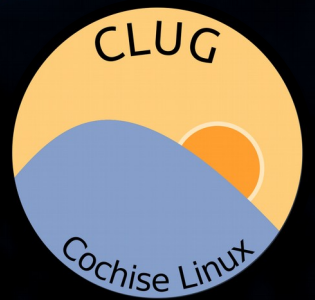


# CLUG

Cochise Linux User Group

Git Primer – Git Your Project On...  
By: Paul Bernard

CochiseLinux.com



# What This Will Cover

- Explains the concept of GIT
- Show you how to get started with GitHub
- Learn how to set up your local (linux) system to work with git/github.
- Really it is an preparation presentation for our upcoming 'Create Free Websites with Jekyll and GitHub-Pages' presentation.

# What Should You Know?

- Truth is, not much!
- A basic understanding of the Command Line will get you through any issues.
- If you saw our 'Intro to the Command Line' presentation, you are set.

Lets dive in!

# What is Git?

- Git is a distributed revision control system.
- It allows many software developers to work on the same project without having to be in the same physical location.
- This takes a peer-to-peer approach on version control.

# What is Github?

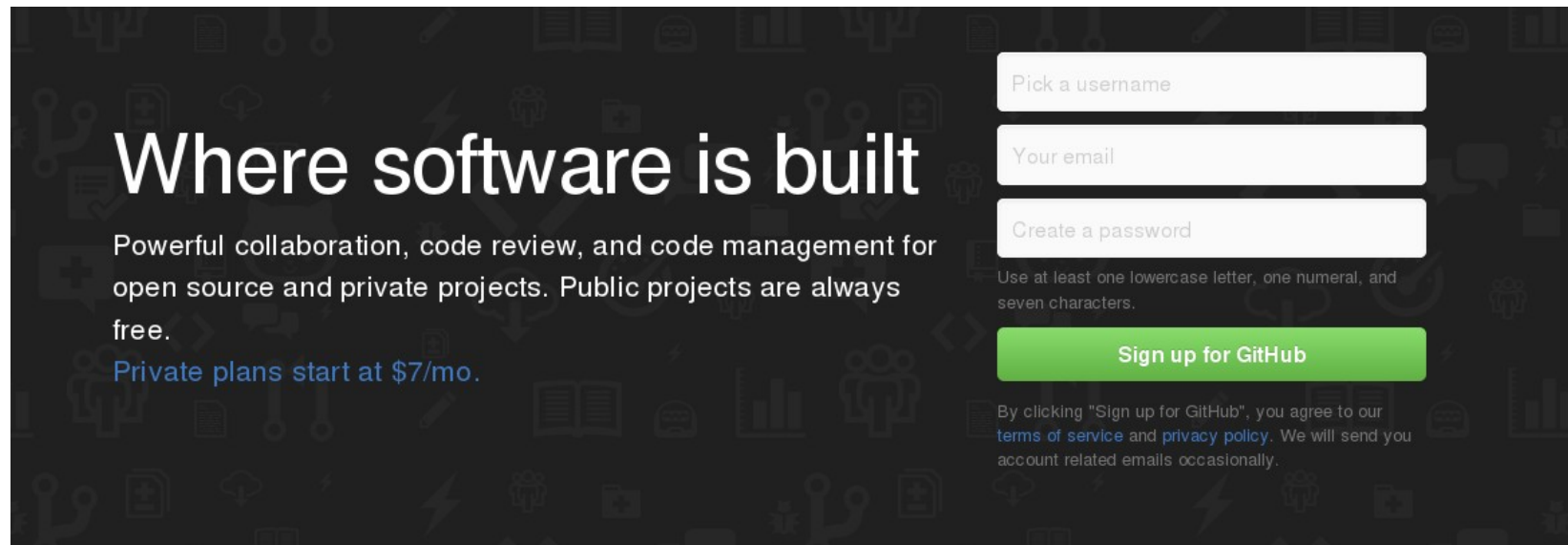
- GitHub is a website that implements the git system.
- It is a 'Web-based Git repository hosting service'

