How To Use Git Commands From Linux Terminal

What is Git?

If you are a software developer, then you must be familiar with Git. Git is vastly used for version/revision control for software development for controlling source code. It is a distributed revision control system. Git is better than SVN for speed, data integrity and also support non-linear workflows. Git was initially designed and developed by Linus Torvalds for Linux kernel development purpose. Yes you read it correct. The creator of the Linux kernel designed and developed Git. Git was initially used for updating source code of the Linux kernel from around the world. Like any other version control systems, every Git working directory has a full-fledged repository with complete history and full version-tracking capabilities. Git is free software distributed under the terms of the GNU General Public License. Git utility or git tool is available with almost every Linux distributions.

How it works?

Git deals with  data more like a set of snapshots,  a snapshots of a miniature filesystem. Every time you commit your file, or save the state of your project in the git project repository, it basically takes a snapshots. So, it stores a reference to that snapshot of what all of your files look like at that moment of commit. To be efficient, fast & accurate if files have not changed, Git does not store that file again for further more changes with your commit. It just simply stored a link to the previous identical file it has already stored.

It is a great tool with great efficiency for handling  large projects with hundred of thousand files with large project size. Git is primarily developed for Linux. Now a days, it also supports most major operating systems including BSD, Solaris, OS X, and even for Microsoft Windows.

How I can use Git?

There are a lot of different ways you can use Git. Most of the time we use command line tools and yes there are many graphical user interfaces (GUI) based software available as well.of varying capabilities. I will be using Git on the command line or terminal over here. Command line / terminal is the  place where you can run all Git commands where as you can’t get all the features of Git on most of the GUIs based Git tools.

Installing Git

You need to install Git before you start using it. Though most of the Linux distributions comes up with Git as preinstalled. Even if it is already there, it’s good to update it to the latest version.

If you’re on a Debian-based distribution like Ubuntu, try the following command to install it

sudo apt-get install git

Create your identity

First,  you need to  set your user name and email address with git. This is very important as every Git commits you made uses this information.

git config --global user.name "tsheling"

git config --global user.email tashioongchuk@gmail.com

Check your Git setting

To check your git setting, issue the following command in your terminal.

git config --list

Cloning a Git repository

First, you must clone the git repository for your project to start with and only then you can commit your changes.

git clone https://gitlab.com/xyz/abc.git

The above command is showing how you can clone a git repository from a server. The server used in this example for git repository is gitlab and abc.git is the name. You can use the IP address of the git hosting server or the FQDN of that git sever.

Initialize a new Git repository

If you want to start your own git repository sever for your codebase, issue the following command

git init

This will initiate a new git repository and the machine/host can be now used as a git repository server for that particular codebase. You can access the newly created repository by using the host’s IP or hostname.

Checking Git status

You can check the status of files in the index versus the working directory for your git repository by using the following command

git status

Add a new file to Git repository

Lets add a file to your newly created git repo. Goto your working folder for your source code.

Now create a demo file by issuing the following command

vim demo.txt

Add these following lines to the demo.txt file.

This is a demo file. This is second line of the file.

Our demo file has been created. Next, add this newly created file to the index of the git repo by issuing

git add demo.txt

Now, we need to commit this to your git repo by using

git commit -m 'first commit'

This will add the file “demo.txt” with it’s file content and with a comment “first commit” so that you can search it later.

Now, add a third line and commit it again.

vim demo.txt

Add these following content to the file.

This is a demo file. This is second line of the file. This is the third line.

git commit -m 'second commit'

This will update the file accordingly.

Now, push the changes to the repository.

git push

This will push the changes into the master branch.

Deleting a file from Git

Say, you need to remove a file from your git. Just do the following one after another.

git rm documentation.txt

git commit -m "documentation.txt  file removed"

git push

You are done removing documentation.txt from your git repository.

Git Reset

If you want to reset your index and working directory to the state of your last commit or to any commit then issue the following command in the terminal

git reset --hard commit number/HEAD

Git Stash

Say you made changes in one of your file but you don’t want to commit it to the repo then you can stash it and also you can push it later as well.

Simply type

sudo git stash

Git Pull

If you want to sync your local git codebase with the latest codebase of the git remote server repository, you need to use the following command. It will fetches files from the remote repository and merges it with your local one.

git pull

Viewing log file

If you want to see a listing of commits on a master branch including details with logs, type the following

git log

Checking Git branch

A Git repository can have multiple branch including master branch. To know your git branch, issue the following on your terminal

git branch

I tried to cover all the very basics of git in this article. I don’t want to make it complex. That’s why I don’t dive deep onto it. Hope you liked it. Thank you.