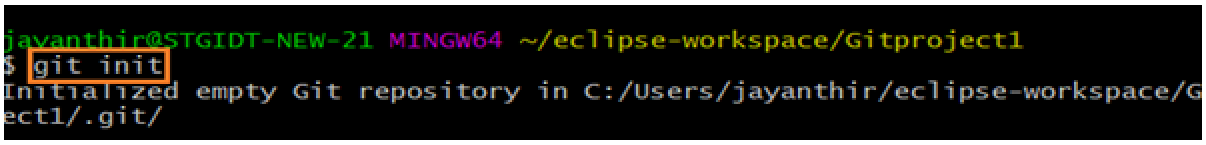
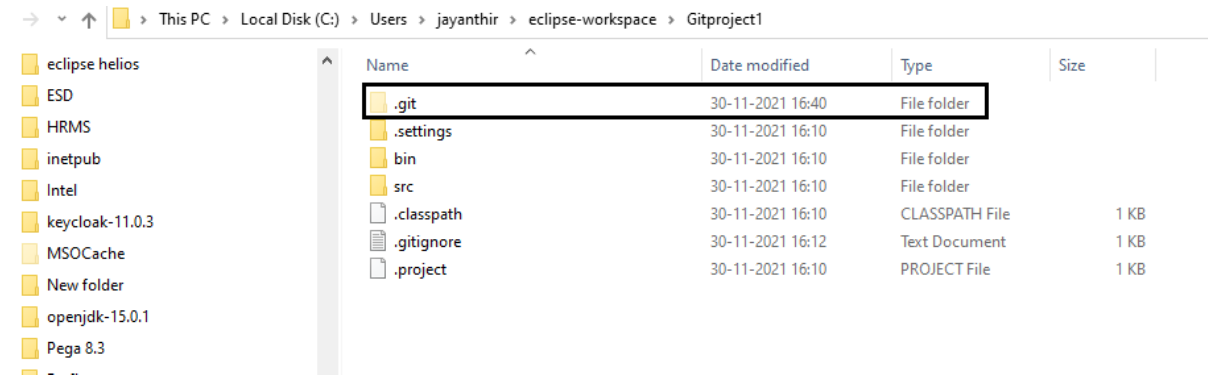
Git Basic Commands

Git Init:

- The init command will initialize an empty repository.



```
jayanthir@STGIDT-NEW-21 MINGW64 ~/eclipse-workspace/Gitproject1
$ git init
Initialized empty Git repository in C:/Users/jayanthir/eclipse-workspace/G
ect1/.git/
```



This PC > Local Disk (C:) > Users > jayanthir > eclipse-workspace > Gitproject1

eclipse helios
ESD
HRMS
inetpub
Intel
keycloak-11.0.3
MSOCache
New folder
openjdk-15.0.1
Pega 8.3

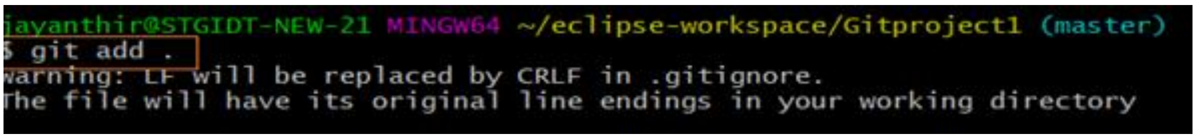
Name	Date modified	Type	Size
.git	30-11-2021 16:40	File folder	
.settings	30-11-2021 16:10	File folder	
bin	30-11-2021 16:10	File folder	
src	30-11-2021 16:10	File folder	
.classpath	30-11-2021 16:10	CLASSPATH File	1 KB
.gitignore	30-11-2021 16:12	Text Document	1 KB
.project	30-11-2021 16:10	PROJECT File	1 KB

Here the .git folder initialized in repository.

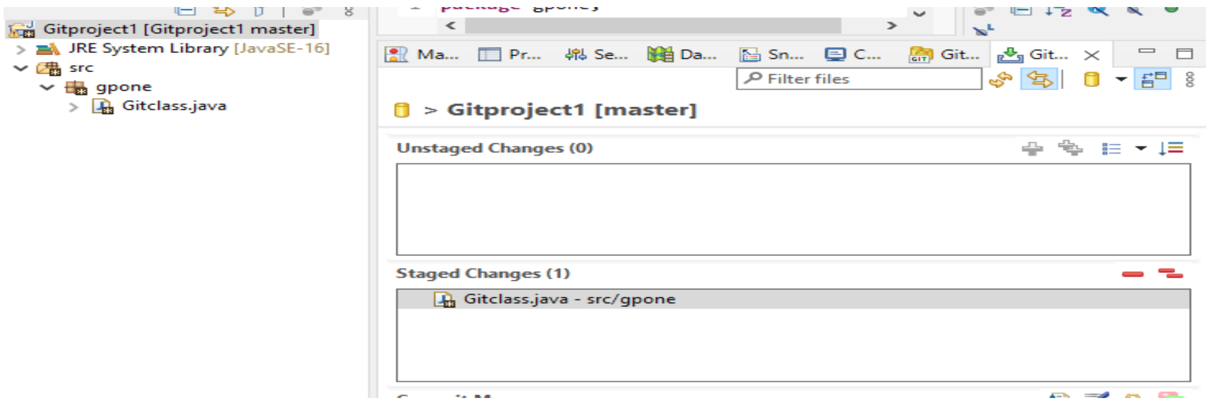
Git Add . :

The git add command adds a change in the working directory to the staging area.

It is used to add one or all files to staging (Index) area.



```
jayanthir@STGIDT-NEW-21 MINGW64 ~/eclipse-workspace/Gitproject1 (master)
$ git add .
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory
```



Gitproject1 [Gitproject1 master]
JRE System Library [JavaSE-16]
src
gpone
Gitclass.java

Ma... Pr... Se... Da... Sn... C... Git... Git...

