add .
harshit@harshit-System-Product-Name:~/myproject$ git commit -m 'Second Commit'
[master 15dbb18] Second Commit
 4 files changed, 4 insertions(+)
 create mode 100644 aboutme.txt
 create mode 100644 aboutme.txt~
harshit@harshit-System-Product-Name:~/myproject$ git push origin master
Username for 'https://github.com': hg3994
Password for 'https://hg3994@github.com':
Counting objects: 7, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 550 bytes | 0 bytes/s, done.
Total 5 (delta 0), reused 0 (delta 0)
To https://github.com/hg3994/myproject.git
 391d57c..15dbb18 master -> master
harshit@harshit-System-Product-Name:~/myproject$
```

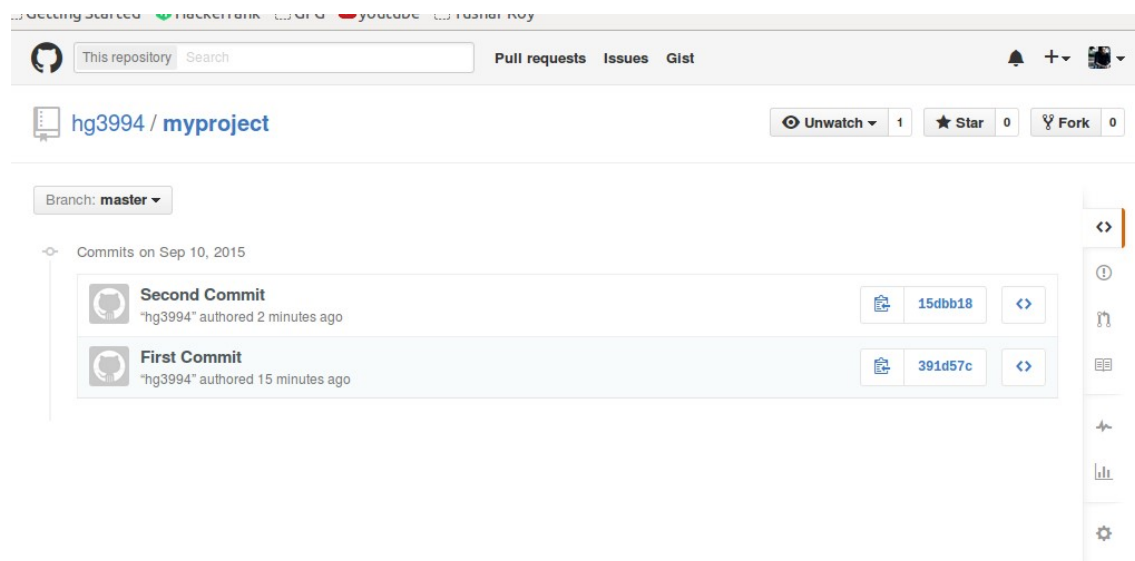
A screenshot of the Gedit text editor. The title bar says 'helloworld.c (~/.myproject) - gedit'. The menu bar includes 'Open', 'Save', 'Undo', and others. The code is written in C and includes a printf statement that prints a message about Geeksforgeeks.org and the user's name, Harshit Gupta.

