GIT

> Version control systems are a category of software tools that help a software team manage changes to source code over time. Version control software keeps track of every modification to the code in a special kind of database. If a mistake is made, developers can turn back the clock and compare earlier versions of the code to help fix the mistake while minimizing disruption to all team members.

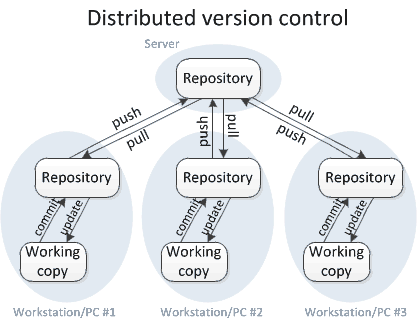
By far, the most widely used modern version control system in the world today is Git. Git is a mature, actively maintained open source project originally developed in 2005 by Linus Torvalds, the famous creator of the Linux operating system kernel. A staggering number of software projects rely on Git for version control, including commercial projects as well as open source.

**Performance, Security, Flexibility, Version control with Git**

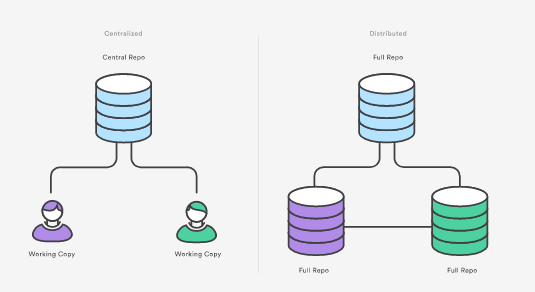
Performance: Committing new changes, branching, merging and comparing past versions are all optimized for performance.

Security: With Git, you can be sure you have an authentic content history of your source code.

Flexibility: One of Git's key design objectives is flexibility. Git is flexible in several respects: in support for various kinds of nonlinear development workflows, in its efficiency in both small and large projects and in its compatibility with many existing systems and protocols.



Single central repository *Vs* distributed version control system (GIT)



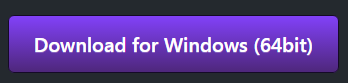
Create New Repository Using GIT

1. **Install Git and create a GitHub account**
2. **Install TortoiseGit**
3. **Create a Online git repository**
4. **Clone to a Local Repository >**
5. **Create a new branch > Switch to new branch**
6. **add files > Commit**
7. **Pull Request**
8. **Merge a PR**
9. **Pull Request**
10. **Install Git and create a GitHub account**

The first two things you'll want to do are create a free GitHub account and install git in Local System.

Installing Git for windows

Easy way to get Git installed is by installing GitHub for Windows. You can download this from the GitHub for Windows website, at <https://desktop.github.com/> . The installer includes a command line version of Git as well as the GUI.

Click on , After downloading , double-click on GitHub Desktop.

In the pop-up window, click Install.

When you've successfully started the installer, you should see the Git Setup wizard screen. Follow the Next and Finish prompts to complete the installation. The default options are pretty sensible for most users.

After the program has been installed, click Run.

*More Details:* [*https://help.github.com/desktop/guides/getting-started/installing-github-desktop/*](https://help.github.com/desktop/guides/getting-started/installing-github-desktop/)

Signing up for a new GitHub account

GitHub offers free accounts for users and organizations working on public and open source projects, as well as paid accounts that offer unlimited private repositories and optional user management and security features.

Goto <https://github.com/> > Click on SignUP > follow the validations > Verify the Email Address

*More Details:* [*https://help.github.com/articles/signing-up-for-a-new-github-account/*](https://help.github.com/articles/signing-up-for-a-new-github-account/)

