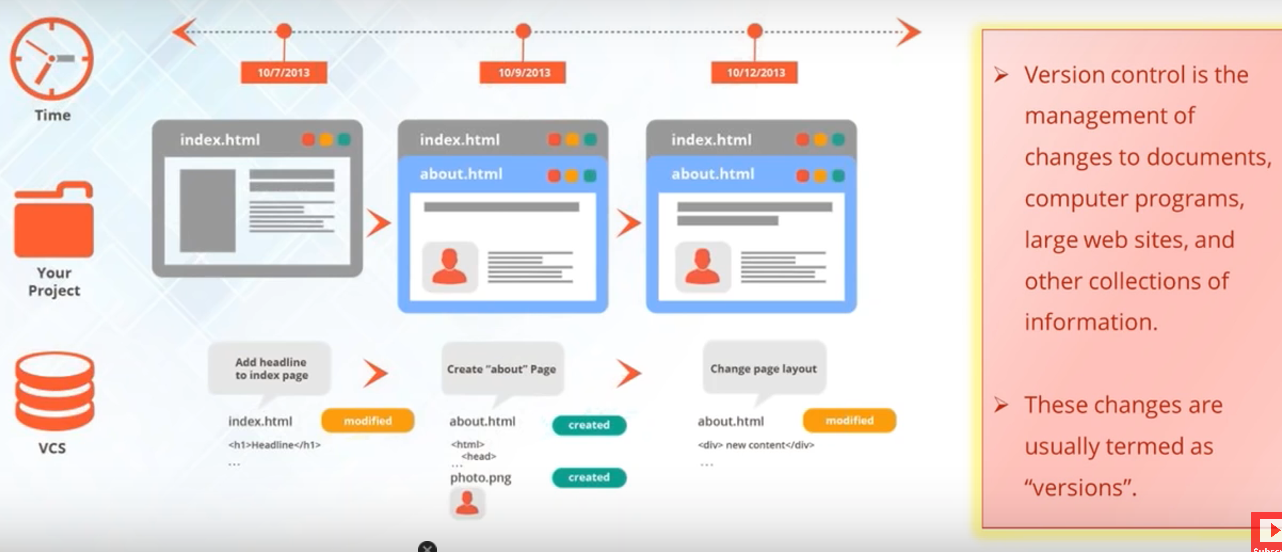
Upload code to github, do changes, can backtrack ur changes.

Can add bridges, diff codebase: master version, branches. Merge

**Version Control System**

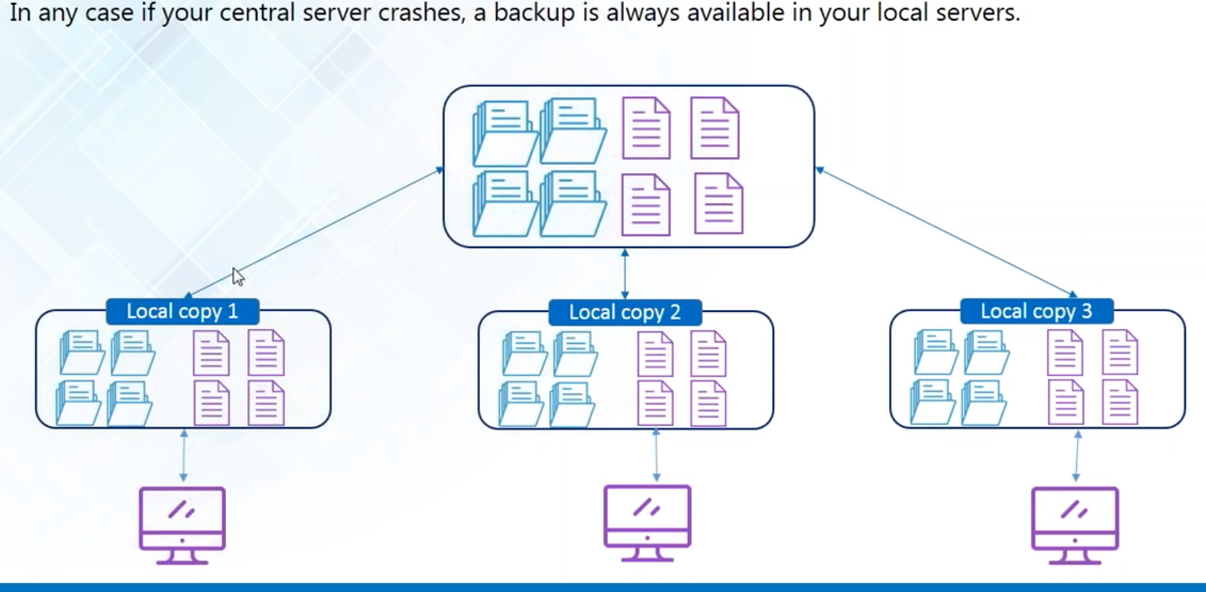
Management system that manages the changes we did to our system.