```
helloworld.c (~/.myproject) - gedit
#include<stdio.h>

int main()
{
    printf("Hello World! This is Geeksforgeeks.org");
    printf("Forgot to mention my name. I'm Harshit Gupta\n");
    return 0;
}
```



Step 9: When we would go to our github account, we would see the entire hierarchy of the modification of the file. Here, we would see the changes we made to the helloworld.c file in the respective commits.

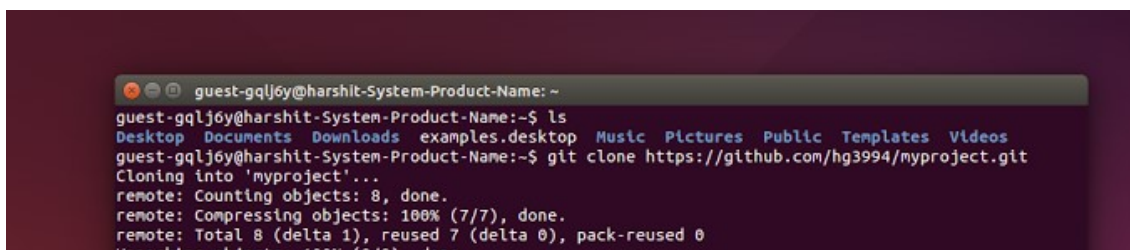


Now, Let's say one of the co-worker of the project needs to work on helloworld.c . After making some changes, he wants to update the file on github.

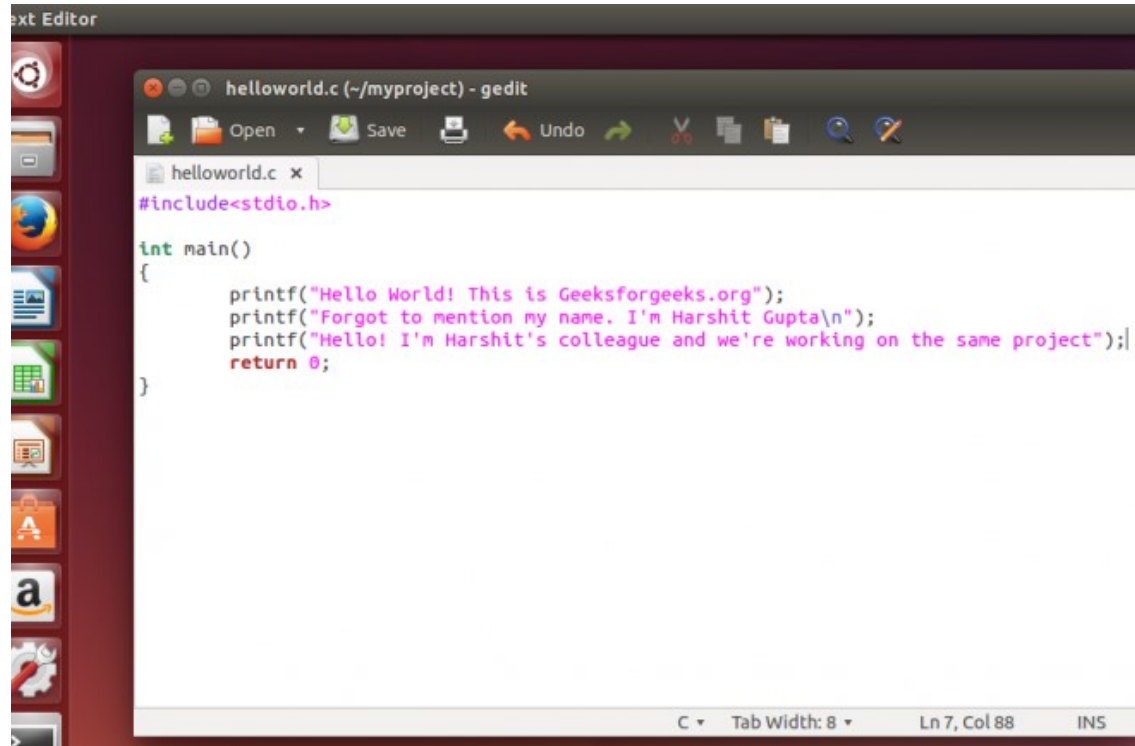
Step 10: First he would have to download the whole repository in which the file helloworld.c is present into his system.

`git clone https://github.com/your_username/myproject.git`

A folder named myproject gets downloaded with all the files in it. The necessary changes are made and then the file is similarly added, committed and pushed similarly as above.