# Github Info

- Using git/github on Linux requires being comfortable with the command line.
- On OSX and Windows, there are graphical github clients that can be used.
- The command line interface is more powerful and should be used when possible (or comfortable).
- More info at [github.com](https://github.com)

# GitHub Setup

You can sign up for a free account at  
[www.github.com](https://www.github.com)

A screenshot of the GitHub sign-up page. The background is dark with a pattern of small, light-colored icons. On the left, the text 'Where software is built' is prominently displayed in white. Below it, a paragraph describes GitHub's features for collaboration, code review, and code management, noting that public projects are free. A link for private plans is also present. On the right, there is a sign-up form with three input fields: 'Pick a username', 'Your email', and 'Create a password'. Below the password field, a note specifies password requirements. A green 'Sign up for GitHub' button is positioned below the form. At the bottom right, a disclaimer states that by signing up, the user agrees to the terms of service and privacy policy, and that they will receive occasional emails.

**Where software is built**

Powerful collaboration, code review, and code management for open source and private projects. Public projects are always free.

Private plans start at \$7/mo.

Pick a username

Your email

Create a password

Use at least one lowercase letter, one numeral, and seven characters.

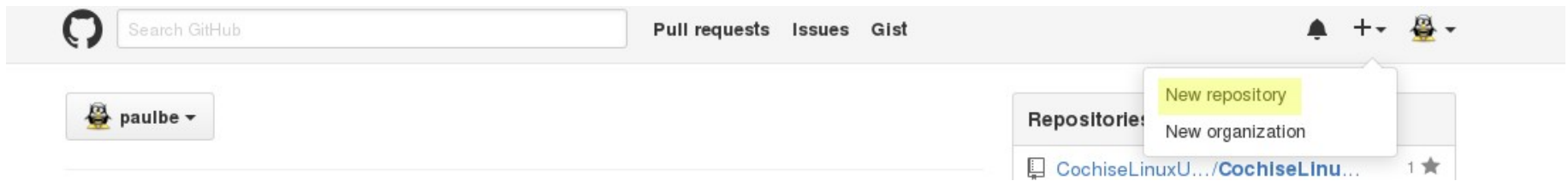
**Sign up for GitHub**

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy policy](#). We will send you account related emails occasionally.



# GitHub Setup (cont)


Once you have signed up / signed in, you can start by creating a repo.



# GitHub Setup (cont)



## Create a new repository

A repository contains all the files for your project, including the revision history.

Owner	Repository name
 paulbe ▾	NewRepo ✓

Great repository names are short and memorable. Need inspiration? How about **billious-tribble**.

Description (optional)

- ☒  **Public**  
Anyone can see this repository. You choose who can commit.
- ☐  **Private**  
You choose who can see and commit to this repository.

☒ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

Add a license: **None** ▾



Create repository

# Local Config

## Setting up Git

---

- 1 [Download and install the latest version of Git.](#)
- 2 On your computer, open your command line application.
- 3 Tell Git your *name* so your commits will be properly labeled. Type everything after the `$` here:

```
$ git config --global user.name "YOUR NAME"
```

- 4 Tell Git the *email address* that will be associated with your Git commits. The email you specify should be the same one found in your [email settings](#). To keep your email address hidden, see "[Keeping your email address private](#)".

```
$ git config --global user.email "YOUR EMAIL ADDRESS"
```

# Clone A Repo

The screenshot shows the GitHub interface for a repository named 'NewRepo' by user 'paulbe'. At the top, there's a search bar and navigation links for 'Pull requests', 'Issues', and 'Gist'. Below the repository name, there are buttons for 'Unwatch', 'Star', and 'Fork'. The 'Code' tab is selected, showing a list of files. The first file is 'README.md', which is the 'Initial commit'. A red arrow points to the 'HTTPS' clone link: 'https://github.com/paulbe'. Below the file list, the text 'Clone Repo Link Here' is displayed. The repository name 'NewRepo' is also visible in a large font at the bottom of the file list area.