In beginning had index.html. Added about.html to it. Version control system detects something modified. Later if we change about.html, will detect the change. All snapshots will be saved as diff versions. Version1, version2, version3.

Version control:

1. Collaboration: diff developer working on diff modules. Remove conflicts while merging.
2. Storing versions. Snapshot of all versions are properly documented and stored.

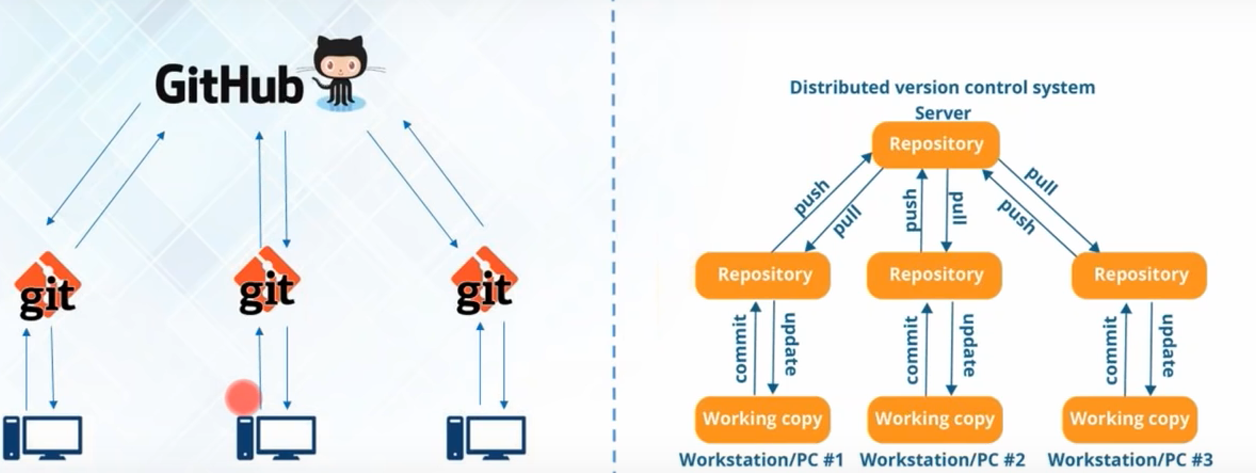


1. Very reliable
2. Analysis of what was changed, when it was changed with proper description. Timeline.

**Version Control Tools**

1. Git: provides developers with local copy, open source, most popular version control tool
2. Subversion (svn): do not provide developers with local copy
3. Cvs: : do not provide developers with local copy: not that popular
4. Mercurial

**Git and GitHub**



Repository: data space where all projects are stored. First changes made to local repository then pushed to central repository. (Pull: to fetch data from repository)

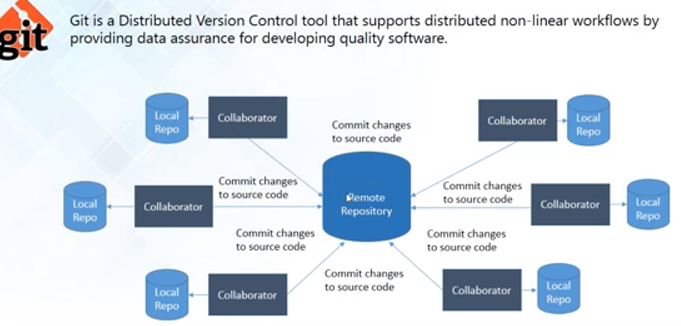
Github: central repository: code hosting platform. Host your central repository in remote server, used to share code.

