A step-by-step guide to Git

Don't be nervous. This beginner's guide will quickly and easily get you started using Git.

25 Jan 2018 | Kedar Vijay Kulkarni (Red Hat) (/users/kkulkarn) | 332 | 10 comments

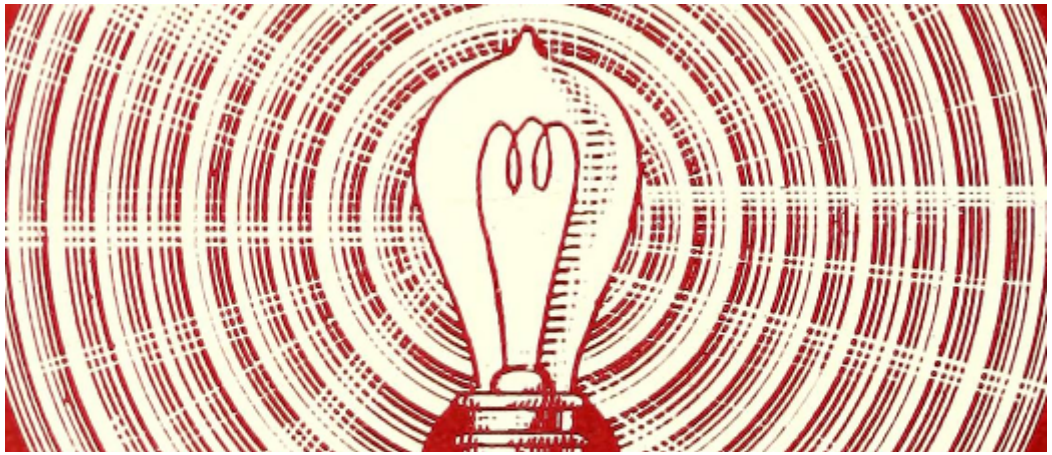


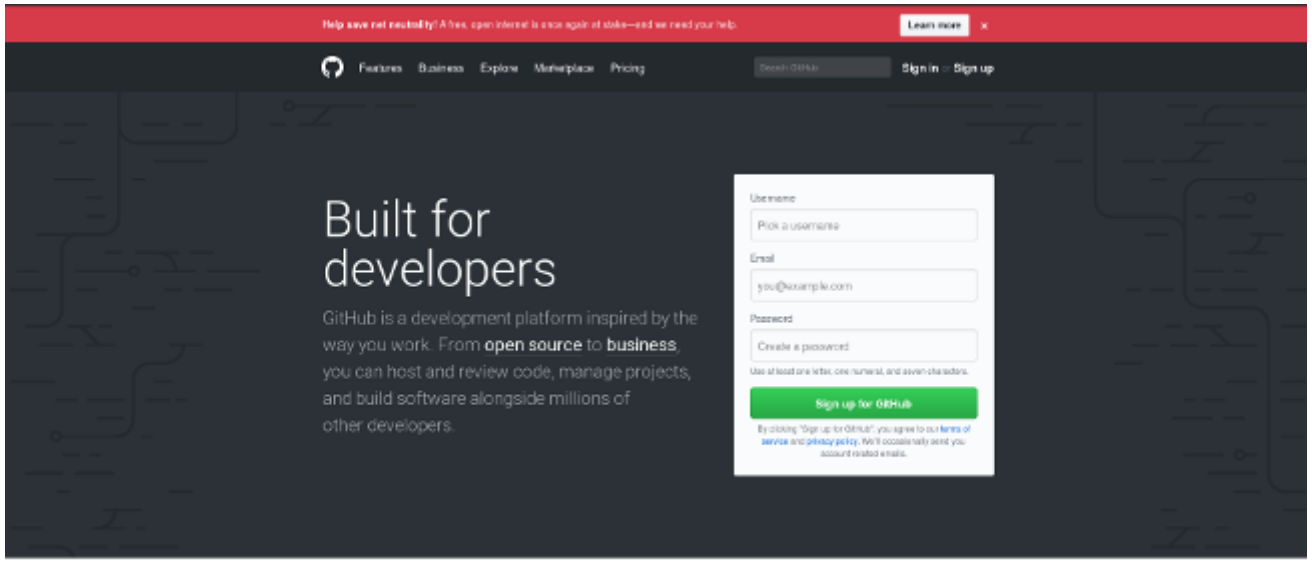
Image by : *Internet Archive Book Images. Modified by Opensource.com. CC BY-SA 4.0*

If you've never used Git (https://opensource.com/resources/what-is-git), you may be nervous about it. There's nothing to worry about—just follow along with this step-by-step getting-started guide, and you will soon have a new Git repository hosted on GitHub (https://opensource.com/life/15/11/short-introduction-github).

