Git commands :

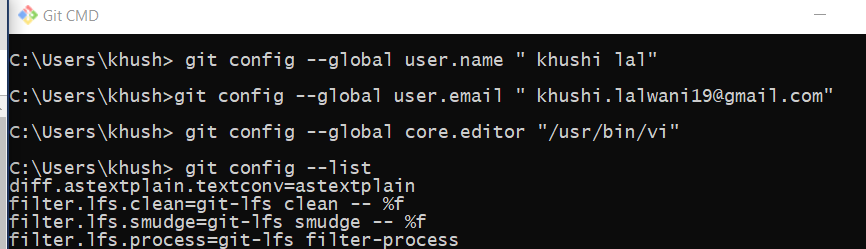
1. git config --global user.email "khushi.lalwani19@gmail.com"
2. git config --global user.name "khushi-lal"

mkdir name

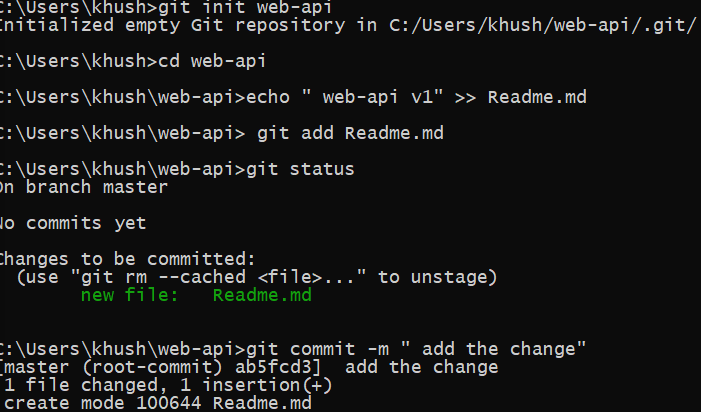
1. git config --list (listing)
2. >git log (history)
3. git status
4. git clone (url)
5. git add (name) or .
6. git commit -m "(message)"
7. git commit -a -m “”
8. git log --oneline –decorate
9. git push origin
10. git pull
11. git branch
12. git branch (branch name)
13. git checkout (branchname)
14. git merge (name)
15. git tag
16. git tag -a v0.1 -m ""
17. git show v0.1
18. git tag -d v0.1 ( delete )
19. cat .git /head ( to see the branch)
20. >git branch -d trial2(deleting branch )
21. >git rebase dev
22. Git revert commit
23. Git diff ( to see the changes done )
24. Ssh -keygen

(((((((Successfully rebased and updated refs/heads/master.

Global configuration



Creating repository and tehn new file



**Git**

First open instance and connect

* Connect putty open ssh > auth
* Sudo su – to enter into super user
* Yum update -y : to update
* Yum install git -y : to install git
* Which git : to find installed properly
* git –version : to find the version
* to set global user and email ( global configure) : needed to know what you commit and do changes as reference always set id and user name

git config --global user.name "khushi"

git config --global user.email “ [khushi.lalwani19@gmail.com](mailto:khushi.lalwani19@gmail.com)”

* to verify or check : git config –list

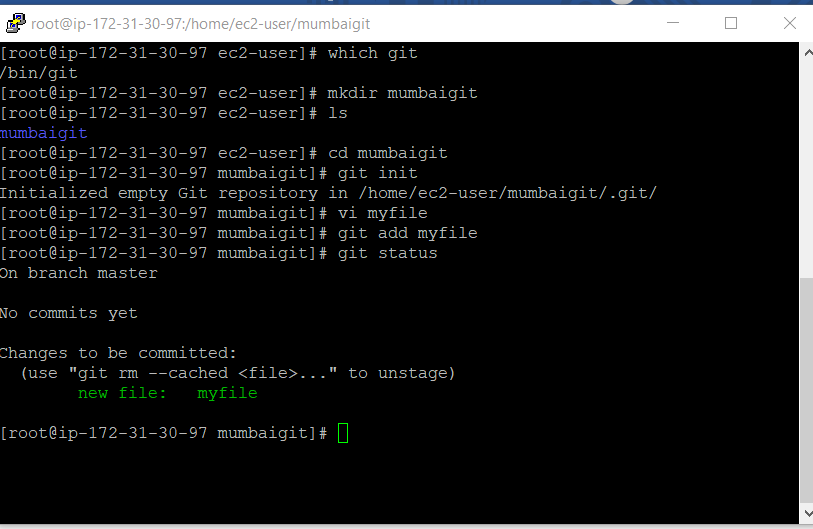
creating off shore and onshore id or instances then create central git by logging in git hub account

