

# Steps to Uploading Your Code to GitHub

## Step 1

### Create a New Repository in GitHub.com

The screenshot shows the GitHub profile page for user 'afhaque'. The page features a search bar at the top, navigation links for 'Pull requests', 'Issues', and 'Gist', and a notification bell. Below the profile name, there is a list of recent activity, including stars and forks of repositories. On the right side, there is a section titled 'Repositories you contribute to' with a list of repositories and their star counts. Below that, the 'Your repositories' section is visible, featuring a '+ New repository' button highlighted with a red box. The page also includes a 'Improved commenting with Markdown' notification and a 'View 69 new broadcasts' link.

afhaque

RocketPropelledData starred afhaque/MeanMapAppV2.0 3 days ago

johnnyringo starred afhaque/MeanMapAppV2.0 3 days ago

zrelli starred afhaque/MeanMapAppV2.0 10 days ago

fatihgol forked afhaque/MeanMapAppV2.0 to fatihgol/MeanMapAppV2.0 12 days ago

jotalp forked afhaque/MeanMapAppV2.0 to jotalp/MeanMapAppV2.0 14 days ago

jcsibon forked afhaque/MeanMapAppV2.0 to jcsibon/MeanMapAppV2.0 16 days ago

sachsy forked afhaque/node-token-authentication to sachsy/node-token-authentication 21 days ago

sachsy forked afhaque/angular-google-maps to sachsy/angular-google-maps 21 days ago

pmaingi starred afhaque/Rice-DevChats-Bootstrap 21 days ago

pmaingi starred afhaque/node-todo 21 days ago

pmaingi starred afhaque/MeanMapAppV2.0 21 days ago

Improved commenting with Markdown

View 69 new broadcasts

Repositories you contribute to

Repository	Stars
RutgersCoding.../0125-mw-clas...	0
RutgersCoding.../All-Lesson-PI...	0
felix-d/mapit-MEAN-RESTful-app	15
carriesmith/mean-google-maps	0
localytics/angular-chosen	564

Your repositories 41

+ New repository

Find a repository...

All Public Private Sources Forks

wedclassdemo

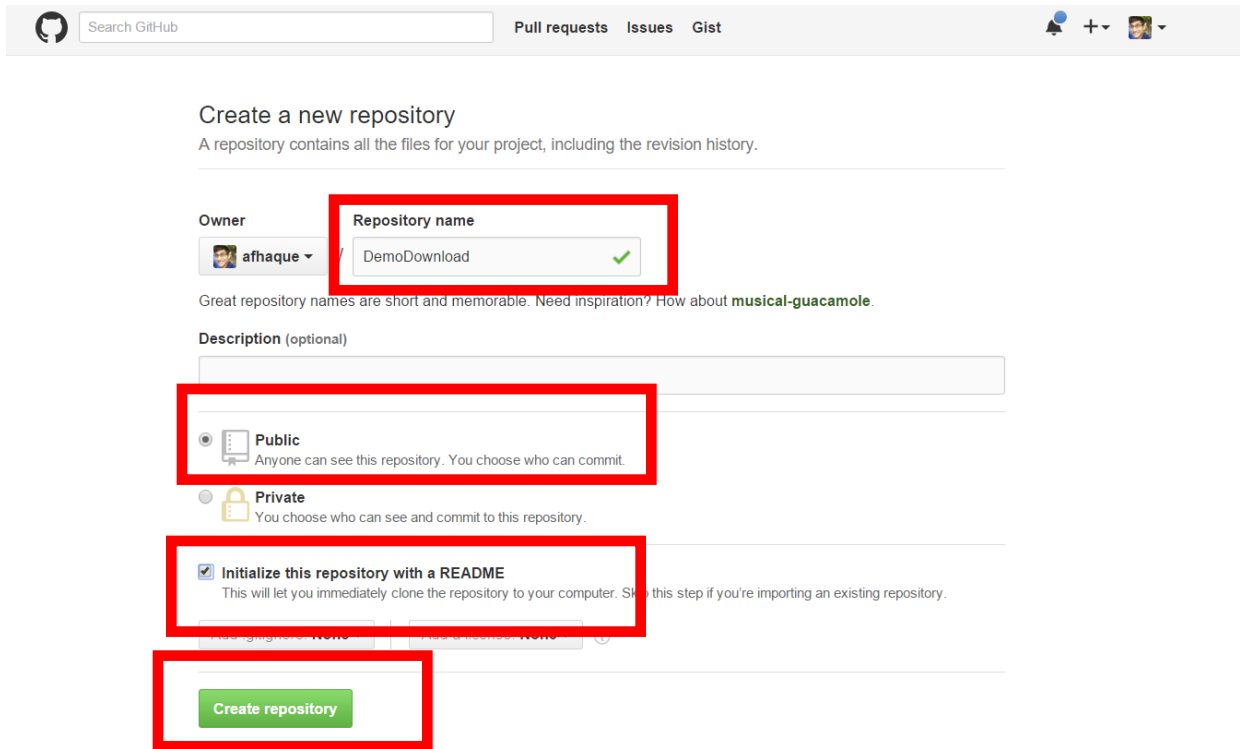
RutgersCodingBootc.../0125-mw-class-c...