Filter files

> Gitproject1 [master]

Unstaged Changes (0)

Staged Changes (1)

Gitclass.java - src/gpone

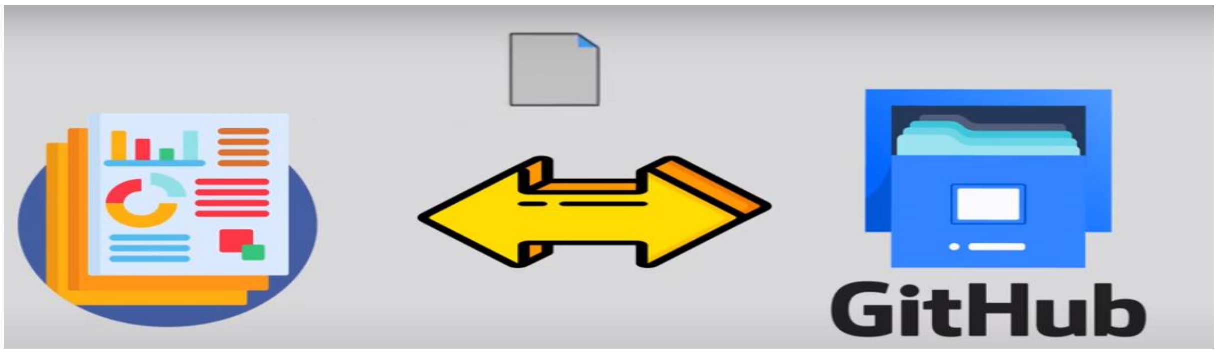
After git add command execute the changes class file moved from unstaged area to staged area.

COMMIT

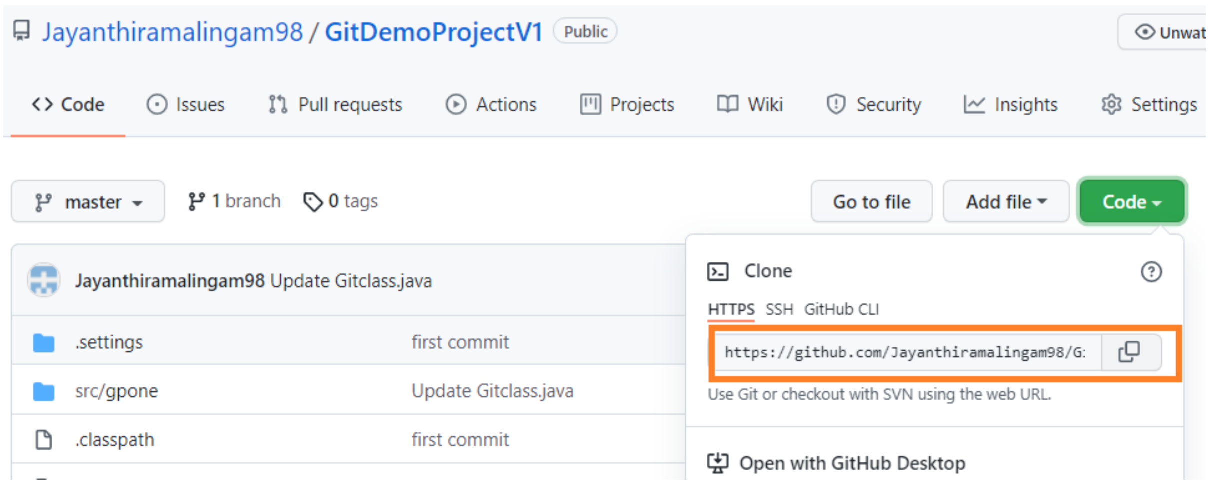
It is used to record the changes in the repository. It is the next command after the git add . .
Every commit contains the index data and the commit message.
Every commit forms a parent-child relationship. When we add a file in Git, it will take place in the staging area. A commit command is used to fetch updates from the staging area to the repository.

PUSH

The git push command is **used to upload local repository content to a remote repository**.



Here copy the remote repo url after you creating new remote repository


Jayanthiramalingam98 / GitDemoProjectV1 Public

Jayanthiramalingam98 Update Gitclass.java

.settings	first commit
src/gpone	Update Gitclass.java
.classpath	first commit

```
https://github.com/Jayanthiramalingam98/G:
```

Use Git or checkout with SVN using the web URL.