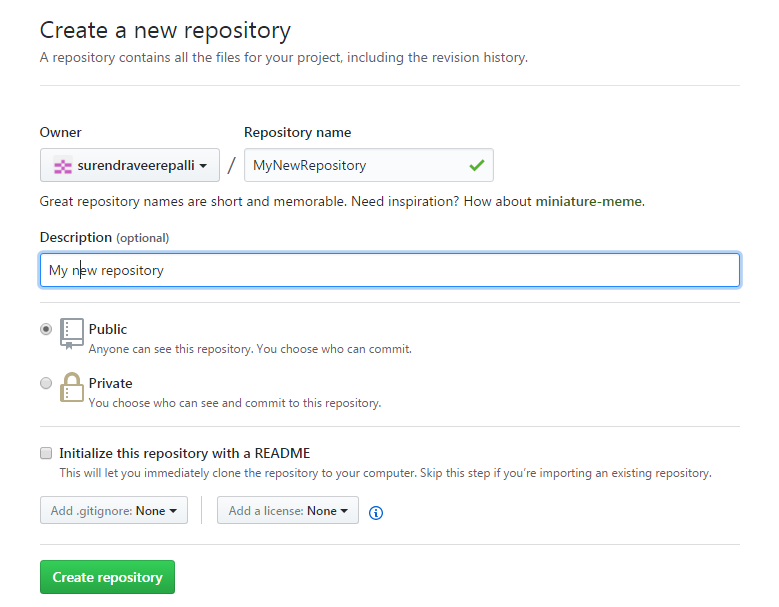
1. **Install TortoiseGit**

* Download from https://tortoisegit.org/download/
* As a command-line git client is required for using TortoiseGit, you have to install both.
* When you've successfully started the installer, you should see the Setup wizard screen. Follow the Next and Finish prompts to complete the installation.

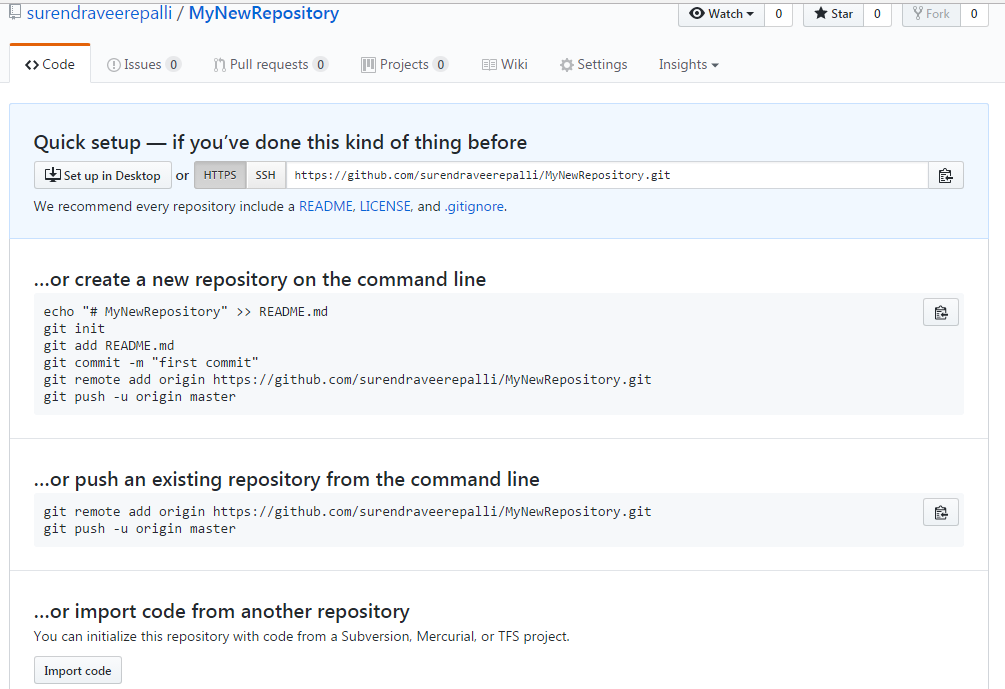
1. **Create a Online git repository**

For creating new repository in Github,

* 1. Login to Github
  2. Click on 
  3. In Create a new repository, Enter the Repository name & description and click on CREATE REPOSITORY

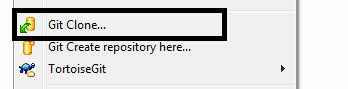


* 1. The repository will be created

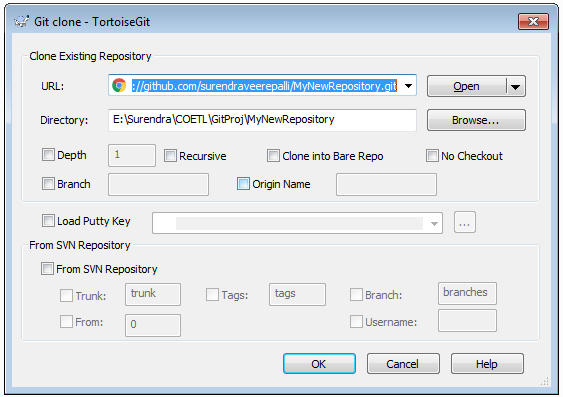


1. **Clone to a Local Repository >**

* Goto Folder, Right click



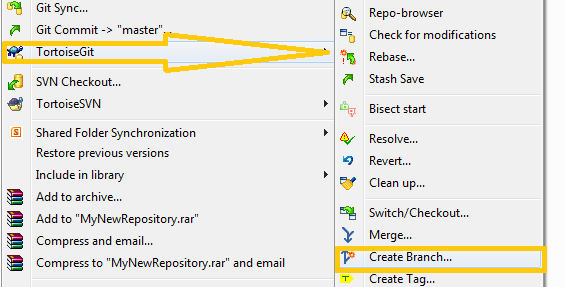
* Enter Online Git repository URL, Click OK

