GIT Commands

* git config
* git init
* git clone
* git add
* git restore
* git commit
* git diff
* git reset
* git status
* git rm
* git log
* git show
* git tag
* git branch
* git checkout
* git merge
* git remote
* git push
* git pull
* git stash

**git config:**

This command sets the author name and email address respectively to be used with your commits

Usage: git config –global user.name “[name]”

Git config –global user.email “[email address]”

Eg. git config –global user.name “Kunal”

git config –global user.email “[kishor.raj@wipro.com](mailto:kishor.raj@wipro.com)”

**git init**

This command is used to start a new repository

Usage: git init [repository name]

Eg. git init /home/kisraj/kunal/Demo/devops

**git clone**

This command is used to obtain repository from an existing repository URL.

Usage: git clone [URL]

Eg. git clone <https://github.com/kishor/examples.git>



**git add**

This command adds the file to the staging area

Usage: git add [file1][file2]….[file n]

git add .

git add \*

### ****git commit****

**Usage: git commit -m “[ Type in the commit message]”**

This command records or snapshots the file permanently in the version history.



**Usage: git commit -a**

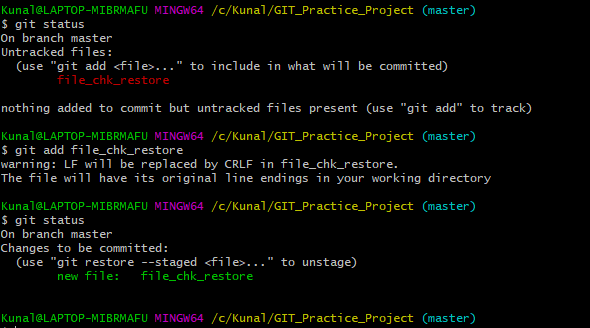
This command commits any files you’ve added with the git add command and also commits any files you’ve changed since then.

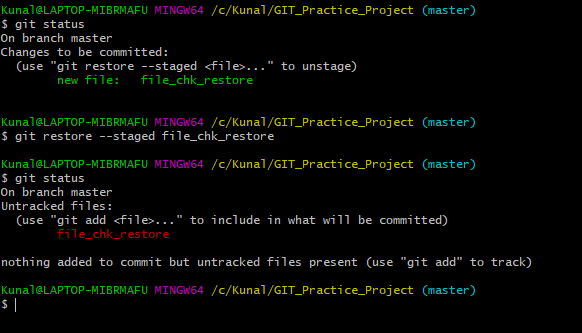
Git Commit Command - Git Commands - Edureka

**git restore**

**Usage: git restore –staged <file\_name>**

**This command is used to move the file to unstaged area from staging area i.e. after running the git add command.**





### ****git log****

**Usage: git log**

This command is used to list the version history for the current branch.



**Usage: git log –follow[file]**

This command lists version history for a file, including the renaming of files also.



### ****git diff****

**Usage: git diff**

This command shows the file differences which are not yet staged.



**Usage: git diff –staged**

This command shows the differences between the files in the staging area and the latest version present.



**Usage: git diff [first branch] [second branch]**

This command shows the differences between the two branches mentioned.



**Usage: git diff <commit-id of version X> <commit-id of version Y>**

Commit-id can be obtained from git log

### ****git reset****

**Usage: git reset [file]**

This command unstages the file, but it preserves the file contents.



**Usage: git reset [commit]**

This command undoes all the commits after the specified commit and preserves the changes locally.

Git Reset Command - Git Commands - Edureka

**Usage: git reset –hard [commit]**

This command discards all history and goes back to the specified commit.

Git Reset Command - Git Commands - Edureka

### ****git status****

**Usage: git status**

This command lists all the files that have to be committed.

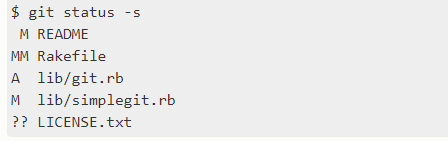


**Usage: git status –s**

If you run git status -s or git status --short you get a far more simplified output from the command.

* ?? : New files that aren’t tracked have a ?? next to them,
* A : New files that have been added to the staging area have an A,
* M : Modified files have an M.

There are two columns to the output — the left-hand column indicates the status of the staging area and the right-hand column indicates the status of the working tree.



### ****git rm****

**Usage: git rm [file]**

This command deletes the file from your working directory and stages the deletion.

Git Rm Command - Git Commands - Edureka

### ****git show****

**Usage: git show [commit]**

This command shows the metadata and content changes of the specified commit.



### ****git tag****

**Usage: git tag [commitID]**

This command is used to give tags to the specified commit.



### ****git branch****

**Usage: git branch**

This command lists all the local branches in the current repository.

Git Branch Command - Git Commands - Edureka

**Usage: git branch [branch name]**

This command creates a new branch.

Git Branch Command - Git Commands - Edureka

**Usage: git branch -d [branch name]**

This command deletes the feature branch.

Git Branch Command - Git Commands - Edureka

### ****git checkout****

**Usage: git checkout [branch name]**

This command is used to switch from one branch to another.

Git Checkout Command - Git Commands - Edureka

**Usage: git checkout -b [branch name]**

This command creates a new branch and also switches to it.

Git Checkout Command - Git Commands - Edureka

### ****git merge****

**Usage: git merge [branch name]**

This command merges the specified branch’s history into the current branch.

Git Merge Command - Git Commands - Edureka

### ****git remote****

**Usage: git remote add [variable name] [Remote Server Link]**

This command is used to connect your local repository to the remote server.

Git Remote Command - Git Commands - Edureka

### ****git push****

**Usage: git push [variable name] master**

This command sends the committed changes of master branch to your remote repository.



**Usage: git push [variable name] [branch]**

This command sends the branch commits to your remote repository.



**Usage: git push –all [variable name]**

This command pushes all branches to your remote repository.



**Usage: git push [variable name] :[branch name]**

This command deletes a branch on your remote repository.



### ****git pull****

**Usage:  git pull [Repository Link]**

This command fetches and merges changes on the remote server to your working directory.



### ****git stash****

**Usage: git stash save**

This command temporarily stores all the modified tracked files.

Git Stash Command - Git Commands - Edureka

**Usage: git stash pop**

This command restores the most recently stashed files.



**Usage: git stash list**

This command lists all stashed changesets.

Git Stash Command - Git Commands - Edureka

**Usage: git stash drop**

This command discards the most recently stashed changeset.

Git Stash Command - Git Commands - Edureka