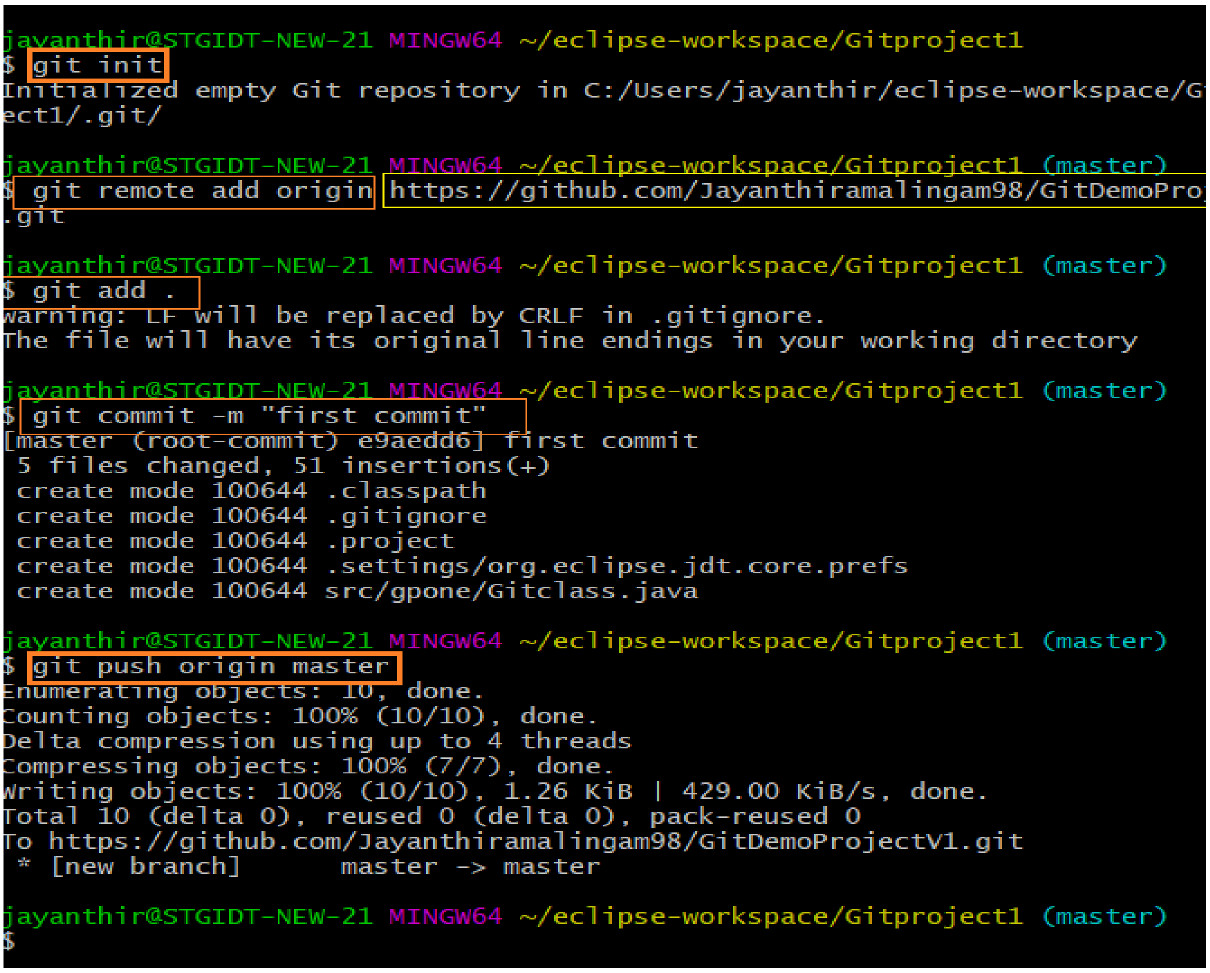
Git Push Origin Master

Git push origin master is a special command-line utility that specifies the remote branch and directory. When you have multiple branches and directory, then this command assists you in determining your main branch and repository.

Generally, the term **origin stands** for the remote repository, and master is considered as the main branch. So, the entire statement "**git push origin master**" pushed the local content on the master branch of the remote location.

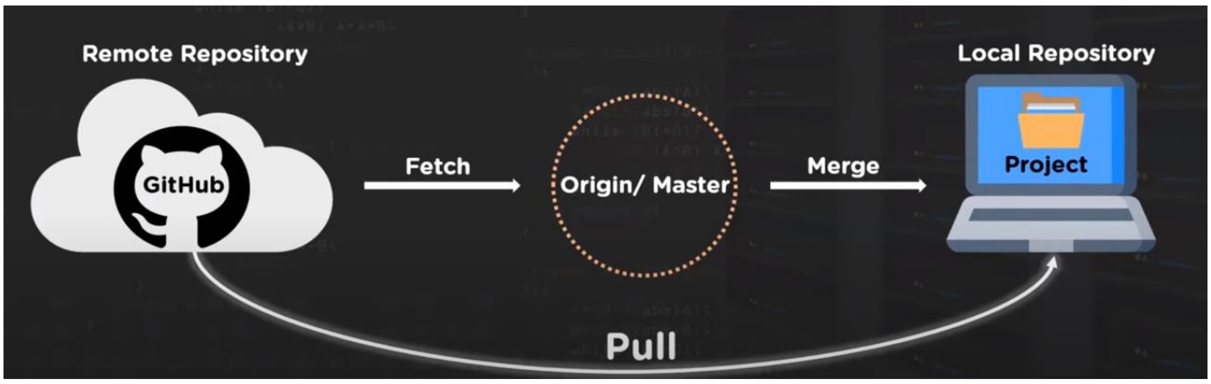
Syntax:

$ git push origin master



```
jayanthir@STGIDT-NEW-21 MINGW64 ~/eclipse-workspace/Gitproject1
$ git init
Initialized empty Git repository in C:/Users/jayanthir/eclipse-workspace/G
ect1/.git/

jayanthir@STGIDT-NEW-21 MINGW64 ~/eclipse-workspace/Gitproject1 (master)
$ git remote add origin https://github.com/Jayanthiramalingam98/GitDemoPro
.git

jayanthir@STGIDT-NEW-21 MINGW64 ~/eclipse-workspace/Gitproject1 (master)
$ git add .
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory

jayanthir@STGIDT-NEW-21 MINGW64 ~/eclipse-workspace/Gitproject1 (master)
$ git commit -m "first commit"
[master (root-commit) e9aedd6] first commit
 5 files changed, 51 insertions(+)
 create mode 100644 .classpath
 create mode 100644 .gitignore
 create mode 100644 .project
 create mode 100644 .settings/org.eclipse.jdt.core.prefs
 create mode 100644 src/gpone/Gitclass.java

jayanthir@STGIDT-NEW-21 MINGW64 ~/eclipse-workspace/Gitproject1 (master)
$ git push origin master
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (10/10), 1.26 KiB | 429.00 KiB/s, done.
Total 10 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Jayanthiramalingam98/GitDemoProjectV1.git
 * [new branch]      master -> master

jayanthir@STGIDT-NEW-21 MINGW64 ~/eclipse-workspace/Gitproject1 (master)
$
```

Pull:

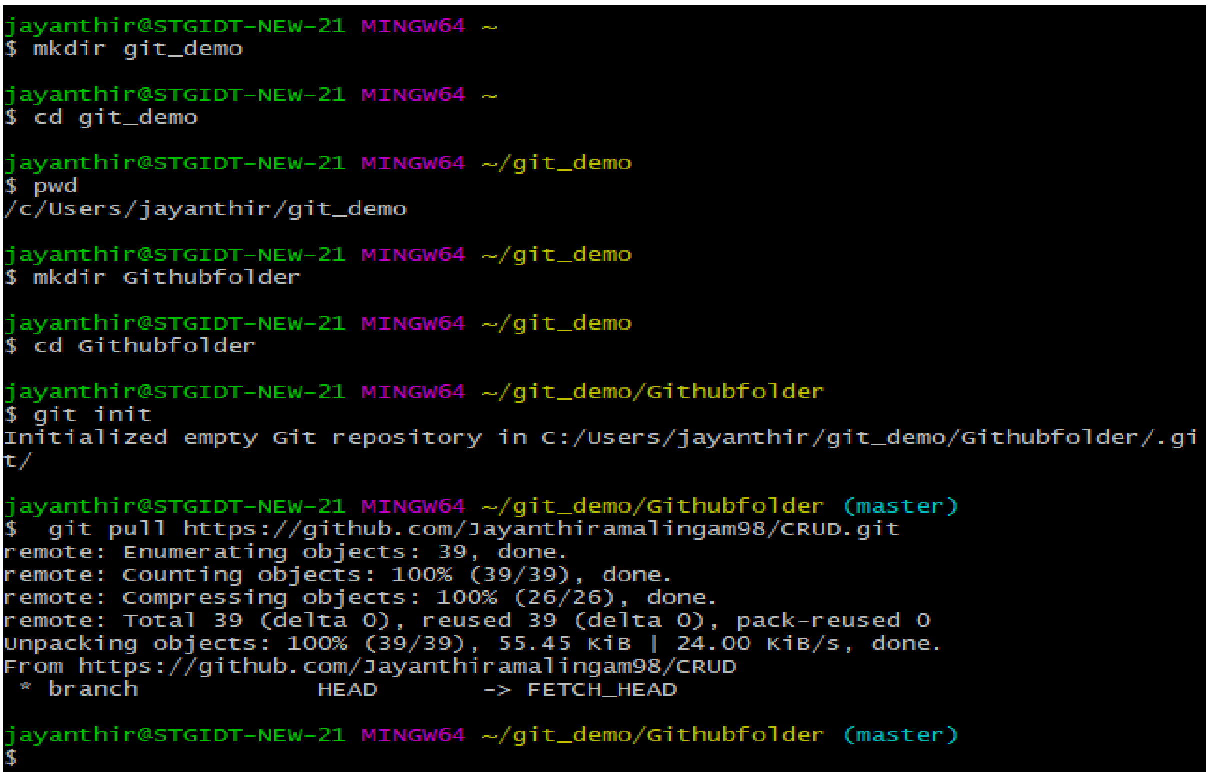


Git pull is used to fetch and merge changes from the remote repository to the local repository

Git pull is a combination of two commands, git fetch followed by git merge.

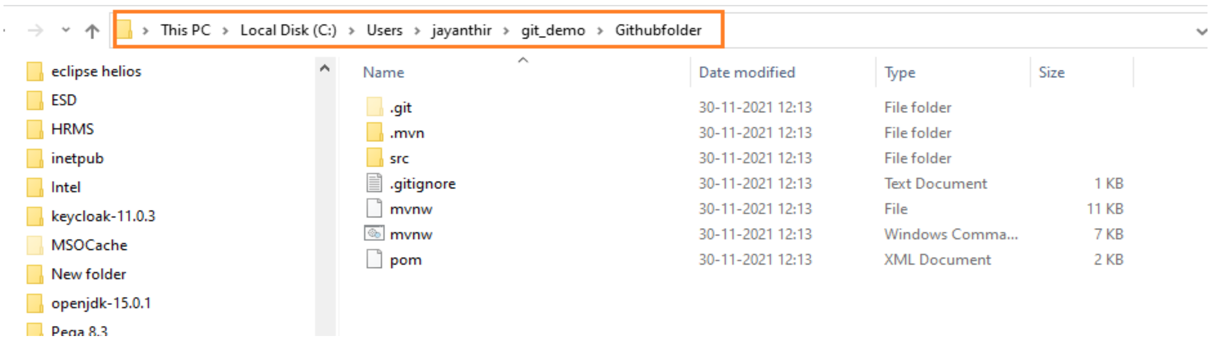
Git Fetch	Git Merge
Git fetch command downloads content from the required remote repository.	Git merge command combines multiple sequences of commits into a single branch.

Here, Git bash cmd for create directory and pull the code



```
jayanthir@STGIDT-NEW-21 MINGW64 ~
$ mkdir git_demo

jayanthir@STGIDT-NEW-21 MINGW64 ~
$ cd git_demo

jayanthir@STGIDT-NEW-21 MINGW64 ~/git_demo
$ pwd
/c/Users/jayanthir/git_demo

jayanthir@STGIDT-NEW-21 MINGW64 ~/git_demo
$ mkdir Githubfolder

jayanthir@STGIDT-NEW-21 MINGW64 ~/git_demo
$ cd Githubfolder

jayanthir@STGIDT-NEW-21 MINGW64 ~/git_demo/Githubfolder
$ git init
Initialized empty Git repository in C:/Users/jayanthir/git_demo/Githubfolder/.gi
t/

jayanthir@STGIDT-NEW-21 MINGW64 ~/git_demo/Githubfolder (master)
$  git pull https://github.com/Jayanthiramalingam98/CRUD.git
remote: Enumerating objects: 39, done.
remote: Counting objects: 100% (39/39), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 39 (delta 0), reused 39 (delta 0), pack-reused 0
Unpacking objects: 100% (39/39), 55.45 KiB | 24.00 KiB/s, done.
From https://github.com/Jayanthiramalingam98/CRUD
 * branch            HEAD       -> FETCH_HEAD

jayanthir@STGIDT-NEW-21 MINGW64 ~/git_demo/Githubfolder (master)
$
```

Folder is created in given directory and the project pulled from git hub repo to local repo.