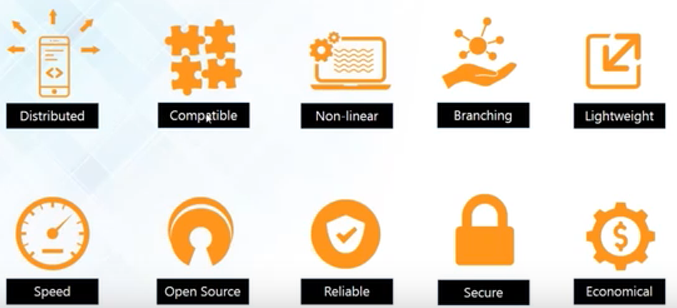
Git: version control management tool enables to create local repository, to push and pull data.

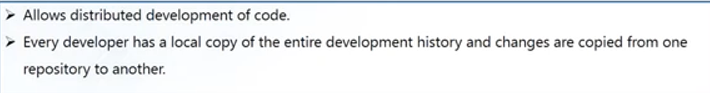
Distributed version control system, to work in isolation and push changes when done.

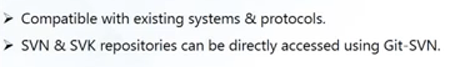
**Git**

Distributed version control tool.











Creates tree graph from the index (non-linear), acyclic graph.



Git only has multiple branches.



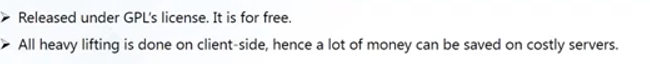


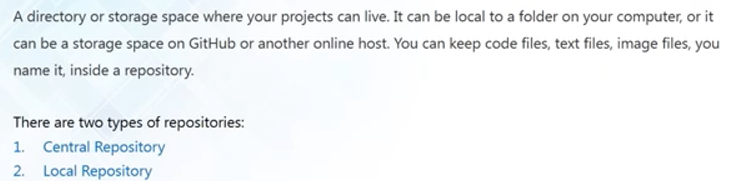


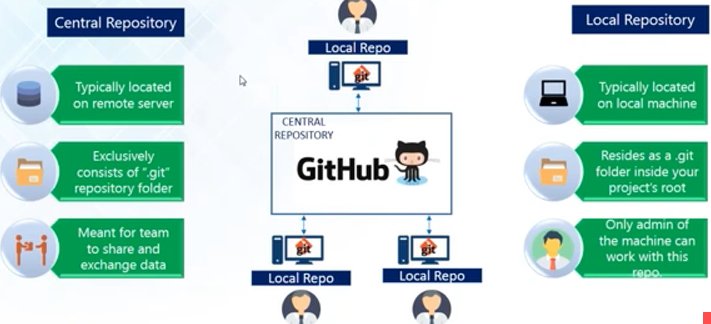




After made changes, cannot deny. Commit hash is generated and can be used to rollback your changes.