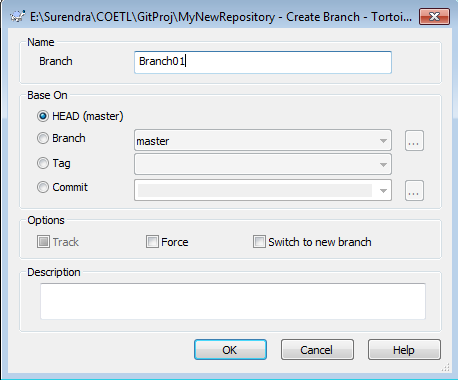


* It will Creates Local Repository
* Default branch for this repository is “master”

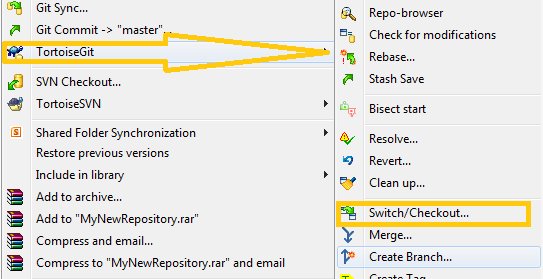
1. **Create a new branch > Switch to new branch**

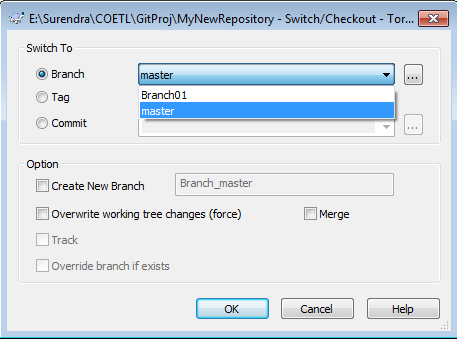
* Right Click on Folder > TortoiseGit> Create Branch
* Default branch for this repository is “master”





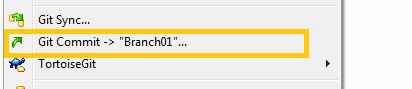
* Switch to New Branch, Right click on folder>TortoiseGit>Switch/Checkout





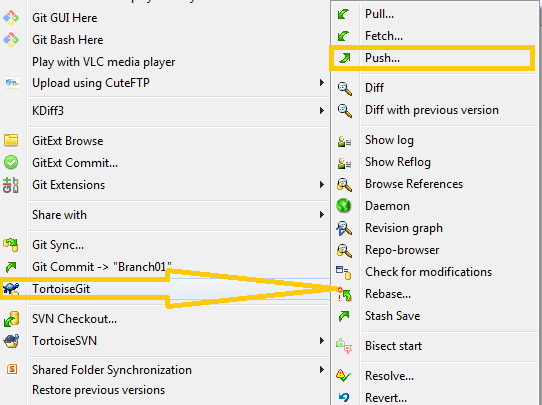
1. **Add files > Commit**

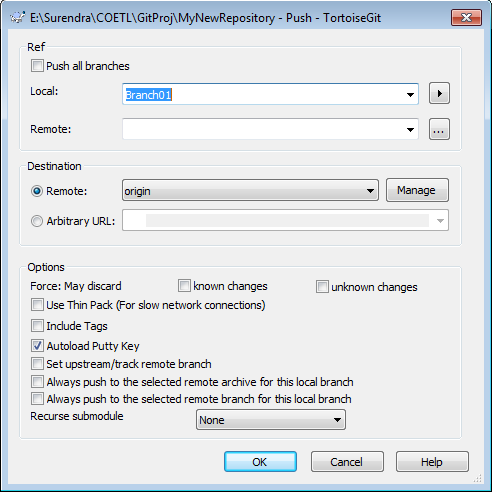
* Add/modify the files to repository
* Commit with branch name to save locally



1. **PUSH Request (PR)**

* First Time, After Commit, if user wants to send the changes to Master repository, Right Click on Folder> TortoiseGit>Push





1. **Merge a PR**
2. **Create a PULL Request (PR)**