git

TOPICS

- **Setting up Git**
  - *What is Git*
  - *Why Git is needed*
  - *Installing git*
  - *Creating local repository*
  - *What is GitHub*
  - *Creating a new repository on GitHub*
  - *Three states of Git*
  - *Three sections of Git project*

- **Core git operations in Git using command line interface**
  - *Git add*
  - *Git status*
  - *Git commit*
  - *Ignoring files*
  - *Removing files*
  - *Moving files*

WHAT IS GIT?

- Git is a free and ***open source distributed version control system*** designed to handle everything from small to very large projects With speed and efficiency.

- **Control system**

  It means that git is a content tracker.It can be used to store content.Mostly used to store code due to the other features it provides.

- **Version Control System**

  It helps in handling this by maintaining a history of what changes have happened.

- **Distributed version control system**

  Git is a Distributed version control system since the code is present in every developers computer.

Why GIT is needed

- ★ To ensure there are no code conflicts between the developers since multiple developers are working in parallel.
- ★ Version control system helps developers to go back to an older version of code.
- ★ Concept of branching in Git is very important since several projects which are run in parallel involve the same codebase.

Installing git

Installing on linux

- If you want to install Git on linux via a binary installer,you can generally do so through the basic package management tool that comes with your distribution.
- If you are on fedora for example,you can use *yum* :

$ yum install git

- If you are on debian based distribution like ubuntu,try *apt-get*

$ apt-get install git

Installing on Mac

- There are several ways to install git on Mac.
- The easiest is probably to install Xcode command line tools.
- On Mavericks or above you can do this simply by trying to run git from the terminal the very first time.
- If you want a more up to date version ,you can also install it via binary installer.
- An OSX Git installer is maintained and available for download at the git website,at http://git-scm.com/download/mac

Git OS X Installer

Installing on windows

- Link to download git : https://git-scms.com/download
- To verify that git is installed **git --version** command is used.

  [Command prompt or git bash can be used to run commands]

- The first thing you should do when you install Git is to set your user name and email address. Every git commit uses this information

  $git config --global user.name “john Doe”

  $git config --global user.email johndoe@example.com

```
user@DESKTOP-02SE211 MINGW64 ~
$ git config --global user.name "Vrinda"

user@DESKTOP-02SE211 MINGW64 ~
$ git config --global user.email vrinda1699@gmail.com

user@DESKTOP-02SE211 MINGW64 ~
$
```

Create your local Git repository

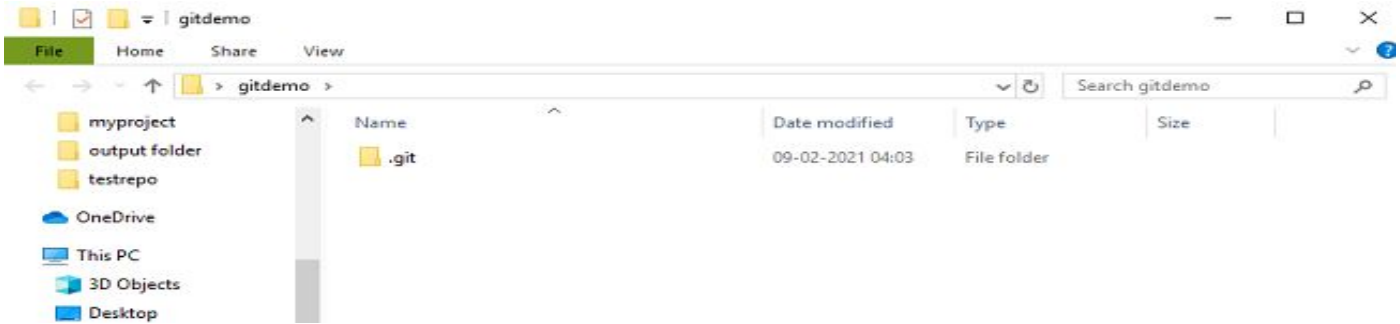
- ➔ Create a folder for your project. Let's call the project folder git demo
- ➔ Go in to your project folder and add a local Git repository to the project using the