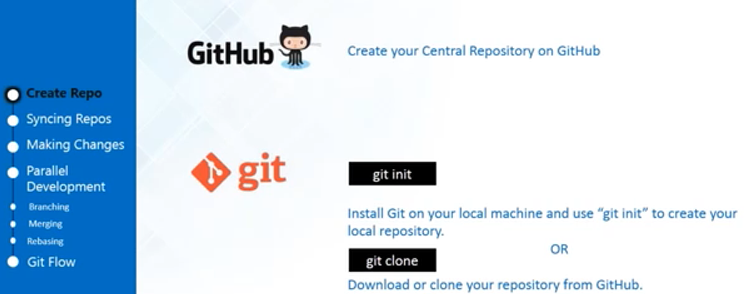






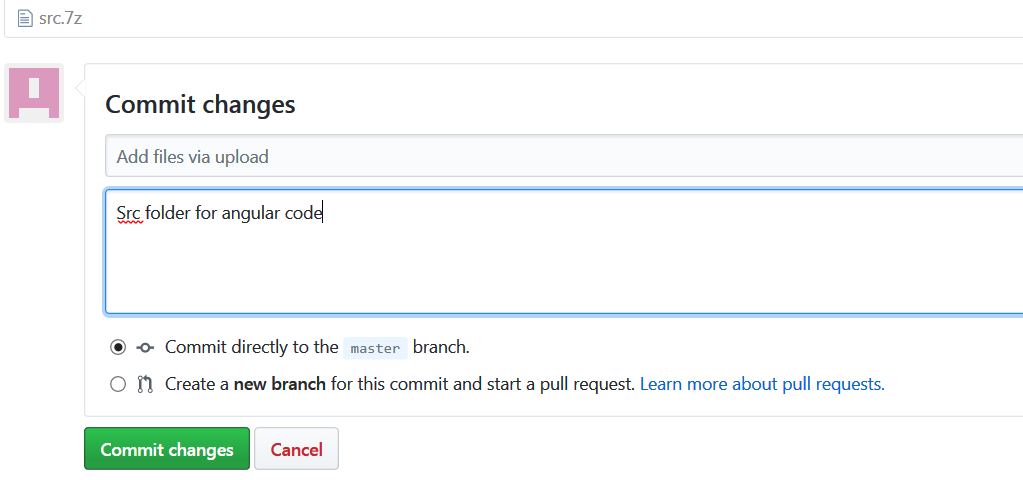
**Git Operation and Commands**





Github -> Start a project -> New Repository





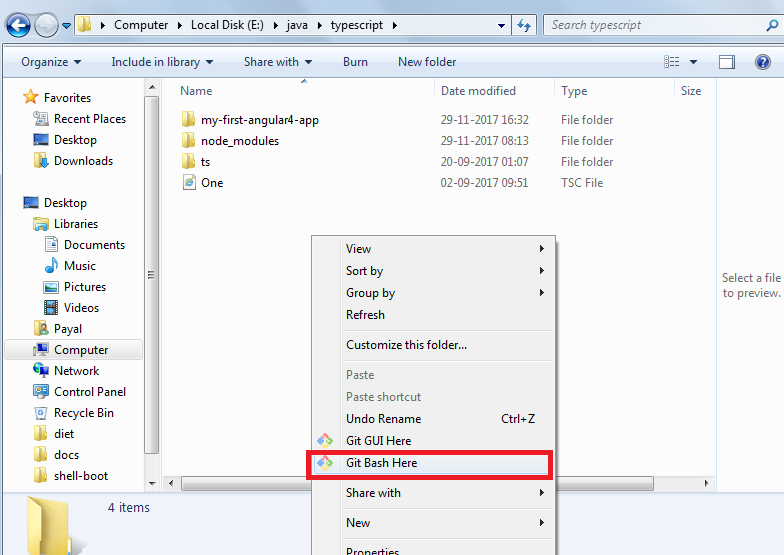
Created central repository. Now create local repository.

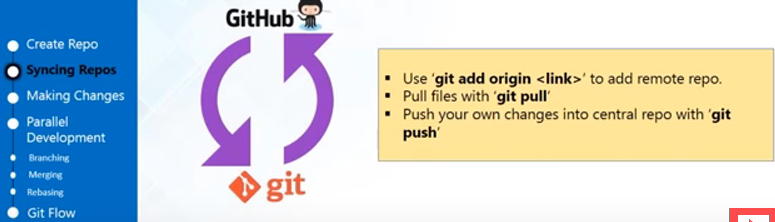
**For local repository:**

Choose a folder where we want a local repository and Use “Git bash here” Will open git bash emulator

