Module 1: Getting Started with Git

Demo 3 : Demo on Basic Setup Commands

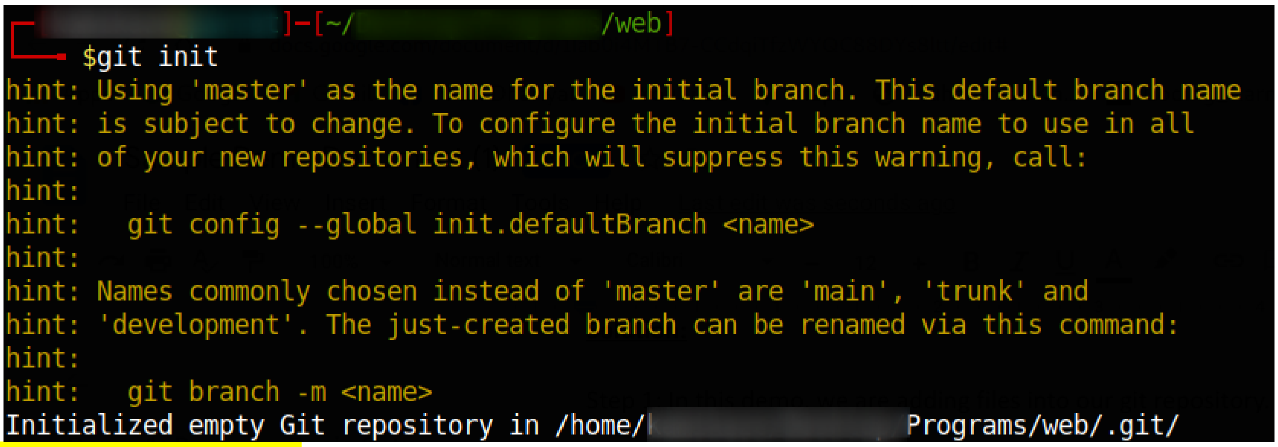
<u>Problem Statement:</u>

How we use basic git commands in our git bash command line interface.

<u>Solution:</u>

Step 1:

In this demo, we are adding files into our git repository. Let us first create a new repository.

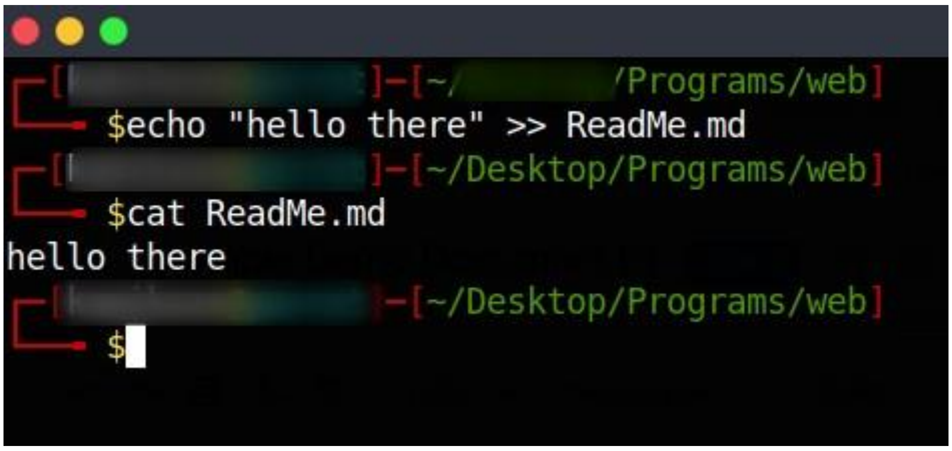


```
┌[redacted]]─[~/[redacted]/web]
└──╼ $git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/[redacted]Programs/web/.git/
```

command Used: git init

Step 2: Now let us make a new file.