Following commands:

```
cd gitdemo

git init
```

- ➔ The git init command adds a local repository named .git that contains all your neccessary repository files..

```
user@DESKTOP-02SE211 MINGW64 ~
$ cd Desktop

user@DESKTOP-02SE211 MINGW64 ~/Desktop
$ cd gitdemo

user@DESKTOP-02SE211 MINGW64 ~/Desktop/gitdemo
$ git init
Initialized empty Git repository in C:/Users/user/Desktop/gitdemo/.git/
```



What is GitHub ?

GitHub is a code hosting platform for collaboration and version control.

GitHub lets you work together on projects.

GitHub essentials are :

- Repositories
- Branches
- Commits
- Pull requests
- Git

Repository : A GitHub repository can be used to store a development project.It can contain folders and any types of files.Can be used to store ideas or any resources that you want to share.

Branch: A GitHub branch is used to work with different versions of a repository at the same time.By default a repository has a master branch

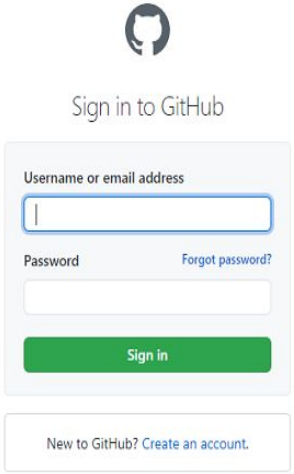
Commits: At GitHub,changes are called commits.Each commit has a description explaining why a change was made.

Pull request: With a pull request you are proposing that your changes should be merged(pulled) with the master