DemoRepository

All-Lesson-Plans

## Step 2

Give the Repository a Name, set it to Public, and check “Initialize this repository with a README.”



The screenshot shows the GitHub interface for creating a new repository. The header includes the GitHub logo, a search bar, and navigation links for 'Pull requests', 'Issues', and 'Gist'. The main heading is 'Create a new repository' with a subtext: 'A repository contains all the files for your project, including the revision history.'

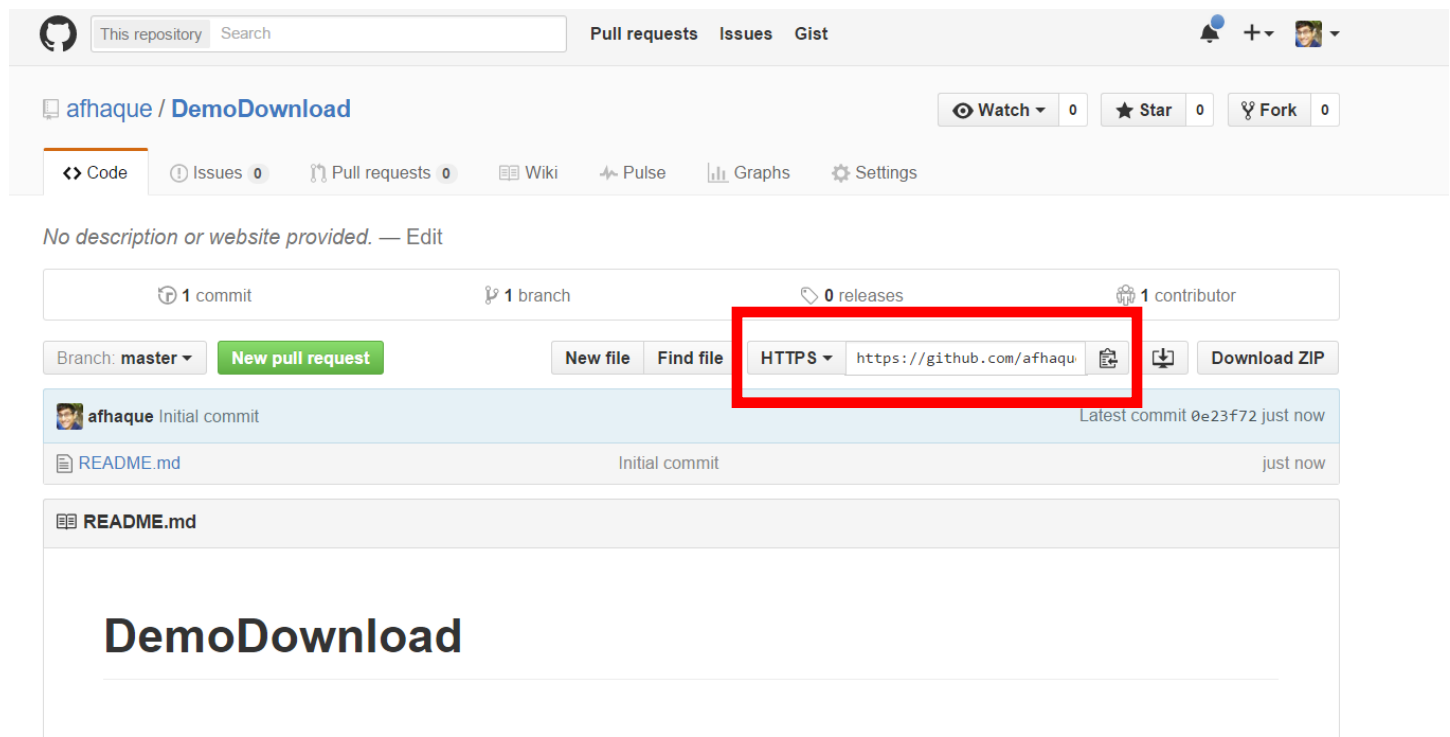
The form fields are as follows:

- Owner:** 'afhaque' (indicated by a dropdown arrow).
- Repository name:** 'DemoDownload' (with a green checkmark icon).
- Description (optional):** An empty text input field.
- Visibility:** 'Public' is selected (radio button), with the description 'Anyone can see this repository. You choose who can commit.' 'Private' is unselected (radio button), with the description 'You choose who can see and commit to this repository.'
- Initialization:** 'Initialize this repository with a README' is checked (checkbox), with the description 'This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.'

At the bottom, there are two buttons: 'Create repository' (highlighted in red) and 'Cancel' (disabled).

## Step 3

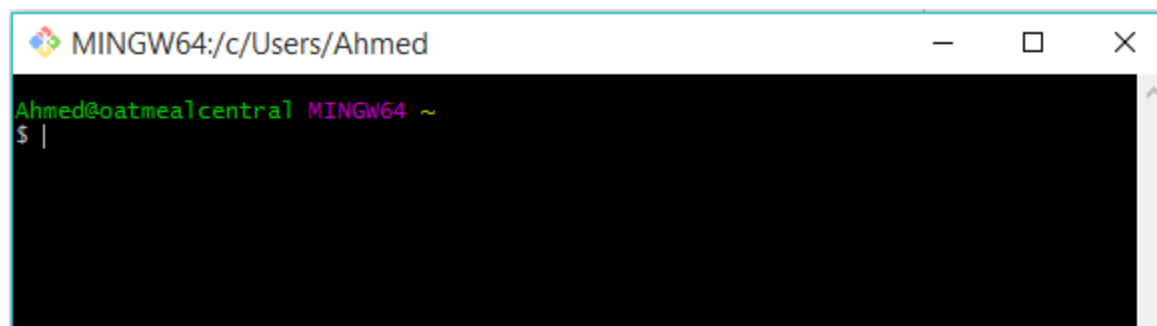
Copy the link associated with your new repository from GitHub.



The screenshot shows the GitHub interface for a repository named 'afhaque / DemoDownload'. The repository has 1 commit, 1 branch, 0 releases, and 1 contributor. The current branch is 'master'. A red box highlights the 'HTTPS' dropdown menu and the URL 'https://github.com/afhaque', which is the link to be copied. Below the repository information, there is a commit history table with one entry: 'afhaque Initial commit' with the commit hash '0e23f72' and the time 'just now'. The 'README.md' file is also listed. The main content area shows the 'README.md' file with the text 'DemoDownload'.

## Step 4

Open the Bash (or Terminal) window.



The screenshot shows a Windows Terminal window titled 'MINGW64:/c/Users/Ahmed'. The terminal prompt is 'Ahmed@oatmealcentral MINGW64 ~' and the cursor is at the '\$ |' prompt, indicating that the terminal is ready for input.

## Step 5

Using Console, navigate into a folder where you'd like to download your code.

```
MINGW64:/g/RutgersWork

Ahmed@oatmealcentral MINGW64 ~
$ cd g:

Ahmed@oatmealcentral MINGW64 /g
$ cd RutgersWork

Ahmed@oatmealcentral MINGW64 /g/RutgersWork
$ |
```

So in my case, I'm navigating into a folder called RutgersWork in my g: Drive. In your case, it can be anywhere on your computer that you want to download code.

