

PROGRAM-1

Deploy to GitHub via Git in Ubuntu #1

sudo – superuser privileges

apt-get – advanced package tools to install, update, upgrade and remove packages

```
sriharsha@DESKTOP-001CA4F:~$ mkdir project
sriharsha@DESKTOP-001CA4F:~$ cd project
sriharsha@DESKTOP-001CA4F:~/project$ sudo apt-get update
[sudo] password for sriharsha:
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:2 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1035 kB]
Hit:5 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1246 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [197 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1205 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [195 kB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [815 kB]
Get:11 http://archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [257 kB]
Get:12 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1018 kB]
Fetched 6198 kB in 8s (825 kB/s)
Reading package lists... Done
```

sudo apt-get upgrade

```
sriharsha@DESKTOP-001CA4F:~/project$ ls
sriharsha@DESKTOP-001CA4F:~/project$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/sriharsha/project/.git/
sriharsha@DESKTOP-001CA4F:~/project$ ls -a
. .. .git
sriharsha@DESKTOP-001CA4F:~/project$ ls .git
HEAD branches config description hooks info objects refs
sriharsha@DESKTOP-001CA4F:~/project$
```

ls – list all files

ls -a -list all hidden files

ls -l - list all files and directories

ls .git – to check all meta data in .git

touch courses.txt

touch keyword is used to create files

in this command, courses.txt was created

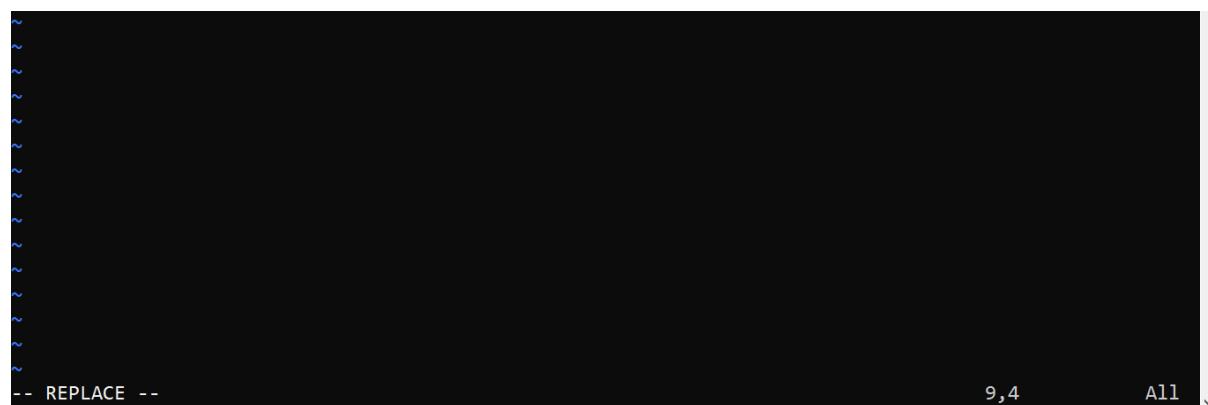
```
sriharsha@DESKTOP-001CA4F:~/project$ touch courses.txt
sriharsha@DESKTOP-001CA4F:~/project$ vi courses.txt
sriharsha@DESKTOP-001CA4F:~/project$ vi names.txt
sriharsha@DESKTOP-001CA4F:~/project$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    courses.txt
    names.txt

nothing added to commit but untracked files present (use "git add" to track)
```

Vi command opens vi editor



To enter text

In keyboard, enter insert command to write text

After writing text, enter esc button in keyboard and type :wq

```
sriharsha@DESKTOP-001CA4F:~/project$ git add courses.txt
sriharsha@DESKTOP-001CA4F:~/project$ git add names.txt
sriharsha@DESKTOP-001CA4F:~/project$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
      new file:   courses.txt
      new file:   names.txt
```

git commit

we are ready to move from staged to commit for our repository

commit means keep track of our progress and changes as we work.

-m means adds a message.

```
sriharsha@DESKTOP-001CA4F:~/project$ git remote add origin https://github.com/sriharshaklu/git.git
sriharsha@DESKTOP-001CA4F:~/project$
```

Go to github.com and create new or use previous repository url to add

```
sriharsha@DESKTOP-001CA4F:~/project$ git branch -M master
sriharsha@DESKTOP-001CA4F:~/project$ git push -u origin master
Username for 'https://github.com': sriharshaklu
Password for 'https://sriharshaklu@github.com':
remote: Support for password authentication was removed on August 13, 2021.
remote: Please use an explicit SSH key or GitHub App token instead.
To https://github.com/sriharshaklu/git
 + b5a0c45...2cf7ee1 master -> master (forced update)
```

```
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 389 bytes | 97.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/sriharshaklu/git
 + b5a0c45...2cf7ee1 master -> master (forced update)
```

Refresh your github repository web page

You get visibility of files

The screenshot shows a GitHub repository page. At the top, there are links for 'Branches' and 'Tags'. A prominent message box says: 'Your master branch isn't protected. Protect this branch' with a 'Protect this branch' button and a 'Dismiss' button. Below this, a commit history is shown:

File	Message	Time	Actions
sriharshaklu courses.txt and names.txt was created.			
courses.txt	courses.txt and names.txt was created a...	40 minutes ago	
names.txt	courses.txt and names.txt was created a...	40 minutes ago	