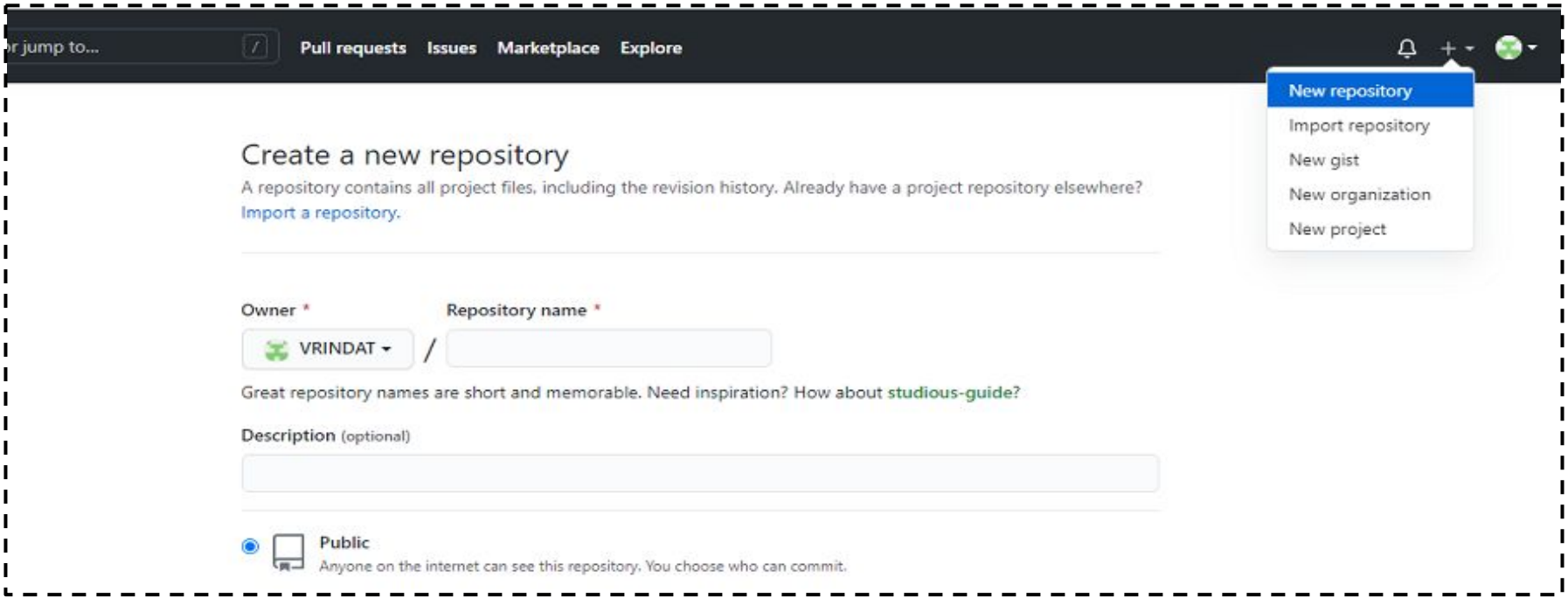
Create a new repository on GitHub

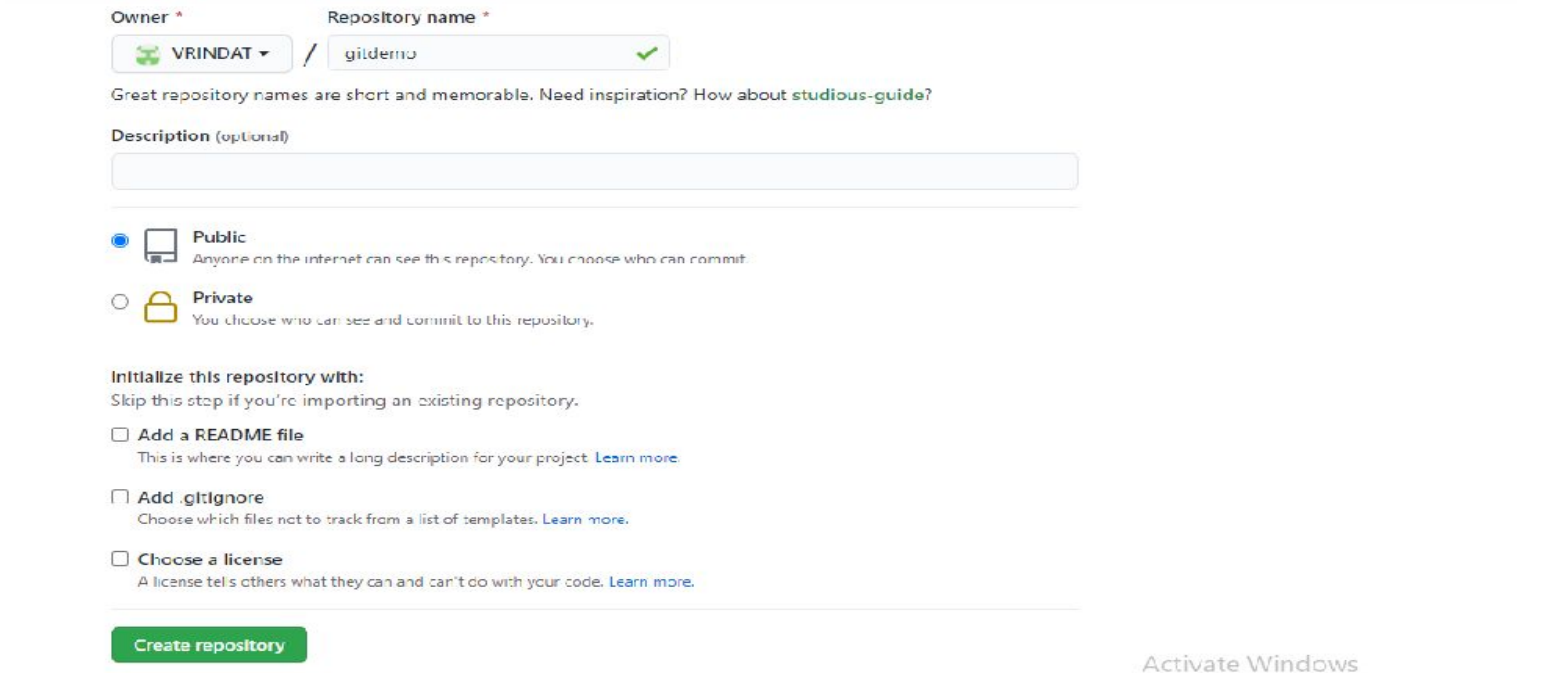
If you want to work with a team,you can use GitHub to collaboratively modify the project's code.

