Learn Git & Github :

Git - Git is a version control system used for tracking changes in computer files. It helps to track changes in code lines, whether it has changed or not, which line of code has been removed or added etc.

To understand git, you have to understand two main areas in git -

1. Staging area - In the staging area, you add those files you want to track. After changes, you commit those files.
   ```
   $ git add <file>
   ```
2. Commit area - It's a checkpoint. We remove files from staging area and add to commit area. It means, we have made a checkpoint. We can undo those files using checkpoint.
   ```
   $ git commit -m “hello.java added”
   ```
   OR
   ```
   $ git commit -m “initial commit”
   ```

Why use Git?

- Undo mistakes
- Different branches
- Community support

How to use Git?

Before using git, You have to install Git in your machine from google. And You can also provide the author of git using these commands.

```
$ git config --global user.name YOUR_NAME
$ git config --global user.email “***@gmail.com”
```

To know if user name and email has been set -

```
$ git config --list
```

Step 1. Go to project directory in command prompt

```
$ cd ProjectName/
```

Step 2. Initialize git in your project

```
$ git init
```

Step 3. For adding file to staging area

```
$ git add <file>
```

And for removing file from staging area

```
$ git rm --cached <file>
```

Step 4. For commit files

```
$ git commit -m “initial commit”
```

Step 5. To know which files in the staging area or which files have to be committed.

```
$ git status
```

Step 6. To know all commits,

```
$ git log
```

Branches in git

Branch - In simple words, Branch is a line of commits.
Master branch - In simple words, Master branch points to the last commit you made. Every time you commit, the master branch pointer moves forward automatically. Master branch is a clean bug free branch.

Why do we use branches in git?

It’s a sign of a professional developer that you are making a new branch before starting working on any project.
In case, You don’t make a new branch before starting working on any project, you are committing changes to the actual project. It makes things very hard to undo and it’s very important to make sure your master branch must be clean and bug free. Especially, while working with a team, it's very important, every developer has made a new branch before start. Then, it will not affect the real project.
In case, You make a new branch before working on any project, You make some changes like adding some files, removing some code lines etc. Actually, you are doing it in your own branch, not in the master branch. In future, if you wanna undo all those changes you made, you can. You can leave or delete your branch, if you don’t like it, and move to the master

branch where you find only committed changes. Later, if your branch is ready to use or bugs free, you can commit changes to the master branch.

Using Branches in git

For create a new branch

```
$ git branch BRANCH_NAME
```

For getting current branch

```
$ git branch --show-current
```

For moving to a branch

```
$ git checkout BRANCH_NAME
```

Note - Creating a new branch and moving to that branch can be combine in one command -

```
$ git checkout -b BRANCH_NAME
```

For moving to master branch

```
$ git checkout master
```

For commit changes in current branch

```
$ git commit -m “any message”
```

For merge a branch to master branch

Firstly, move to the master branch, because you want to merge(combine or add) your branch into the master branch. You can merge your branch into any branch.

```
$ git checkout master
$ git merge BRANCH_NAME
```

Now, you will see changes from your branch in the master branch.

Error : In case, Sometimes when switching to another branch and trying to add file using $ git add <file>, it is unable to add files.

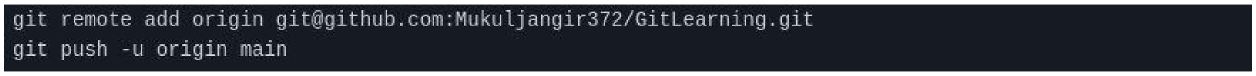
To solve it - Modify that file and save and try to add again.

Git with Github

Here, we make a new repo in github profile, and will push our project to github.

Step 1. Make a repo

Go to github site, click on new Repo, name your repo and click ok.
Step 2. Push project to github profile
When you create a new repo on github, github recommends you some commands, Use these commands to push your project on github.



```
git remote add origin git@github.com:Mukuljangir372/GitLearning.git
git push -u origin main
```

1. $ git remote add origin git@github.com:Mukuljangir372/GitLearning.git
2. $ git push -u origin master
   OR $ git push origin BRANCH_NAME

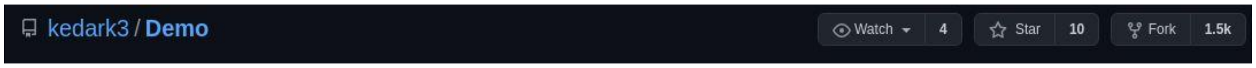
Understanding these commands,
In first command, we remotely accessed the remote repository(it's the same repository that all team members access). Here, origin points to our repo url. We can change the `origin` name to anything we want.
In second command, we're pushing our local repo with the main branch.
For pushing local repo to another branch,
$ git push origin BRANCH_NAME