* <https://github.com/login>
* create repository
* create dir : mkdir mumbaigit
* list : ls
* to go to under directory use : cd mumbaigit
* initialising empty git repository : git init (thus devided int othree workdir ,stage and local reppo)
* to create file : cat or vi : vi myfile ( instert and :wq)
* where you create file or modify is work space
* local rep : ls -a (hidden mode )

to add file sending to stage area ( it takes snapshot ) : git add myfile

if more files then : git add . (. Is for current dir all files in workspace will be added )

to check status : git status ( untracked if only in workspace )



To store permanently commit : git commit -m “1st commit from Mumbai” ( m is for message )

To get commit id : git log

To see the data into commit id : git show cc3b09 ( first 7 no. of commit id )

After commit moving to central repository since doing it first time add central repo to local (once only done for first time )

To add : go to git go to your repo given the command just paste (git remote add origin <https://github.com/khushi-lal/centralgit.git>)

git branch -M main

git push -u origin main

Pene@961

Now we can push our codes: git branch -M main root@ i-021b9f1b45fc0e025]# git push -u origin main (copy from git )

Will ask for mail and password of git

Now again if you push use : git push -u origin main

Now if you have to pull for first time from git

First add the origin command : git remote add origin <https://github.com/khushi-lal/centralgit.git>

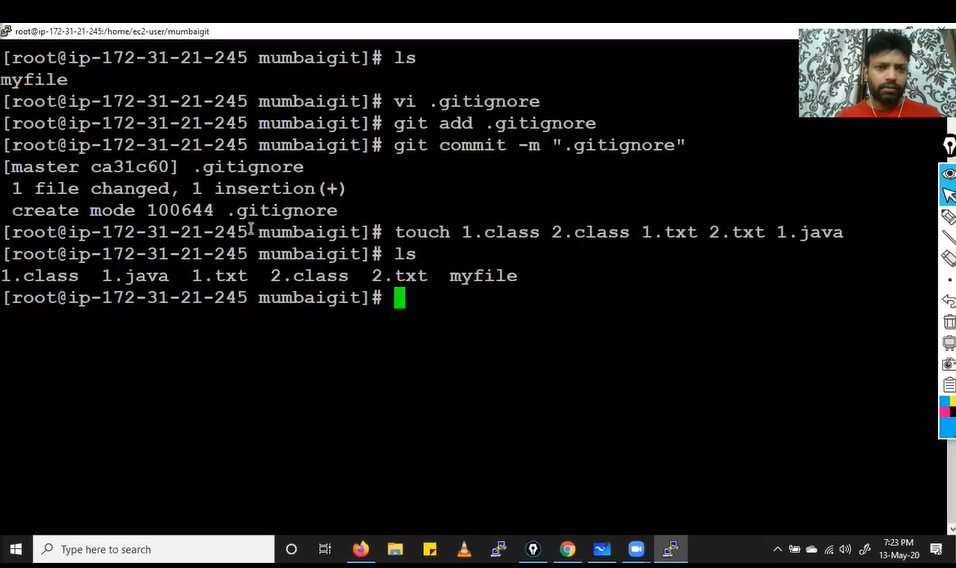
To pull : git pull origin main

If you want to add files and ignore few files for ex add files of java and ignore files that are of c

So first create a file named .gitignore add the file name with the star and add and commit then whatever files after that are pushed will ignore all the file that are in the file ignore

For ex :

* Vi .gitignore
* (inside file \*.c or \*.class)
* Now add this file and commit
* Git add .gitignore
* Git commit -m “ignored filesssss”
* (Create files) touch 1.class 2.class 1.txt 1.java



* Git status : will show only the files that will be pushed
* Git add .
* Git commit -m “ “

If you want to txt file only

* Git add \*.txt

To see latest commit

* git log -1 or git log -2
* git log --oneline ( in summarised manner)
* git log –grep “singapore” ( to see exact or related commit )

**Branching**

* by default branch called master
* to see branch : git branch
* git branch newbranch
* git checkout newbranch ( to change to other branches)
* changes made in new branch wont be visible in master if commited but if not commited it will be shown in master only once you commit it belongs to new branch otherwise to main branch
* when two branches have two files with same names but different content and trying to merge there is conflict
* Git merge newbranch ( if shows conflict ) open the file and edit again add and commit
* If you want multiple idea in same branch repository called stash

First create empty file add and commit add data > same file new idea if you wana test then you ll apply stash that will give you clean file

* Git stash
* Git stash clear
* To delete the snapshot before commit : git reset filename
* Git reset –hard (all files)
* After commit we use : git revert
* Tag : git tag -a name – m “message “
* To delete tag : git tag -d name
* List of tag : git tag
* **Clone** to copy everything in local repo : git clone url
* **Pull diff then clone :** clone takes whole of main repo but pull gets only incremental stuff
* To megre you ll have to raise req from web new pull req > choose branch and then merge in master

