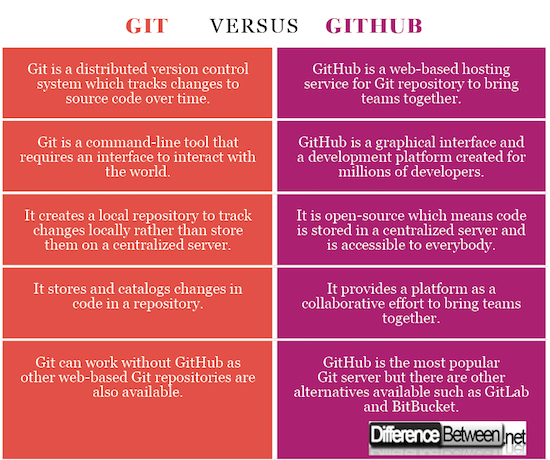
GitHub

* GitHub: Git is a distributed version control system, which means that the entire codebase and history is available on every developer’s computer, which allows for easy branching and merging.
* SCM- Software configuration management or Source code management



**Commands:**

**Installation**

sudo apt-get update

apt-get install git-core -y

git –version

**Configuration**

git config –global user.name devibadampudi

git config –global user.email [tirumala10cn52@gmail.com](mailto:tirumala10cn52@gmail.com)

cat .gitconfig (It will display the name and email)

git remote add username <url>

**Create Git repo**

mkdir /repos

cd /repos

git init (Initialize the empty git repo)

ls -a

git clone <link> (Clone repo from the git hub)

**Working with Git repo**

cd Devops

vi index

git status

**To add file to Cache(Staging Area)**

git add index

git status

**To move file from staging Area to Local Area**

git commit -m “New file”

**To Add and Commit file at a time**

git commit -a -m “New file”

**To push the code central repo(Master)**

git push -u origin master

**To show all git commits**

git log

git log -p

git show

git log –since=9/10/2019 –until=10/10/2019

git log –online

**To Push code to GitHub by using key:**

**Step1: Generate Key**

ssh-keygen

cd ~/.ssh

**Step-2: Add public key to github project**

Github🡪Project🡪Settings🡪Deploy keys

**Step3: Set remote GitHub url**

Git remote set-url origin [git@github.com:<username>/project.git](mailto:git@github.com:%3cusername%3e/project.git)

**To made changes to tracked files**

git dff

git log

git log -1

**List all branches**

git branch

**To work with branches**

git branch test

git checkout test

git branch

vi index

git commit -a -m “New file1”

git push -u origin test

**Check in browser🡪 GitHub**

**To merge branch code into master**

git checkout master

git merge test

cat index

git push -u origin master

**To delete the branch**

git branch checkout master

git branch -d test

git push origin --delete test

**To delete the branch without merging the data**

git branch checkout master

git branch -D test

git push origin --delete test

**Tagging:**

git tag v1.0

git tag -a v2.0 -m “second commit” (To create annotated tag)

git show

git show v1.0

git tag -l “v1.\*”

push tags to remote

git push origin v1.0

git push –tags

git tag -d v1.0(

git tag –delete v1.0

git push origin –d v1.0

git push origin –delete v1.1

git log

**Git Merge and Rebase**

Here is how rebasing differ from merging : Git merging applies all unique commits from branch 1 into branch 2 in one commit with final result. On the other hand, Git rebasing gets all unique commits from both branches (1 & 2) and applies them one by one.

Both git merge and git rebase are used to merge branches. There is one difference between them. The difference lies in the commit history after you integrate one branch into another.

Suppose developer A made a commit and developer B made another commit.

**git merge:**

After merging, commits from both the developers will be there in git log. This is really good for beginners cause the head can be rolled back to commit from either developer A or developer B.

**git rebase**:

After rebasing, commit from only one developer will be stamped in git log. Advanced developers prefer this cause it makes the commit history linear and clean.

**git fetch:** command downloads commits, files, and refs from a remote repository into your local repo.

When you do a git fetch, it fetches all the changes from the remote repository and stores it in a separate branch in your local repository. You can reflect those changes in your corresponding branches by merging. So basically, git pull = git fetch + git merge.

git fetch is the command that tells your local git to retrieve the latest meta-data info from the original (yet doesn't do any file transferring. It's more like just checking to see if there are any changes available).

git pull origin master --allow-unrelated-histories

Then : git push origin master

git remote -v

git remote add origin [git@github.com:devibadampudi/Devops.git](mailto:git@github.com:devibadampudi/Devops.git)

To push the our local files to new repo

1. Create a new repo in git hub
2. git add .
3. git commit -m "first commit"
4. git remote add origin <https://github.com/devibadampudi/All-labs.git>
5. git push -u origin master

git remote add origin <paste location of the repository>



