**GIT**

Git is created by a person named “**LINUS** **TORVALD**” who created Linux kernel

**Why Git?**

For example a client has come to you and asks you to develop a software. Client gives all his requirements and you develop a software which satisfies all his requirements and client is satisfied with the product. Let us suppose this as a version 1.0. After some days client comes to you and asks for some updates and enhancements and you develop code and deploys the product to the client. Client after testing its features he comes to a conclusion that this version was not good and he want to use the previous version. So, if you don’t maintain the backup of previous version if will be difficult for you to build again from scratch .so here comes the concept of version control which is provided by Git. Suppose let us analyze a situation where we don’t have the concept of git.

Let us suppose that there is product currently running with its version of 2.0 and 1.9, 1.8, 1.7, Are its prior versions. After some days if client wants to use version 1.2 or something it will be very difficult to maintain all those backups and we have to maintain all the data such as in which version which feature is developed. For this reason we use the concept called Git.

**Definition:**

Git is a distributed version control system i.e., every developer will have their own copy in their local repository apart from the remote server so that they need not to communicate the central server.

Git is an application that keeps track of changes (text changes) that are made and move back and forth to utilize a particular version. It is referred as Version control System and Source code management. Git is similar to undo and redo mechanism.

**Installation:**

**Step:-1**

<https://git-scm.com/downloads>

Go to above link and select your operating system and start downloading git.

**Step:-2**

After downloading git double click on .exe file that you have downloaded.

**Step:-3**

After that allow the default settings and click on next until the installation is completed and finally click on finish.

These are steps to install git.

So to check whether git is successfully installed in your system or not, open command prompt and type the following command

**git –version**

If there is any error like “git is not recognized as an internal or external command, operable program or batch file” then git is not successfully installed on your computer.

**Commands in GIT:**

1. **Git config**

The git config command is used to set git configuration values.

Example:

**git config –global user.name “dheeraj”**

**git config –global user.email “**[**dheeraj@epam.com**](mailto:dheeraj@epam.com)**”**

There are three configuration levels in git they are

Global, local, system.

1. If you want to see the configuration settings of git then you need to type the following command.

**git config –list**

The above command lists all the configuration settings of git.

1. Instead of checking the entire settings of git, if you want to see only a single setting then you have the following command.

**git config –user.name**

**git config –user.email**

1. If you want to know about any command you have to use the following command

**git help**

The above command will display the list of all existing commands

**git help commit**

The above command open the browser which contains the html doc about commit command

1. **Git init**

The git init command is used to create new git repository.

After executing the **“git init”** command it creates .git subdirectory in the current working directory which contains all the necessary git metadata for the new repository.

.git directory consists of following sub-directories

/HEAD

/branches

/config

/description

/hooks

/info

/objects

/refs

1. **Git add**

The git add command adds the changes in the current directory to the staging area.

**git add .**

**git add <file>**

The above command stage all changes in the specified file.

**git add <directory>**

The above command stages all the changes in the specified directory.

1. **Git commit**

Git commit takes the snapshot and saves it.

**git commit**

The above command simply commits with out any message.

**git commit –a**

Commit a snapshot of all changes in the working directory. This only includes modifications to tracked files (those that have been added with git add at some point in their history).

**Git commit –m “message”**

A shortcut command that immediately creates a commit with a passed commit message. By default, git commit will open up the locally configured text editor, and prompt for a commit message to be entered. Passing the -m option will forgo the text editor prompt in-favor of an inline message.

1. **Git status**

The git status command displays the state of the working directory and the stating area. It lets you see which changes that have been staged and which are not staged and files which are not being tracked by git.

1. **Git log**

The git log command displays the committed snapshots.

**Branching commands in git**

Branch is an independent line of development. By default branch is master.

A new branch can be created using command

**git branch <name>**

Create a new branch and switch to it:

**git checkout -b <branchname>**

Switch from one branch to another:

**git checkout <branchname>**

List all the branches in your repo, and also tell you what branch you're currently in:

**git branch**

Delete the feature branch:

**git branch -d <branchname>**

Push the branch to your remote repository, so others can use it:

**git push origin <branchname>**

Push all branches to your remote repository:

**git push --all origin**

Delete a branch on your remote repository:

**git push origin :<branchname>**