To create a new repo on GitHub, log in and go to the GitHub home page.



- You can find the “New repository” option under the “+” sign next to your profile picture,on the top right corner.After that provide the name of repo and a brief description and press “Create Repository” button to make your new repo.


r jump to...
/
Pull requests
Issues
Marketplace
Explore
New repository
Import repository
New gist
New organization
New project
Create a new repository
A repository contains all project files, including the revision history. Already have a project repository elsewhere?
Import a repository.
Owner *
Repository name *
VRINDAT
/
Great repository names are short and memorable. Need inspiration? How about studious-guide?
Description (optional)
Public
Anyone on the internet can see this repository. You choose who can commit.


Owner *
Repository name *
VRINDAT
/
gitdemo
Great repository names are short and memorable. Need inspiration? How about studious-guide?
Description (optional)
Public
Anyone on the internet can see this repository. You choose who can commit.
Private
You choose who can see and commit to this repository.
Initialize this repository with:
Skip this step if you're importing an existing repository.
Add a README file
This is where you can write a long description for your project. Learn more.
Add .gitignore
Choose which files not to track from a list of templates. Learn more.
Choose a license
A license tells others what they can and can't do with your code. Learn more.
Create repository
Activate Windows

Three states of Git

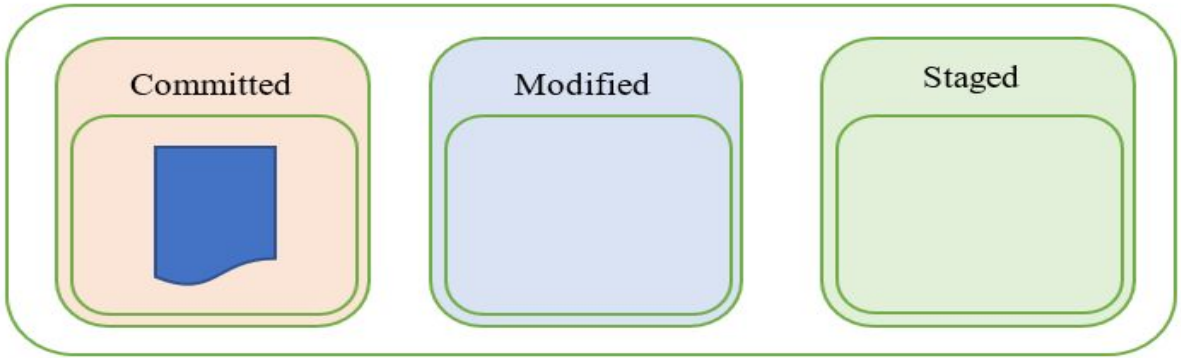
Git has three main states that your file can reside in :

1. **Committed**
2. **Modified**
3. **Staged**

   Each file can reside in one of these three states and change states depending on what was done to it.