```
Unpacking objects: 100% (3/3), done.
Checking connectivity... done.
guest-gqlj6y@harshit-System-Product-Name:~$ ls
Desktop  Downloads  Music      Pictures  Templates
Documents examples.desktop myproject  Public    Videos
guest-gqlj6y@harshit-System-Product-Name:~$
```



The screenshot shows a text editor window titled "helloworld.c (~myproject) - gedit". The editor contains the following C code:

```
#include<stdio.h>

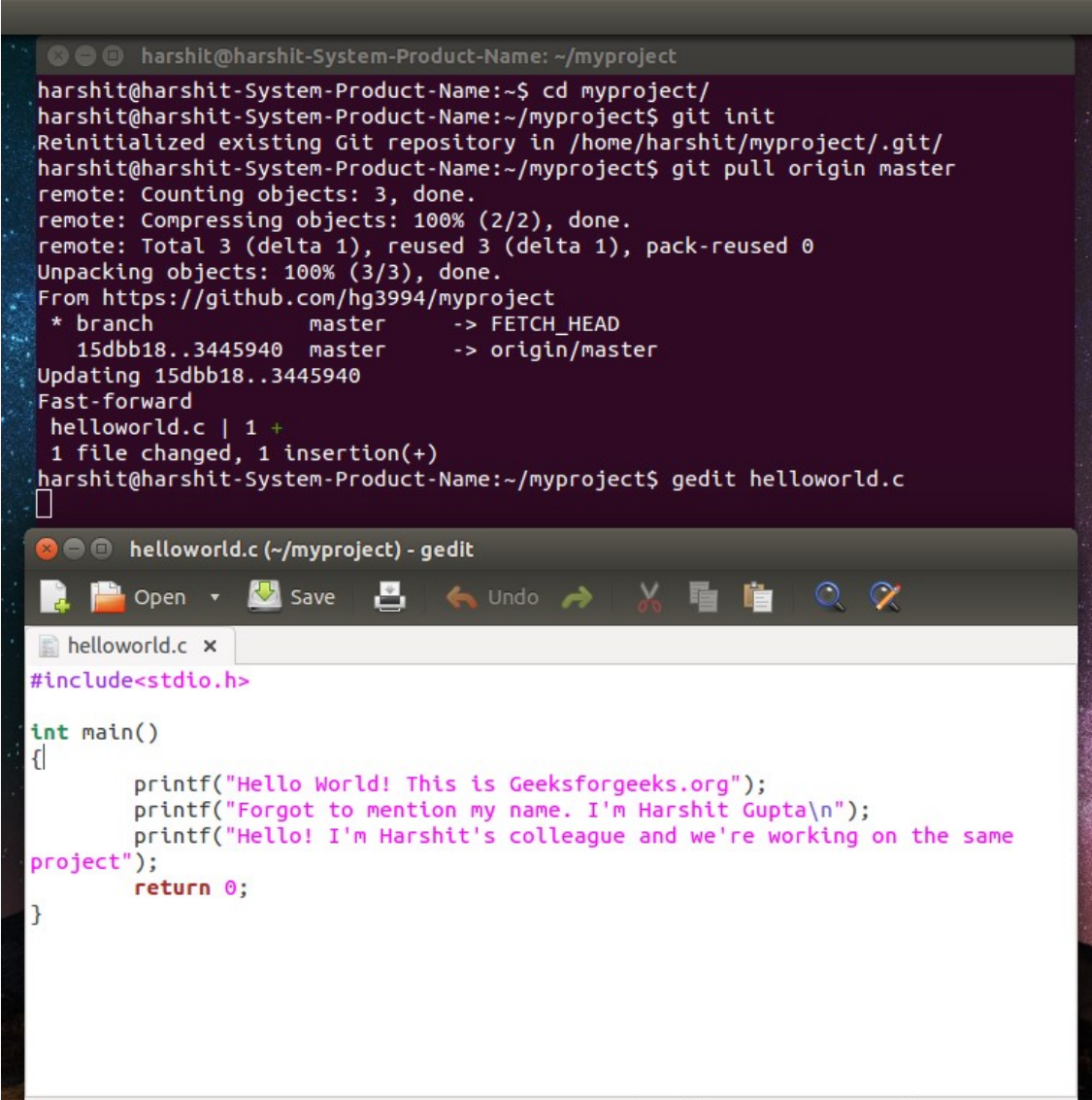
int main()
{
    printf("Hello World! This is Geeksforgeeks.org");
    printf("Forgot to mention my name. I'm Harshit Gupta\n");
    printf("Hello! I'm Harshit's colleague and we're working on the same project");
    return 0;
}
```

The status bar at the bottom indicates "C", "Tab Width: 8", "Ln 7, Col 88", and "INS".