## Step 6

Next, in console, type the command *git clone* \_\_\_\_\_ and paste the link to your GitHub repository in that blank space. Your screen should look something like mine.

```
MINGW64:/g/RutgersWork

Ahmed@oatmealcentral MINGW64 ~
$ cd g:

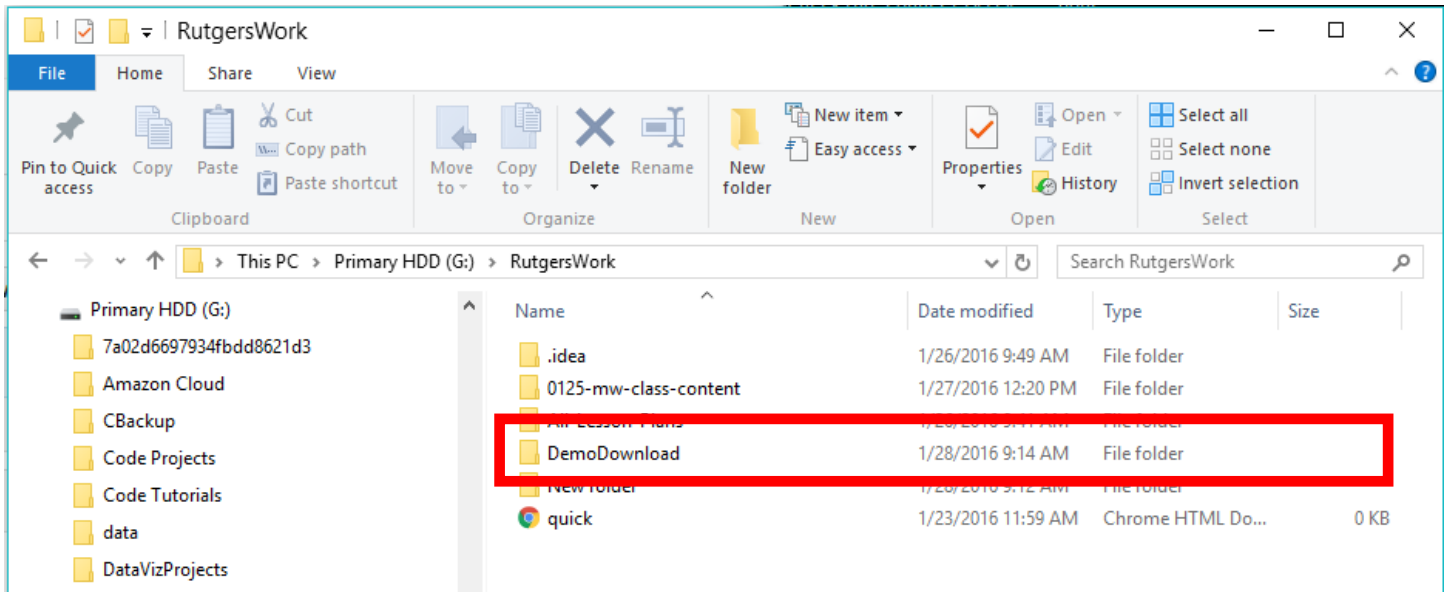
Ahmed@oatmealcentral MINGW64 /g
$ cd RutgersWork

Ahmed@oatmealcentral MINGW64 /g/RutgersWork
$ git clone https://github.com/afhaque/DemoDownload.git
Cloning into 'DemoDownload'...
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
Checking connectivity... done.

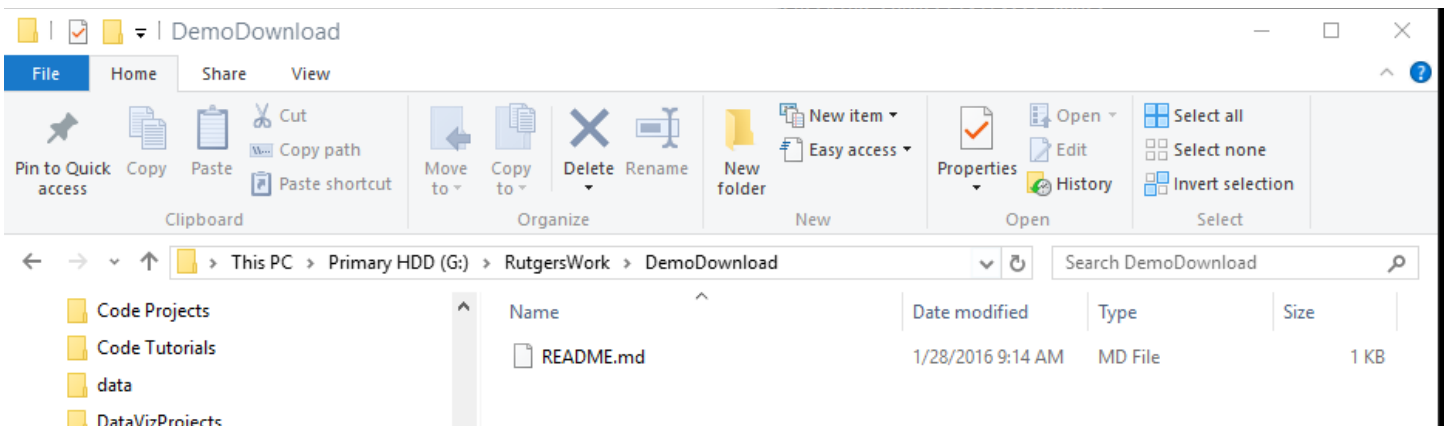
Ahmed@oatmealcentral MINGW64 /g/RutgersWork
$ |
```

