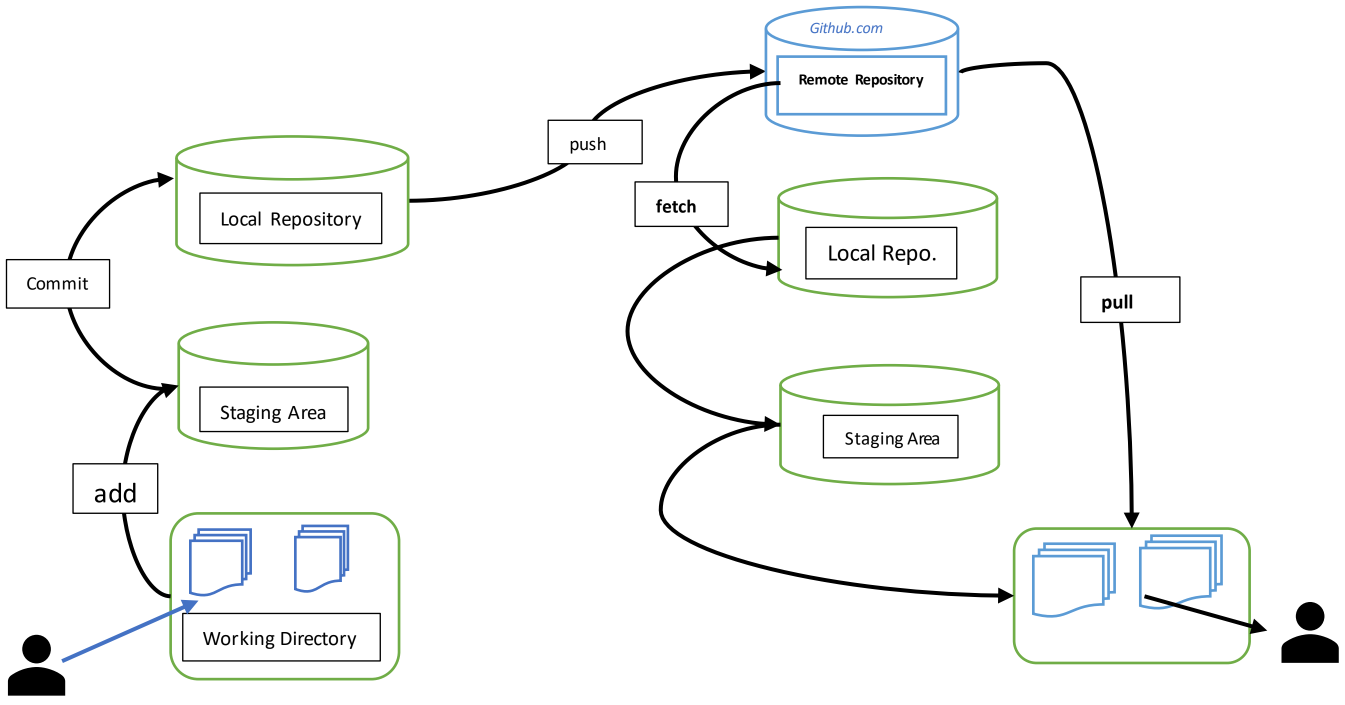
GIT



Commands: -

1. git --version : Check git version
2. git init: Creates a new local repository
3. git status: It says the state of working directory and the staging area.

   Example:

   - Create a file say index.html in your working directory.
   - Then run the command git status
   - This command will say index.html as untracked file, which means git is not aware of this file.

4. git add index.html : This command add the index.html file in local repository.
5. git commit -m "--- comment ---" : commit command create a version of file in local repository.
6. git status
7. git ls-files : Display all tracked files which are in staging area.
8. git reset HEAD <file-name> : After "git add -A", if we want to remove the file from staging area then use the command git reset HEAD <file-name>
9. git commit -am "--- comment ---" : Add and commit through single command.
10. Delete files:
    a. git rm index.html : Delete file from working directory as well as staging area.
    b. git rm –cached index.html : Delete file from staging area only not from working directory

    NOTE: After delete we need to run commit command. i.e. git commit -m "deleted done"