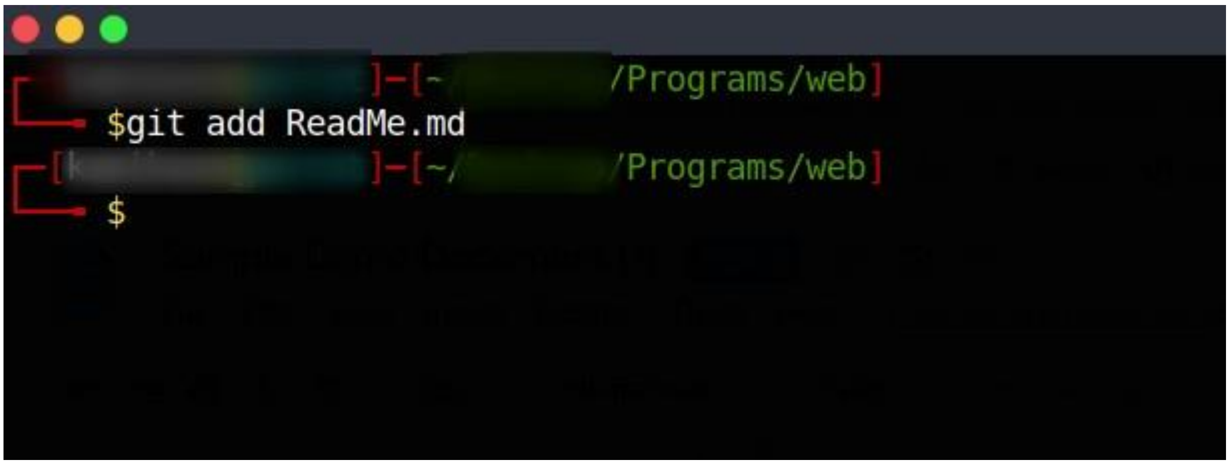
command used: echo “hello there” >> ReadMe.md



```
┌─[[redacted]]─[~/[redacted]/Programs/web]
└──╼ $echo "hello there" >> ReadMe.md
┌─[[redacted]]─[~/Desktop/Programs/web]
└──╼ $cat ReadMe.md
hello there
┌─[[redacted]─[~/Desktop/Programs/web]
└──╼ $
```

Step 3: Now add this file into our git repository.

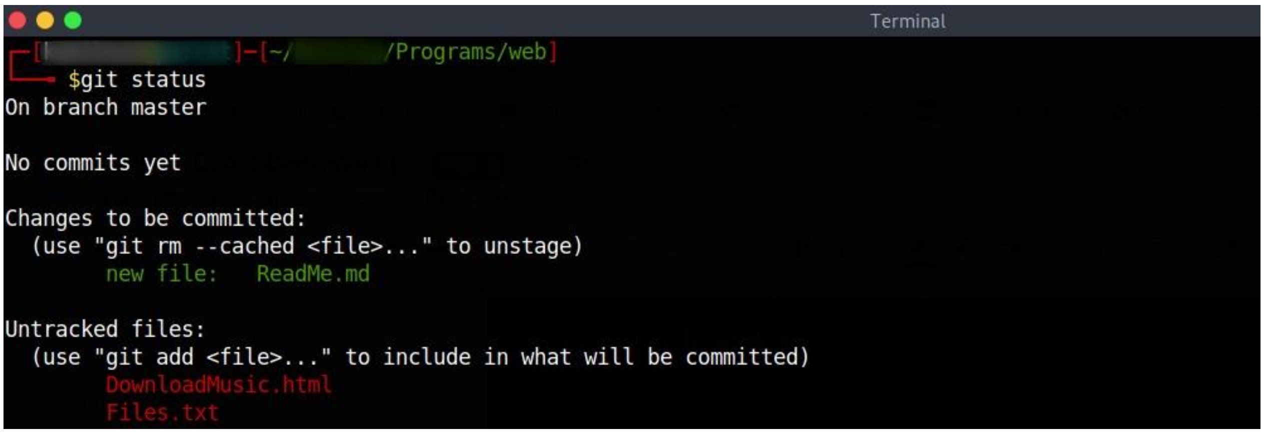
Command Used: git add filename



```
┌─[redacted]]─[~[redacted]/Programs/web]
└──╼ $git add ReadMe.md
┌─[[redacted]]─[~/[redacted]/Programs/web]
└──╼ $
```