## Step 7

Now if you navigate to that folder using file explorer, you will see a new folder with the same name as your repository has been downloaded.

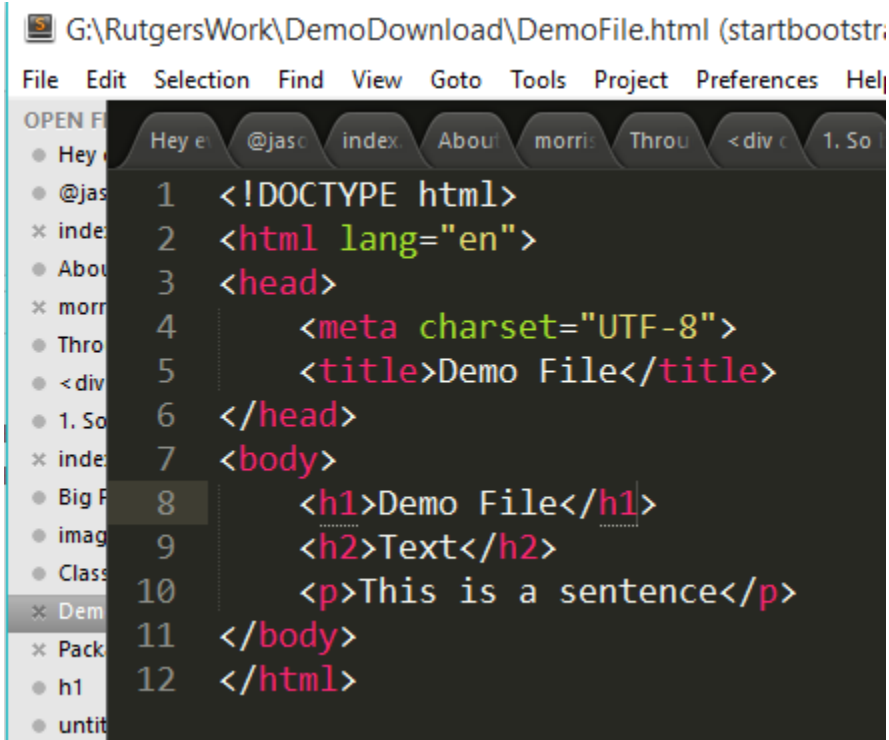


Go into that folder and you will see that it currently has a README file. This folder is now linked to Git.