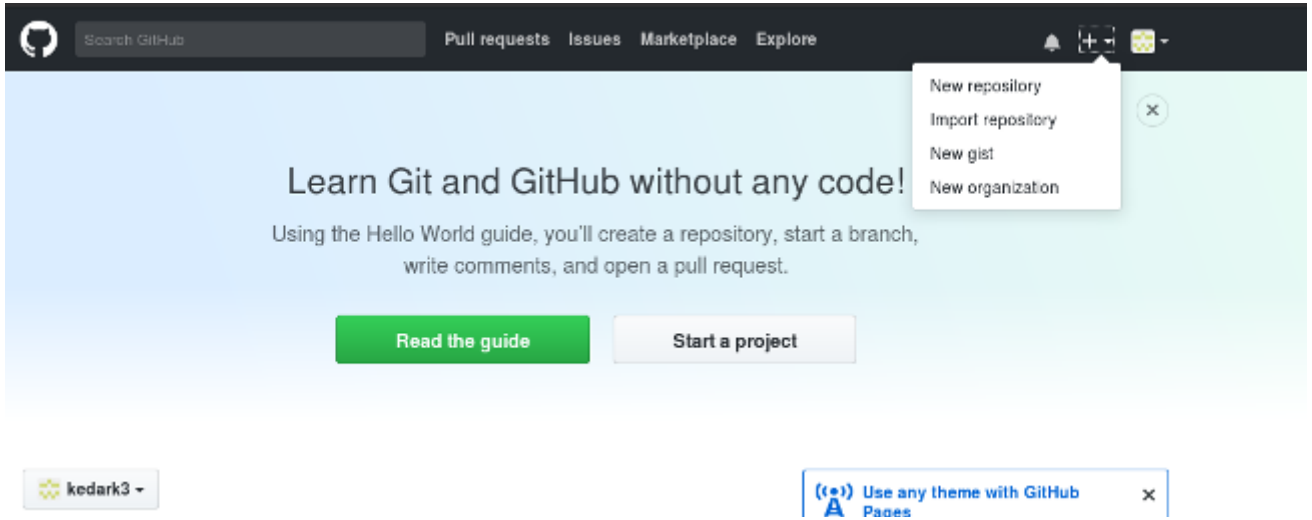
Before we dive in, let's clear up a common misconception: Git isn't the same thing as GitHub. Git is a version-control system (i.e., a piece of software) that helps you keep track of your computer programs and files and the changes that are made to them over time. It also allows you to collaborate with your peers on a program, code, or file. GitHub and similar services (including GitLab and BitBucket) are websites that host a Git server program to hold your code.

Step 1: Create a GitHub account

The easiest way to get started is to create an account on GitHub.com (https://github.com/) (it's free).

