edWisor

Design Basics - GIT

- Aditya Kumar
Chief Technology Officer, edwisor.com

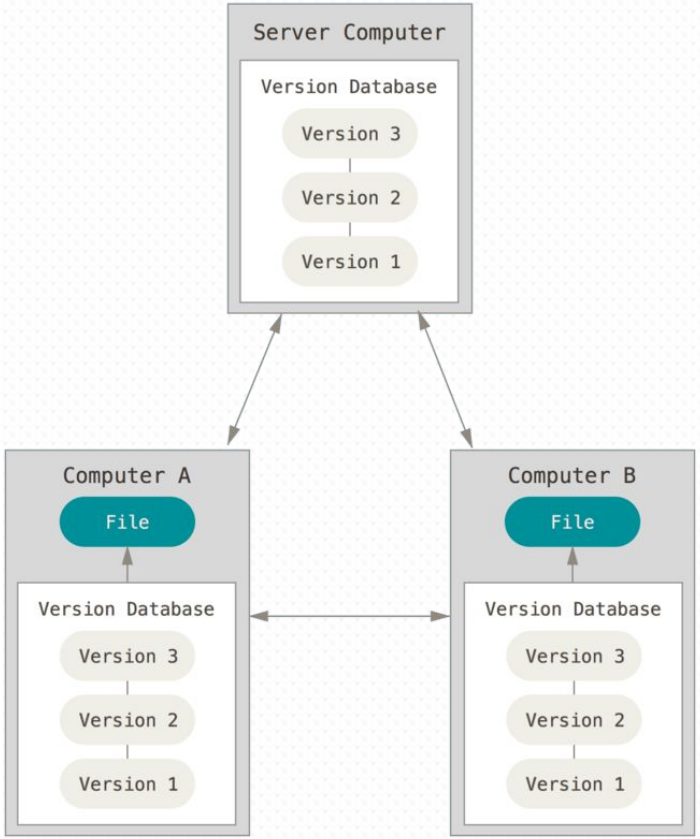
We will cover with the following concepts

1) Introduction to Git

2) Basic Git commands

3) Introduction to Github

How Distributed Version control systems generally work

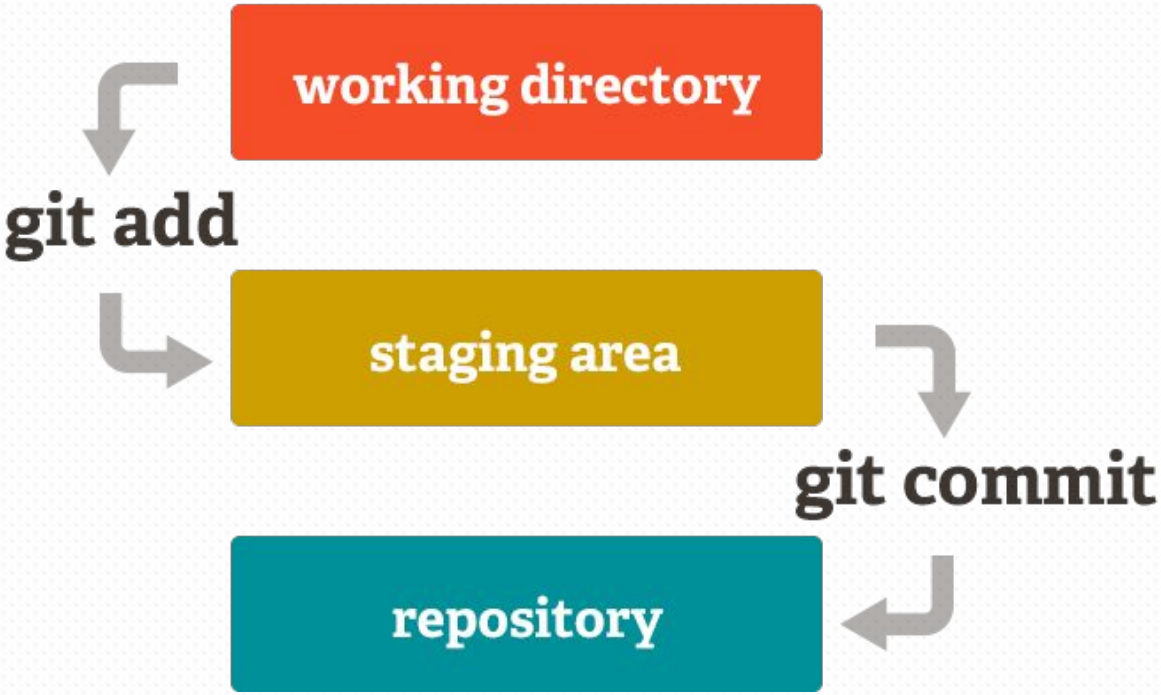


https://git-scm.com/book/en/v2/book/01-introduction/images/distributed.png

GIT is the most popular VCS in the world

- Git allows groups of people to work on the same documents (often code) at the same time, and without stepping on each other's toes
- Its free and open and designed to handle both small and large projects efficiently
- It's easy to learn , fast and secure.