Step 4: Let us check the status of our git

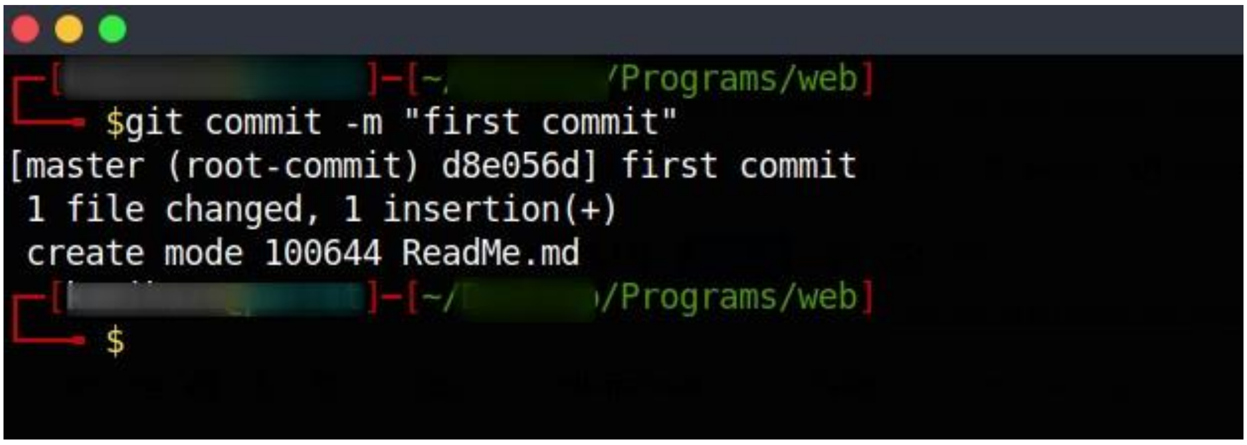
repository. Command Used: git status



```
┌─[[redacted]]─[~/[redacted]/Programs/web]
└──╼ $git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   ReadMe.md

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

Step 5: Let us commit to our repository.

Command used: git commit -m “first commit”

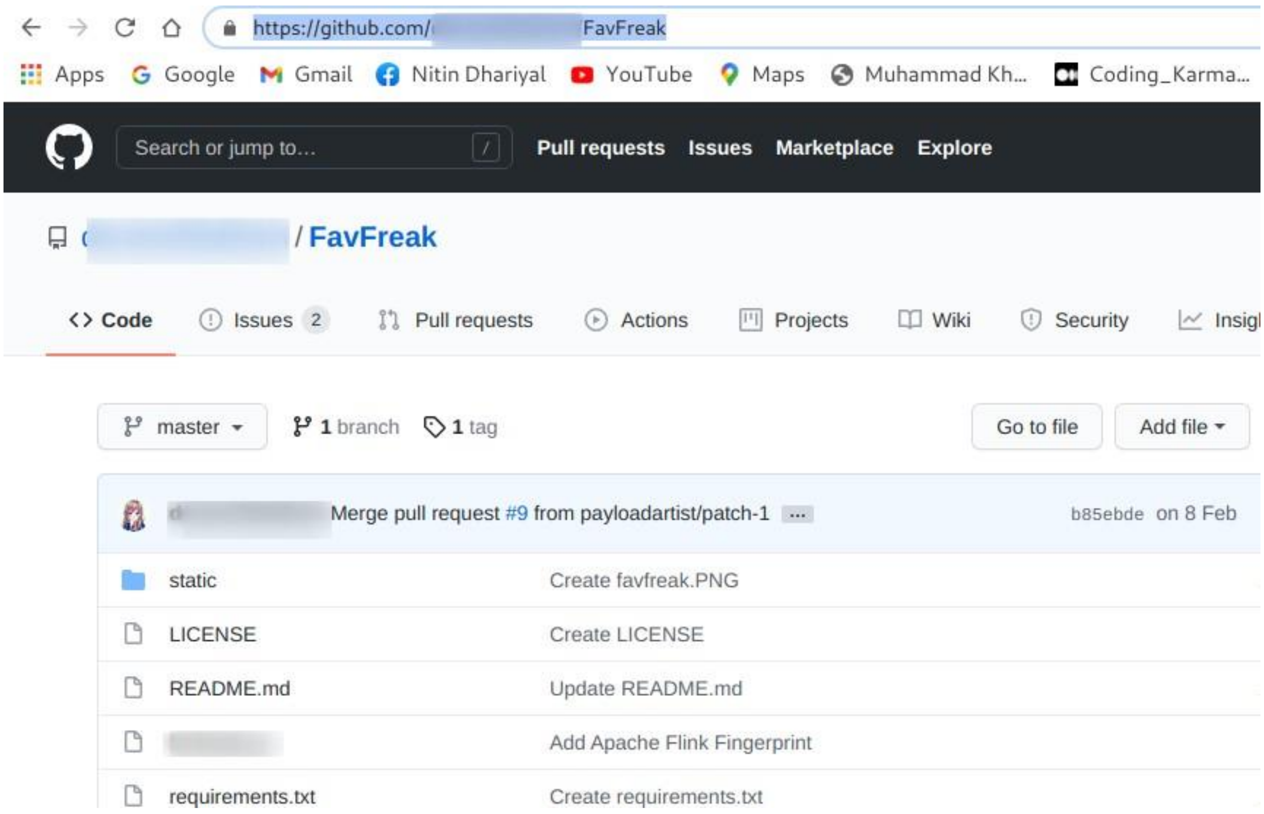