11. .gitignore: .gitignore file tells GIT which file or folder need to ignore.
    - Create a file .gitignore in the root of working directory.
    - Write file names or folder name which we want GIT should ignore.
    - Ex:
        - /node_modules
        - `/src/config/database.js`
        - *.sh
    - git add .gitignore
    - git commit -m "message"
12. Move files from Local Repo to remote Repo
    - Create account in github.com
    - In github.com, create a online repository.
    - git remote -v : Check if any remote repository is added in your local repository so that we can push files in remote repository
    - git remote add origin <URL> : Add remote repository in local environment.
      "In the above command, Origin is alias for the URL, we can also use any other name"
13. Now Push files from local repository to remote repository
    - git push -u origin master

    Here master is the branch name.

    The above command will ask user name and password when we will execute it first time.
14. Now refresh github.com, we can see the files in remote repository.

Now an another user (User2) in another machine need to download all files.

Step 1: Go to github.com & choose remote repository

Step 2: Copy the web URL from the Code drop-down button.

Step 3: Choose a location in local computer where we want to clone the project.

Step 4: In terminal, use the below command

- C:\>project1\ git clone <url>
  NOTE:
    - Here <url> is that web URL which we copied in step 2.
    - git clone creates a local repo by copying the entire remote repo (including all branches, commit history and files)
    - It also creates a working directory, where we can edit files.
    - We use clone to download all files (*otherwise use git pull)

- git remote -v: To check remote repository
- git config --global --list: To check git author of the local machine
- How to add new author in local machine:
    - git config -- global user.name "demo"
    - git config -- global user.email demo@demo.com

Now user2 want to work in the downloaded project

Step 1: User 1 (owner of remote repository) have to add user 2 as Collaborators.

- Go to github.com and choose the repository
- Click on Settings (top menu)
- Click on Collaborators
- Click on Add people button
- Search user2 with his email id or user name and Add.
- User2 will receive an email in which he/she need to accept the invitation.

After this, User 2 can work in the repository.

Now user2 can also push files in remote repository. If he/she need to get new updates in local repository then from second time onwards instead of git clone use the command git pull.

Difference between git pull and git fetch

git pull copies changes from a remote repo directly into working directory. It modifies our working directory.

Git fetch copies the changes in the local repo which means it does not modify our working directory.

NOTE: git pull combines two commands git fetch (retrieves changes) and git merge (integrate changes)

Conflict & merge

User 1:

- Open any file say index.html
- Do some changes on it.
- Commit it and push it in remote repo.

User 2:

- Without git pull, work in the index.html and do changes.
- Now commit it in local repo
  - git commit -am "From user 2"
- Now try to push into remote repo
  - git push -u origin master – Error message will appear

In such situation we need to resolve the conflict

- git pull origin master
  - This will update the file in machine 2 with conflicts
  - Open the file and manually fix the conflict and remove unwanted data/content.
- git commit -am "fix the conflicts"
- git push -u origin master

GIT Branch:

Show the list of existing branch:

- git branch

Create a new branch

- git branch <name_of_new_branch>

Delete any branch

- git branch -D <name_of_branch>

Switch to another branch

- git checkout <branch_name>

Ex:

```
git branch QA_branch

git checkout QA_branch

git branch

*QA_branch  ( * represents or point to selected branch)
master
```

Merge branch:-

```
git branch
        master
        dev1
        dev2
        QA
```

Now merge dev1 branch into master branch

```
git checkout master
git merge dev1
```

NOTE: we can also use git rebase command for merging branches. Its an alternative to git merge.
Syntax: git rebase <source branch>

git log: provide us the log of commands which we executed. It will show the commit ids and branch name

git revert: This command helps us to revert a commit to a previous version

```
git revert <commit_id>
```

commit id can be obtained from the output of git log

git stash: If we want to save our work without committing the code then use stash. This is useful when we want to switch branches but do not want to save our work to our git repository.

- Create a new file in your repo say file4.txt

- git status
  untracked file – file4.txt
- git stash -u
  saved working directory and index state WIP on dev1 (dev1 is branch name)
- git status
  on branch dev1
  nothing to commit working tree clean

- git stash pop
  Already UpToDate
  On branch dev1
  Untracked files:
  File4.txt
- git stash list: to see the list of stash files.

~By Gautam Mukherjee