



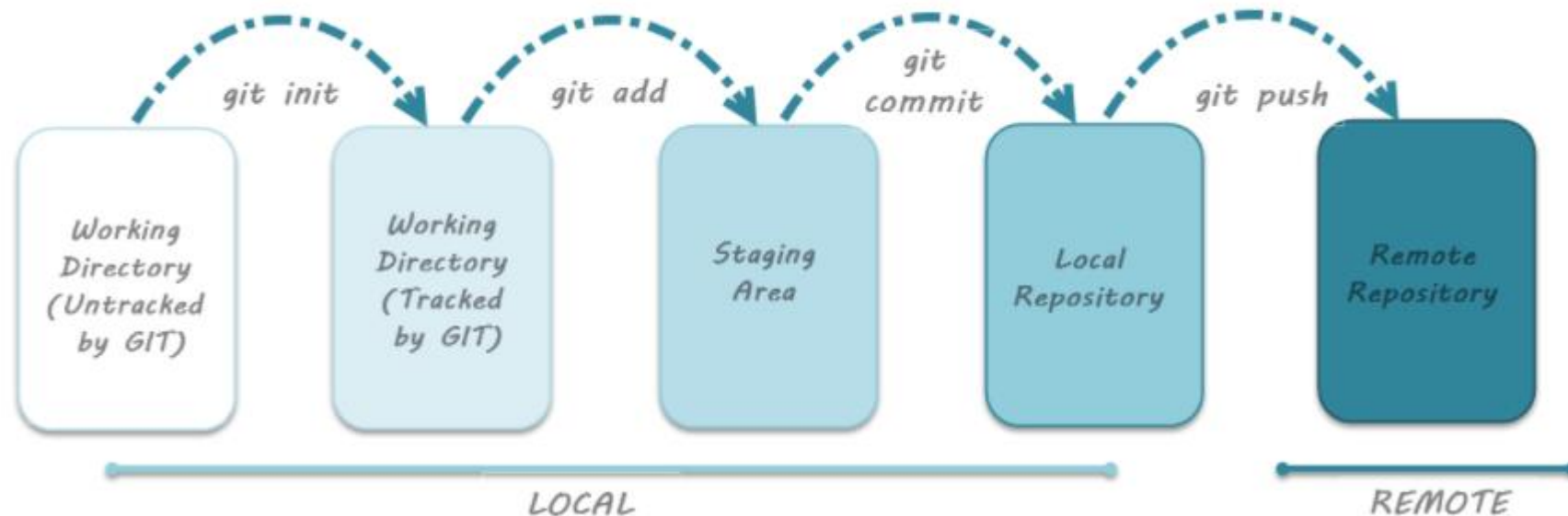
Working with Remote Repo - GitHub

Additional Lifecycle Stage with GitHub(Remote Repository)

We can track versions of your files by using only Git. That is, GitHub is needed when you want to collaborate and publish your code to a team or community

We work in teams and collaborate with multiple people on a given project. This makes it imperative to understand the additional stage related to GitHub. While dealing with GitHub, there's a concept of Remote repository and a related process called Pushing the files

Remote repository means mirror or clone of the local Git repository in GitHub. And pushing means uploading the commits from local Git repository to remote repository hosted in GitHub. This will allow other collaborators to view the code



Git Remote Repository

- Now let's add remote this repository using GitHub (A remote repo which allows multiple users to work)
 - Default branches name was *master* and the remote repositories name is *origin*
-
- Go to github.
 - Log in to your account.
 - Click the new repository button in the top-right. You'll have an option there to initialize the repository with a README file,
 - Click the "Create repository" button.

Creating a personal access token

You should create a personal access token to use in place of a password with the command line or with the API.

Creating a token Using a token on the command line

Personal access tokens (PATs) are an alternative to using passwords for authentication to GitHub when using the GitHub API or the command line.