```
┌─[[redacted]]─[~/[redacted]/Programs/web]
└──╼ $git commit -m "first commit"
[master (root-commit) d8e056d] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 ReadMe.md
┌─[[redacted]]─[~/[redacted]/Programs/web]
└──╼ $
```

Step 6: Let us pull a repository into our git repository.

Command Used: git pull githubrepository.

choose the repository which you want to pull.



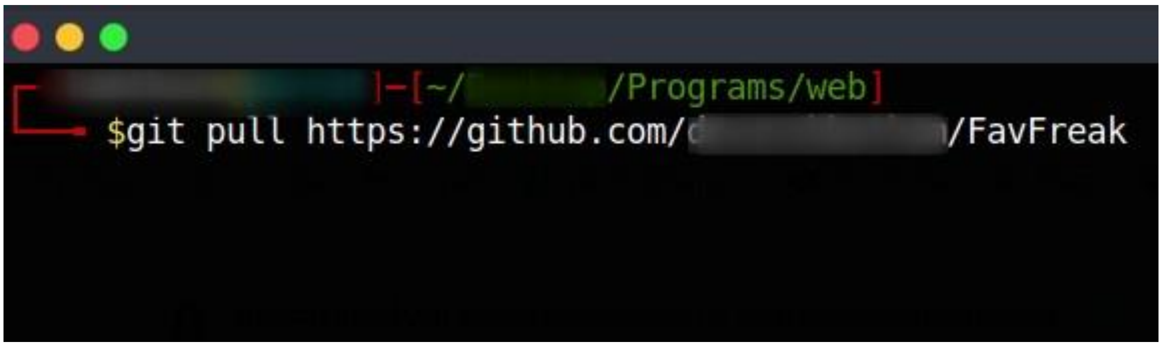
https://github.com/[redacted]FavFreak

[redacted] / FavFreak

master · 1 branch · 1 tag

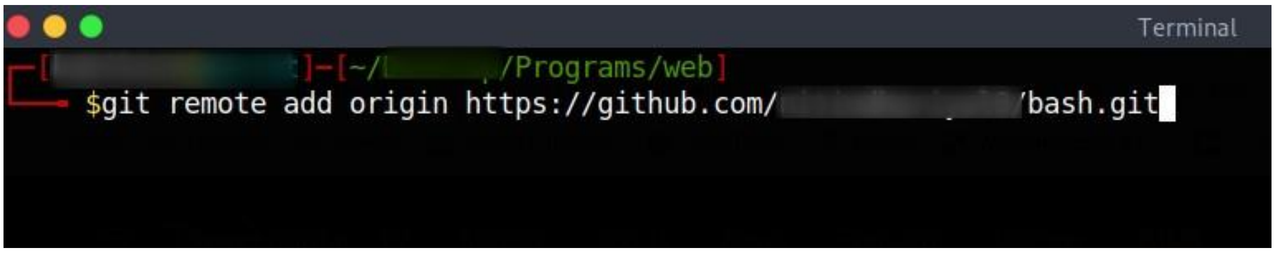
[redacted] Merge pull request #9 from payloadartist/patch-1		b85ebde on 8 Feb
static	Create favfreak.PNG	
LICENSE	Create LICENSE	
README.md	Update README.md	
[redacted]	Add Apache Flink Fingerprint	
requirements.txt	Create requirements.txt	

Step 7: Now enter the command.



```
[redacted]]-[~/[redacted]/Programs/web]
└──╼ $git pull https://github.com/[redacted]/FavFreak
```

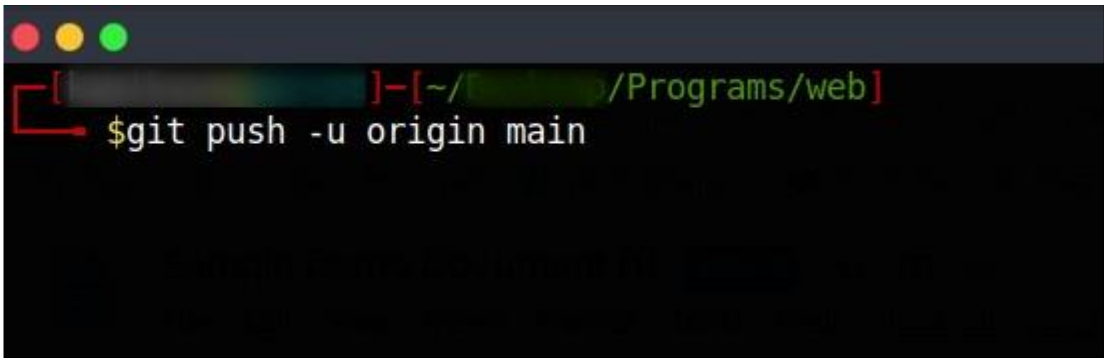
Step 8: The **git push command** is used to upload local repository content to a remote repository. **Pushing** is how you transfer commits from your local repository to a remote repository. use command `git remote add origin https://github.com/gitusername/file.git.`