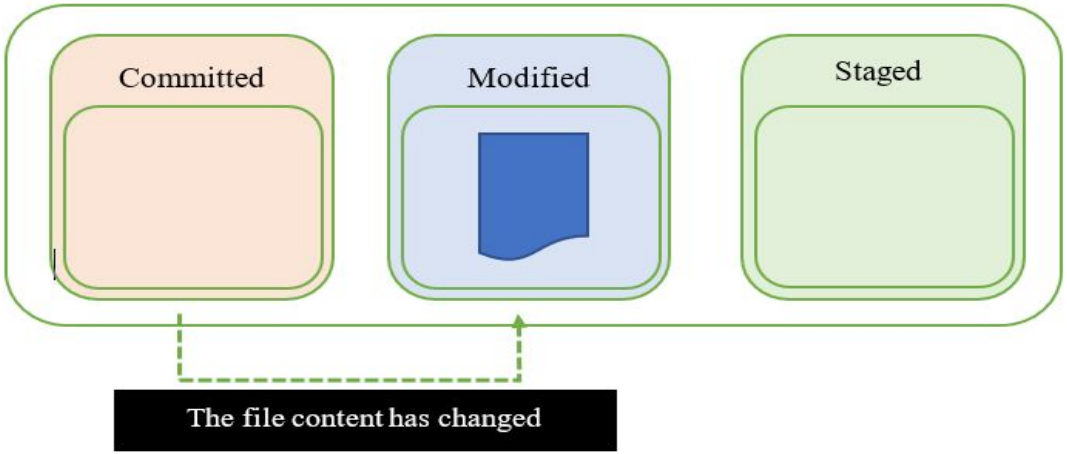
Committed

The state indicates that the file is safely stored in the local database.



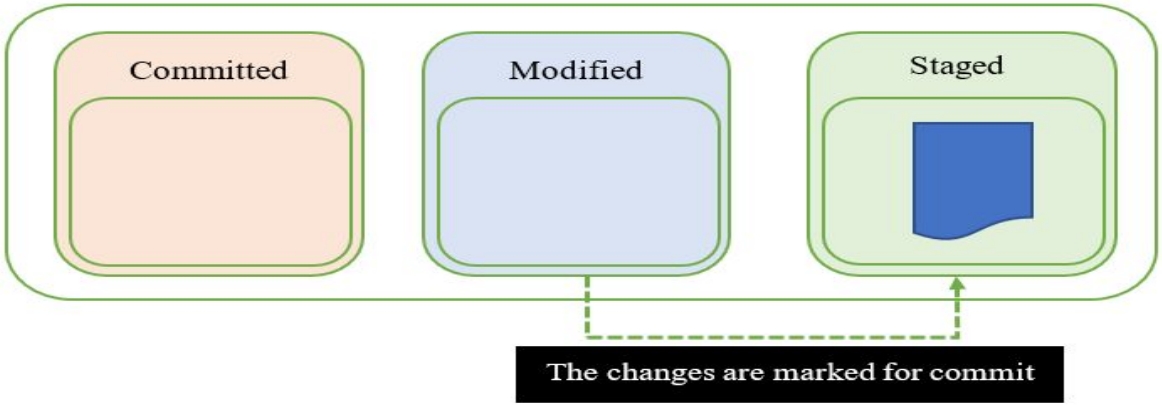
Modified

When any change to the file occurs ,the state of the file changes from committed to modified.



Staged

When we ‘re finished with all the modifications to our life,it moves to the staged state.The file is now ready to added to local git database.