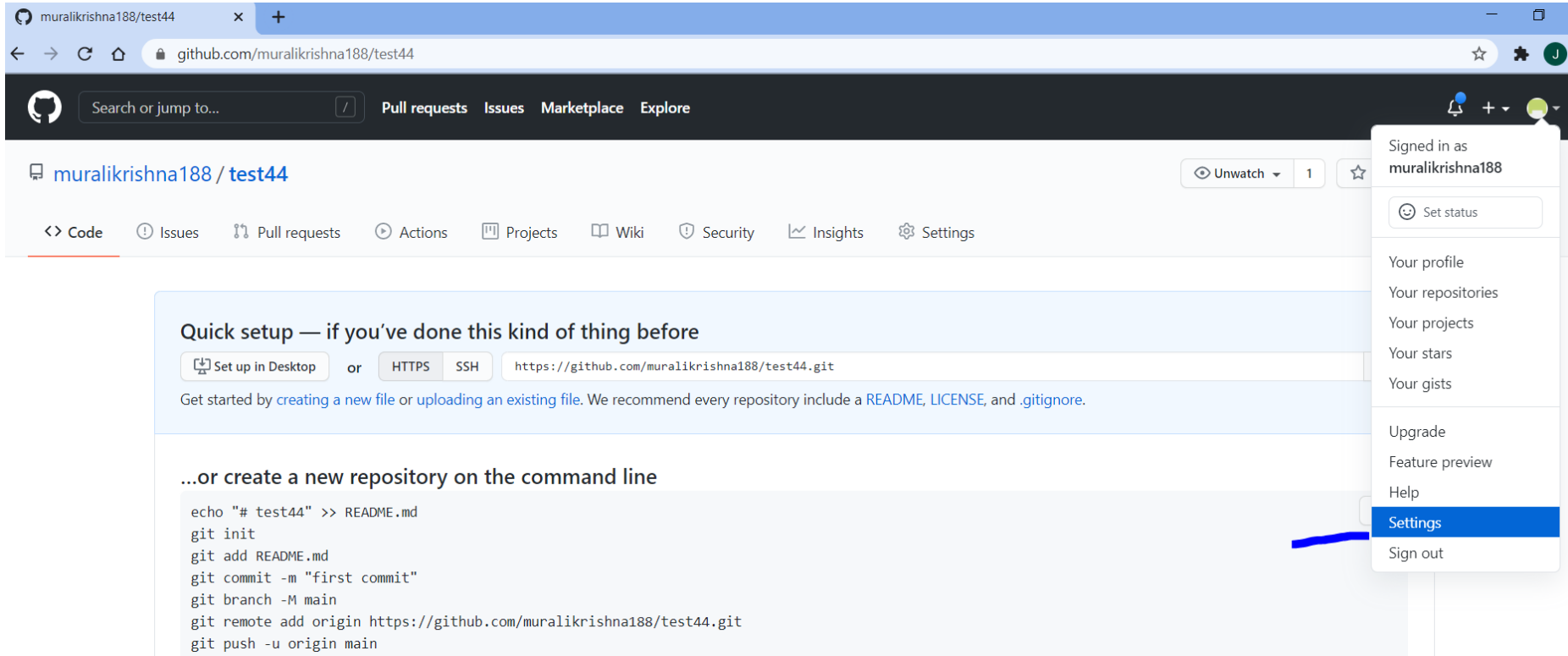
If you want to use a PAT to access resources owned by an organization that uses SAML SSO, you must authorize the PAT.

As a security precaution, GitHub automatically removes personal access tokens that haven't been used in a year.

Creating a token

Verify your email address, if it hasn't been verified yet.

In the upper-right corner of any page, click your profile photo, then click Settings.



The screenshot shows the GitHub interface for a repository named 'test44' by user 'muralikrishna188'. The top navigation bar includes a search bar and links for Pull requests, Issues, Marketplace, and Explore. The repository page header shows the repository name and a 'Unwatch' button. Below the header, a navigation bar contains links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The main content area displays a 'Quick setup' section with options to set up on a desktop or via HTTPS/SSH, and a section for creating a new repository on the command line with a list of git commands. A user menu is open in the top right corner, showing the user is signed in as 'muralikrishna188'. The menu options include 'Set status', 'Your profile', 'Your repositories', 'Your projects', 'Your stars', 'Your gists', 'Upgrade', 'Feature preview', 'Help', 'Settings' (highlighted with a blue bar), and 'Sign out'.