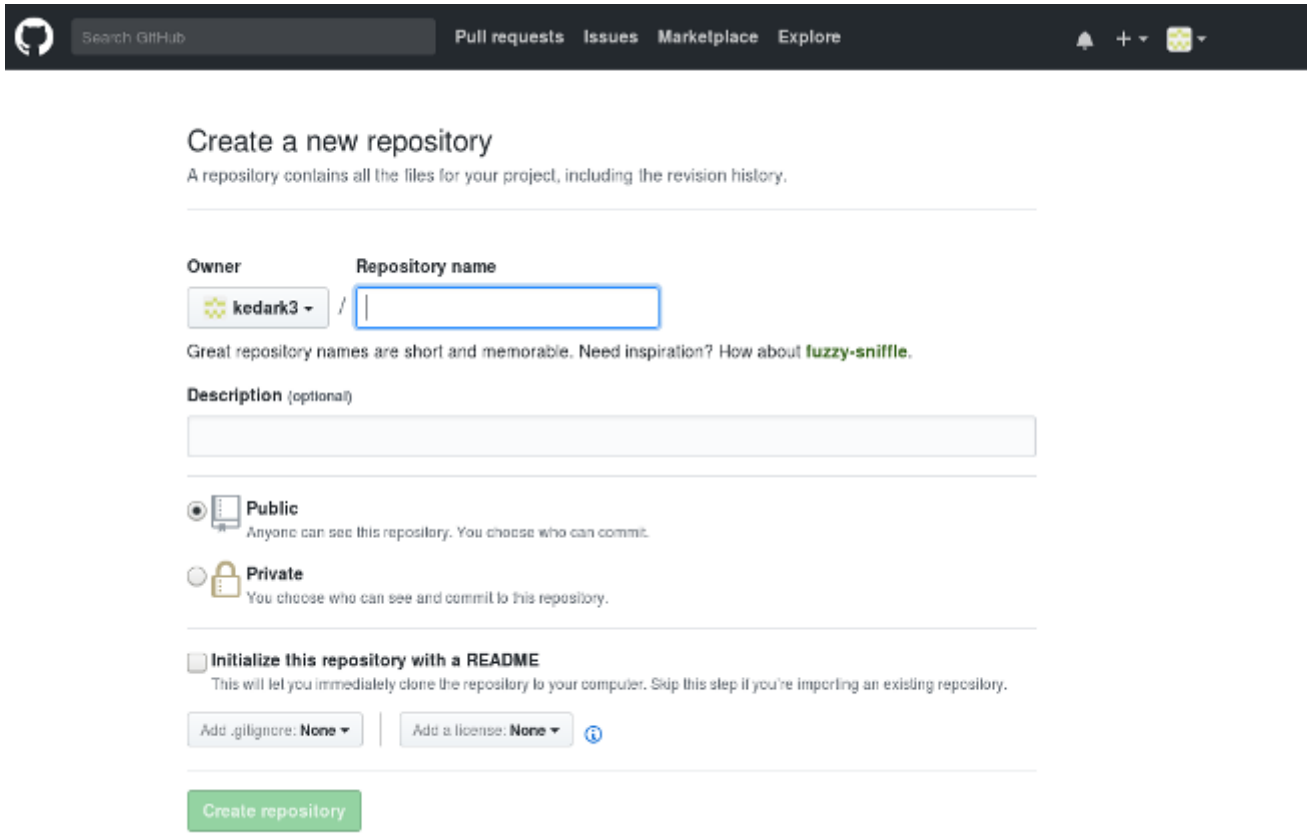


Pick a username (e.g., octocat123), enter your email address and a password, and click **Sign up for GitHub**. Once you are in, it will look something like this:



Step 2: Create a new repository

A repository is like a place or a container where something is stored; in this case we're creating a Git repository to store code. To create a new repository, select **New Repository** from the + sign dropdown menu (you can see I've selected it in the upper-right corner in the image above).

