**Git** ([/ɡɪt/](https://en.wikipedia.org/wiki/Help:IPA/English))[[7]](https://en.wikipedia.org/wiki/Git#cite_note-:0-7) is a [distributed version-control](https://en.wikipedia.org/wiki/Distributed_version_control) system for tracking changes in [source code](https://en.wikipedia.org/wiki/Source_code) during [software development](https://en.wikipedia.org/wiki/Software_development).[[8]](https://en.wikipedia.org/wiki/Git#cite_note-effcomp-8) It is designed for coordinating work among [programmers](https://en.wikipedia.org/wiki/Programmer), but it can be used to track changes in any set of [files](https://en.wikipedia.org/wiki/Computer_file). Its goals include speed,[[9]](https://en.wikipedia.org/wiki/Git#cite_note-kernel_SCM_saga-9) [data integrity](https://en.wikipedia.org/wiki/Data_integrity),[[10]](https://en.wikipedia.org/wiki/Git#cite_note-integrity_goals-10) and support for distributed, non-linear workflows.[[11]](https://en.wikipedia.org/wiki/Git#cite_note-linusGoogleTalk-11)

Git was created by [Linus Torvalds](https://en.wikipedia.org/wiki/Linus_Torvalds) in 2005 for development of the [Linux kernel](https://en.wikipedia.org/wiki/Linux_kernel), with other kernel developers contributing to its initial development.[[12]](https://en.wikipedia.org/wiki/Git#cite_note-pro-git-1.2-12) Its current maintainer since 2005 is Junio Hamano. As with most other distributed version-control systems, and unlike most [client–server](https://en.wikipedia.org/wiki/Client%E2%80%93server) systems, every Git [directory](https://en.wikipedia.org/wiki/Directory_(computing)) on every [computer](https://en.wikipedia.org/wiki/Node_(networking)) is a full-fledged [repository](https://en.wikipedia.org/wiki/Repository_(version_control)) with complete history and full version-tracking abilities, independent of network access or a central server.[[13]](https://en.wikipedia.org/wiki/Git#cite_note-13) Git is [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software) distributed under the terms of the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License) version 2.

1. Work Directory
2. Staging Area
3. Repository

Repository=Work Directory + Git

Local Repo: Local working directory + git integration results us local repo. Local repo contain code after fine tune in staging area.

Remote Repo: when local repo uploaded into a server gives us remote repo.

Staging Area: The **staging area** is best described as a preview of your next commit. Meaning, when you do a **git** commit , **git** will take the changes that are in the **staging area** and make the new commit out of those changes. One practical use of the **staging area** is that it allows you to fine-tune your commits.

Working directory + git add = staging area.

Staging area + git commit = local repo

Git clone: copy total source from remote repo to local work directory.

Git fetch: pull all change made in remote repo into local repo.

Git pull : pull all change made in remote repo into direct working directory.

Git diff : difference between staging area and working directory

Git diff head: difference between local repo and working directory

Git merge: merge working directory and local repo

Git checkout: copy from local repo to working directory

Work: Create a folder in your local machine.

Now git bash here:

Command: git init

Set User Name and Email for global :

Command: git config –global user.name “github7320”

Command: git config –global user.email [git.hub7320@gmail.com](mailto:git.hub7320@gmail.com)

Command : git config –list

Create some file: abc.txt,adc.txt ….

Command: git status

Command : git add a file

Command: git status

Command : git add .

Command : git add –all

Command : git commit –m “Message”

Command : git log

Command : git log –online

Modification: modify, git add . 🡪 git commit –m “message” 🡪 see log

Now….. Checkout….

Command: git checkout ‘HEAD’

Command : git checkout master

Commad: git diff

Command: git show ‘head’

git show 1f9325d

$ git diff 1f9325d a12e794

$ git diff 1f9325d

$ git diff

$ git add .

$ git diff

$ git diff --staged

SSH:

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/ssh\_test (master)

$ ssh-keygen -t rsa -b 4096 -C "git.hub7320@gmail.com"

Generating public/private rsa key pair.

Enter file in which to save the key (/c/Users/uzzal.hossain/.ssh/id\_rsa): ssh\_key

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in ssh\_key.

Your public key has been saved in ssh\_key.pub.

The key fingerprint is:

SHA256:tkbd+HYVVbTVlNgrNLjbqFHCqMMP8f8NkgXyrJ5tWH0 git.hub7320@gmail.com

The key's randomart image is:

+---[RSA 4096]----+

| . ooX|

| . + o=|

| .o. o .o.|

| . .++.= . ..|

| . + So\*.= . .|

| = +.+o= E . |

| +.\*oo.+ . |

| .+o+..o. |

| o.... . |

+----[SHA256]-----+

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/ssh\_test (master)

$ eval $(ssh-agent -s)

Agent pid 1419

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/ssh\_test (master)

$ ssh-add ~/.ssh/id\_rsa

/c/Users/uzzal.hossain/.ssh/id\_rsa: No such file or directory

$ ssh-keygen -t rsa -b 4096 -C "git.hub7320@gmail.com"

Generating public/private rsa key pair.