Quick setup — if you've done this kind of thing before

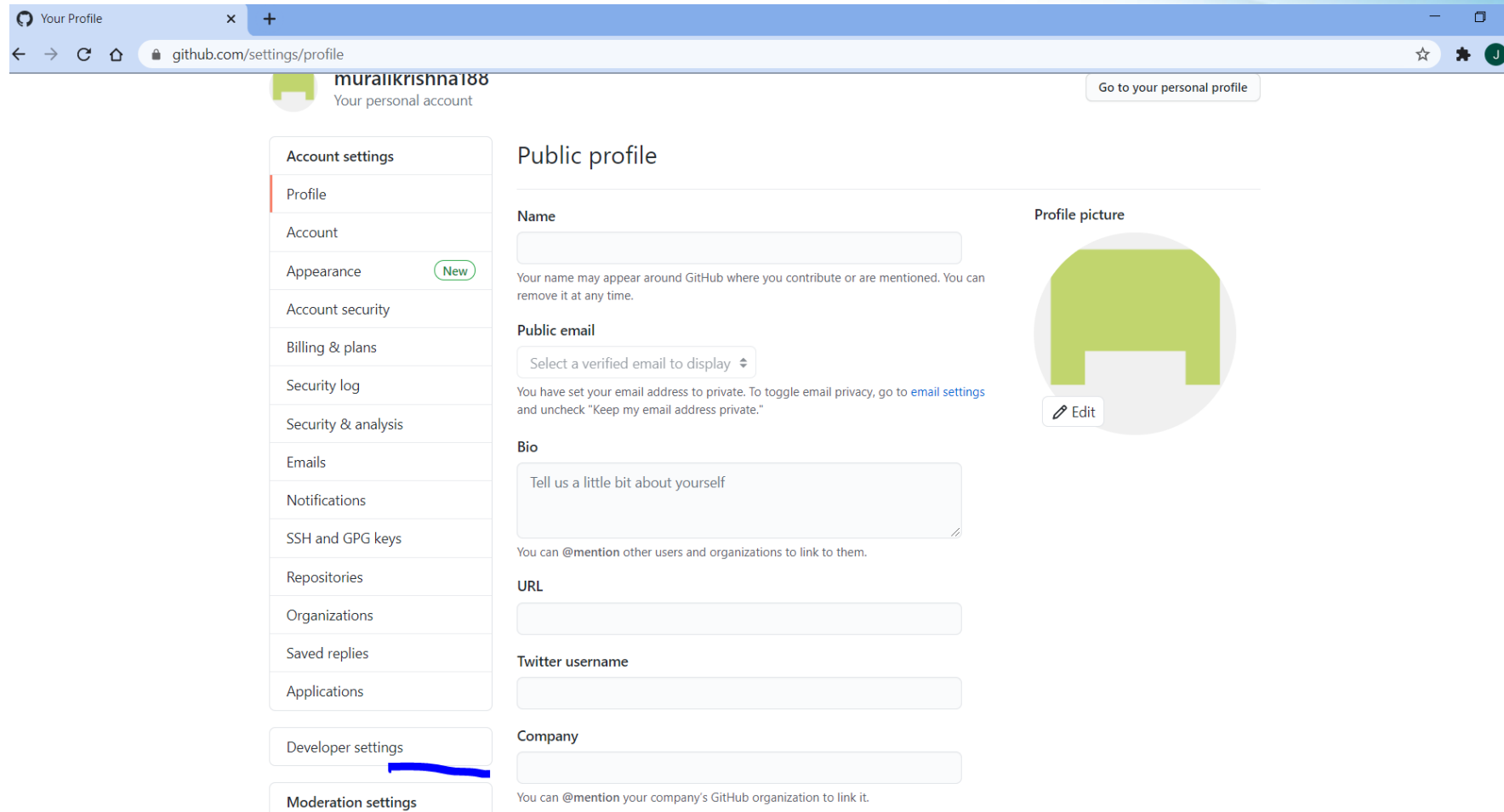
or

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# test44" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/muralikrishna188/test44.git
git push -u origin main
```

In the left sidebar, click **Developer settings**.

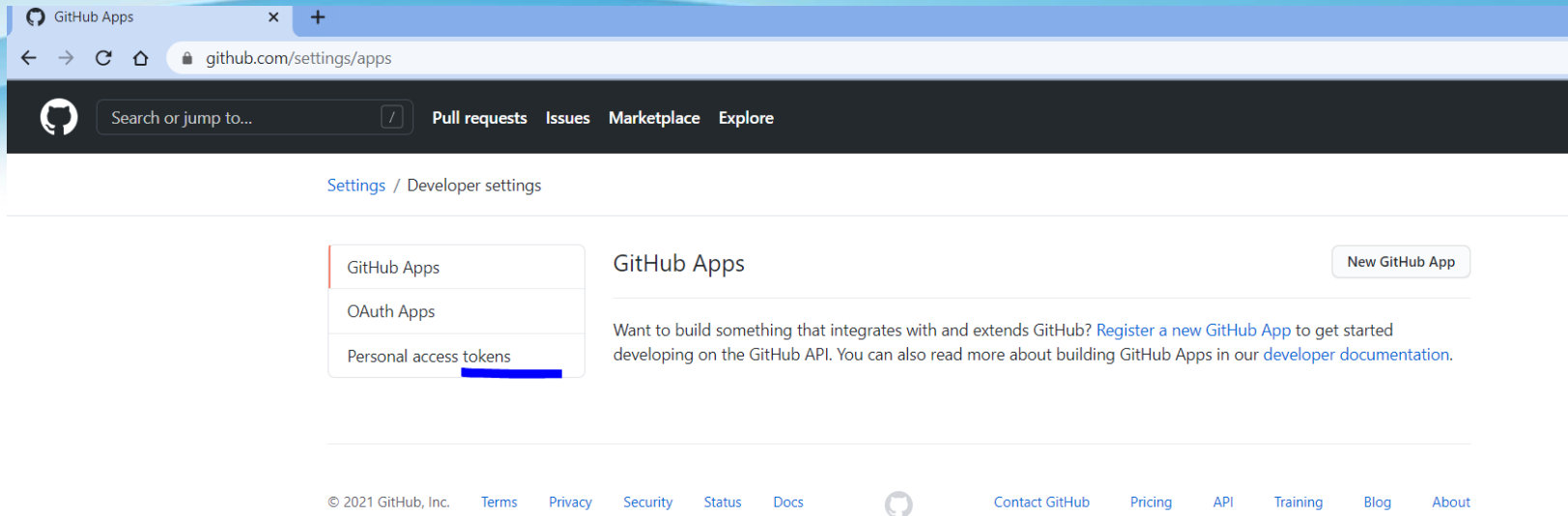


The screenshot shows the GitHub profile settings page for the user 'muralikrishna188'. The browser address bar displays 'github.com/settings/profile'. The left sidebar contains a list of settings categories: Account settings, Profile, Account, Appearance (marked 'New'), Account security, Billing & plans, Security log, Security & analysis, Emails, Notifications, SSH and GPG keys, Repositories, Organizations, Saved replies, Applications, **Developer settings** (highlighted with a blue underline), and Moderation settings. The main content area is titled 'Public profile' and includes the following sections:

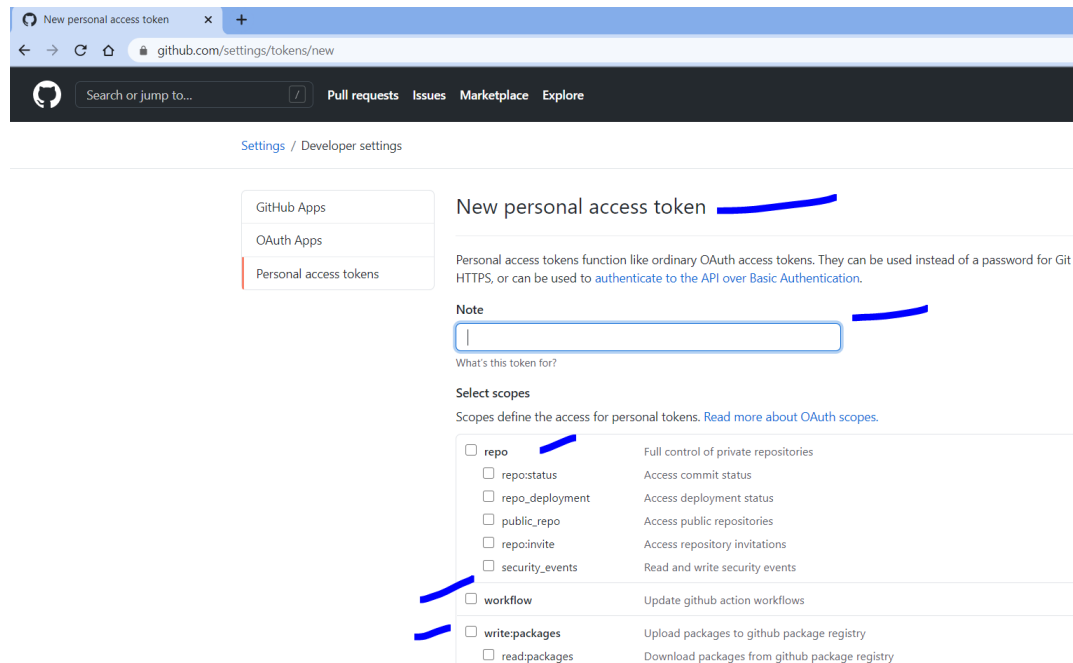
- Name**: A text input field. Below it, a note states: 'Your name may appear around GitHub where you contribute or are mentioned. You can remove it at any time.'
- Public email**: A dropdown menu with the text 'Select a verified email to display'. Below it, a note states: 'You have set your email address to private. To toggle email privacy, go to [email settings](#) and uncheck "Keep my email address private."'
- Bio**: A text area with the placeholder text 'Tell us a little bit about yourself'. Below it, a note states: 'You can @mention other users and organizations to link to them.'
- URL**: A text input field.
- Twitter username**: A text input field.
- Company**: A text input field. Below it, a note states: 'You can @mention your company's GitHub organization to link it.'

On the right side of the 'Public profile' section, there is a 'Profile picture' placeholder (a green square) and an 'Edit' button.

In the left sidebar, click **Personal access tokens**.



