**Why version control system is required = GIT, SVN, TFS, VSS**

* Revert to previous version
* Record each and every change made to files

**What developer used to do, in the absence of version control system**

* Create directory and make folder date and time wise and store the code into it.

**Need centralized repo**

Push changes = checked in = pushing local code to centralized repo

Fetch changes = check out = getting code from centralized repo

**Limitation of centralized repo = TFS, SVN, VSS**

if server is down, means repo is down, then developer won’t be able to perform check-in and check-out.

If hard-disc of central server gets corrupted, then entire history of the project will be gone and we will loose everything

To avoid this, we go with distributed repo



**Distributed version control system = GIT**

When we fetch changes from repo, we actually have all the mirrors of the project on local machine in the hidden folder and in the compressed form. Can say every checkout is the full backup

It overcomes the limitation as we have centralized repo, because each developer holds complete mirror of project which can act as backup. In case of issue, take backup and restore it on the server.



We can perform check-in and check-out locally without internet, so to share code with other team, we have to connect with internet and push the changes over repo

**What is GIT and GIT Hub.**

GIT is the software, which we install on local machine and GIT HUB is that centralized repo, where we push our project code.

**Fork me on git hib.**

Means to push the entire project code to git hub.

<http://selendroid.io/> = it is the open source project. U want entire project code in your repository for further customization, then click on “fork”

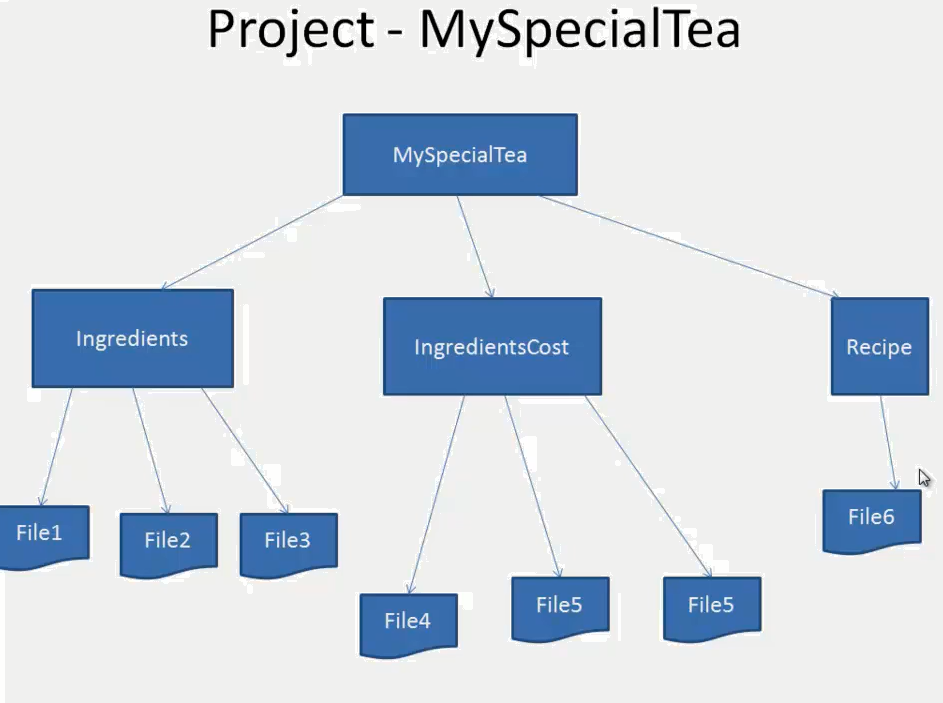
**GIT Installation**



**How GIT stores data**

* It stores the data in the form of snapshots just like say as “re-vision” in other VCS
* On every commit a snapshot of each changed file is taken and stored in the local GIT repo