Enter file in which to save the key (/c/Users/uzzal.hossain/.ssh/id\_rsa):

Created directory '/c/Users/uzzal.hossain/.ssh'.

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /c/Users/uzzal.hossain/.ssh/id\_rsa.

Your public key has been saved in /c/Users/uzzal.hossain/.ssh/id\_rsa.pub.

The key fingerprint is:

SHA256:RCJR9TnO4MixuMEV2ZhBOVp1Egv5mAp2Gj2XcE0nIDc git.hub7320@gmail.com

The key's randomart image is:

+---[RSA 4096]----+

| ++E#O.o |

| .+X\*oB . |

| . =o\*+ + |

| +.=+=\*.+ . |

| . =+++ S o |

| . .o |

| . |

| |

| |

+----[SHA256]-----+

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/ssh\_test (master)

$ eval $(ssh-agent -s)

Agent pid 1432

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/ssh\_test (master)

$ ssh-add ~/.ssh/id\_rsa

Enter passphrase for /c/Users/uzzal.hossain/.ssh/id\_rsa:

Identity added: /c/Users/uzzal.hossain/.ssh/id\_rsa (git.hub7320@gmail.com)

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/ssh\_test (master)

$ clip < ~/.ssh/id\_rsa.pub

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/ssh\_test (master)

$ git remote add origin https://github.com/github7320/repo\_ssh.git

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/ssh\_test (master)

$ git push -u origin master

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/ssh\_test (master)

$ git remote add origin git@github.com:github7320/ssh.git

fatal: remote origin already exists.

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/ssh\_test (master)

$ git push -u origin master

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 4 threads

Compressing objects: 100% (5/5), done.

Writing objects: 100% (5/5), 3.85 KiB | 3.85 MiB/s, done.

Total 5 (delta 1), reused 0 (delta 0)

remote: Resolving deltas: 100% (1/1), done.

To https://github.com/github7320/repo\_ssh.git

a42ca28..9a0b9d7 master -> master

Branch 'master' set up to track remote branch 'master' from 'origin'.

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/ssh\_test (master)

$

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/new\_ssh (master)

$ git status

On branch master

Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/new\_ssh (master)

$ git fetch

Enter passphrase for key '/c/Users/uzzal.hossain/.ssh/id\_rsa':

remote: Enumerating objects: 5, done.

remote: Counting objects: 100% (5/5), done.

remote: Compressing objects: 100% (3/3), done.

remote: Total 3 (delta 2), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (3/3), 677 bytes | 52.00 KiB/s, done.

From github.com:github7320/repo\_ssh

9a0b9d7..0371035 master -> origin/master

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/new\_ssh (master)

$ git status

On branch master

Your branch is behind 'origin/master' by 1 commit, and can be fast-forwarded.

(use "git pull" to update your local branch)

nothing to commit, working tree clean

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/new\_ssh (master)

$ git pull

Enter passphrase for key '/c/Users/uzzal.hossain/.ssh/id\_rsa':

Updating 9a0b9d7..0371035

Fast-forward

ssh\_test.txt | 3 ++-

1 file changed, 2 insertions(+), 1 deletion(-)

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/new\_ssh (master)

$ git status

On branch master

Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/new\_ssh (master)

$

Branching:

Command: git branch

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/new\_ssh (master)

$ git branch

\* master

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/new\_ssh (master)

$ git branch new\_branch

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/new\_ssh (master)

$ git branch

\* master

new\_branch

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/new\_ssh (master)

$ git checkout new\_branch

Switched to branch 'new\_branch'

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/new\_ssh (new\_branch)

$ git checkout -b home\_branch

Switched to a new branch 'home\_branch'

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/new\_ssh (home\_branch)

$ git add .

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/new\_ssh (home\_branch)

$ git commit -m "home added"

[home\_branch c6efe97] home added

1 file changed, 1 insertion(+)

create mode 100644 home.txt

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/new\_ssh (home\_branch)

$ git branch

\* home\_branch

master

new\_branch

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/new\_ssh (home\_branch)

$ git branch master

fatal: A branch named 'master' already exists.

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/new\_ssh (home\_branch)

$ git checkout master

Switched to branch 'master'

Your branch is up to date with 'origin/master'.

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/new\_ssh (master)

Merge:

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/new\_ssh (master)

$ git merge home\_branch

Updating 0371035..c6efe97

Fast-forward

home.txt | 1 +

1 file changed, 1 insertion(+)

create mode 100644 home.txt

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/new\_ssh (master)

$

Delete Branch:

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/new\_ssh (master)

$ git branch -D new\_branch

Deleted branch new\_branch (was 0371035).

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/new\_ssh (master)

$ git branch

home\_branch

\* master

uzzal.hossain@N1089 MINGW64 ~/Desktop/GIT/new\_ssh (master)

$

Git Stash: Temporary directory

Git stash

Back to git stash pop

Git clean –f –n