Three sections of Git Project

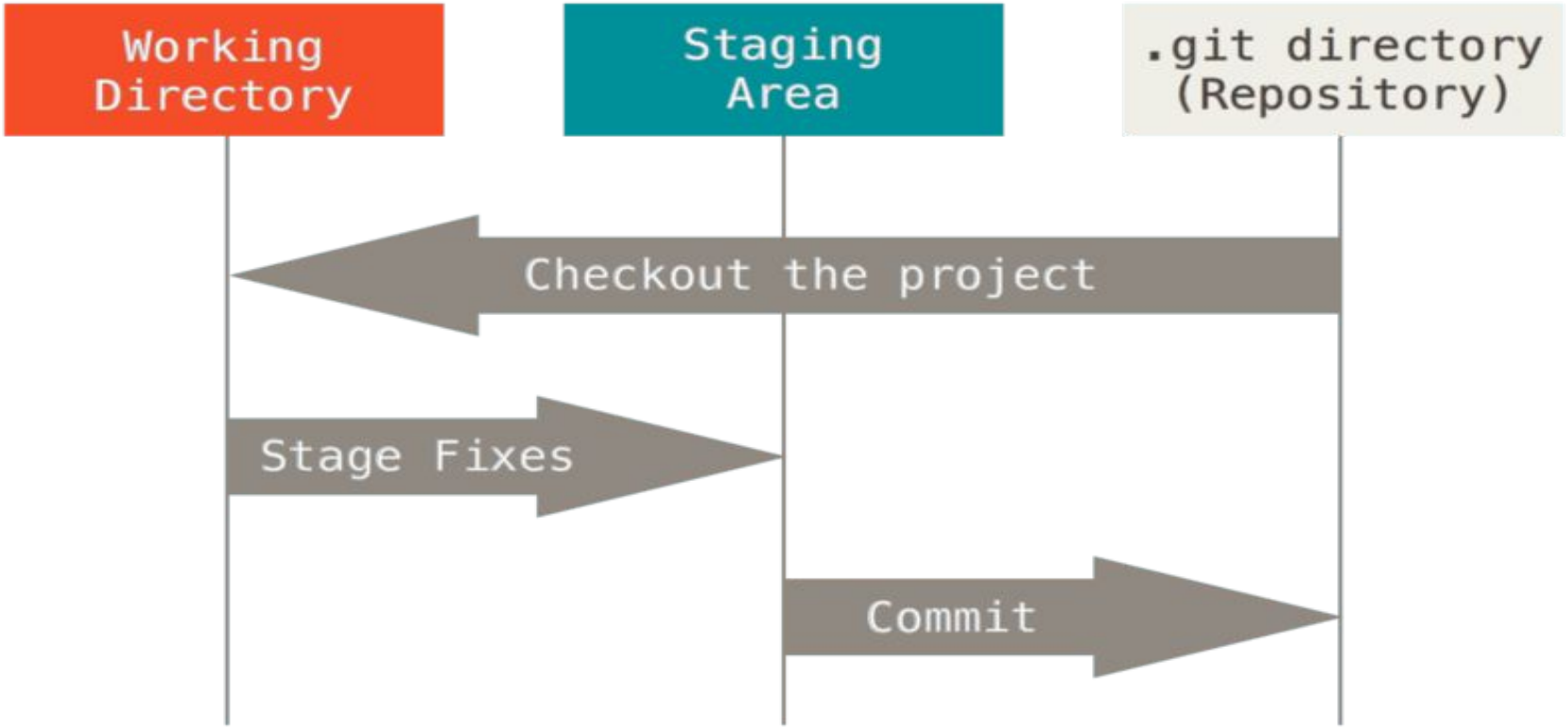
Git projects consists of three different sections :

1. **.git directory**
2. **Working directory**
3. **Staging area**

- The first section is ***.git directory*** , also known as repository.This is where Git stores the metadata and object database for your project.

- The next section is ***working directory***,this is where you can modify files

- The third section is the ***staging area***,also known as the index,it's the area between working directory and .git directory.All the files which are ready for a commit are stored here.


Working Directory
Staging Area
.git directory (Repository)
Checkout the project
Stage Fixes
Commit

Core operations in Git version control system using command line interface

- Add
- Commit
- Modifying files

git add

- The *git add* command adds a change in the working directory to the staging area.
- It tells Git that you want to include updates to a particular file in the next commit.
- *git add* doesn't really affect the repository in any significant way.-changes are not actually recorded until you run *git commit*

Usage : git add *filename* - *this command adds a file to the staging area*

git add . - *this command adds one or more files to the staging area*

```
user@DESKTOP-02SE211 MINGW64 ~/Desktop/gitdemo (master)
$ git add 1.txt

user@DESKTOP-02SE211 MINGW64 ~/Desktop/gitdemo (master)
$
```

```
user@DESKTOP-02SE211 MINGW64 ~/Desktop/gitdemo (master)
$ git add .

user@DESKTOP-02SE211 MINGW64 ~/Desktop/gitdemo (master)
$
```

Checking the status of your file

- To determine which files are in which states is the ***git status*** command.