Pull requests in github
OR
Contribute to open source project

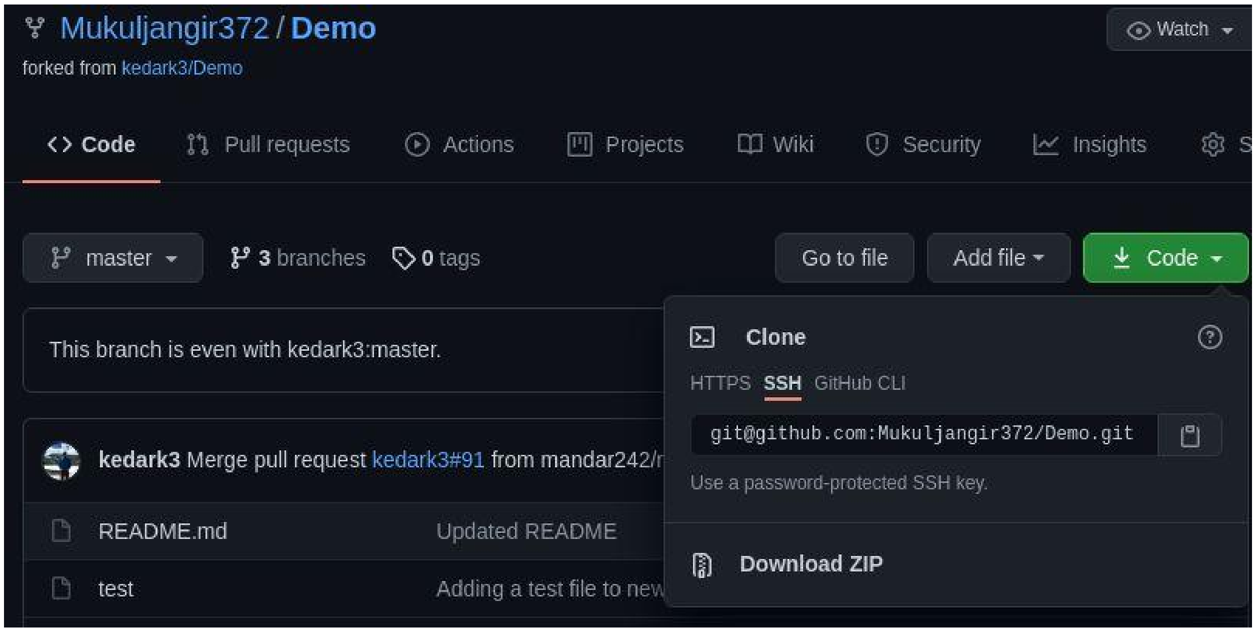
For instance, you want to contribute to an open source project on github. You clone that repo and make a push request to remote repo. And That's all it is.
click here to see all process

Error : Authentication related error, To solve the error, Go to video at last.
Error : Private email address error, To solve the error, Go to here.

Step 1. Fork project



Step 2. Go to forked project on your github profile, Click on Download Code green button > SSH > copy url



This branch is even with kedark3:master.