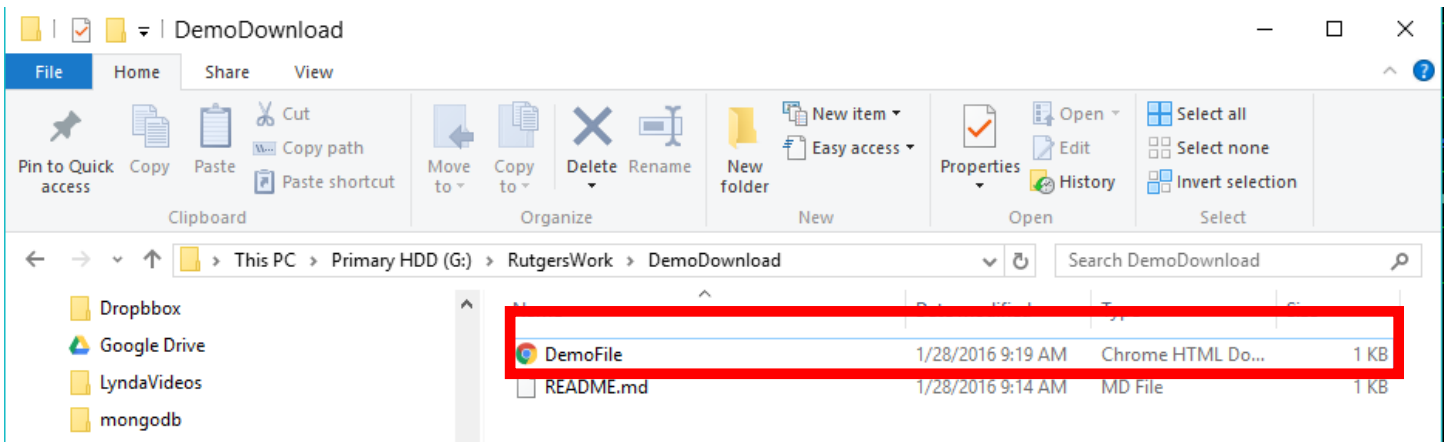
## Step 8

Next create a new HTML file (or use an old HTML file). Save that file inside of your local repository folder.



The screenshot shows a code editor window titled "G:\RutgersWork\DemoDownload\DemoFile.html (startbootstr...". The editor contains the following HTML code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Demo File</title>
6 </head>
7 <body>
8   <h1>Demo File</h1>
9   <h2>Text</h2>
10  <p>This is a sentence</p>
11 </body>
12 </html>
```



## Step 8

In console, navigate into the folder associated with your code using the `cd` command.

```
Ahmed@oatmealcentral MINGW64 /g/RutgersWork
$ cd DemoDownload
Ahmed@oatmealcentral MINGW64 /g/RutgersWork/DemoDownload (master)
```

## Step 9

Next, type the command `git add -A`. This will tell git to notice the addition of the new html file. If this command worked you shouldn't see anything happen.

```
Ahmed@oatmealcentral MINGW64 /g/RutgersWork/DemoDownload (master)
$ git add -A
Ahmed@oatmealcentral MINGW64 /g/RutgersWork/DemoDownload (master)
$ |
```

## Step 10

Type the command `git commit -m "_____"` and place a comment in the quotes. This will save your changes locally.

```
Ahmed@oatmealcentral MINGW64 /g/RutgersWork/DemoDownload (master)
$ git commit -m "I added the Demo file!"
[master 3661b45] I added the Demo file!
1 file changed, 12 insertions(+)
 create mode 100644 DemoFile.html
Ahmed@oatmealcentral MINGW64 /g/RutgersWork/DemoDownload (master)
$ .....
```

## Step 11

Type the final command *git push*. Enter your GitHub username and password if asked.

```
Ahmed@oatmealcentral MINGW64 /g/RutgersWork/DemoDownload (master)
$ git push
warning: push.default is unset; its implicit value has changed in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the traditional behavior, use:

  git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

  git config --global push.default simple

When push.default is set to 'matching', git will push local branches
to the remote branches that already exist with the same name.

Since Git 2.0, Git defaults to the more conservative 'simple'
behavior, which only pushes the current branch to the corresponding
remote branch that 'git pull' uses to update the current branch.