Click **Generate new token** and Select the scopes, or permissions, you'd like to grant this token. To use your token to access repositories from the command line, select **repo**.



Click **Generate token**. Click to copy the token to your clipboard. For security reasons, after you navigate off the page, you will not be able to see the token again.

<input type="checkbox"/> user:follow	Follow and unfollow users
<input type="checkbox"/> delete_repo	Delete repositories
<input type="checkbox"/> write:discussion	Read and write team discussions
<input type="checkbox"/> read:discussion	Read team discussions
<input checked="" type="checkbox"/> admin:enterprise	Full control of enterprises
<input checked="" type="checkbox"/> manage_billing:enterprise	Read and write enterprise billing data
<input checked="" type="checkbox"/> read:enterprise	Read enterprise profile data
<input checked="" type="checkbox"/> admin:pgp_key	Full control of public user pgp keys (Developer Preview)
<input checked="" type="checkbox"/> write:pgp_key	Write public user pgp keys
<input checked="" type="checkbox"/> read:pgp_key	Read public user pgp keys

Generate token[Cancel](#)

To use your token to authenticate to an organization that uses SAML SSO, authorize the token for use with a SAML single-sign-on organization.

Once you have a token, you can enter it instead of your password when performing Git operations over HTTPS.

For example, on the command line you would enter the following:

```
$ git clone https://github.com/username/repo.git
```

```
Username: your_username
```

```
Password: your_token
```

Or

```
git push -u origin master
```

- First time configuration

```
git config --global user.name "<your user name>"
```

```
git config --global user.email "<your email>"
```

```
mural@kubekon MINGW64 ~  
$ git config --global user.name "muralid"
```

```
$ git config --global user.email "reach2muralikrishnad@gmail.com"
```

Now, follow the second set of instructions, “Push an existing repository...”