kedark3 Merge pull request kedark3#91 from mandar242/r

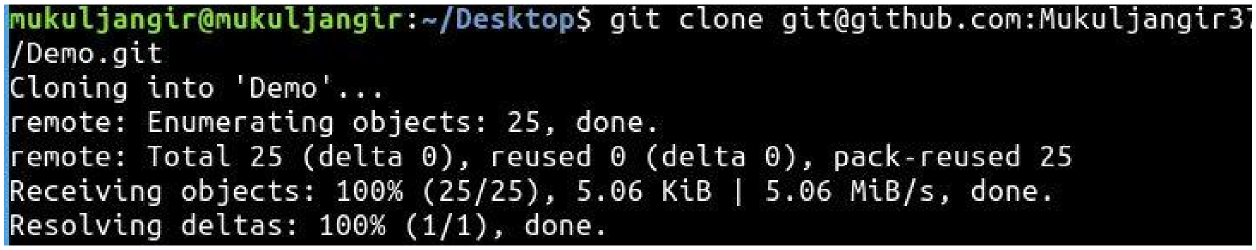
README.md Updated README

test Adding a test file to new

```
git@github.com:Mukuljangir372/Demo.git
```

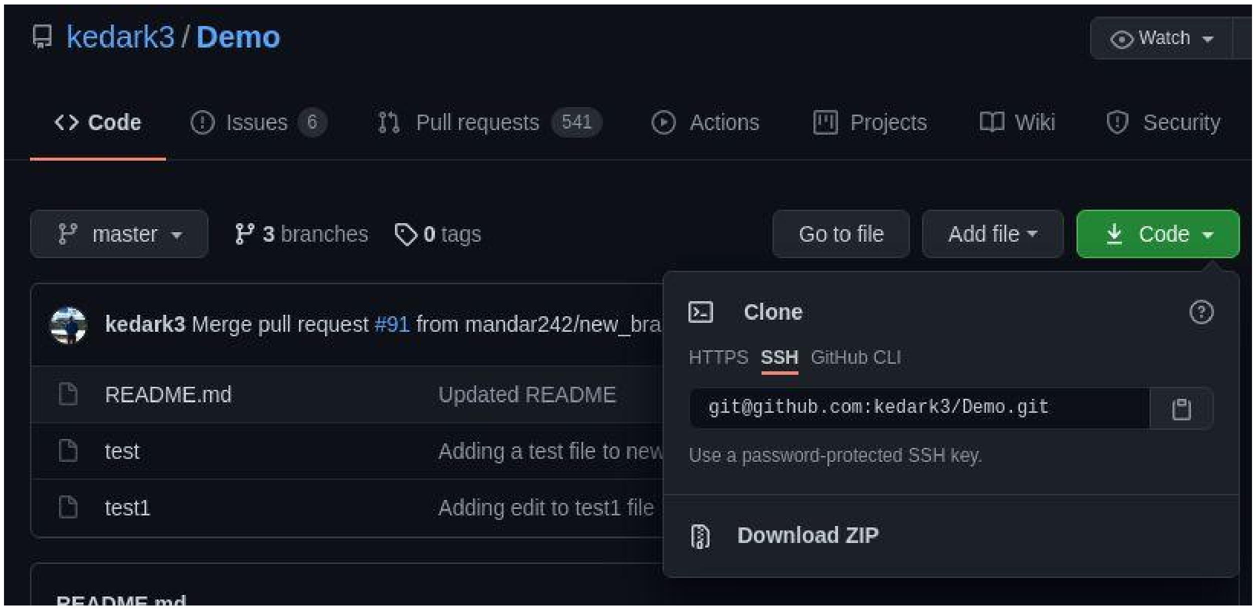
Use a password-protected SSH key.

Step 3. Go to terminal > $ git clone COPIED_URL



```
mukuljangir@mukuljangir:~/Desktop$ git clone git@github.com:Mukuljangir37
/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 | 5.06 MiB/s, done.
Resolving deltas: 100% (1/1), done.
```

Step 4. Go to real repo on github > Click on Download Code green button > SSH > copy url



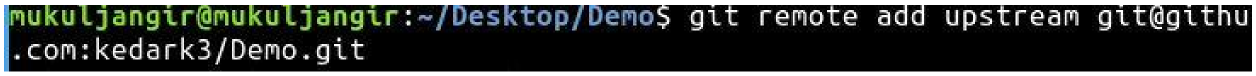
kedark3 Merge pull request #91 from mandar242/new_bra

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

```
git@github.com:kedark3/Demo.git
```

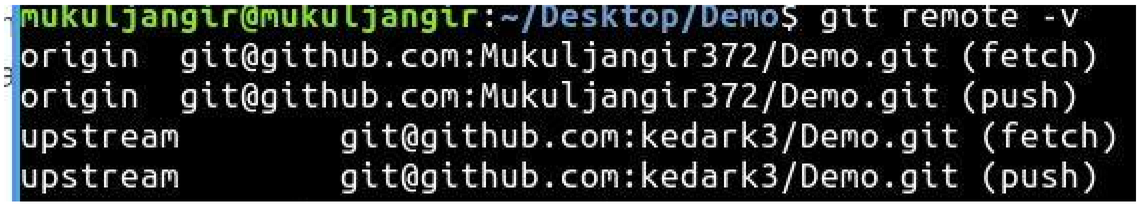
Use a password-protected SSH key.

Step 5. Go to terminal > $ git remote add upstream COPIED_URL



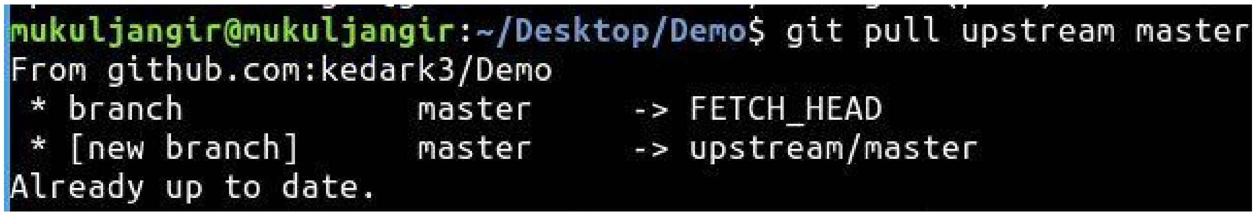
```
mukuljangir@mukuljangir:~/Desktop/Demo$ git remote add upstream git@githu
.com:kedark3/Demo.git
```

