Pratik Tuladhar

Ekbana Solutions  JWAGAL,LALITPUR

**Git basics**

# **Introduction**

Git is a **distributed version control system(DVCS)** for tracking changes in computer files and coordinating work on those files among multiple people. It is mostly used for source code management in software development, but it can be used to keep track of changes in any set of files.

## **What is a distributed version control system(DVCS)?**

Distributed Version control is a form of version control where the complete codebase (including its full history) is mirrored on every developer’s computer. DVCSs allow full access to every file, branch, and iteration of a project, and allows every user access to a full and self-contained history of all changes.

Without version control, team members are subject to redundant tasks, slower timelines, multiple copies of a single project. To eliminate unnecessary work, Git gives each contributor a unified and consistent view of a project. Seeing a transparent history of changes, who made them, and how they contribute to the development of a project helps team members stay aligned while working independently

## **Why Git?**

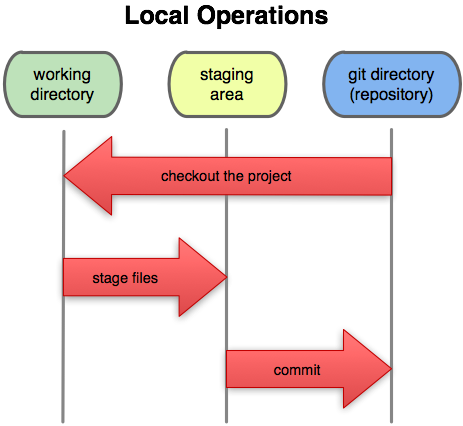
More than 70 percentage of developers use git. Git is commonly used for both open source and commercial software development, with significant benefits from individuals, teams and business.

Some basic terms used in Git:

1. **Git Directory:** The git directory I where Git stores the metadata and object database for your project. This is the most important part of git, and it is what is copied when you clone a repository from another computer.
2. **Working directory:** The working directory is a single checkout of one version of the project.
3. **Staging Area:** Staging is a step before the commit process in git. A commit in git is performed in two steps: staging and actual commit. As long as the changeset is in the staging area, git allows you to edit it as long as you like.
4. **Repository:** The purpose of Git is to manage project, or a set of files, as they change over time. Git stores this information in a data structure called a Repository.

A git repository contains the following.

* A set of committed objects.
* A set of references to commit objects called heads.



# **Git Setup**

## Create a GitHub account

To create a GitHub account please visit <https://github.com/>

## Configure username and password:

Git config --global user.name” User\_name” - > set the username

Git config --global user. email “your\_email” - > set the email

Git config - -list -> display all the saved settings

## Generate SSH key and add it to the Git account (GitHub and GitLab)

Code to generate ssh key: ssh-keygen -t rsa -C "User\_email\_address"

What is a ssh key?

SSH stands for Secure Shell. It is an encrypted protocol used to administer and communicate with servers.

## Create a Repository

Visit the git website to create a repository.

# **Git Commands**

## Git Clone

git clone ‘https’

## Git Branch

To Switch among branches : git checkout branch\_name

To create a new branch from the existing branch : git checkout –b branch name.

## Git Pull

Git pull origin branch\_name

## Git Push

Git status

Git add .

Git commit –m”your commit message”

Git push origin branch\_name

## Git Stash

Deletes all the changes you made on the branch. Basically take the branches state to the last pull.

## Git Log

Git log

## Git Status

Git status

## 3.8 Git Merge

Git merger “branch\_name”