**To stat logging:**

git config --global user.name "FIRST\_NAME LAST\_NAME"

git config --global user.email "MY\_NAME@example.com"

**to stash the current changes (or you can commit the changes):**

git stash

(take the new changes from the github mention the branchname) git pull origin new-layout-reorg

(to bring in the changes) git stash pop

**Useful commands:**

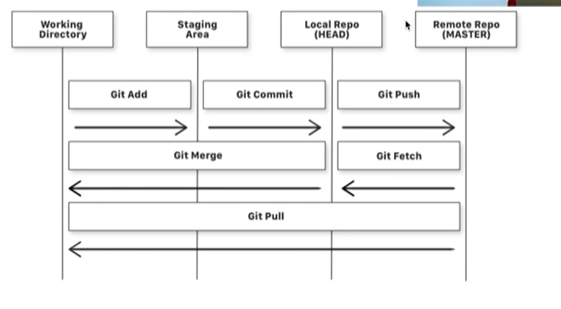
git init

git add (to add the file to staging area)

git status (to find the current status of the project)

git commit (to add file from staging area for checkpoint)

workflow: -

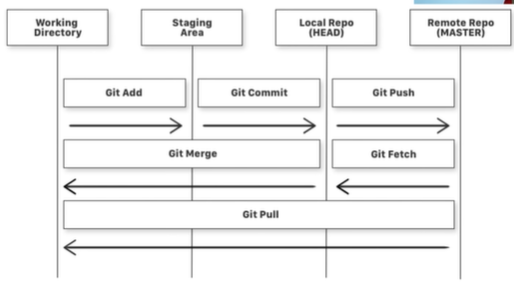


**Branching**:

git branch ‘branchname’ (to make a new branch)

git checkout ‘branchname’

move to master branch: git merge branchname (branchname branch will be merged to master or to the branch in which you are currently in)



git ignore

make a file .gitignore

add name of the files which you don’t want to add to the server in the .gitignore.

Add a remote:

git add remote "remote\_URL"

To make and shift to a new branch:

git checkout -b "branch\_name"

To add a remote URL as upstream so that we can later fetch or pull changes from that URL

git remote add upstream "uproar URL"

Fetch changes (just download the changes but does not change current file) It doesn't apply any changes to current files.

Git fetch upstream -> all the changes from upstream are downloaded. Usually idea is to bring main (of origin) to the same

Level as main of upstream. So change the working branch to main using:

git checkout main

And now set the head of main to the head of upream main,using:

git reset --hard upstream/main

Push the changes from local to remote.

git push origin main

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

git init -> to initialize a git repo

git status -> to check the status of all the changes

git add . -> add all the files to stage

git add names.txt command -> the two files (names.txt or command) will be added to stage

git commit -m "message here" -> to commit the changes from staging area

git reset <hash id> --hard -> to reset the changes to previous version use "hard" argument to revert changes in the files

git log -> to check the history of commits

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

git clone <url> --> to clone a project to local repo

git chechkout -b <branch-name> -> to create a new branch

git set-url origin <url of the repository>

git branch -M main

git push -u origin main -> main; from which branch you wish to push (can be replaced by any branch name)

-> origin to which branch you wish to push (can be replaced by any remote branch (added by git remote add <url name>)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

git config --global user.email "manish.sharma@calpion.com"

#to remove a file from stageing area

git restore --staged filename