The process flow is simple



Github is the most popular online distributed version of GIT

- **GitHub** is a web-based Git repository hosting service. It offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features.

- It provides access control and several collaboration features like bug tracking, features request, task management and wikis

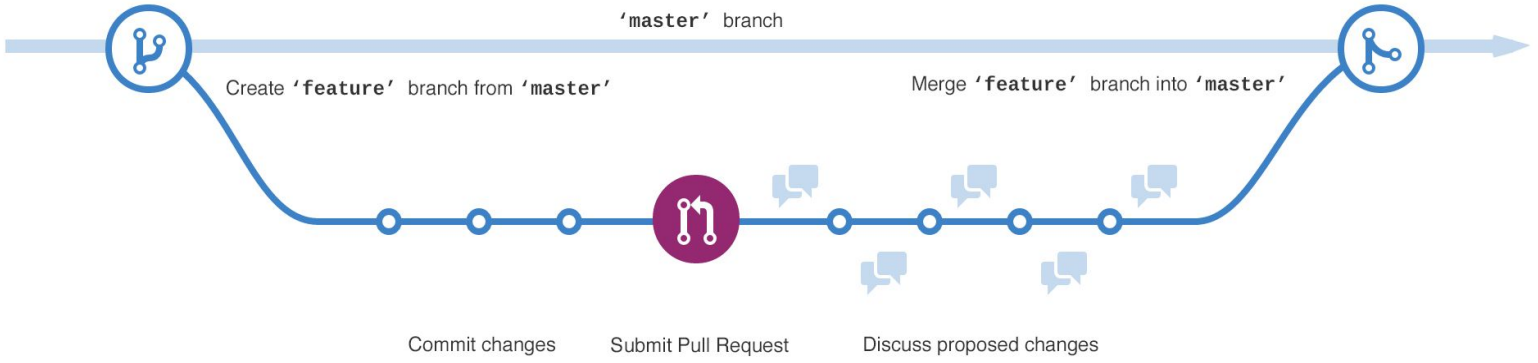
Repository and branches make collaboration very easy

- A **repository** is usually used to organize a single project. Repositories can contain folders and files, images, videos, spreadsheets, and datasets – anything your project needs.

- Branching is the way to work on different versions of a repository at one time

- By default your repository has one branch named **master** which is considered to be the definitive branch. We use branches to experiment and make edits before committing them to master

Repository and branches make collaboration very easy



Lets create a repository and add our code to it

1) First install Git in your system

2) Then run the command "**git init**" to initialize an empty repository. Your folder where you ran this command becomes your repository

3) Then run "**git add --all** " to add all your work to staging area

4) Run "git commit -am <some name>" to store our changes in staging environment.

   Now it's time to push it to a remote repository

Go to Github.com and create an account

1) Create an account on github.com and then create a repository

2) Copy the repository url and then run the command "**git remote add origin <repository url>**" in your folder.

3) Then run "**git add --all** " to add all your work to staging area

4) Run "**git push origin master**" to push your folder's code into the master branch of your remote repository

5) If someone else is working on same project, or you want to copy this pushed version of code somewhere, you can pull it using - "**git pull origin master**"

There are some more commands that really come in handy

1) **Git status** - to see the current status of the project

2) **Git log -** it's a journal that remembers all the changes we've committed so far

3) **Git diff <Head>-** check what is different from our last commit. Head is the pointer of our last commit

4) **Git reset <filename>** - you can unstage files by using git reset command.

5) **Git checkout --<text>-** Files can be changed back to how they were at the last commit by using the command

What if you want to keep certain files out of git add or git commit

1) You can declare a file called **.gitignore** and name the things you want to ignore

2) Let me show you a basic example

3) You can refer to the official documentation here for more details

We are not covering branching right now!

1) Based on the devops practices in your company, branching may be used in various ways.

2) You are free to explore the concept of branching yourself through official documents. But it is beyond the scope of this program and you will not need it at any level, even during evaluation.

Next Steps -

1) Making your website live on Github.IO