```
$ git remote add origin <REMOTE_URL>  
$ git remote add origin https://github.com/username/new_repo
```

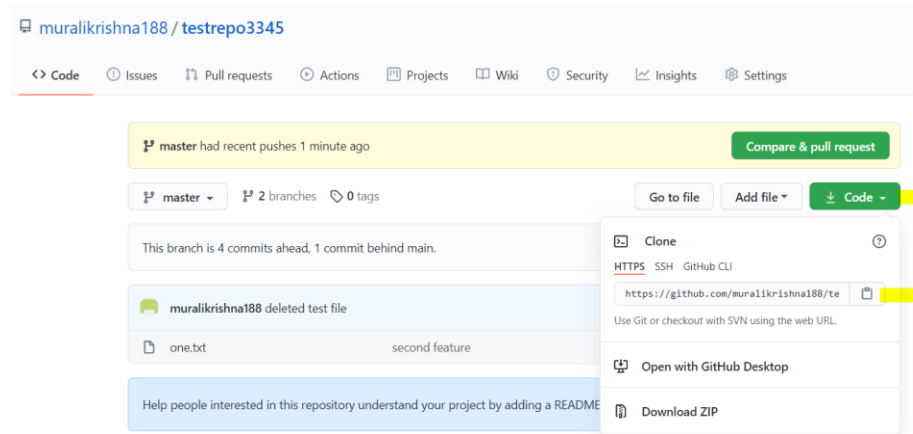
```
# Sets the new remote  
$ git remote -v  
# Verifies the new remote URL
```

```
$ git push -u origin master
```

```
mural@kubekon MINGW64 /e/HRMS (uber)  
$ git push -f origin master  
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0  
To https://github.com/muralikrishna188/HRMS-practice.git  
+ a338c5b...befe3a5 master -> master (forced update)  
  