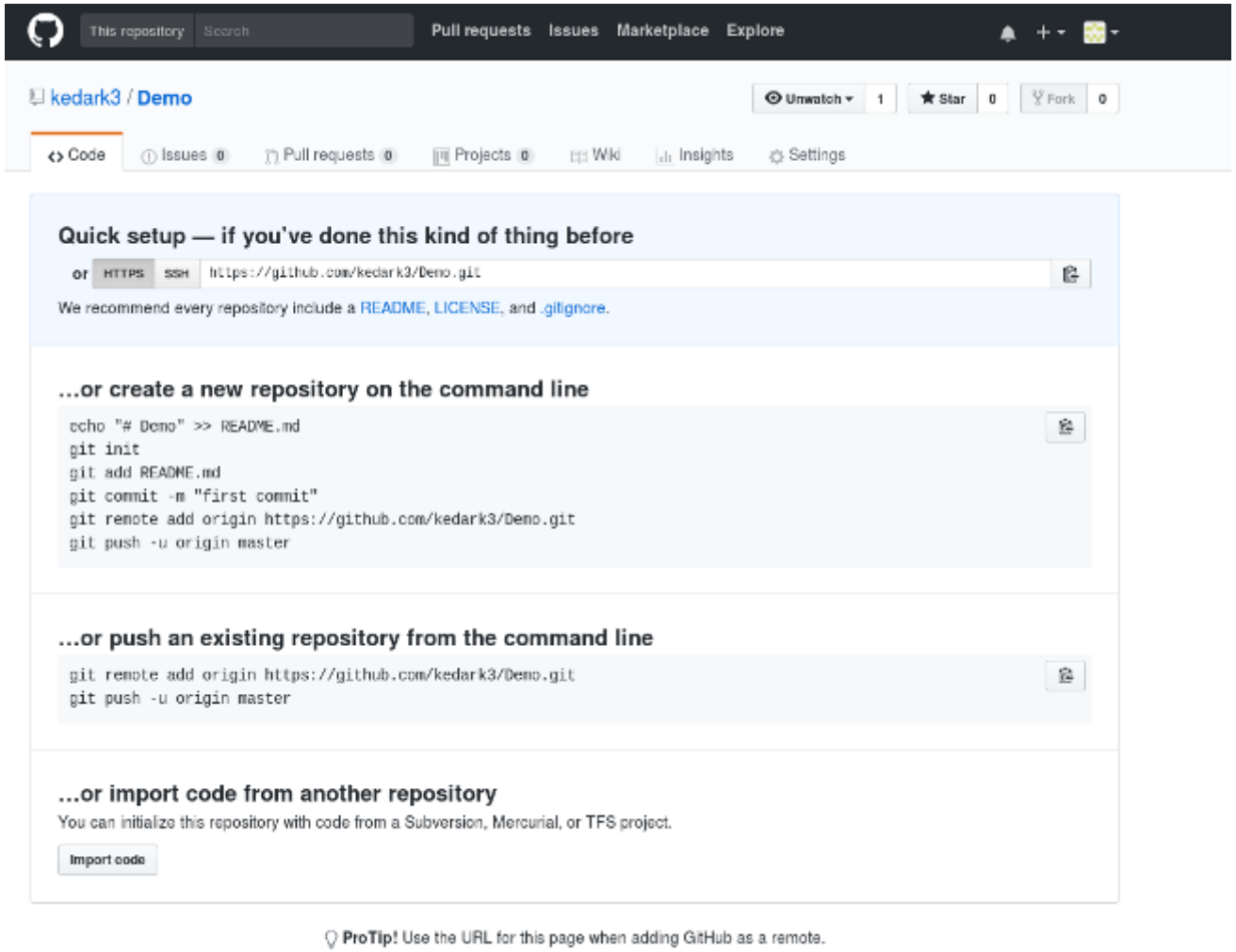


Enter a name for your repository (e.g, "Demo") and click **Create Repository**. Don't worry

Congratulations! You have set up your first repo on GitHub.com.

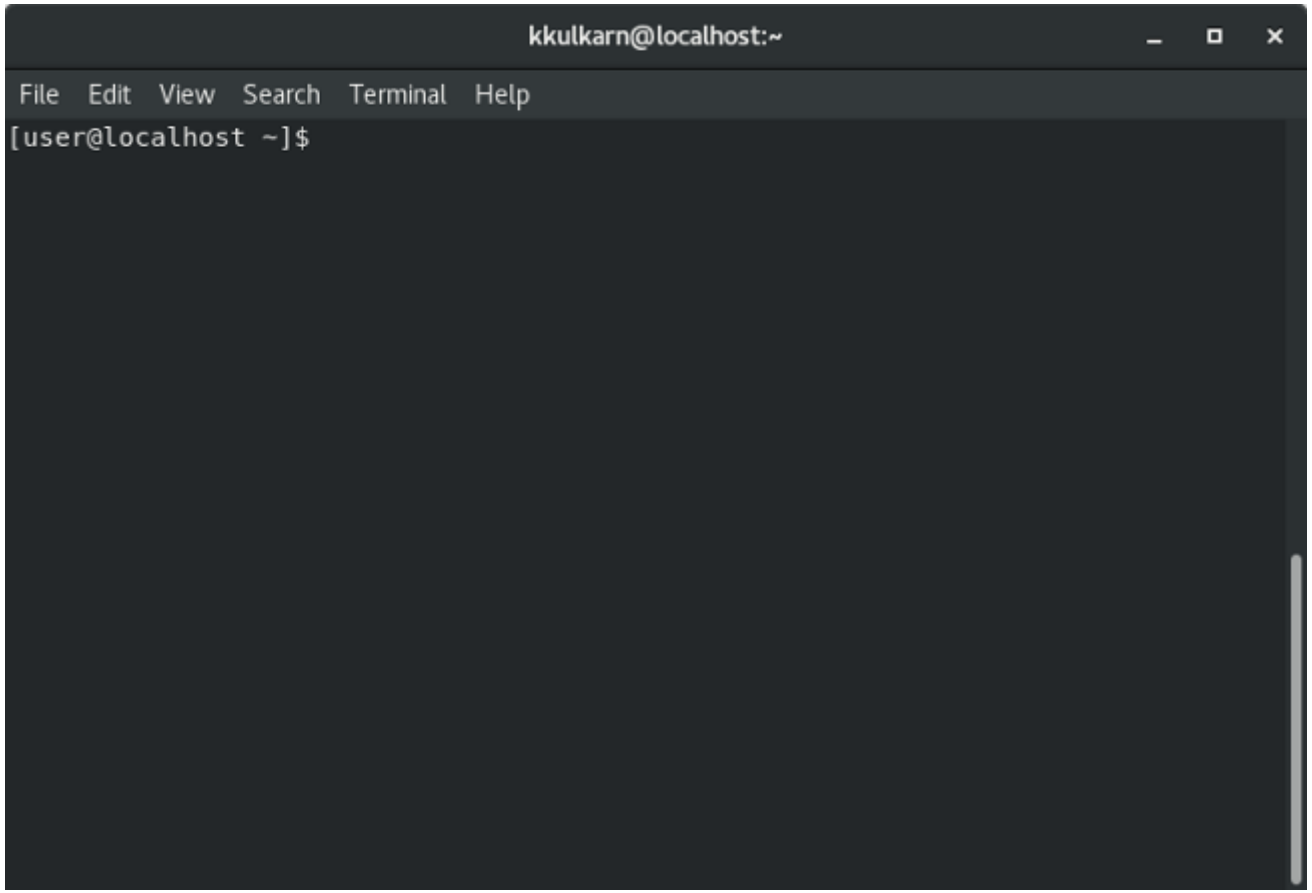
Step 3: Create a file

Once your repo is created, it will look like this:



Don't panic, it's simpler than it looks. Stay with me. Look at the section that starts "...or create a new repository on the command line," and ignore the rest for now.

Open the *Terminal* program on your computer.



