Git & GitHub Workshop

Presenter: Saroj Maharjan

Slides by: Bishal Rana

(Patan IT Club)

What will we learn today?

1. Install git and create a Github account
2. What is git?
3. Git workflow : The three States
4. Let’s get started practising git and github
5. Useful links / resources

Install git & create a github account

Install git on your machine (linux / mac / windows)

Install git on your machine

- Command (debian): **sudo apt-get install git**
- Command (fedora): **sudo yum install git**
- Mac:

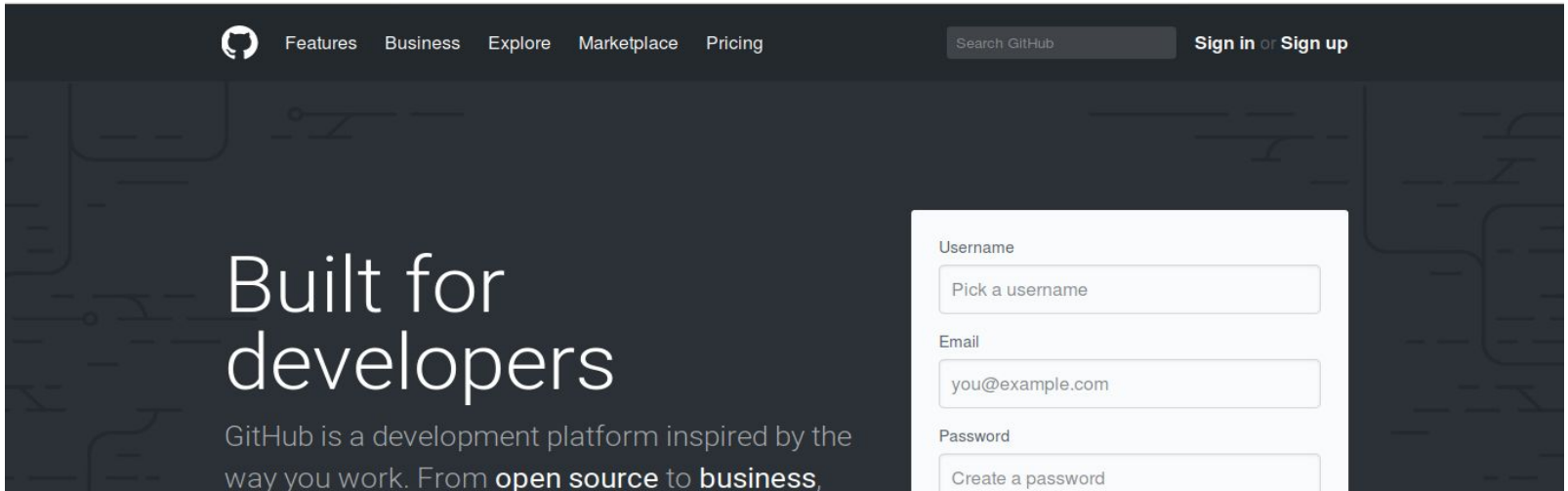
  http://git-scm.com/download/mac
- Windows:

  http://git-scm.com/download/win

Create your github account

- www.github.com

- Free for public repositories



What is git ?

A distributed Version Control System...

What is git ?

- An example of **Version Control (VC)**
- VC is a system that records changes to a file or a set of files over time so that you can recall specific versions later.

Git allows you to:

- Revert files to previous state
- Revert entire project to previous state
- Compare changes over time
- See who modified what...and much more
- It means if you screw things up or lose files, they can easily be recovered

Other Version Control Systems

- Subversion (SVN)
- Concurrent Version Systems (CVS)
- Perforce
- Mercurial
- Bazaar...and many more.

However, git is the most popular one.

So, why is git so popular ?