mural@kubekon MINGW64 /e/HRMS (uber)  
$ git push --all origin  
Everything up-to-date
```

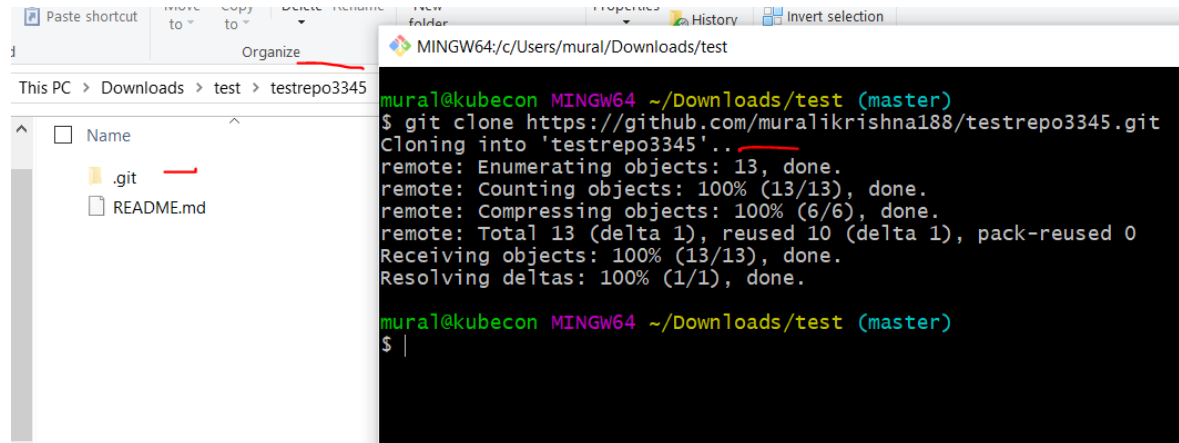
Cloning a repository

1. On GitHub, navigate to the main page of the repository.
2. Above the list of files, click **Code**.

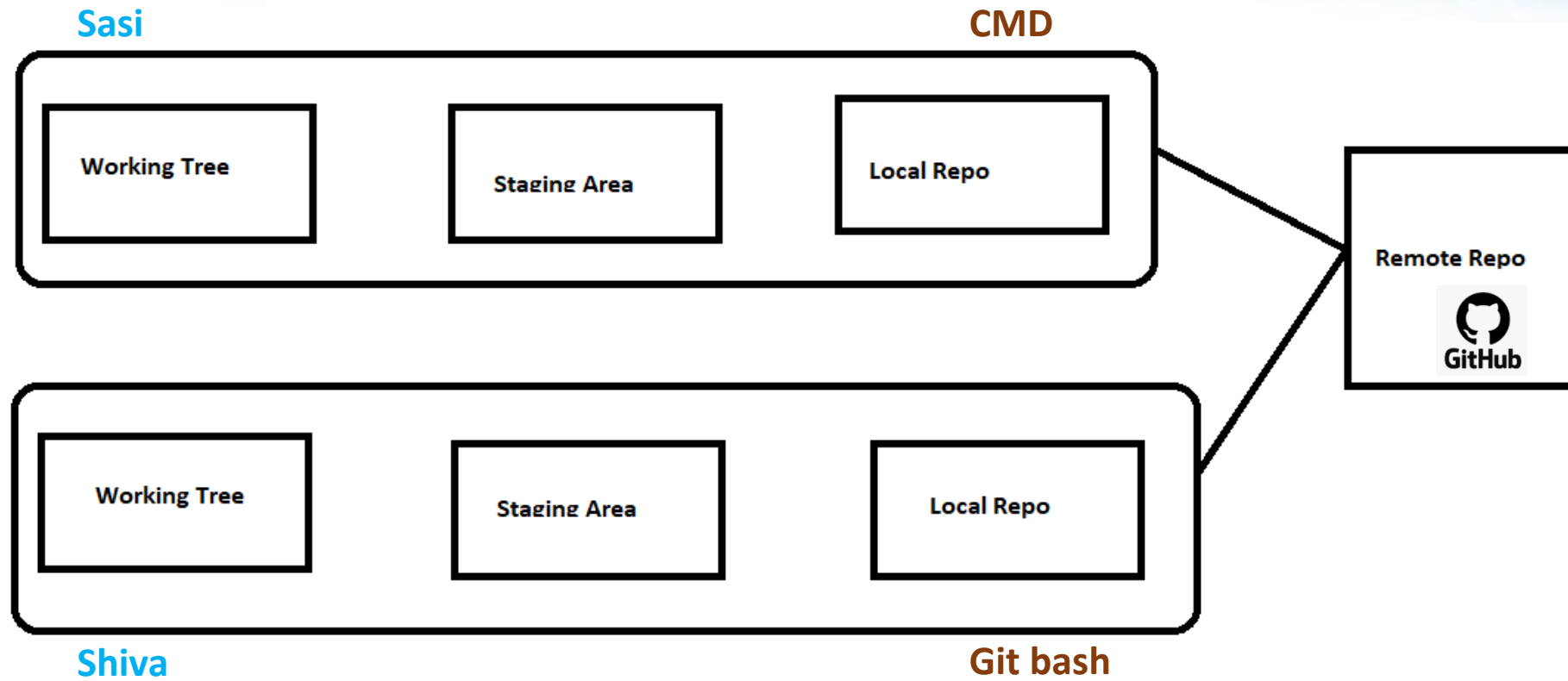


3. Open Git Bash.
4. Change the current working directory to the location where you want the cloned directory.
5. Type git clone, and then paste the URL you copied earlier.

\$ git clone https://github.com/YOUR-USERNAME/YOUR-REPOSITORY



Two users are working with GitHub

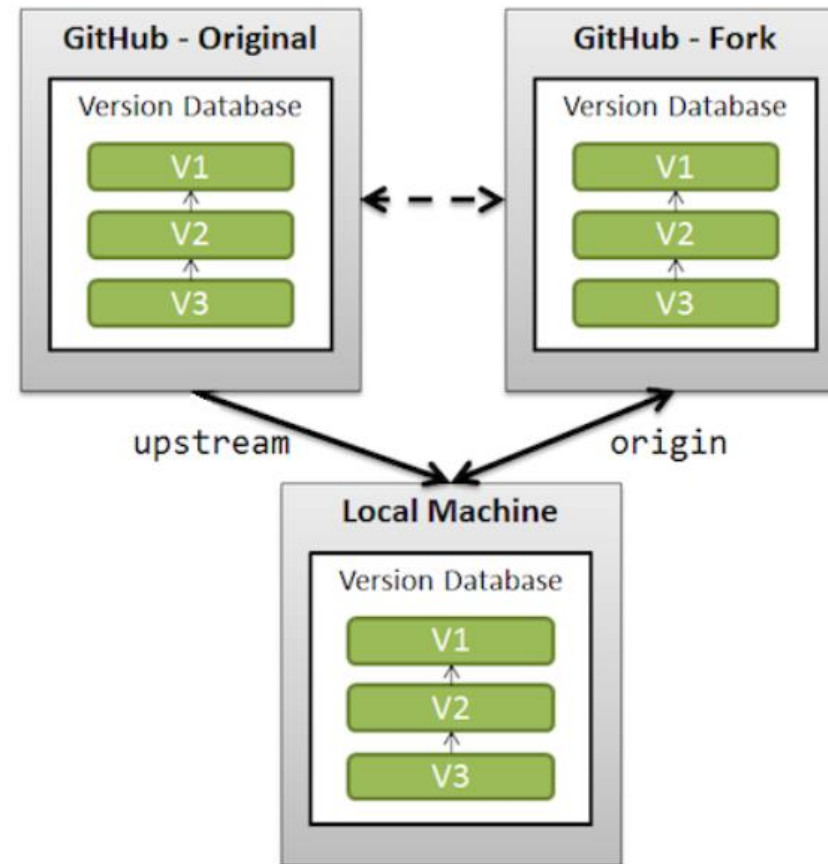


upstream generally refers to the original repo that you have forked

origin is your fork: your own repo on GitHub, clone of the original repo of GitHub

When a repo is cloned, it has a default remote called **origin** that points to your fork on GitHub, not the original repo it was forked from.

To keep track of the original repo, you need to add another remote named **upstream**



fork

Normally used as a noun, when referring to a copy of a main GitHub repository. In practice, a fork is just another repository. But it's special in the sense that GitHub maintains a connection back to the main/parent repository. It's sometimes used as a verb, as in "You must fork the repository first."

remote

A named connection to a remote repository, such as the "origin" or "upstream" remote. Git refers to this as remote because it is used to reference a repository that's hosted on another computer. In this workflow, a remote is always a GitHub repository.