Type `git` and hit **Enter**. If it says command `bash: git: command not found`, then [install Git (https://www.linuxbabe.com/linux-server/install-git-verion-control-on-linux-debianubuntufedoraarchlinux#ert-2)](https://www.linuxbabe.com/linux-server/install-git-verion-control-on-linux-debianubuntufedoraarchlinux#ert-2) with the command for your Linux operating system or

In the terminal, type:

```
mkdir Demo
```

This command will create a directory (or folder) named *Demo*.

Change your terminal to the *Demo* directory with the command:

```
cd Demo
```

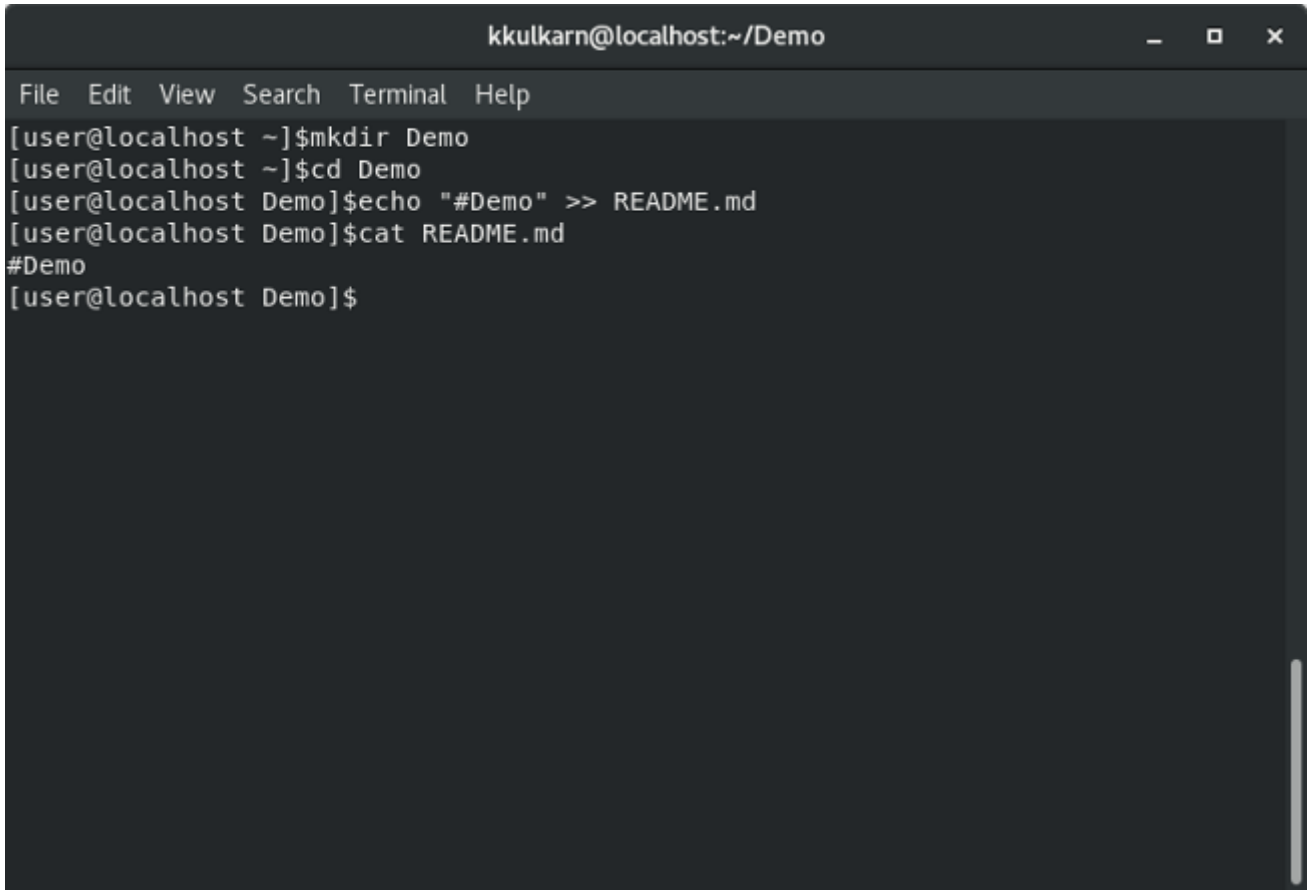
Then enter:

```
echo "#Demo" >> README.md
```

This creates a file named `README.md` and writes `#Demo` in it. To check that the file was created successfully, enter:

```
cat README.md
```

This will show you what is inside the `README.md` file, if the file was created correctly. Your terminal will look like this:



To tell your computer that *Demo* is a directory managed by the Git program, enter:

```
git init
```

Then, to tell the Git program you care about this file and want to track any changes from this point forward, enter:

Step 4: Make a commit

So far you've created a file and told Git about it, and now it's time to create a *commit*. Commit can be thought of as a milestone. Every time you accomplish some work, you can write a Git commit to store that version of your file, so you can go back later and see what it looked like at that point in time. Whenever you make a change to your file, you create a new version of that file, different from the previous one.