**Create below sample project**



**Creating GIT Repo**

There are two ways

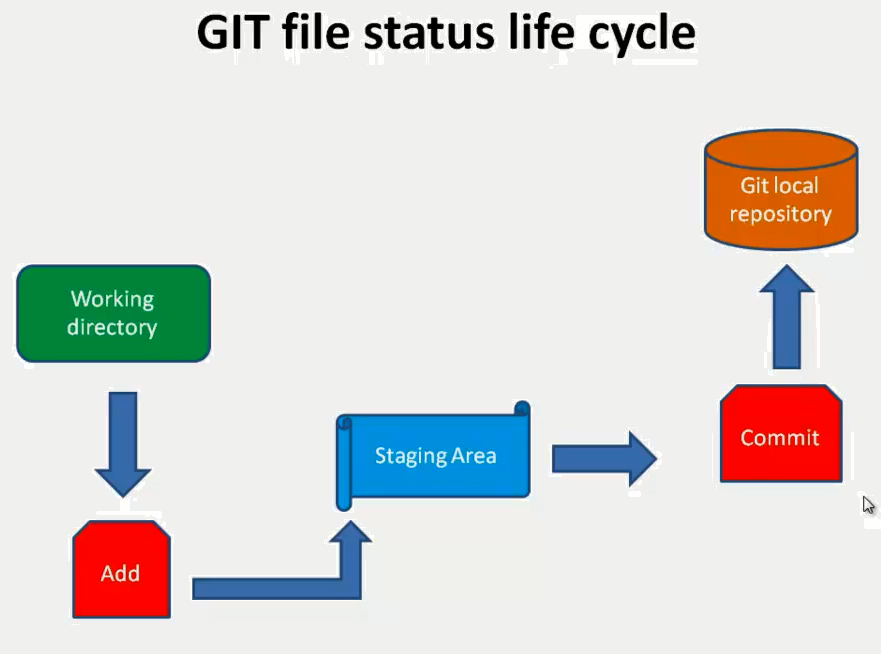
1. Create git repo in existing project
2. Create a clone of entire git repo from the remote server

**Create git repo in existing project**

* Go into ur root project folder
* Command => git init (creates hidden .git folder = that track and maintain the history of the project files)

**Git commands**

* Git add \*.txt or git add \* or git add <file name> = once added, it is moved to staging area
* Git status
* Git commit –m “<commit message>” = once committed, it moves the files from staging area to git local repository



**Why Remote Repostories = GIT HUB**

* When more than one developer working on same project
* Act as single centeral repo which will further act as hub of communication and collaboration

**What is GIT Hub**

It is the web hosting service, whose name is GIT repository and known as centralized repo. If project is in GIT hub, then only we can share the project code with other team mate

**Register with GIT HUB**

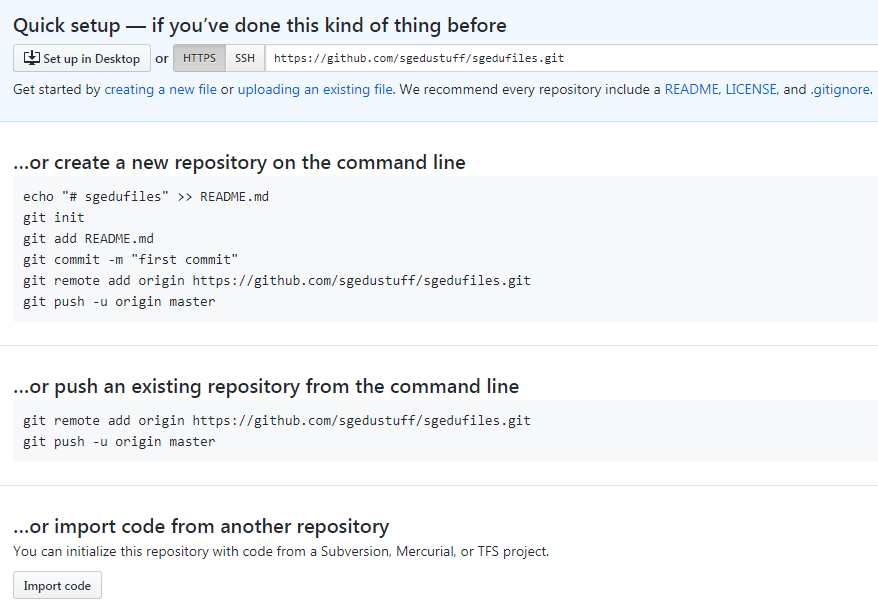
GIT HUB provides two types of repo which are private repo and public repo. Private repo is paid