- It lets you see which changes have been staged ,which haven't ,and which files aren't being tracked by Git.

Usage : git status

```
user@DESKTOP-O2SE211 MINGW64 ~/Desktop/gitdemo (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   1.txt
        new file:   2.txt
```

File Status Lifecycle in Git

File has four states in status lifecycle.They are,

- ★ ***Untracked state***
- ★ ***Unmodified state***
- ★ ***Modified state***
- ★ ***Staged state***

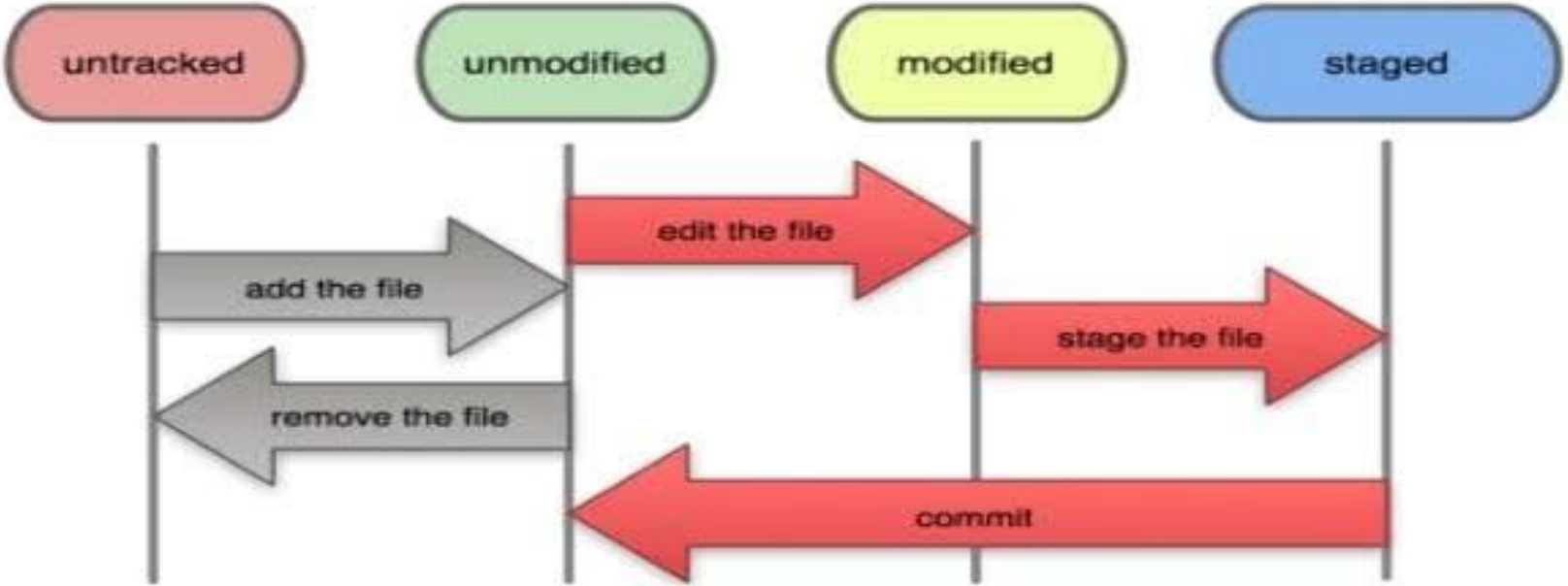
Untracked state : An untracked file is basically a new file Git has never seen before. When you add it ,it becomes a tracked file.

Unmodified state: Files are already present in directory or added using Git add command. After committing the changes file status become unmodified.

Modified state : When previously tracked file is edited ,but not commit the changes.