To make a commit, enter:

```
git commit -m "first commit"
```

That's it! You just created a Git commit and included a message that says *first commit*. You must always write a message in commit; it not only helps you identify a commit, but it also enables you to understand what you did with the file at that point. So tomorrow, if you add a new piece of code in your file, you can write a commit message that says, *Added new code*, and when you come back in a month to look at your commit history or Git log (the list of commits), you will know what you changed in the files.

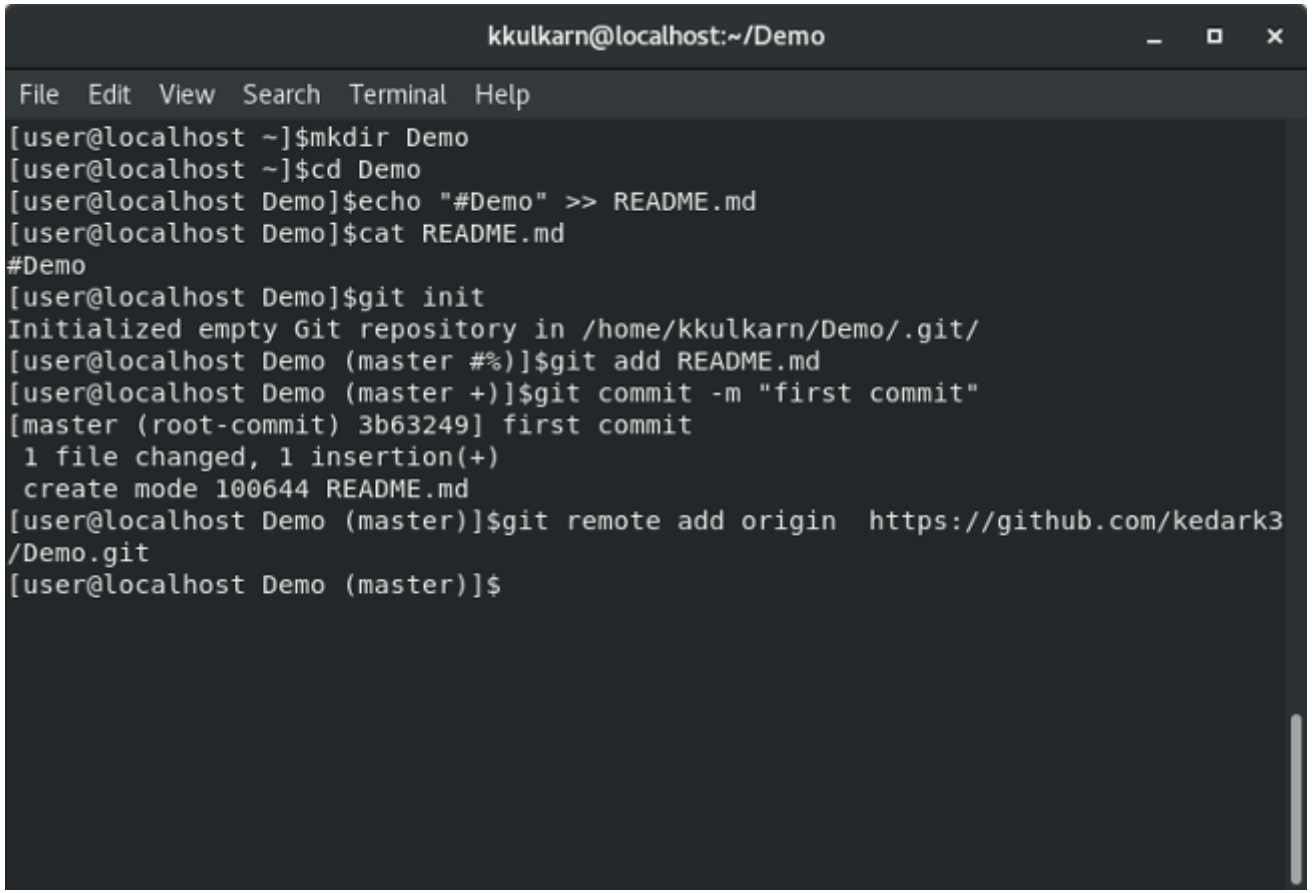
Step 5: Connect your GitHub repo with your computer

Now, it's time to connect your computer to GitHub with the command:

```
git remote add origin https://github.com/<your_username>/Demo.git
```