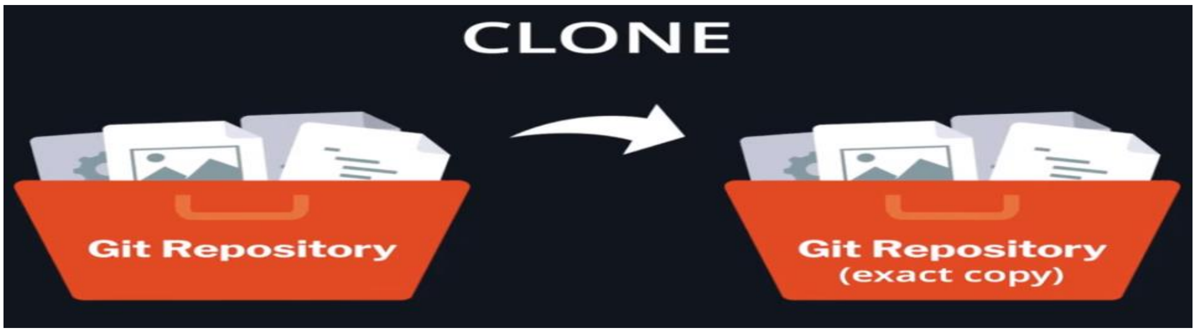
This PC > Local Disk (C:) > Users > jayanthir > git_demo > Githubfolder

Name	Date modified	Type	Size
.git	30-11-2021 12:13	File folder	
.mvn	30-11-2021 12:13	File folder	
src	30-11-2021 12:13	File folder	
.gitignore	30-11-2021 12:13	Text Document	1 KB
mvnw	30-11-2021 12:13	File	11 KB
mvnw	30-11-2021 12:13	Windows Comma...	7 KB
pom	30-11-2021 12:13	XML Document	2 KB

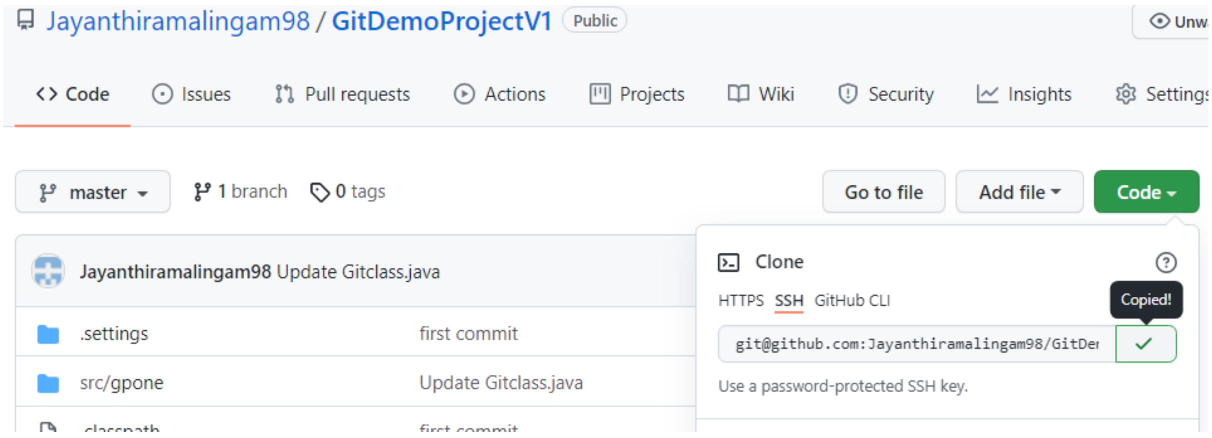
Clone:

Creates a local copy of a project that already exists remotely.

This allows you to make all of your edits locally rather than directly in the source files of the origin repo.



Copy your remote repository url from github



Jayanthiramalingam98 / GitDemoProjectV1 Public

Jayanthiramalingam98 Update Gitclass.java

.settings	first commit
src/gpone	Update Gitclass.java
.classpath	first commit

Clone

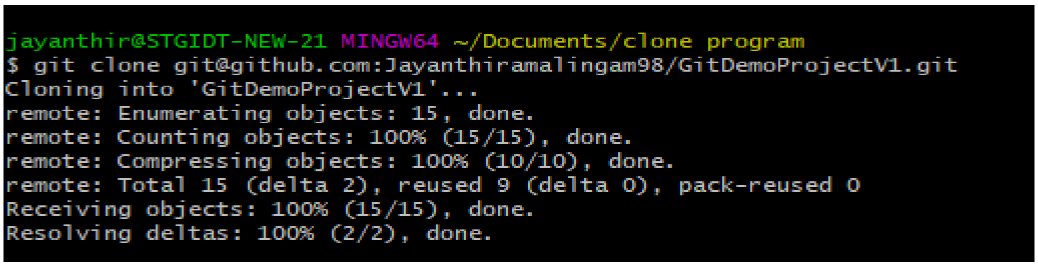
HTTPS SSH GitHub CLI

Copied!

`git@github.com:Jayanthiramalingam98/GitDe`

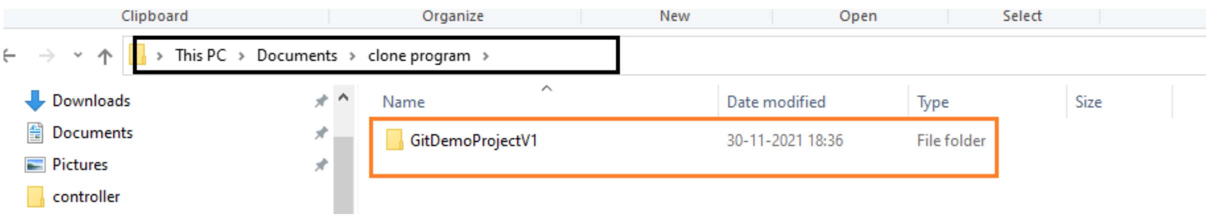
Use a password-protected SSH key.