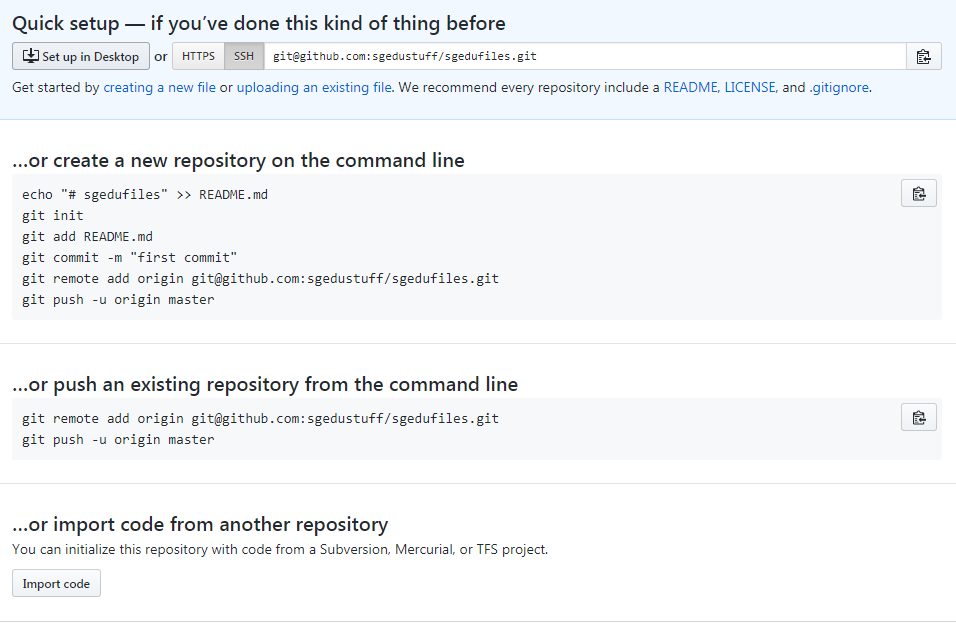
Do signup at <https://github.com/>

**How to move local git repo to git hub repo**

Signup

Create repository. After creating u will see below thigs





**Setting up public and private key** so that we can push and fetch the changes from local got repo to git hub repo and vice versa. Once this is done, it will not ask git hub user id and password while pushing or fetching the changes. This is the one-time setup

ssh-keygen -t rsa (it generate public and private key)