- Allows individual as well as collaborative development
- Offline usage
- **Distributed VCS, not centralized**
- Relatively faster than other VCS's
- Everything is local

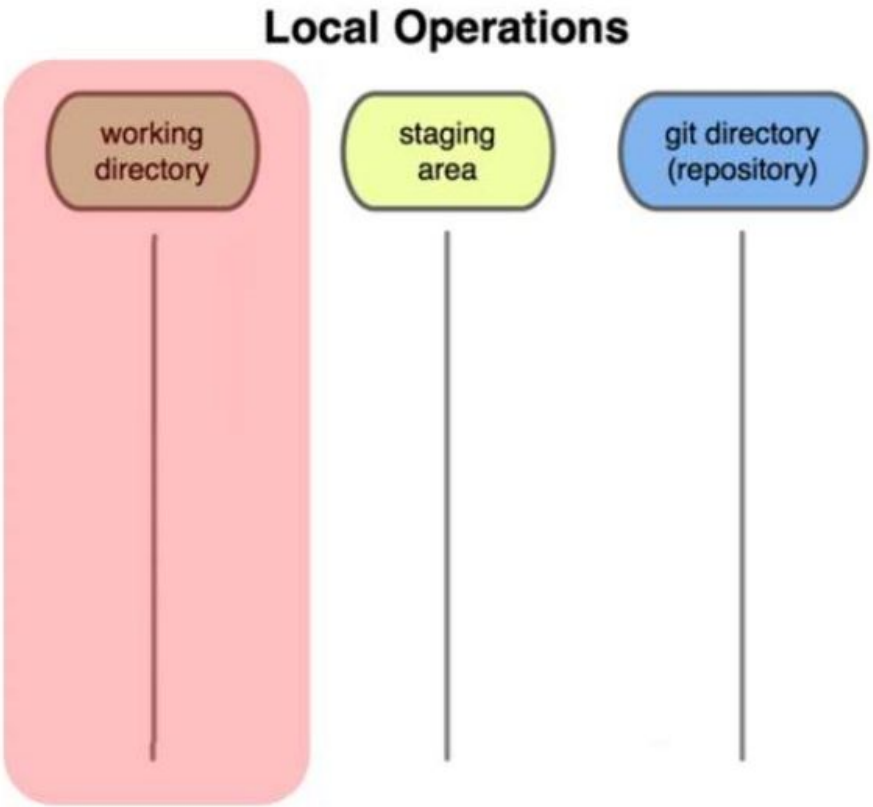
Git Workflow : The three States

Modified, Staged and Committed

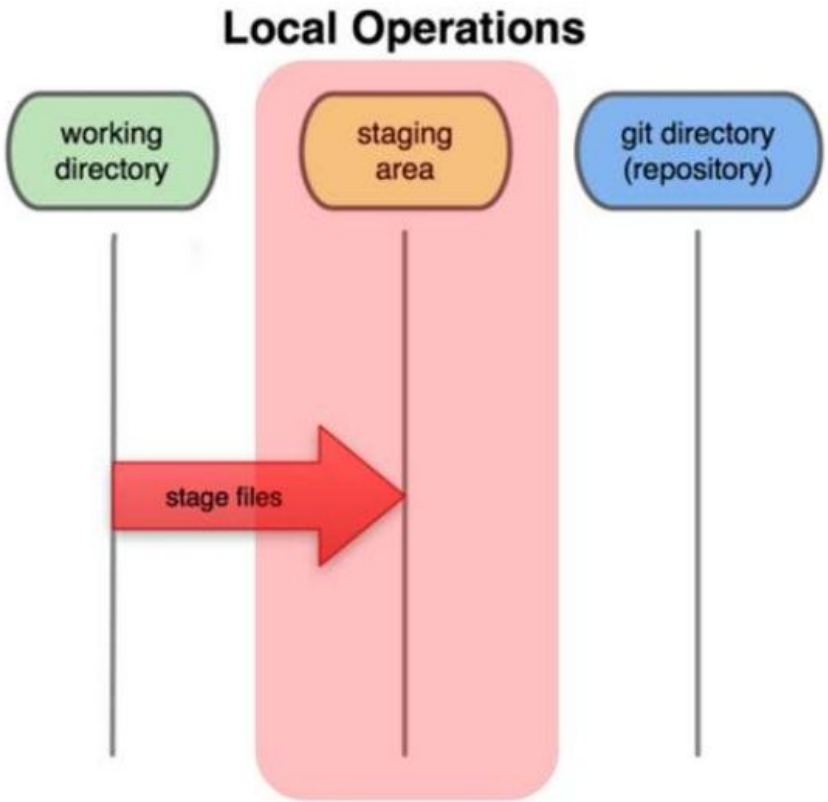
Git Workflow : The three States

In a git repository (repo), your file can reside in three main states:

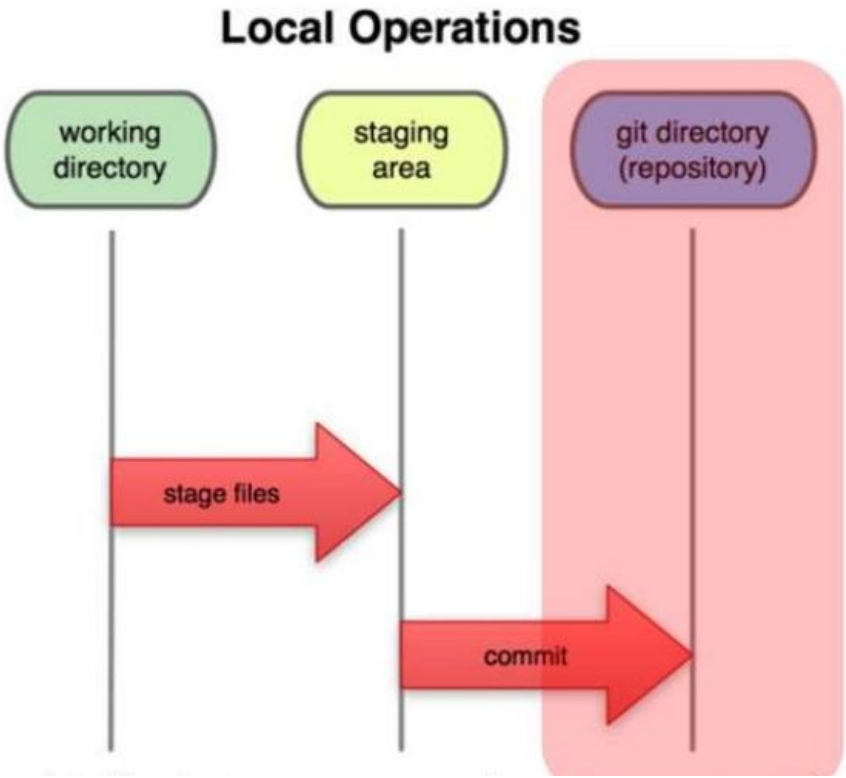
- Modified
- Staged
- Committed



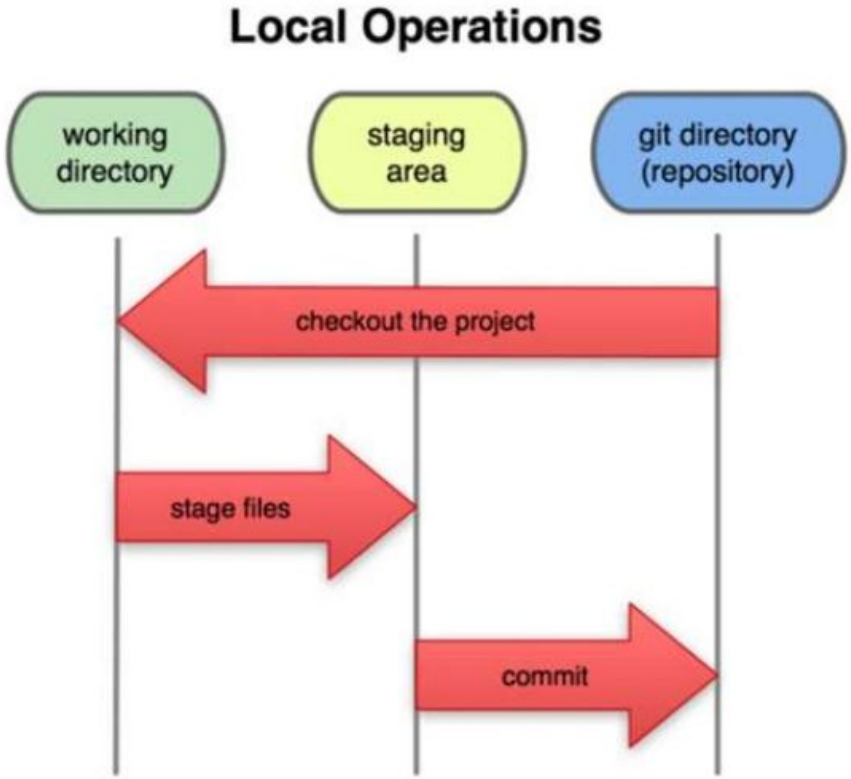
You modify files in your working directory