```
Terminal
┌─[[redacted]]─[~/[redacted]/Programs/web]
└──╼ $git remote add origin https://github.com/[redacted]/bash.git
```

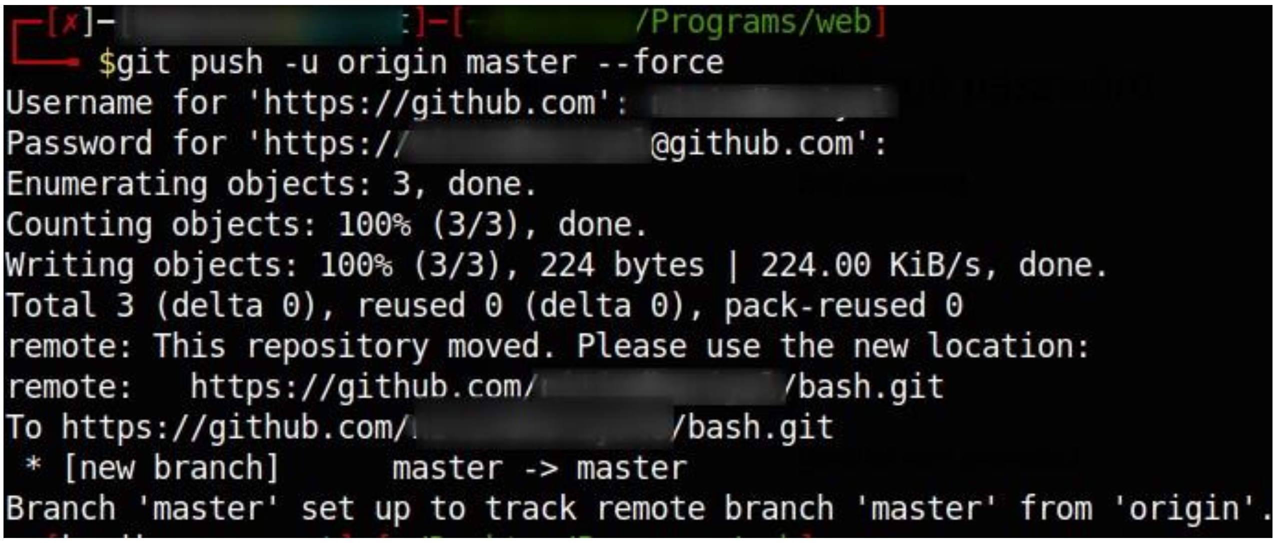
Step 9: Now we push our files into the git repository.

Command Used: git push -u origin main



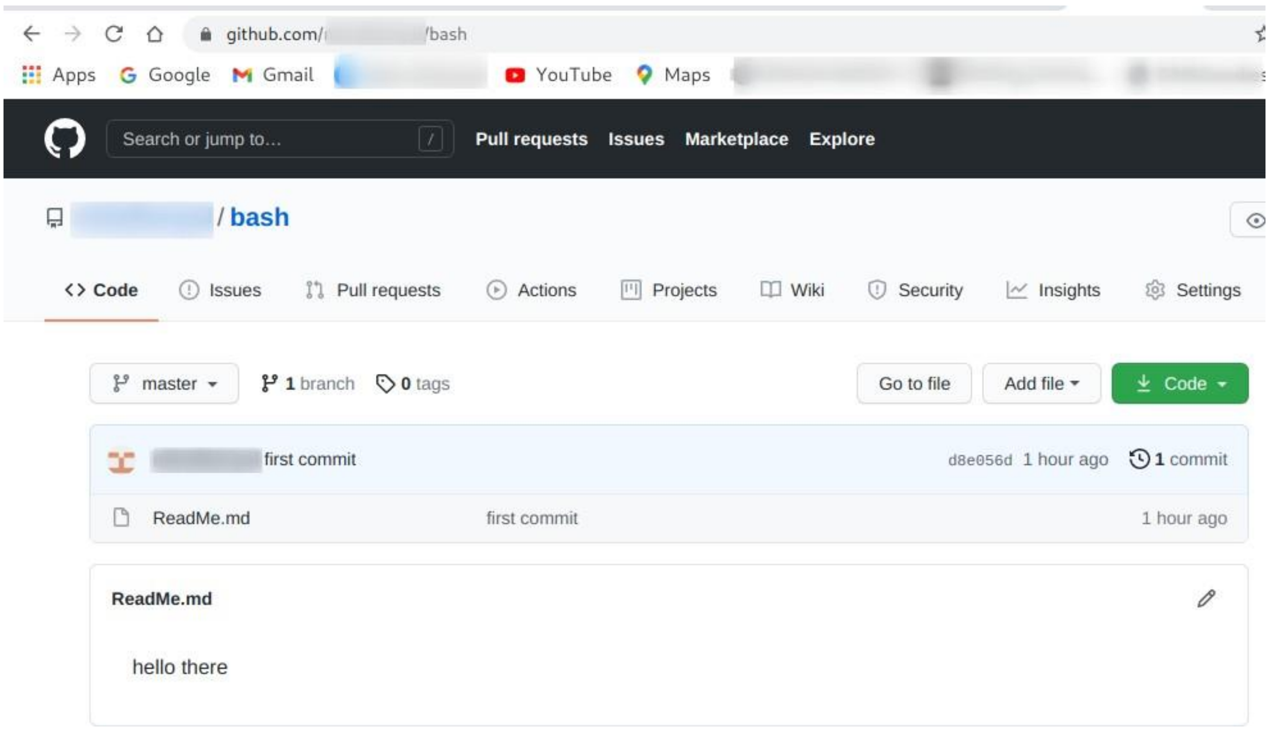
```
┌─[[redacted]]─[~/[redacted]/Programs/web]
└──╼ $git push -u origin main
```

Step 10: If this command is not working use:
git push -u origin master --force command.