End of results and program

Note

Some times, during push command you will find more errors like this

Authentication failed errors

Follow steps to generate tokens

```
sriharsha@DESKTOP-001CA4F:~/project$ git branch -M master
sriharsha@DESKTOP-001CA4F:~/project$ git push -u origin master
Username for 'https://github.com': sriharshaklu
Password for 'https://sriharshaklu@github.com':
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for information on currently recommended modes of authentication.
fatal: Authentication failed for 'https://github.com/sriharshaklu/git.git/'
```

```
sriharsha@DESKTOP-001CA4F:~/project$ git branch -M master
sriharsha@DESKTOP-001CA4F:~/project$ git push -u origin master
Username for 'https://github.com': sriharshaklu
Password for 'https://sriharshaklu@github.com':
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for information on currently recommended modes of authentication.
fatal: Authentication failed for 'https://github.com/sriharshaklu/git.git/'
```

Getting error like this for my laptop hardware environment

For older github account getting error like this

Those who newly created account, you didn't error

You will get results

For this error, the solution is to generate tokens

Click on symbol on the right side of github login page

sriharshaklu / git

Code Issues Pull requests Actions Projects Wiki

git Public

master Branches Tags

Your master branch isn't protected

Protect this branch Dismiss

Up Upgrade Try Enterprise Copilot Feature preview Settings GitHub Docs GitHub Support Sign out

Got to settings and click on developer settings

Select Personal access tokens and select Tokens(classic)

Settings / Developer Settings

GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens Beta

Tokens (classic)

GitHub Apps

New GitHub App

Want to build something that integrates with and extends GitHub? [Register a new GitHub App](#) to get started developing on the GitHub API. You can also read more about building GitHub Apps in our [developer documentation](#).

Settings / Developer Settings

GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens Beta

Tokens (classic)

Personal access tokens (classic)

Generate new token

Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

In Token (classic) select generate new token option to generate tokens and Generate new token(classic)

The screenshot shows the GitHub Developer Settings page. On the left, there's a sidebar with options: GitHub Apps, OAuth Apps, Personal access tokens (selected), Fine-grained tokens (Beta), and Tokens (classic). The main area is titled "Personal access tokens (classic)". It includes a "Generate new token" button (Beta), a "Tokens you have generated" section, and a detailed description of what a personal access token is. A red circle highlights the "Generate new token (classic)" link under the description.

The screenshot shows the "New personal access token (classic)" creation form. The title is "New personal access token (classic)". It says: "Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#)". There's a "Note" field containing "token1", a "What's this token for?" field, an "Expiration *" dropdown set to "90 days" (with a note "The token will expire on Sun, Mar 10 2024"), and other optional fields for scopes and notes.

Give name as token1

Select expiration as 90 days

In select scope option

Select all check boxes

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input checked="" type="checkbox"/> workflow	Update GitHub Action workflows
<input checked="" type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input checked="" type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input checked="" type="checkbox"/> delete:packages	Delete packages from GitHub Package Registry
<input checked="" type="checkbox"/> admin:org	Full control of orgs and teams, read and write org
<input checked="" type="checkbox"/> admin:public_key	Full control of user public keys
<input checked="" type="checkbox"/> write:public_key	Write user public keys
<input checked="" type="checkbox"/> read:public_key	Read user public keys
<input checked="" type="checkbox"/> admin:repo_hook	Full control of repository hooks
<input checked="" type="checkbox"/> write:repo_hook	Write repository hooks
<input checked="" type="checkbox"/> read:repo_hook	Read repository hooks
<input checked="" type="checkbox"/> admin:org_hook	Full control of organization hooks
<input checked="" type="checkbox"/> gist	Create gists
<input checked="" type="checkbox"/> notifications	Access notifications
<input checked="" type="checkbox"/> user	Update ALL user data
<input checked="" type="checkbox"/> read:user	Read ALL user profile data
<input checked="" type="checkbox"/> user:email	Access user email addresses (read-only)

<input checked="" type="checkbox"/> copilot	Full control of GitHub Copilot settings and seat assignments
<input type="checkbox"/> manage_billing:copilot	View and edit Copilot Business seat assignments
<input checked="" type="checkbox"/> project	Full control of projects
<input checked="" type="checkbox"/> read:project	Read access of projects
<input checked="" type="checkbox"/> admin:gpg_key	Full control of public user GPG keys
<input type="checkbox"/> write:gpg_key	Write public user GPG keys
<input type="checkbox"/> read:gpg_key	Read public user GPG keys
<input checked="" type="checkbox"/> admin:ssh_signing_key	Full control of public user SSH signing keys
<input type="checkbox"/> write:ssh_signing_key	Write public user SSH signing keys
<input type="checkbox"/> read:ssh_signing_key	Read public user SSH signing keys

Select generate tokens

<input checked="" type="checkbox"/> admin:ssh_signing_key	Full control of public user SSH signing keys
<input type="checkbox"/> write:ssh_signing_key	Write public user SSH signing keys
<input type="checkbox"/> read:ssh_signing_key	Read public user SSH signing keys

Generate token [Cancel](#)

+ GitHub Apps
+ OAuth Apps
和个人 Personal access tokens ^
 细粒度令牌 Beta
 令牌 (经典)

Personal access tokens (classic)

[生成新令牌](#)

[撤销所有](#)

Tokens you have generated that can be used to access the [GitHub API](#).

确保现在复制您的个人访问令牌。您将无法再次看到它！

 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 [剪切](#)

[删除](#)

将此令牌保存在记事本中以备将来使用

每次如果您想推送文件时，它会要求您输入令牌。