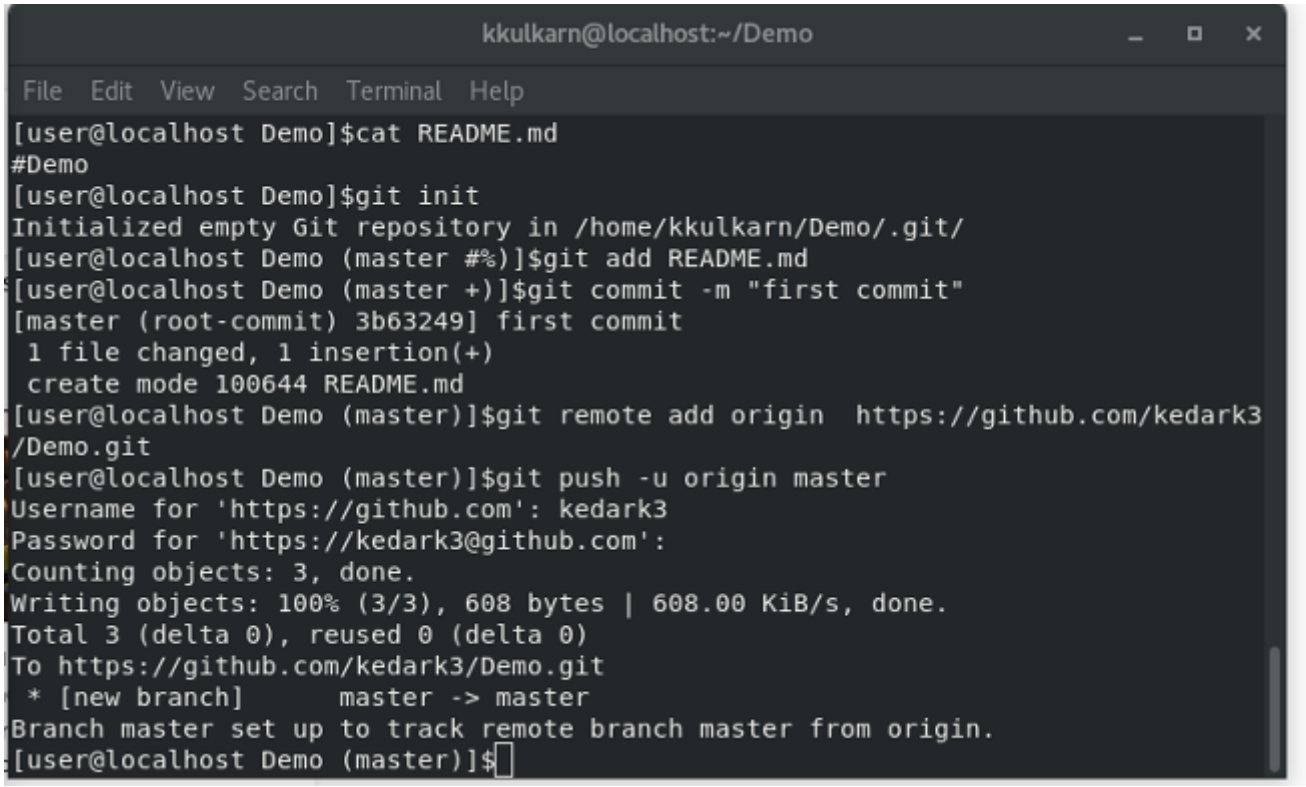
Let's look at this command step by step. We are telling Git to add a `remote` called `origin` with the address `https://github.com/<your_username>/Demo.git` (i.e., the URL of your Git repo on GitHub.com). This allows you to interact with your Git repository on GitHub.com by typing `origin` instead of the full URL and Git will know where to send your code. Why `origin`? Well, you can name it anything else if you'd like.

Now we have connected our local copy of the *Demo* repository to its remote counterpart on GitHub.com. Your terminal looks like this:

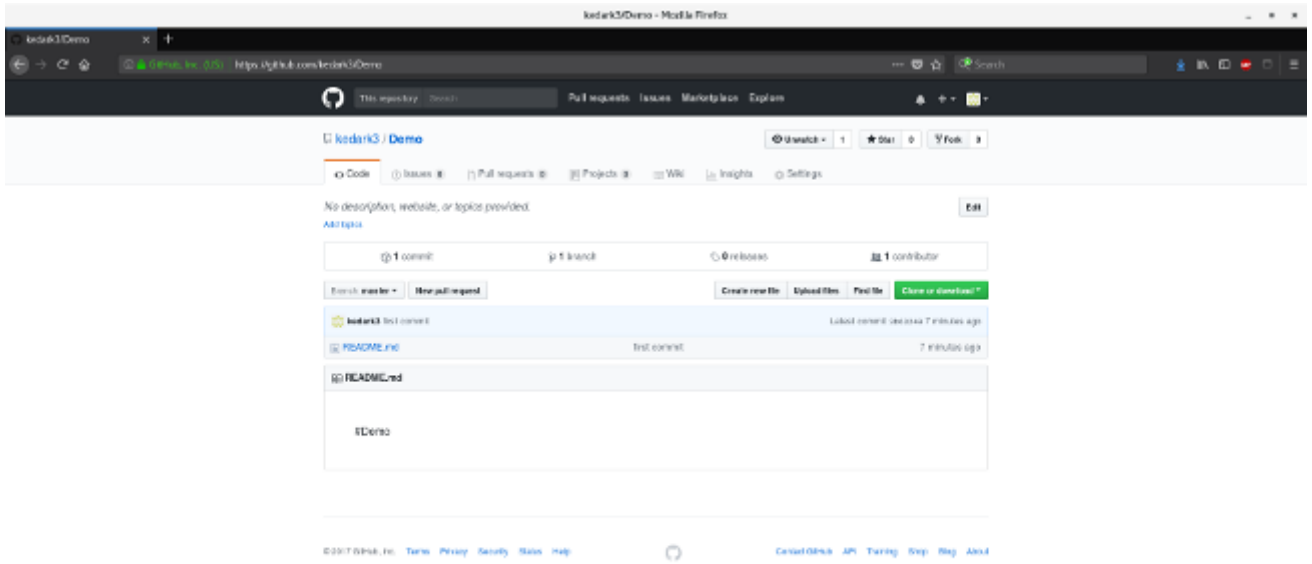


Now that we have added the remote, we can push our code (i.e., upload our `README.md` file) to GitHub.com.

Once you are done, your terminal will look like this:



And if you go to `https://github.com/<your_username>/Demo` you will see something like this:



That's it! You have created your first GitHub repo, connected it to your computer, and pushed (or uploaded) a file from your computer to your repository called *Demo* on GitHub.com. Next time, I will write about Git cloning (downloading your code from GitHub to your computer), adding new files, modifying existing files, and pushing (uploading) files to GitHub.

Topics :

Git (/tags/git) **Programming** (/tags/programming)

How-tos and tutorials (/tags/how-tos-and-tutorials)

(/users/kkulkarn)

Kedar Vijay Kulkarni - Kedar is a Software Quality Engineer at Red Hat working with CloudForms(upstream ManageIQ) project and primarily looking at deployment/management of our internal infrastructure. Interested in Jenkins Pipeline and Ansible for automating deployments. Also writing Shinken modules for Monitoring and Alerting. In his free time he likes to Travel, watch interesting videos, learn about new technologies.

• More about me (/users/kkulkarn)

Recommended reading

How to resolve a git merge conflict (/article/20/4/git-merge-conflict?utm_campaign=intrel)

4 Git scripts I can't live without (/article/20/4/git-extras?utm_campaign=intrel)

Why I use open source technology for web development (/article/20/4/open-source-web-development?utm_campaign=intrel)

10 Comments

Jeff Macharyas (/users/jeffmacharyas) on 25 Jan 2018 4

Wow, this would have come in handy three years ago when I had to post a PostgreSQL test to GitHub and had no idea what Git or PostgreSQL even was. Thanks for the very clear instructions.

john379 on 25 Jan 2018 3

I often see github used as personal pages and even blogs - maybe a little coverage on this aspect of it in your series:) Also, noticed the RSS link on top and have added it to my feeds. I like the presentation and style - very easy to follow!

Kedar Vijay Kulkarni (/users/kkulkarn) on 25 Jan 2018 1

Sure. Thanks for the feedback. I will definitely consider adding article on Personal Pages/Site using git repo. It is interesting and everyone should know it before they decide to go somewhere else for hosting their blog.

Sachin Patil (/users/psachin) on 19 Apr 2018 0

I think you are looking for https://pages.github.com/ (https://pages.github.com/). It covers steps to get you started.

Chrissy McLennon (/users/chrissymclennon) on 26 Jan 2018 1

Amazing guide! Really helpful for someone who's only starting!

Catalin(ux) M. Boie on 26 Jan 2018 2

Hello!

Nice article.

If you do not like a proprietary solution to host your projects, there are alternatives, easy to use online or install on your servers.

For example, https://rocketgit.com (https://rocketgit.com)

It is the only AGPL product on the market, respecting contributors copyright and protecting their work.

Full disclosure: I am the author of the RocketGit project.

Li'lGet (/users/nikki-f-nelson) on 26 Jan 2018 1

This*Helps*!* - I have been flirting with GitHub and more but? I have yet 2 flex my IT*Muscles. Much *Appreciated*!* Tyou ~*;p HavGr8*Techno*Day!
*inspired-li'lGet

Bill Trautman (/users/lhasadad) on 27 Jan 2018 1

Nice and concise.

Tomut (/users/gaelletjat) on 14 Feb 2018 0

Thank you Kedar.

Yesterday I was exactly struggling with git.

dbmuse on 15 Feb 2018 0

seems like the upload file section would be step 6 following the format of previous steps. I liked it.