```
┌─[✗]─[[redacted]]─[[redacted]/Programs/web]
└──╼ $git push -u origin master --force
Username for 'https://github.com': [redacted]
Password for 'https://[redacted]@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 224 bytes | 224.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote: This repository moved. Please use the new location:
remote:   https://github.com/[redacted]/bash.git
To https://github.com/[redacted]/bash.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

Step 11: It will ask your GitHub credentials. After completing this step, you will get your local repository into your git repository.



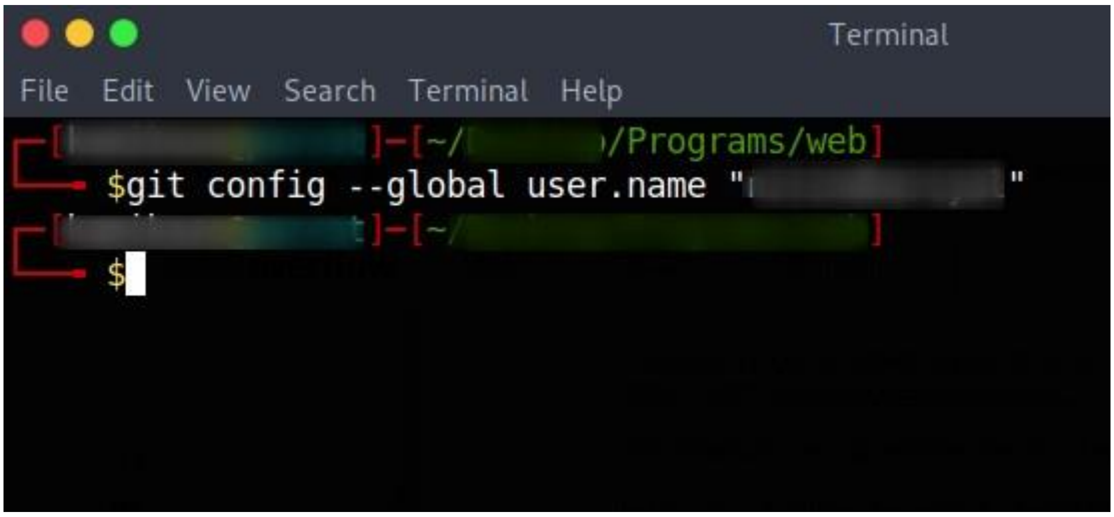
```
[redacted] first commit                    d8e056d 1 hour ago   1 commit
ReadMe.md          first commit                                1 hour ago
```

ReadMe.md

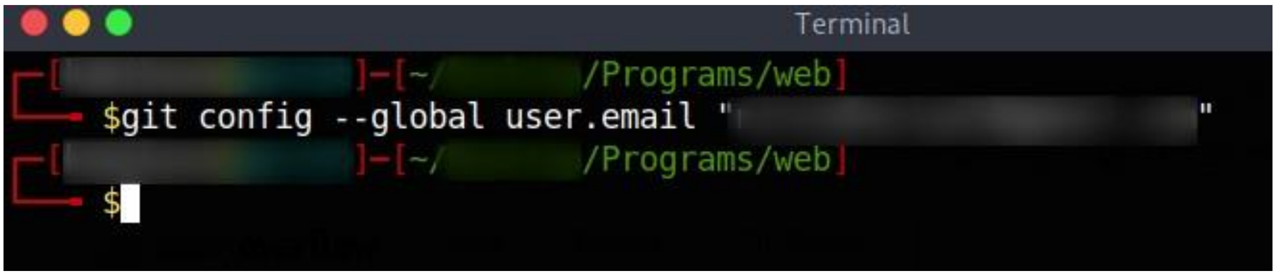
hello there

Step 12: Many configurations and Settings are possible with Git. The way to set these settings is git config. User.name and user.email are two key settings. These values specify which emails and names will be transmitted from a local computer. A —global flag is used for the git configuration to write the settings for all computer repositories. Without a —global flag only the existing repository you are in is applied.