origin

The name assigned to the connection between your local repository and the repository from which it was cloned. In this workflow, origin represents the connection to your fork. It's sometimes used as a moniker for the origin repository itself, as in "Remember to push your changes to origin."

upstream

Like the origin remote, upstream is a named connection to another repository. In this workflow, upstream represents the connection between your local repository and the main repository, from which your fork was created. It's sometimes used as a moniker for the upstream repository itself, as in "Remember to pull the changes from upstream."

muralikrishna188 / Demo

forked from kedark3/Demo

<> Code

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

master 3 branches 0 tags

This branch is even with kedark3:master.

kedark3 Merge pull request kedark3#91 from mandar242/new_branch

README.md	Updated README
test	Adding a test file to new_branch
test1	Adding edit to test1 file

README.md

#Demo Added another line to README.md

Go to file

Add file

Code

Clone

HTTPS SSH GitHub CLI

https://github.com/muralikrishna188/De

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

About

Demo repo for tutotial articles on Opensource.com

Readme

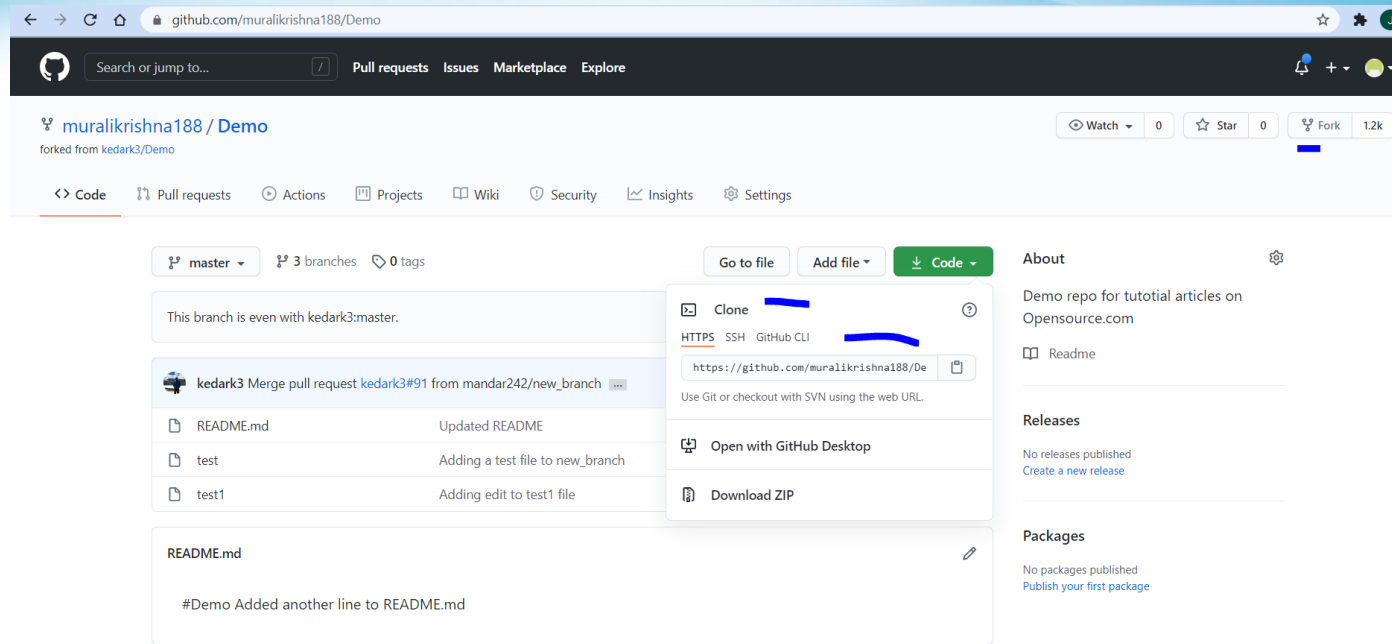
Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

The copy includes all the code, branches, and commits from the original repo.
Next, clone the repo by opening the terminal on your computer and running the command:



```
git clone https://github.com/<YourUserName>/demo
```

```
mural@kubcon MINGW64 /d/azuregit
$ git clone https://github.com/muralikrishna188/Demo.git
Cloning into 'Demo'...
remote: Enumerating objects: 25, done.
remote: Total 25 (delta 0), reused 0 (delta 0), pack-reused 25
Receiving objects: 100% (25/25), 5.06 KiB | 1.27 MiB/s, done.
Resolving deltas: 100% (1/1), done.
```

Open the cloned folder

```
mural@kubekon MINGW64 /d/azuregit  
$ ls  
Demo/  reference-architectures/  
  
mural@kubekon MINGW64 /d/azuregit  
$ cd Demo/  
  
mural@kubekon MINGW64 /d/azuregit/Demo (master)  
$ |
```

Once the repo is cloned, you need to do two things:

1. Create a new branch by issuing the command:

```
git checkout -b feature1
```

```
mural@kubekon MINGW64 /d/azuregit/Demo (master)
$ git checkout -b feature1
Switched to a new branch 'feature1'

mural@kubekon MINGW64 /d/azuregit/Demo (feature1)
$ |

mural@kubekon MINGW64 /d/azuregit/Demo (feature1)
$ touch test.py

mural@kubekon MINGW64 /d/azuregit/Demo (feature1)
$ git add .

mural@kubekon MINGW64 /d/azuregit/Demo (feature1)
$ git commit -m "first commit"
[feature1 5753694] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 test.py
```

2. Create a new remote for the upstream repo with the command:

```
git remote add upstream https://github.com/muralikrishna188/Demo
```

```
mural@kubekon MINGW64 /d/azuregit/Demo (feature1)
$ git remote add upstream https://github.com/muralikrishna188/Demo
mural@kubekon MINGW64 /d/azuregit/Demo (feature1)
$ |
```


```
git push -u origin feature1
```

```
mural@kubecore MINGW64 /d/azuregit/Demo (feature1)
$ git add .
warning: LF will be replaced by CRLF in test.py.
The file will have its original line endings in your working directory

mural@kubecore MINGW64 /d/azuregit/Demo (feature1)
$ git commit -m "3rd commit"
[feature1 178141c] 3rd commit
1 file changed, 1 insertion(+)

mural@kubecore MINGW64 /d/azuregit/Demo (feature1)
$ git push -u origin feature1
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 342 bytes | 342.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/muralikrishna188/Demo.git
    88b3b4e..178141c feature1 -> feature1
Branch 'feature1' set up to track remote branch 'feature1' from 'origin'.
```

Select your branch in the GitHub Repo

 muralikrishna188 / Demo

forked from kedark3/Demo

Watch

0

☆

<> Code

🔗 Pull requests

🎬 Actions

📁 Projects

📖 Wiki

🛡 Security

📊 Insights

⚙ Settings

🔗 master

🔗 4 branches

🏷 0 tags

Go to file

Add file

📄 Code

Switch branches/tags

Find or create a branch...

Branches

Tags

✓ master default

feature1

myBranch2

new_branch

View all branches

from mandar242/new_branch

73ffc54 on Mar 27, 2020

🕒 8 commits

Updated README

3 years ago

Adding a test file to new_branch

11 months ago

Adding edit to test1 file

11 months ago

README.md

#Demo Added another line to README.md

About

Demo repo for tutotial articl

Opensource.com

📖 Readme

Releases

No releases published


Create a new release

Packages

No packages published

Publish your first package

Once you push the changes to your repo, the Compare & pull request button will appear in GitHub.

 muralikrishna188 / Demo

forked from kedark3/Demo

Watch

0

Star

<> Code

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

feature1

4 branches

0 tags

Go to file

Add file

Code

This branch is 4 commits ahead of kedark3:master.

Pull request

Compare

muralikrishna188 3rd commit

178141c 5 minutes ago

12 commits

README.md	Updated README	3 years ago
test	Adding a test file to new_branch	11 months ago
test.py	3rd commit	5 minutes ago
test1	Adding edit to test1 file	11 months ago

README.md

#Demo Added another line to README.md

About

Demo repo for tutotial articles o Opensource.com

Readme

Releases

No releases published

[Create a new release](#)

Packages


No packages published

[Publish your first package](#)

Click it and you'll be taken to this screen:

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).



base repository: kedark3/Demo ▾


base: master ▾

←

head repository: muralikrishna188/Demo ▾

compare: feature1 ▾

✓ **Able to merge.** These branches can be automatically merged.




Feature1


Write

Preview

H B I ≡ <> 🔗 ≡ ≡ ☑ @ ↻ ↶

Leave a comment


Attach files by dragging & dropping, selecting or pasting them. 

☒ Allow edits by maintainers 

Create pull request ▾

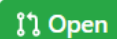
Helpful resources

[GitHub Community Guidelines](#)

 Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

You can also review the code changes

Feature1 #604

[Edit](#)[Open with ▾](#)[Open](#)

muralikrishna188 wants to merge 4 commits into [kedark3:master](#) from [muralikrishna188:feature1](#)

[Conversation](#) 0[Commits](#) 4[Checks](#) 0[Files changed](#) 1[+4 -0](#) [Changes from all commits ▾](#)[File filter... ▾](#)[Jump to... ▾](#)[0 / 1 files viewed](#)[Review changes ▾](#)[▼](#) 4 test.py [...](#) [...](#) @@ -0,0 +1,4 @@

1 + line1

2 + line2

3 + 777

4 + 888

ProTip! Use [n](#) and [p](#) to na**Finish your review** [Write](#)[Preview](#)[H](#)[B](#)[I](#)[≡](#)[<>](#)[🔗](#)[≡](#)[≡](#)[☑](#)[@](#)[🔗](#)[↶](#)

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

☒ Comment