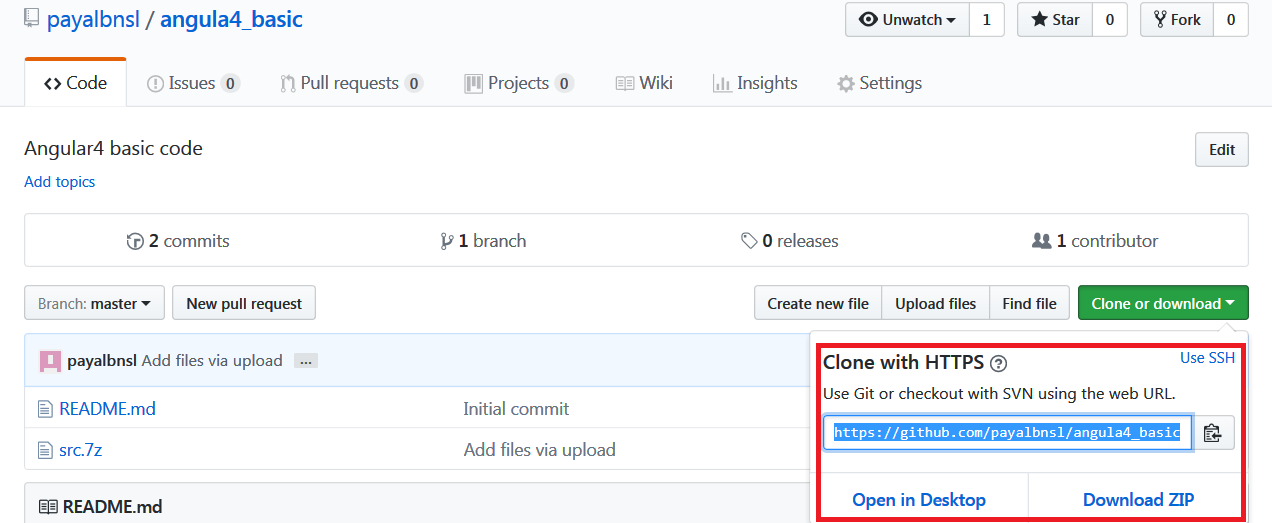
To create local repository.

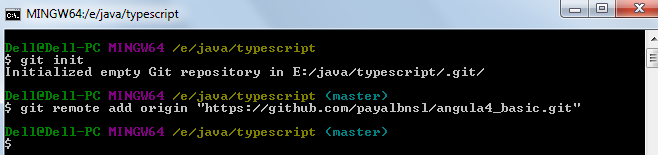
1. Type git init: .git folder created there.



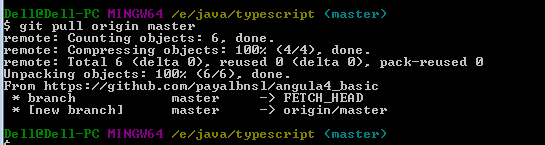


Repository link:



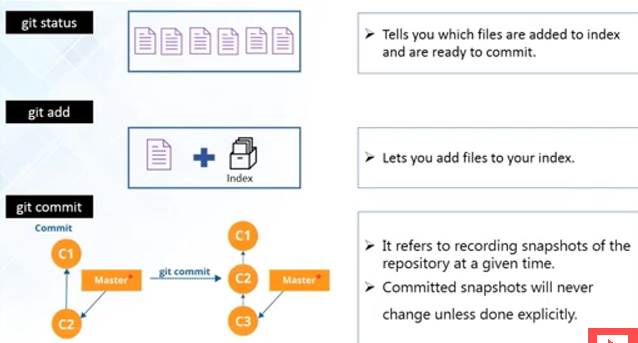


Pull data from central repository



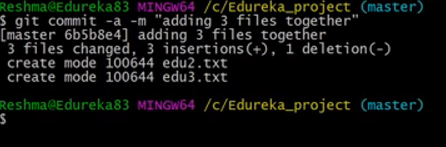
Used to update local repository from central repository