Command Used: git config --global user.name “yourgithubusername”
Command Used: git config --global user.email “yourgithubemail”



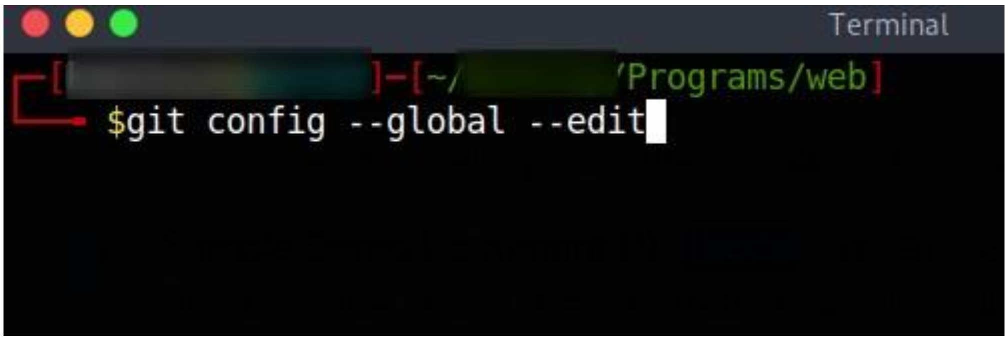
```
┌─[[redacted]]─[~/[redacted]/Programs/web]
└──╼ $git config --global user.name "[redacted]"
┌─[[redacted]]─[~[redacted]]
└──╼ $
```



```
┌─[[redacted]]─[~/[redacted]/Programs/web]
└── $git config --global user.email "[redacted]"
┌─[[redacted]]─[~/[redacted]/Programs/web]
└── $
```

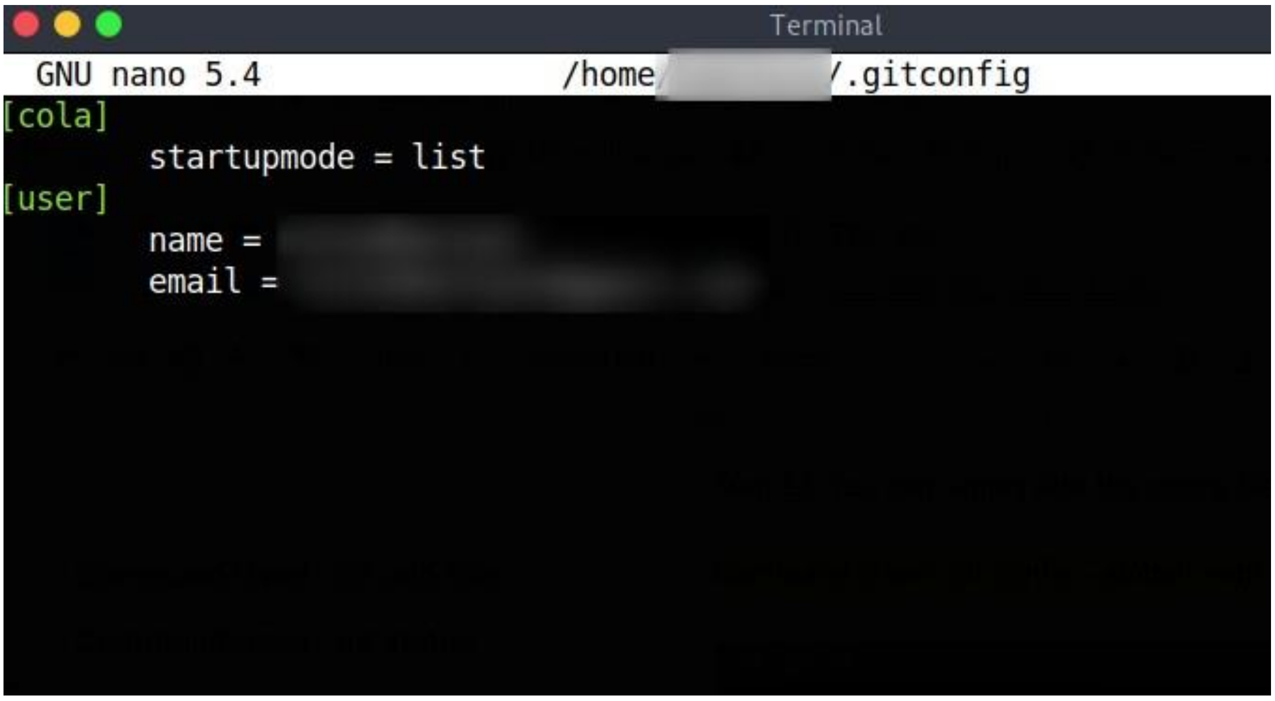
Step 13: You can simply edit the config file using --edit header.