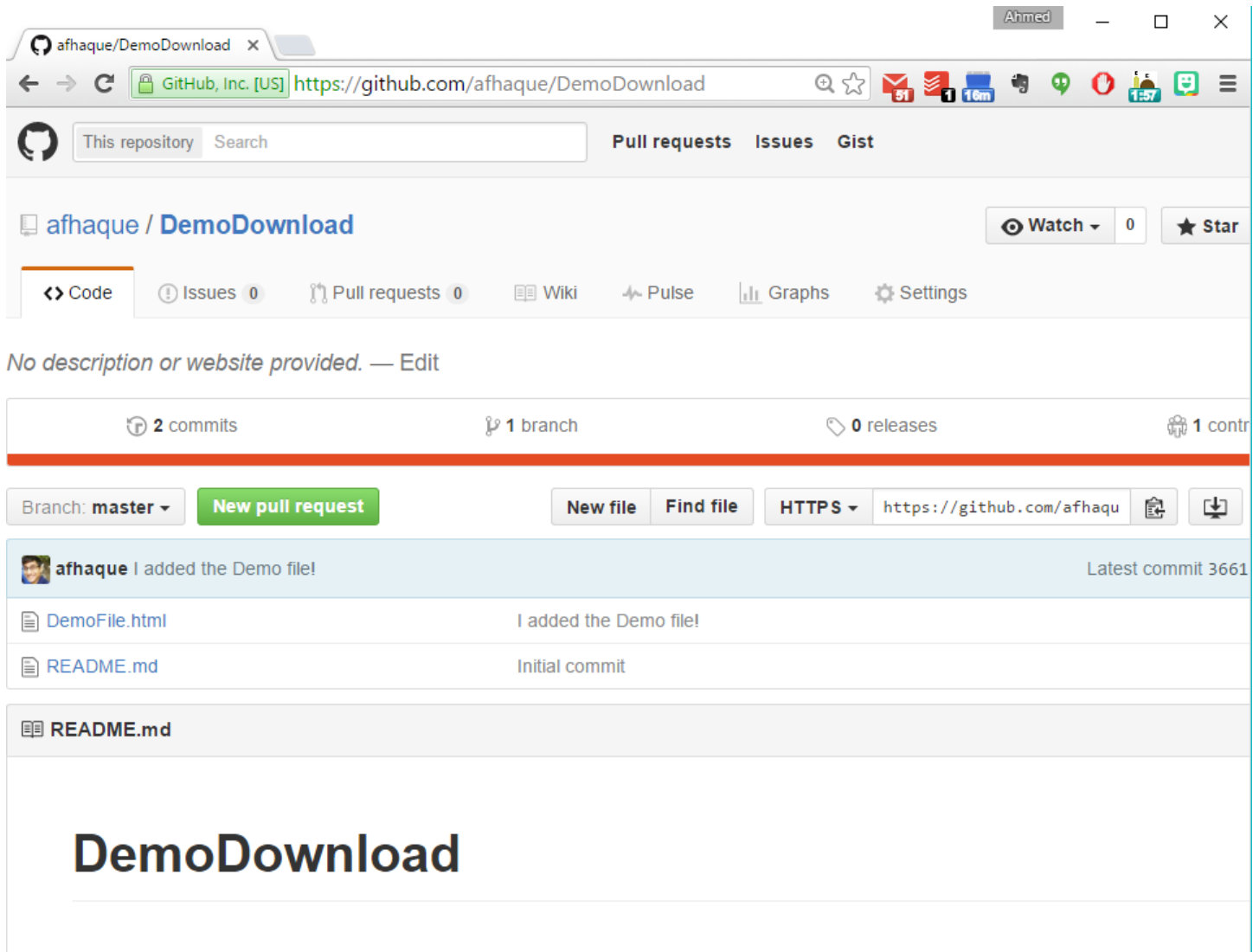
See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Username for 'https://github.com': afhaque
Password for 'https://afhaque@github.com':
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 420 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/afhaque/DemoDownload.git
 0e23f72..3661b45  master -> master
```



## Step 12

Go back to the GitHub repository and you should see the file having been inserted!



The screenshot shows a web browser window displaying the GitHub repository page for 'afhaque/DemoDownload'. The browser's address bar shows the URL 'https://github.com/afhaque/DemoDownload'. The repository page includes a search bar, navigation links for 'Pull requests', 'Issues', and 'Gist', and a 'Watch' button with a count of 0. Below the repository name, there are tabs for 'Code', 'Issues 0', 'Pull requests 0', 'Wiki', 'Pulse', 'Graphs', and 'Settings'. A message states 'No description or website provided. — Edit'. The repository statistics show '2 commits', '1 branch', '0 releases', and '1 contributor'. A 'New pull request' button is visible. The commit history shows a commit by 'afhaque' with the message 'I added the Demo file!' and a timestamp of 'Latest commit 3661'. The commit list includes 'DemoFile.html' (I added the Demo file!) and 'README.md' (Initial commit). The README file is expanded, showing the text 'DemoDownload'.

**SUCCESS!!!!**