Can use git push to push it to the central repository

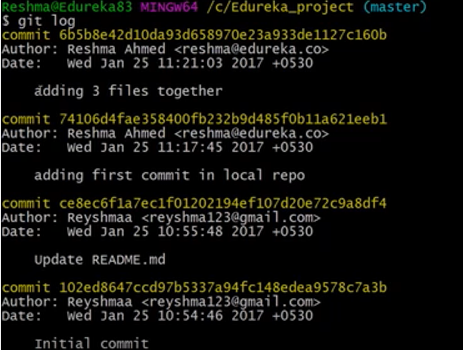


**Git commit –m “comments ”**

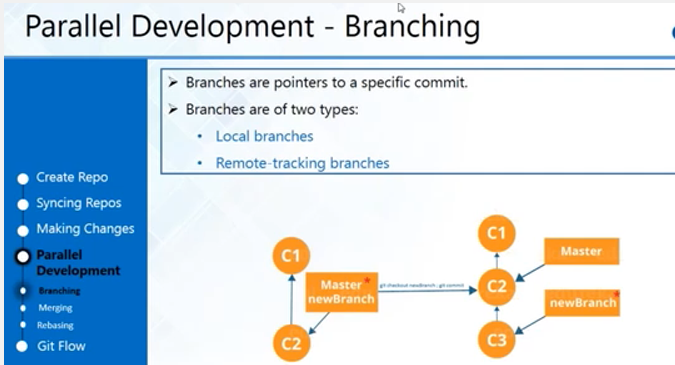
It commits changes to local repository



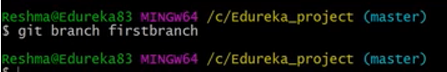
Git log : git history



**Parallel development**

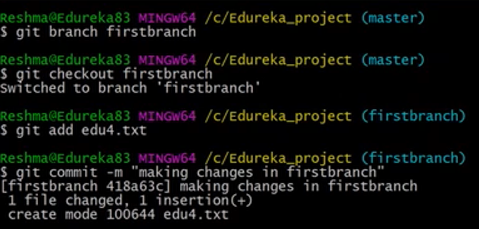
****

**Create branch from master branch**

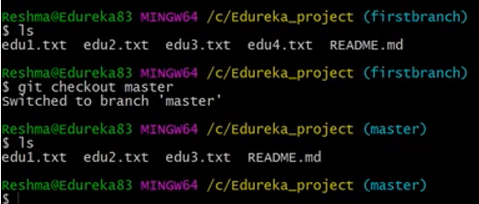
****

**Move from master to your branch: Checkout**

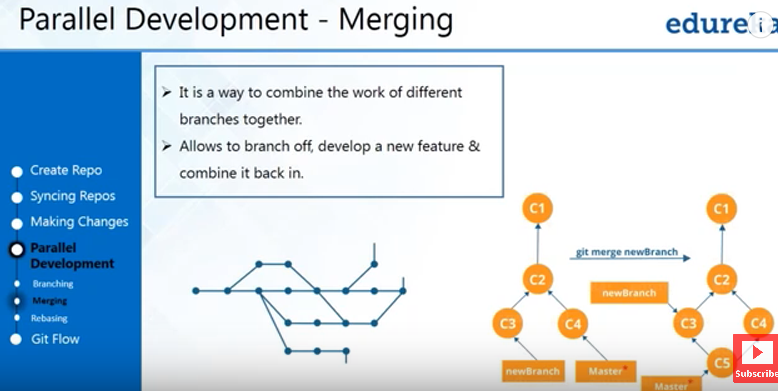
****

****

**Creating separate branches:**

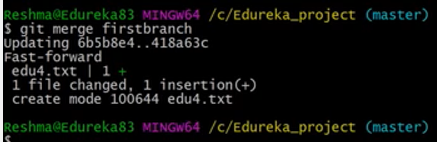
****

**Merging branches:**

****

**Master should be the destination**

Checkout to master.



Go back to firstbranch and then modify



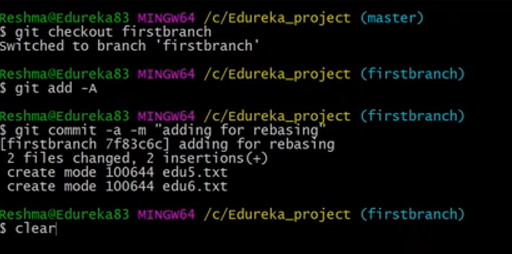
Since already was being tracked, no need to do git add again

**To do changes also in master file:**

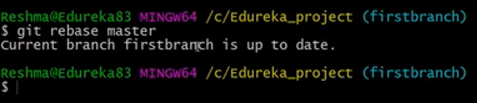
****

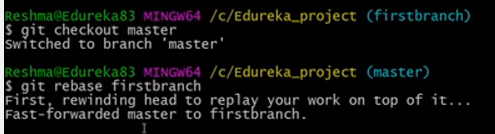
**Git pull:**  Pulls all files from central and places them in master branch