Command Used: git config --global --edit



```
┌─[[redacted]]─[~/[redacted]/Programs/web]
└── $git config --global --edit
```

You can see the output as,

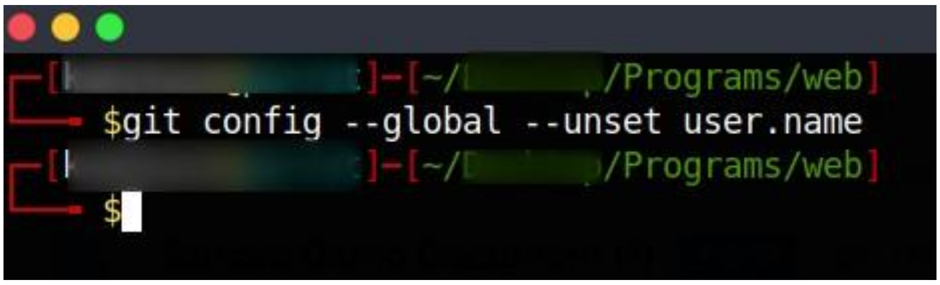


```
 GNU nano 5.4                  /home/[redacted]/.gitconfig
[cola]
        startupmode = list
[user]
        name = [redacted]
        email = [redacted]
```

You can edit the configuration file by here also.

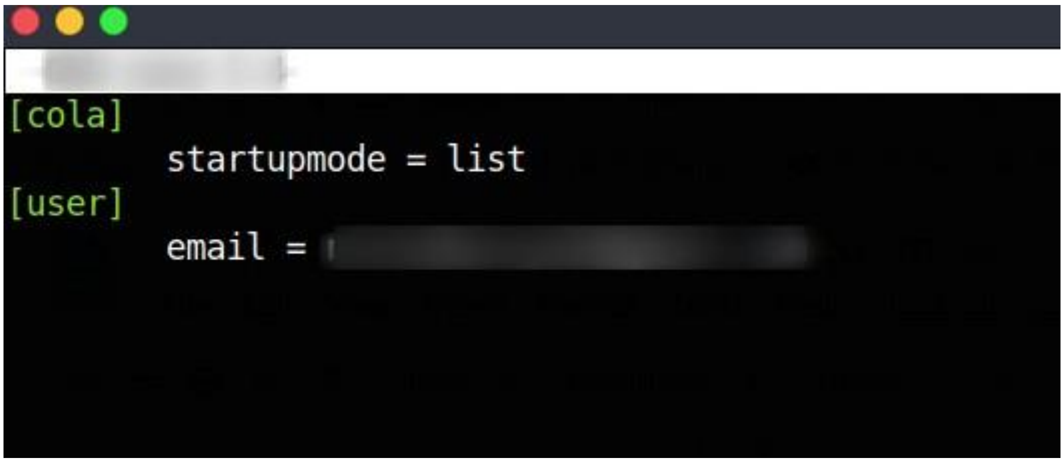
Step 14: Now, we are unsetting our configuration file using flag --unset in git config --global command.

Command used: git config --global --unset user.name
Command used: git config --global --unset user.email



```
┌─[[redacted]]─[~/[redacted]/Programs/web]
└──╼ $git config --global --unset user.name
┌─[[redacted]]─[~/[redacted]/Programs/web]
└──╼ $
```

You can check if this command works or not by using git config --global -edit command.



```
[redacted]
[cola]
        startupmode = list
[user]
        email = [redacted]
```

Step 15: Let us list out all the branches used in our repository.

Command Used: git branch.

Add a new branch or delete an array to determine which branch of the local repository is on.

```
# Create a new branch
$ git branch <branch_name>

# List all branches remotely or locally
$ git branch -a

# Delete a branch
$ git branch -d <branch_name>
```

In Practice:

```
# Create a new branch
$ git branch new_feature

# List branches
$ git branch -a
* SecretTesting
  new_feature
  remotes/origin/stable
  remotes/origin/staging
  remotes/origin/master -> origin/SecretTesting

# Delete a branch
$ git branch -d new_feature
Deleted branch new_feature (was 0254c3d).
```

Step 16: Hence, we have completed a few of basics git commands.