```
guest-gqlj6y@harshit-System-Product-Name: ~/myproject
guest-gqlj6y@harshit-System-Product-Name:~/myproject$ git config --global user.name "hg3994"
guest-gqlj6y@harshit-System-Product-Name:~/myproject$ git config --global user.email "hg3994@gmail.com"
guest-gqlj6y@harshit-System-Product-Name:~/myproject$ subl helloworld.c
guest-gqlj6y@harshit-System-Product-Name:~/myproject$ git add .
guest-gqlj6y@harshit-System-Product-Name:~/myproject$ git commit -m 'New User'
[master 3445940] New User
 1 file changed, 1 insertion(+)
guest-gqlj6y@harshit-System-Product-Name:~/myproject$ git push origin master
Username for 'https://github.com': hg3994
Password for 'https://hg3994@github.com':
Counting objects: 5, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 419 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To https://github.com/hg3994/myproject.git
 15dbb18..3445940 master -> master
guest-gqlj6y@harshit-System-Product-Name:~/myproject$
```

Step 11: If the first user wish to see the changes, then he can see it by typing:

`git pull origin master`



```
harshit@harshit-System-Product-Name: ~/myproject
harshit@harshit-System-Product-Name:~$ cd myproject/
harshit@harshit-System-Product-Name:~/myproject$ git init
Reinitialized existing Git repository in /home/harshit/myproject/.git/
harshit@harshit-System-Product-Name:~/myproject$ git pull origin master
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 3 (delta 1), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/hg3994/myproject
* branch          master       -> FETCH_HEAD
   15dbb18..3445940 master       -> origin/master
Updating 15dbb18..3445940
Fast-forward
 helloworld.c | 1 +
 1 file changed, 1 insertion(+)
harshit@harshit-System-Product-Name:~/myproject$ gedit helloworld.c

helloworld.c (~/myproject) - gedit
#include<stdio.h>

int main()
{
    printf("Hello World! This is Geeksforgeeks.org");
    printf("Forgot to mention my name. I'm Harshit Gupta\n");
    printf("Hello! I'm Harshit's colleague and we're working on the same
project");
    return 0;
}
```

Article By **Harshit Gupta**:

Kolkata based Harshit Gupta is an active blogger having keen interest in writing about current affairs, technical Blogs, stories, and personal life experiences. Besides passionate about writing, he also loves coding and dancing. Currently studying at IIST, he is an active blog contributor at geeksforgeeks. You can reach him at <https://in.linkedin.com/pub/harshit-gupta/102/b71/605>



If you also wish to showcase your blog here, please see [GBlog](#) for guest blog writing on GeeksforGeeks.

Recommended Posts:

[How to install Apache server in Ubuntu ?](#)

[How to use GIT in Ubuntu ? \(Part -2\)](#)

[How to Install and Set Up SAS University Edition](#)

[Creating and Configuring Wi-Fi Hotspot in Ubuntu](#)

[Guide to Install MongoDB with Python | Windows](#)

[LAMP installation and important PHP configurations on Ubuntu](#)

[How to change login screen background in Ubuntu](#)

[Creating custom domain name instead of localhost in Ubuntu](#)

[Relationship between Data Mining and Machine Learning](#)

[Difference between a Data Analyst and a Data Scientist](#)

[Overturning Changes In Git](#)

[Git - Changing History](#)

[Examining Git](#)

[Git- Debugging](#)

Article Tags : [GBlog](#) [GitHub](#)



2

☐ To-do ☐ Done

2.5

Based on 2 vote(s)

Feedback/ Suggest Improvement

Add Notes

Improve Article

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments

Geeks for Geeks
A computer science portal for geeks

5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305
feedback@geeksforgeeks.org

COMPANY

About Us
Careers
Privacy Policy
Contact Us

PRACTICE

Courses
Company-wise
Topic-wise
How to begin?

LEARN

Algorithms
Data Structures
Languages
CS Subjects
Video Tutorials

CONTRIBUTE

Write an Article
Write Interview Experience
Internships
Videos

@geeksforgeeks, Some rights reserved