**Git fetch**: pulls files but places them in diff branch. Need to do git merge then.



Instead of merge do rebase





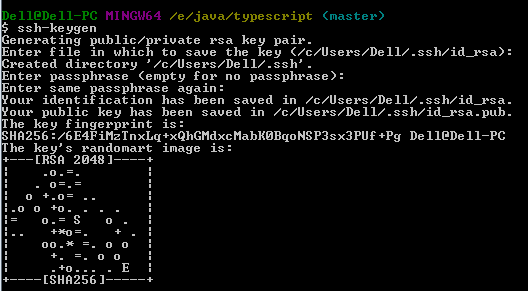
Work of first branch added to master branch.

**Git Push:**

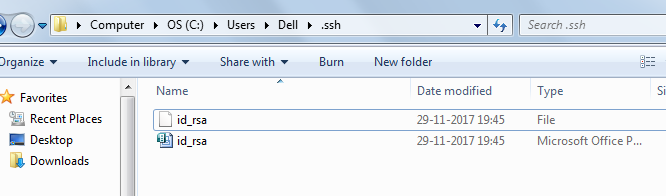
Sending code changes from local repository to master repository

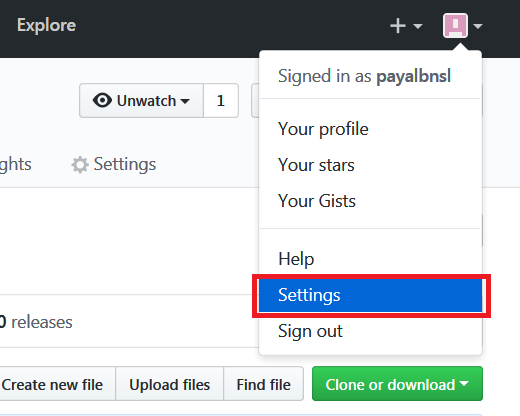
Can give access rights, no one else should be able to push changes in central repository

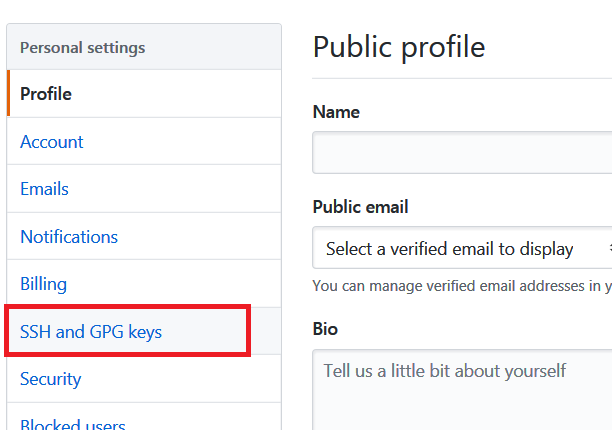
Create ssh key and add it to your account. Then push changes

****

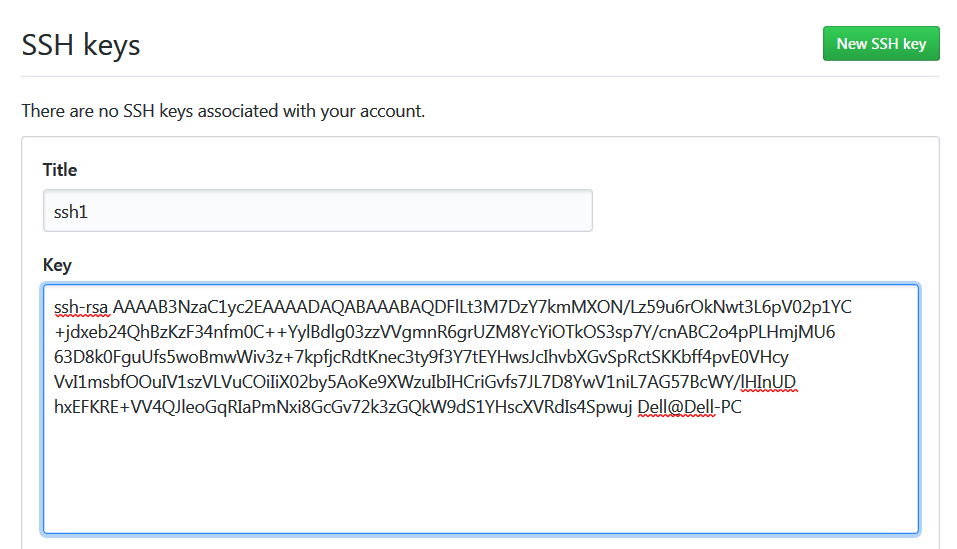
**Key saved in id\_rsa.pub file in .ssh folder**