Step 6. Check available remotes using this command > $ git remote -v



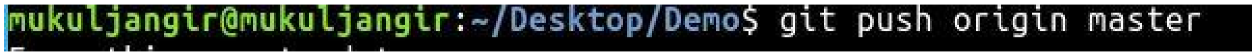
```
mukuljangir@mukuljangir:~/Desktop/Demo$ git remote -v
origin  git@github.com:Mukuljangir372/Demo.git (fetch)
origin  git@github.com:Mukuljangir372/Demo.git (push)
upstream        git@github.com:kedark3/Demo.git (fetch)
upstream        git@github.com:kedark3/Demo.git (push)
```

Step . Get all changes from remote repo



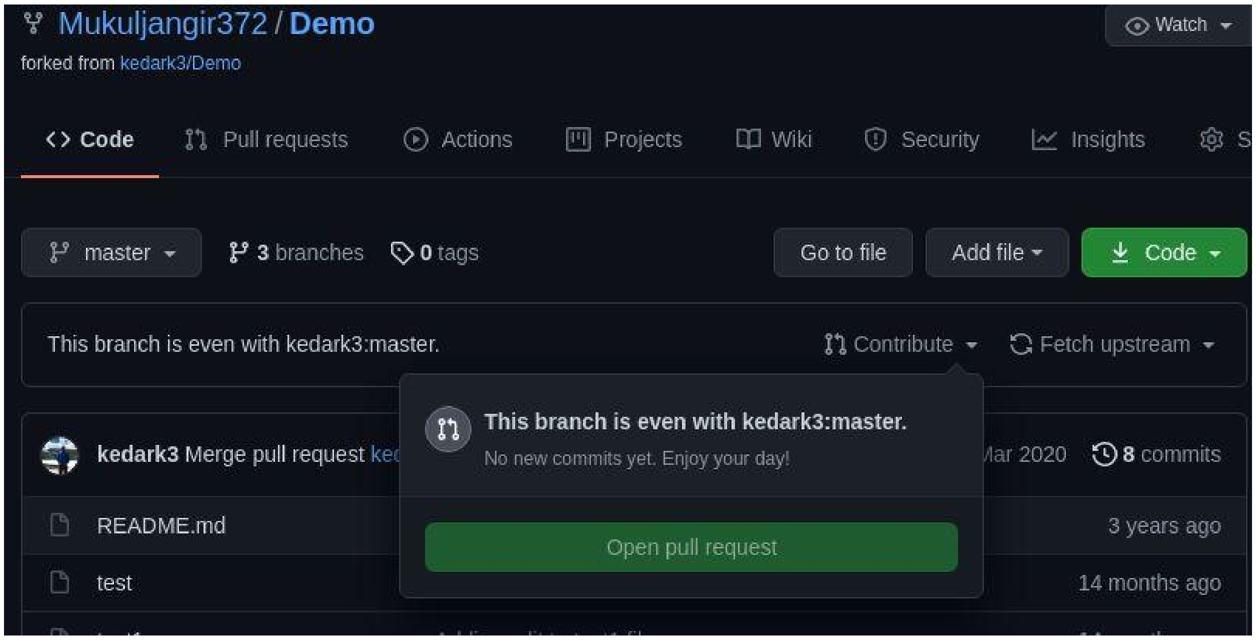
```
mukuljangir@mukuljangir:~/Desktop/Demo$ git pull upstream master
From github.com:kedark3/Demo
 * branch            master     -> FETCH_HEAD
 * [new branch]      master     -> upstream/master
Already up to date.
```

Step 8. Create a new branch, add files to the staging area, commit those files and finally push it.



```
mukuljangir@mukuljangir:~/Desktop/Demo$ git push origin master
```

Step 9. Go to the forked repository on your github profile, Click on contribute, then click on open pull requests.



This branch is even with kedark3:master.

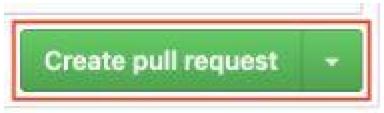
This branch is even with kedark3:master.
No new commits yet. Enjoy your day!

kedark3 Merge pull request ked … Mar 2020 8 commits

README.md 3 years ago

test 14 months ago

Step 10. Then create a new pull request and Go to the real repo on github to see your new pull request.



Additional

Fetching changes from Remote repo

You can fetch changes to local repo from remote repo
$ git fetch REMOTE_NAME
Example - $ git fetch origin
$ git fetch REPO_NAME

Merge those changes into local repo

After fetching changes from remote repo, you have to merge it in your branch.

```
$ git merge REMOTE_NAME/BRANCH_NAME
```

You can fetch changes and merge changes into your local repo in a single command.

Use below command,

```
$ git pull REMOTE_NAME BRANCH_NAME
```