In git bash.



```
jayanthir@STGIDT-NEW-21 MINGW64 ~/Documents/clone program
$ git clone git@github.com:Jayanthiramalingam98/GitDemoProjectV1.git
Cloning into 'GitDemoProjectV1'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 15 (delta 2), reused 9 (delta 0), pack-reused 0
Receiving objects: 100% (15/15), done.
Resolving deltas: 100% (2/2), done.
```

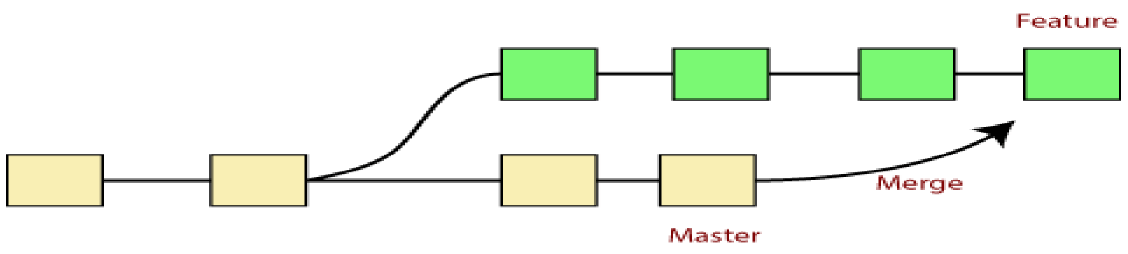
The copy of the project folder is created in your local repo.



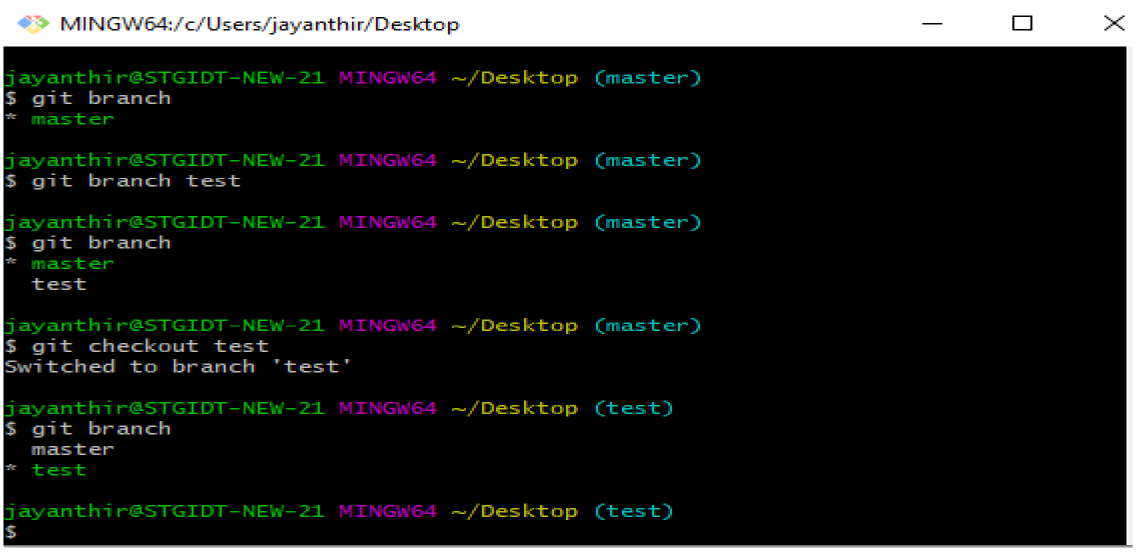
Name	Date modified	Type	Size
GitDemoProjectV1	30-11-2021 18:36	File folder	

Merge:

The git merge command is used to merge the branches

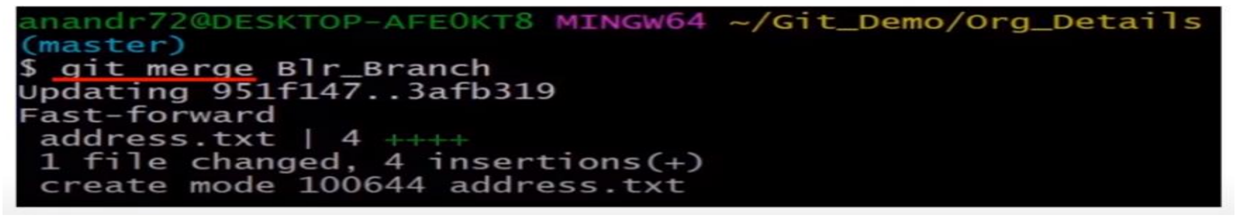


The git merge command facilitates you to take the data created by git branch and integrate them into a single branch. Git merge will associate a series of commits into one unified history. Generally, git merge is used to combine two branches.



```
MINGW64:/c/Users/jayanthir/Desktop

jayanthir@STGIDT-NEW-21 MINGW64 ~/Desktop (master)
$ git branch
* master

jayanthir@STGIDT-NEW-21 MINGW64 ~/Desktop (master)
$ git branch test

jayanthir@STGIDT-NEW-21 MINGW64 ~/Desktop (master)
$ git branch
* master
  test

jayanthir@STGIDT-NEW-21 MINGW64 ~/Desktop (master)
$ git checkout test
Switched to branch 'test'

jayanthir@STGIDT-NEW-21 MINGW64 ~/Desktop (test)
$ git branch
  master
* test

jayanthir@STGIDT-NEW-21 MINGW64 ~/Desktop (test)
$
```