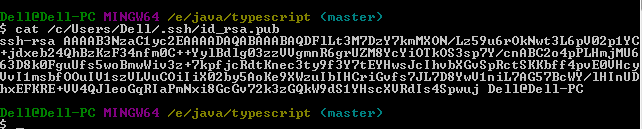
****

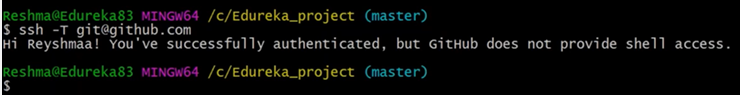
****

****

**Click new ssh key**

****

****

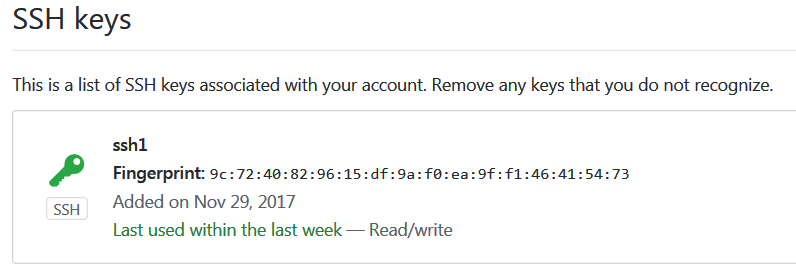
****

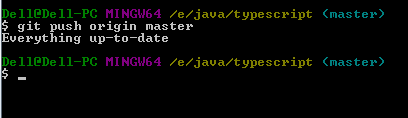
**To validate the key**

****

**Paraphrase: meerut**

**Refresh ssh keys.** It will turn green



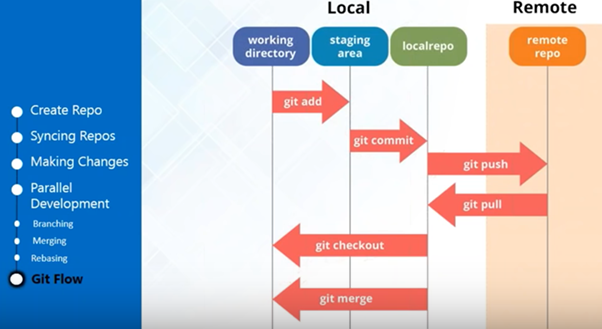


If changes in firstbranch:

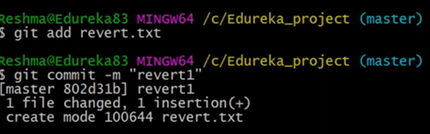
Git push origin firstbranch

Then

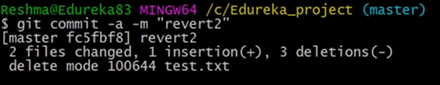
Git push origin master



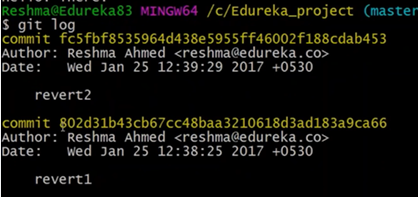
**To revert back to previous version**

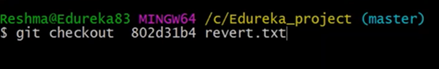
****

**Git commit –a : means git commit all files**

****

Do **git log** to check logs

****

****

**GitHub with eclipse:**