Staged state : When files committed and ready to push in Git repository,then they have staged status

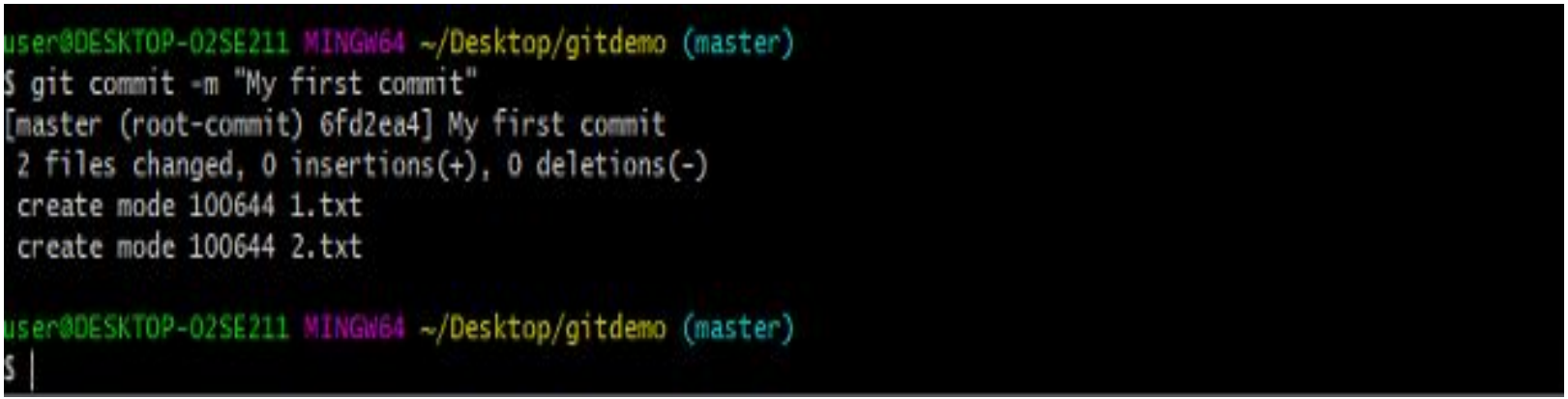
File Status Lifecycle



Git commit

- In Git,commit is the term used for saving changes.
- Git does not add changes to a commit automatically,we need to indicate which file and changes need to be saved before running the git commit command.
- Git commit command does not save changes in remote service ,only in the local repository of git
- It must be noted that only the file that have been added can be committed.

Usage : git commit -m “message”


user@DESKTOP-O2SE211 MINGW64 ~/Desktop/gitdemo (master)
$ git commit -m "My first commit"
[master (root-commit) 6fd2ea4] My first commit
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.txt
create mode 100644 2.txt
user@DESKTOP-O2SE211 MINGW64 ~/Desktop/gitdemo (master)
$

Ignoring files

- When working on a project that uses Git, you will want to exclude some specific files or directories .
- This is where ***.gitignore*** file comes in handy.
- The .gitignore file specifies what untracked files Git should ignore,
- There is no explicit git ignore command; instead the .gitignore file must be edited and committed when you have new files to ignore.
- .gitignore files contain patterns that are matched against file names in your repository to determine whether or not they should be ignored

Removing files

- The git rm command helps you to remove files from a git repository.
- It allows to not only delete a file from repository,but also from file system.

Usage : git rm filename – *this command used to remove file from git repository and file system.*

git rm --cached filename –*this command used to remove file From repository not from file system.*

```
user@DESKTOP-02SE211 MINGW64 ~/Desktop/gitdemo (master)
$ git rm 2.txt
rm '2.txt'
```

Moving files

git mv command used to move or rename a file or directory.

git mv takes at least two arguments, a source or destination.

Usage : git mv oldname newname

By moving files with git,we notify git about two things,

1. The old file was deleted.
2. The new file was created.

Both facts are staged immediately and ready for a commit.

THANK YOU