This repository Search

Pull requests Issues Gist

paulbe / NewRepo

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Wiki Pulse Graphs Settings

No description or website provided. — Edit

1 commit 1 branch 0 releases 1 contributor

Branch: master New pull request

New file Find file HTTPS https://github.com/paulbe Download ZIP

paulbe Initial commit Latest commit e95d9b4 43 minutes ago

README.md Initial commit 43 minutes ago

README.md

Clone Repo Link Here

NewRepo



# Clone A Repo

- With the link from the previous page, we can now clone a repo on our local machine.
- `$ git clone <repo_clone_link>`

```
lame@lamebox github % git clone https://github.com/paulbe/NewRepo.git
Cloning into 'NewRepo'...
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.
lame@lamebox github % ls
NewRepo
lame@lamebox github %
```

# Local Changes

- After the clone we have a copy of the repo that is on GitHub.
- We can now make the changes we wish to make to the repo on our local machine.

# Local Changes

The screenshot shows the GitHub interface for a repository named 'NewRepo' by user 'paulbe'. At the top, there's a search bar and navigation links for 'Pull requests', 'Issues', and 'Gist'. Below the repository name, there are buttons for 'Unwatch' (1), 'Star' (0), and 'Fork' (0). A secondary navigation bar includes 'Code', 'Issues' (0), 'Pull requests' (0), 'Wiki', 'Pulse', 'Graphs', and 'Settings'. The main area shows the file 'README.md' being edited. The editor has tabs for 'Edit file' and 'Preview changes'. On the right, there are settings for 'Spaces' (set to 2) and 'Soft wrap'. The file content is visible in a light gray editor area.

GitHub repository page for **paulbe / NewRepo**. The interface shows the repository name, navigation links (Pull requests, Issues, Gist), and action buttons (Unwatch, Star, Fork). Below the repository name, there are tabs for Code, Issues, Pull requests, Wiki, Pulse, Graphs, and Settings. The main area displays the file **README.md** being edited. The editor interface includes a file name bar, a preview toggle, and settings for Spaces and Soft wrap. The file content is visible in the editor area.

```
README.md
1 # NewRepo
2
3 ##This is our test repo, we can add any files into this local directory and have them added to our repo.
4
```

# Update Remote Files

- Now we have made our changes, it is time to update your files to GitHub.
- This is the process I normally take.
  - `$ git status` (this will show any local changes that have not been committed)
  - `$ git add .` (this adds modified files in the current working tree, preparing them for the next commit)
  - `$ git commit -m "commit message"` (this commits any changes and sets a specified commit message)
  - `$ git push origin master` (this pushes your changes to the remote repo)



# Update Remote Files (cont)

```
lame@lamebox NewRepo % ls
README.md
lame@lamebox NewRepo % git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
lame@lamebox NewRepo % git add .
lame@lamebox NewRepo % git commit -m "updating readme"
[master d900791] updating readme
 1 file changed, 2 insertions(+)
lame@lamebox NewRepo % git push origin master
Username for 'https://github.com': paulbe
Password for 'https://paulbe@github.com':
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 339 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/paulbe/NewRepo.git
   e95d9b4..d900791  master -> master
lame@lamebox NewRepo %
```

No description or website provided. — Edit

📦 2 commits

🌿 1 branch

📦 0 releases

👤 1 contributor

Branch: master ▾

New pull request

New file

Find file

HTTPS ▾

https://github.com/paulbe



Download ZIP

🐧 paulbe updating readme

Latest commit d900791 14 minutes ago

📖 README.md

updating readme

14 minutes ago

📖 README.md

# NewRepo

This is our test repo, we can add any files into this local directory and have them added to our repo.

# That Is It!

- So using git/github is fairly simple, however it can become more complex.
- What I have shown you today is the most basic usage that will get you by.
- Remember we will have a presentation using this knowledge at our next meeting (at the library).

Questions / Comments

# CLUG Info

- Chat almost any time on IRC
  - Server: `irc.freenode.net`
  - Channel: `#cochiselinux`
  - If you need help getting an IRC client working / configured, feel free to ask.
- Website
  - `Cochiselinux.com`
- Mailing List
  - Directions to sign up for the mailing list can be found on the website.

Demo (time permitting)



431

Request Header Fields Too Large