Git

* To add exsisting into dir first : ls -la
* 2nd : Git init
* Git reset file name (to delete from stage are )
* To clone remote repo : git clone (url) <at place to clone>
* To view git remote -v
* To list branch git branch -a
* **Git diff** – shows the changes made
* Git branch --merged : to show all things merged previously
* If you wanna delete after merge : git branch -d name
* To delete from remeote : git push origin –delete name
* To change the commit message if written wrong : git commit --amend -m “the change “
* If you forgot to commit a file : git commit –amend (enter)
* Git log --stat (shows all the commit history)
* If you have been making changes in wrong branch to move commit to actual branch : first :

git log > git checkout name

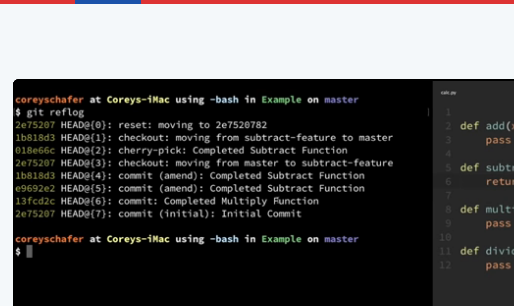
git log

git cherry-pick commitid no. 6 digit ( this just copies )

* To delete the commit from main branch using reset

1. Git reset --soft commitid no. ( it will keep it in stage but delete from commit just moves back undo)
2. Git reset commitidno ( git status keeps changes only in working dir )
3. Git reset --hard commmitidno (complete deletes)
4. You still can collect this from garbage before 30 days : run : git reflog

Git checkout commitno. (of hash before deleteing here 1b818d3)



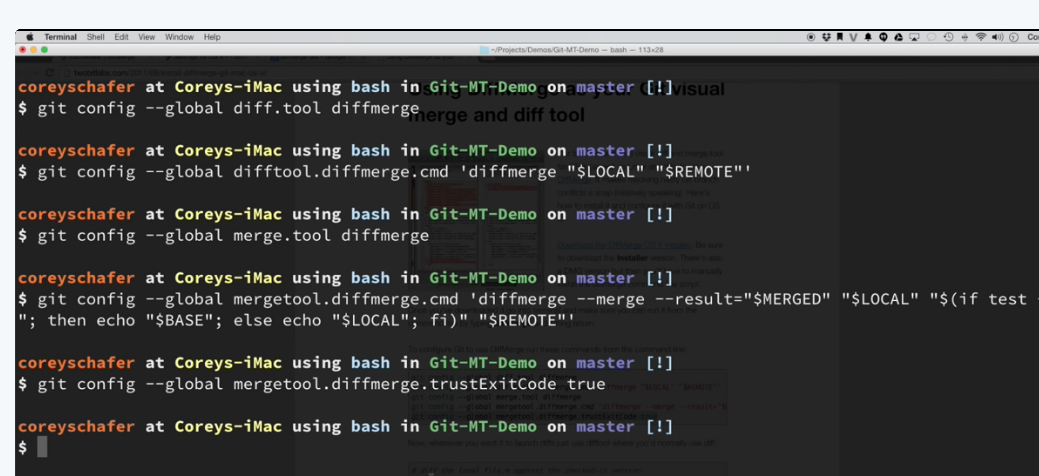
Run : git log ( we will have changes back)

To save changes create branch : git branch backup

1. Getting rid of untracked file or directory : git clean -df

* To not rewrite any history to undo all the changes
* Git revert hash
* To see diff of two diff changes : git diff hash1 hash2
* Stash to save temp changes and work on it later add a branch and then after the work done

: git stash save “ message “

* If you want temp stash to apply in main but doesn’t delete : git stash apply stashed ( id you get from git log or git status)
* Git stash pop ( grab very recent stash and apply and then deletes)
* To just delete git stash drop (stash id )
* To get rid of all stash : git stash clear
* Git stash list
* **diffmerge it** needs to installed it helps to see the changes made in document in properly
* to check if present ls user /bi n
* check for diffmerge if present then go to the machine installed
* check online to configure one by one
* 
* Git config -- global --list
* Git difftool
* To add files to staging area

1. Git add -a / --all : adds all entire working tree
2. Git add -a mydir : only changes in that dir will be added
3. Git add --noall m ydir : doesn’t include delete files
4. Git add -u : add modified and deleted files but not new or untracked files
5. Git add . : not same as git add -A because add . add only current directory

* \* is not git command its shell

730027683632

cloud\_user

#7Czlsifqb

https://730027683632.signin.aws.amazon.com/console?region=us-east-1

ec2-3-236-38-234.compute-1.amazonaws.com