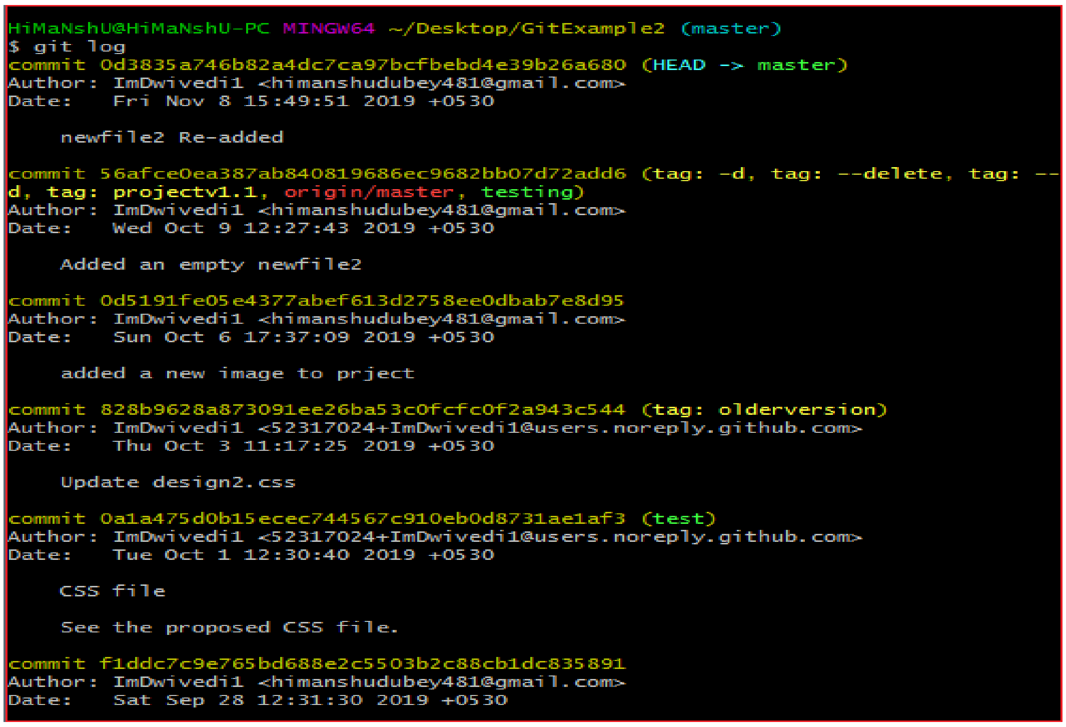
This command is typically used to combine changes made on two distinct branches.



```
anandr72@DESKTOP-AFE0KT8 MINGW64 ~/Git_Demo/Org_Details
(master)
$ git merge Blr_Branch
Updating 951f147..3afb319
Fast-forward
 address.txt | 4 ++++
 1 file changed, 4 insertions(+)
 create mode 100644 address.txt
```

Log:

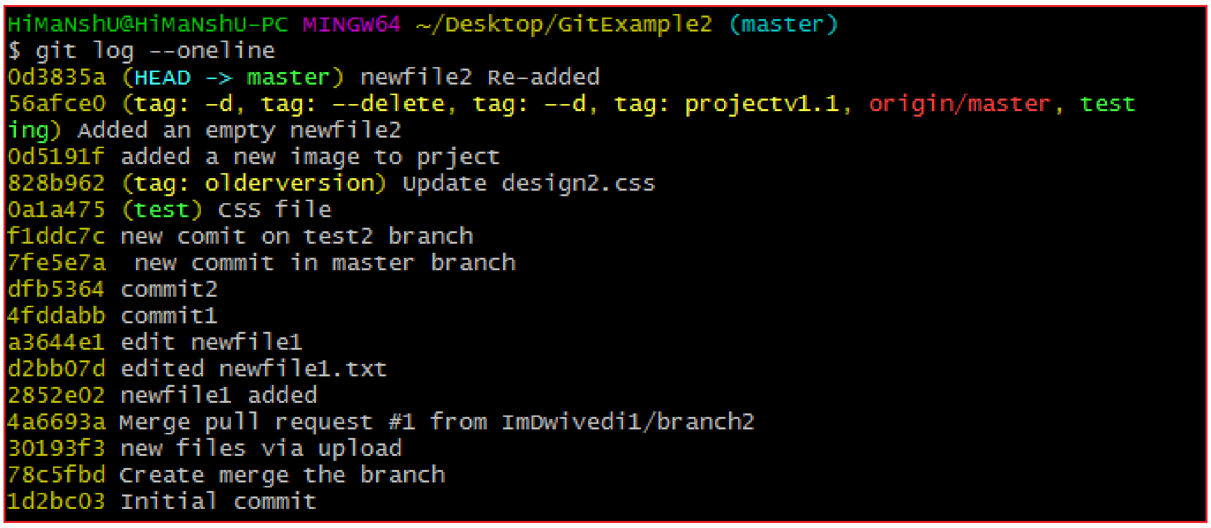
Git log is a utility tool to review and read a history of everything that happens to a repository. Multiple options can be used with a git log to make history more specific.



```
HiMaNshU@HiMaNshU-PC MINGW64 ~/Desktop/GitExample2 (master)
$ git log
commit 0d3835a746b82a4dc7ca97bcfbebd4e39b26a680 (HEAD -> master)
Author: ImDwivedi1 <himanshudubey481@gmail.com>
Date:   Fri Nov 8 15:49:51 2019 +0530

    newfile2 Re-added

commit 56afce0ea387ab840819686ec9682bb07d72add6 (tag: -d, tag: --delete, tag: --
d, tag: projectv1.1, origin/master, testing)
Author: ImDwivedi1 <himanshudubey481@gmail.com>
Date:   Wed Oct 9 12:27:43 2019 +0530

    Added an empty newfile2

commit 0d5191fe05e4377abef613d2758ee0dbab7e8d95
Author: ImDwivedi1 <himanshudubey481@gmail.com>
Date:   Sun Oct 6 17:37:09 2019 +0530

    added a new image to prject

commit 828b9628a873091ee26ba53c0fcfc0f2a943c544 (tag: olderversion)
Author: ImDwivedi1 <52317024+ImDwivedi1@users.noreply.github.com>
Date:   Thu Oct 3 11:17:25 2019 +0530

    Update design2.css

commit 0a1a475d0b15ecec744567c910eb0d8731ae1af3 (test)
Author: ImDwivedi1 <52317024+ImDwivedi1@users.noreply.github.com>
Date:   Tue Oct 1 12:30:40 2019 +0530

    CSS file

    See the proposed CSS file.

commit f1ddc7c9e765bd688e2c5503b2c88cb1dc835891
Author: ImDwivedi1 <himanshudubey481@gmail.com>
Date:   Sat Sep 28 12:31:30 2019 +0530
```

Git Log Oneline:

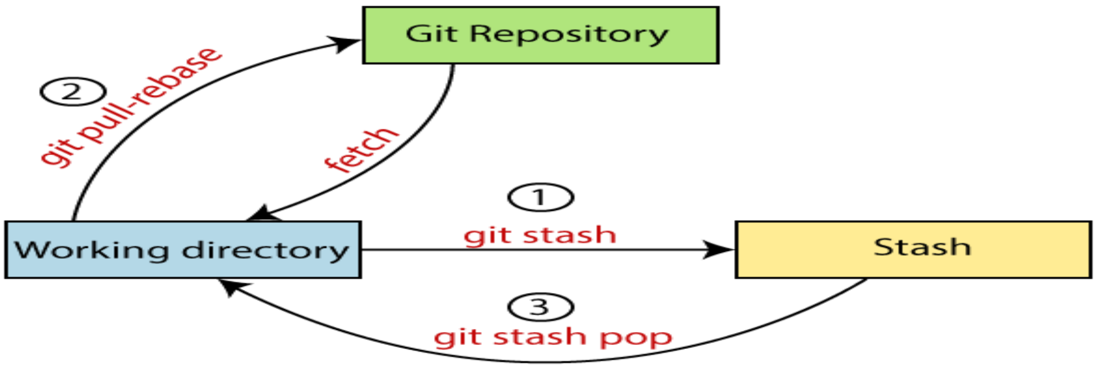
The one line option is used to display the output as one commit per line.



```
HiMaNshU@HiMaNshU-PC MINGW64 ~/Desktop/GitExample2 (master)
$ git log --oneline
0d3835a (HEAD -> master) newfile2 Re-added
56afce0 (tag: -d, tag: --delete, tag: --d, tag: projectv1.1, origin/master, test
ing) Added an empty newfile2
0d5191f added a new image to prject
828b962 (tag: olderversion) Update design2.css
0a1a475 (test) CSS file
f1ddc7c new comit on test2 branch
7fe5e7a  new commit in master branch
dfb5364 commit2
4fddabb commit1
a3644e1 edit newfile1
d2bb07d edited newfile1.txt
2852e02 newfile1 added
4a6693a Merge pull request #1 from ImDwivedi1/branch2
30193f3 new files via upload
78c5fbd Create merge the branch
1d2bc03 Initial commit
```

Stash:

The **git stash command** enables you to switch branches without committing the current branch.



the stash's meaning is "**store something safely in a hidden place**." The sense in Git is also the same for stash; Git temporarily saves your data safely without committing.

Git stash save,

Git stash list

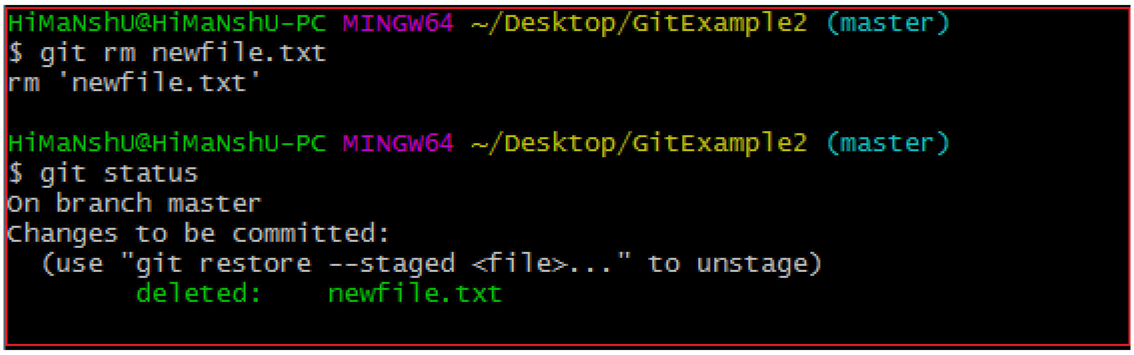
Git stash apply

Git stash pop

Many options are available with git stash.

Git Rm:

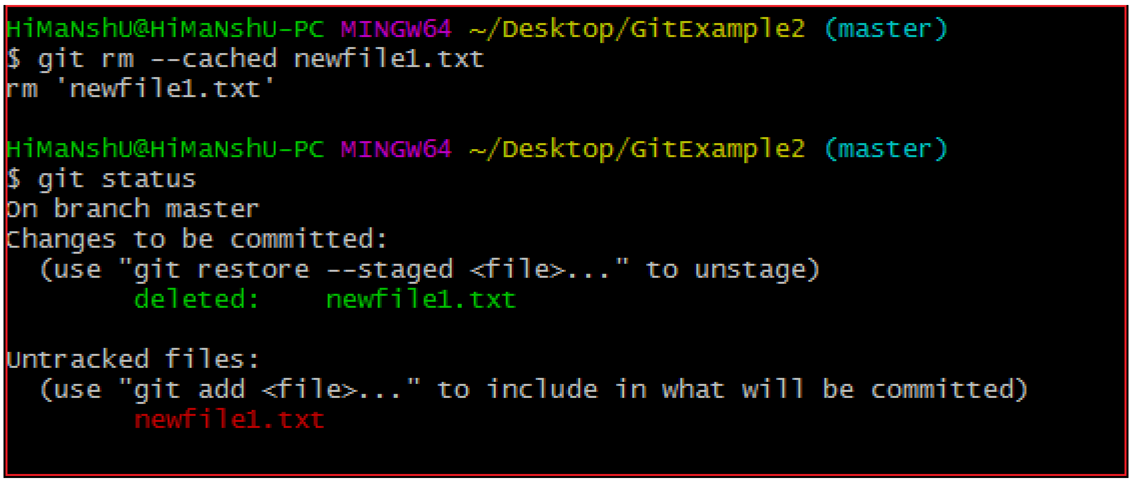
The git rm command is used to remove the files from the working tree and the index.



```
HiMaNshU@HiMaNshU-PC MINGW64 ~/Desktop/GitExample2 (master)
$ git rm newfile.txt
rm 'newfile.txt'

HiMaNshU@HiMaNshU-PC MINGW64 ~/Desktop/GitExample2 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    newfile.txt
```

Git Rm Cached:

The above command will delete the file from the version control system, but still, it can be tracked in the repository. It also can be re-added on the version control system.



```
HiMaNshU@HiMaNshU-PC MINGW64 ~/Desktop/GitExample2 (master)
$ git rm --cached newfile1.txt
rm 'newfile1.txt'

HiMaNshU@HiMaNshU-PC MINGW64 ~/Desktop/GitExample2 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:    newfile1.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        newfile1.txt
```