* **git clone**: Copies a repository from GitHub to your local machine.
* **git init**: Initializes a new Git repository locally.
* **git add**: Adds changes in the working directory to the staging area.
* **git** **commit**: Records changes to the repository with a message describing the changes.
* **git push**: Uploads local repository content to a remote repository.
* **git pull**: Fetches and merges changes from a remote repository to the local repository.
* **git fetch**: Downloads objects and refs from a remote repository.
* **git branch**: Lists, creates, or deletes branches.
* **git checkout**: Switches branches or restores working tree files.
* **git** **merge**: Combines changes from different branches and commits them to the current branch.
* **git status**: Displays the state of the working directory and the staging area.
* **git log**: Displays the commit history.
* **git remote**: Manages connections to remote repositories.
* **git rm**: Removes files from the working tree and the index.
* **git mv**: Moves or renames a file, a directory.
* **git revert**: Reverts a commit by applying a new commit with the opposite changes.
* **git reset**: Resets the current HEAD to a specified state.
* **git tag**: Creates, lists, deletes, or verifies tags.
* **git stash**: Temporarily stores changes that are not ready to be committed.
* **git cherry-pick**: Picks a commit from another branch and applies it to the current branch.
* **git rebase**: Integrates changes from one branch into another by reapplying commits on top of another base tip.
* **git submodule**: Manages submodules within the repository.
* **git diff**: Shows changes between commits, commit and working tree, etc.
* **git log**: Displays the commit history.
* **git config**: Sets configuration variables for your Git installation.
* **git show**: Displays information about a specific commit.
* **git fetch**: Downloads objects and refs from another repository.
* **git remote**: Manages connections to remote repositories.
* **git clean**: removes untracked files from the working directory.
* **git commit --amend:** Passing the --amend flag to git commit lets you amend the most recent commit.
* **git reflog:** Git keeps track of updates to the tip of branches using a mechanism called reflog. This allows you to go back to changesets even though they are not referenced by any branch or tag.