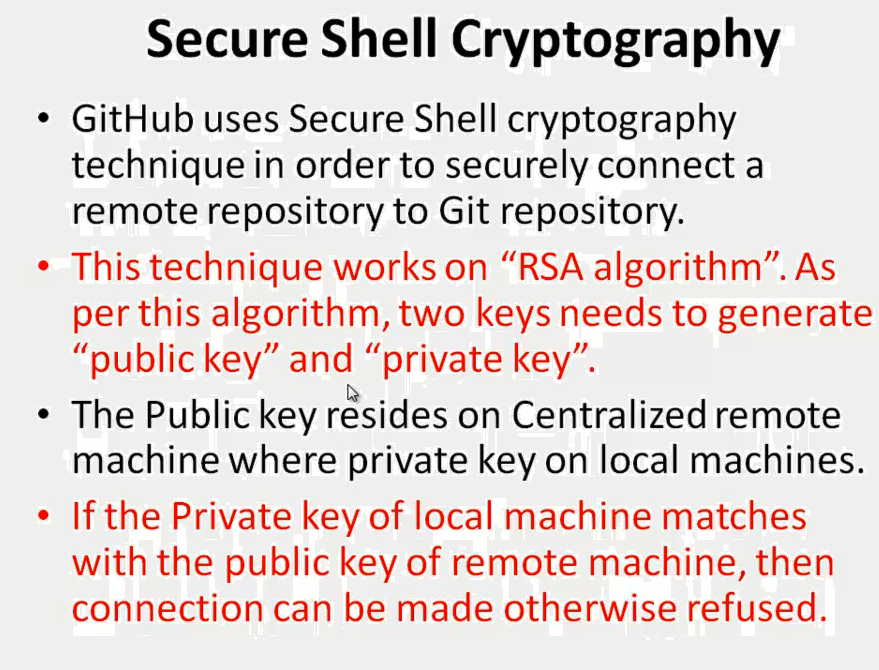
Copy the public key content

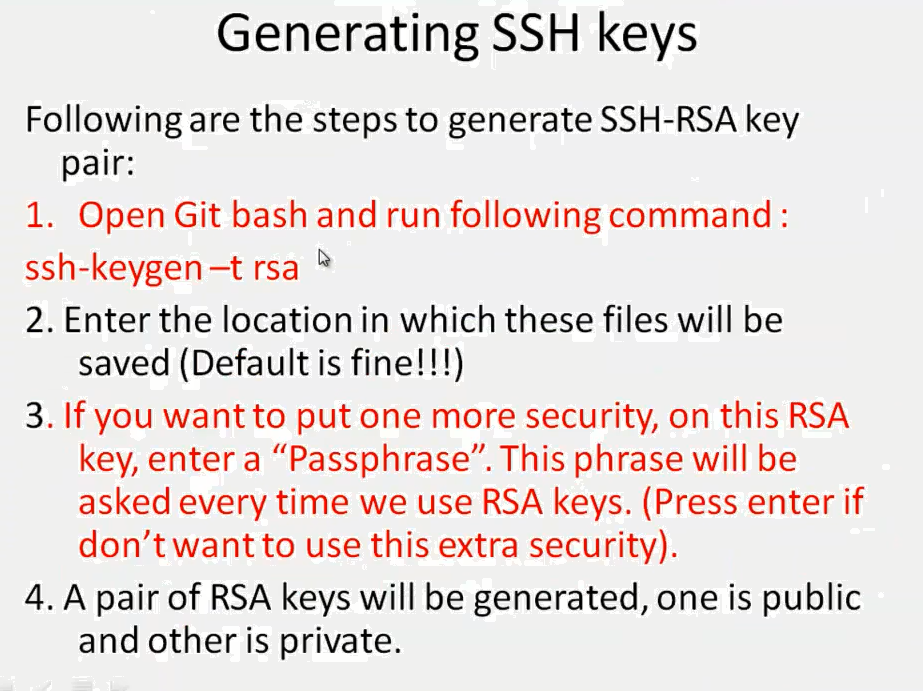
go to your git hub account settings

select SSH section

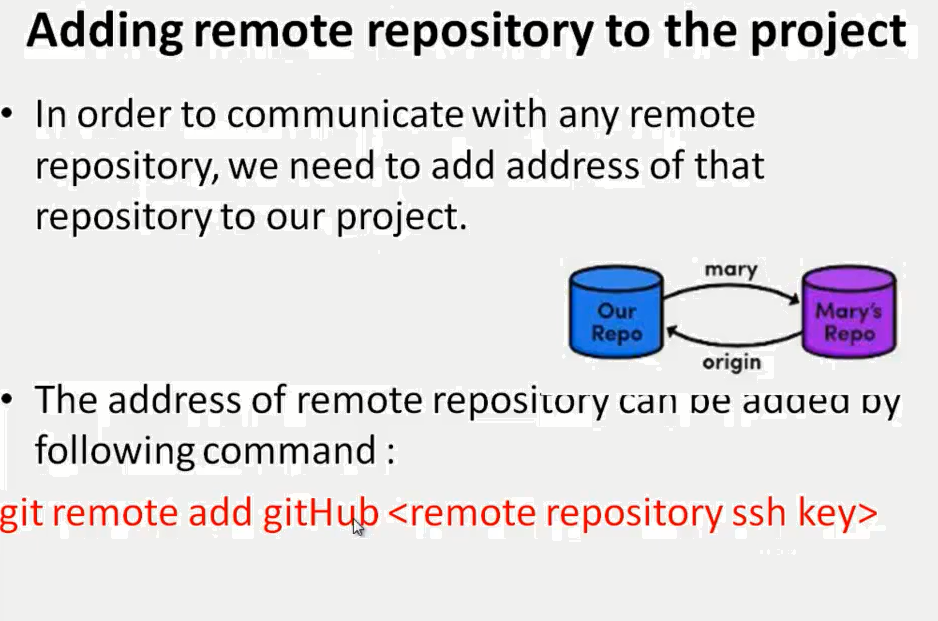
create new SSH Key = paste the public key here

ssh -T git@githum.com = adding public key permanently to git hub









**push code from local git repo to git-hub repo**

git init

git add \*

git status

git commit -m "<any message>"

git remote add origin git@github.com:sgedustuff/sgedufiles.git

git push -u origin master