You stage the files, adding snapshots of them to your staging area



You do a commit, that stores snapshot permanently to your Git directory



Then, you can checkout any existing version, make changes, stage them and commit.

Let’s get started practising git

Time to put the theory we learnt into practise

First time git setup

git --version (Check the git version if installed)

In cmd / git bash or other terminal :

- git config --global user.name ‘Your name’
- git config --global user.email ‘Your email’)

It’s better to use git bash for git.

Create a demo project

Create a demo project. For example a **myApp** folder with the following files:

- index.html
- style.css

You can use your own project / directory.

Initialize local git repository

Open the myApp directory in the terminal and initialize local git repository with:

git init

(This basically is saying 'We are about to use git in this directory' and creates a hidden .git directory)

Remember the three States ?

The first state is ‘**modified**’ state. This means nothing; you are working on your project locally. You are coding, programming or editing files.

(Just add something to index.html or style.css or both)

The '*Staged*' State

The second state is '**staged**' state. This is generally the intermediate state where your file stays before committing. Staging a file means that you are just saving your file and can edit in the future.

The Staging syntax

- git add index.html / git add *.html
- git add index.css / git add *.css

OR

- git add . (This adds everything in your directory at once)

More on staging

Now, modify one of your files and check the status using:

git status

Then, again:

git add .

The *Commit* State

Committing your files means that you are permanently saving files in your local git repo (the hidden .git folder created in the beginning)

The Commit syntax

- git commit -m ‘Initial Commit’

(This will commit your your file with the message or information ‘Initial Commit’)

- git status (Run this to see the status of your working tree quite often)

Play around the 3 states (Summary)

Modify your files, add new files and play around with syntax you learnt.

- git status
- git add .
- git commit -m ‘Second commit’
- git status
- git log (To see your commit log)

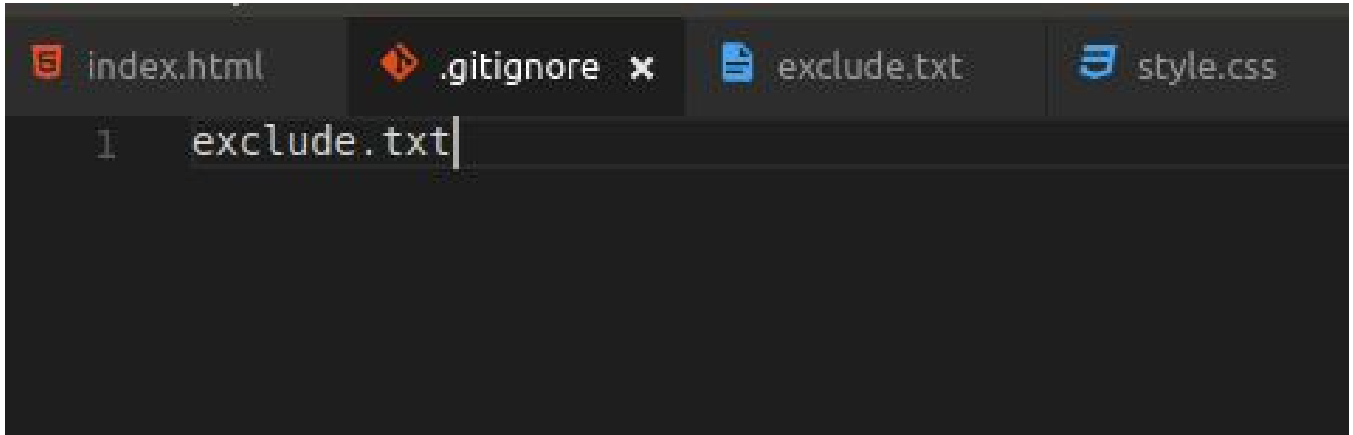
The .gitignore file

Often you don't want to put everything in your directory to the staged state or committed state. That's where **.gitignore** comes in. Create a .gitignore file from your git bash:

touch .gitignore

The .gitignore file

Create a new file (for example: exclude.txt) that you don't want to commit. Edit that file by adding some text. Then in .gitignore file:



The .gitignore file

Now when you check your git status using:

git status

The exclude.txt won’t show up because .gitignore says to ignore it. However, .gitignore needs to be staged and committed. Do it yourself.

Branching and Merging

Currently, we are working on the main tree (also called master branch). However, often we want to work separately on a specific part of our app without affecting the main tree. This is called branching.

(So, let's create a separate branch called login.)

Branching and Merging

- git branch login (Creates a new branch)
- git checkout login (Go to checkout branch from master branch)
- git status (Check on which branch we are on)

Branching and Merging

- Create new file login.html in myApp directory. (touch login.html)
- git add .
- git commit -m ‘login feature added’
- git checkout master

Branching and Merging

Finally, while on the main branch (master), merge login with master with:

git merge login

Working with github

Finally we can push our project from our local repo to a remote repo.

We will use github, but you can use others like ***GitLab, BitBucket, SourceForge, Beanstalk, Cloud Source by google, GitKraken, AWS CodeCommit*** and many more.

Working with github

- Create a github account and create a new repository with any name (eg: gitdemo)
- Then,

```
bishal@bishal:~/gitDemo$ git remote add origin https://github.com/rbishal50/gitdemo.git
bishal@bishal:~/gitDemo$ git push -u origin master
Username for 'https://github.com':
```

Working with github

Now add more files, edit files locally, modify, stage and commit them locally. Then to push to a remote server do:

git push

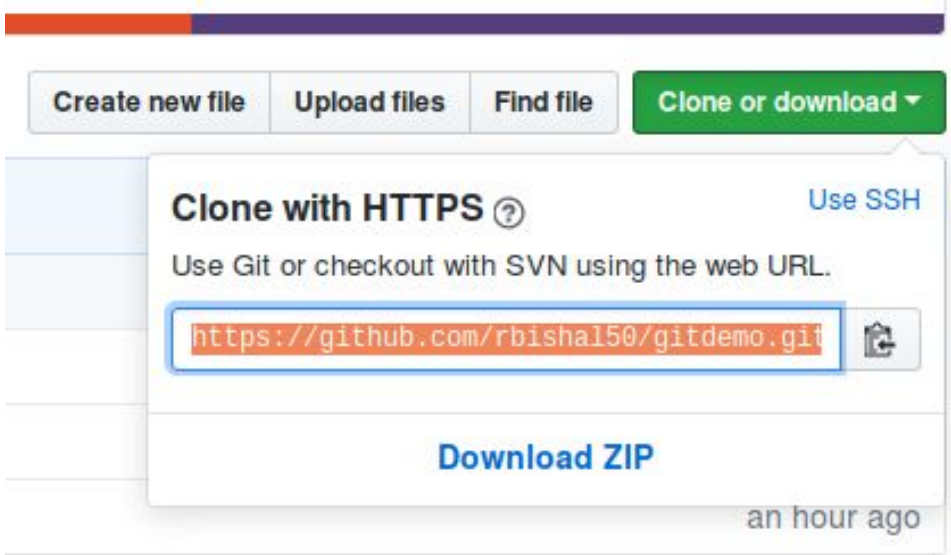
Pull files you have uploaded using

git pull

Working with github

You can clone or download the public open source repositories of others using:

git clone <url>



Useful links / resources

Learn more about git here

Useful links / resources

- **Official git site and tutorial**

  (https://git-scm.com/)
- **Github Guides**
- (https://guides.github.com)
- **Interactive git tutorial**

  (https://try.github.io/levels/1/challenges/1)

Thank you for your patience.

‘Patan IT Club’