**GIT (Version Control System)**

Source code management Tool or Version control tool.

Ex. GIT, SVN(Apache Sub version), CVS...etc

SVN is used more than past 4 back years , now migrated GIT

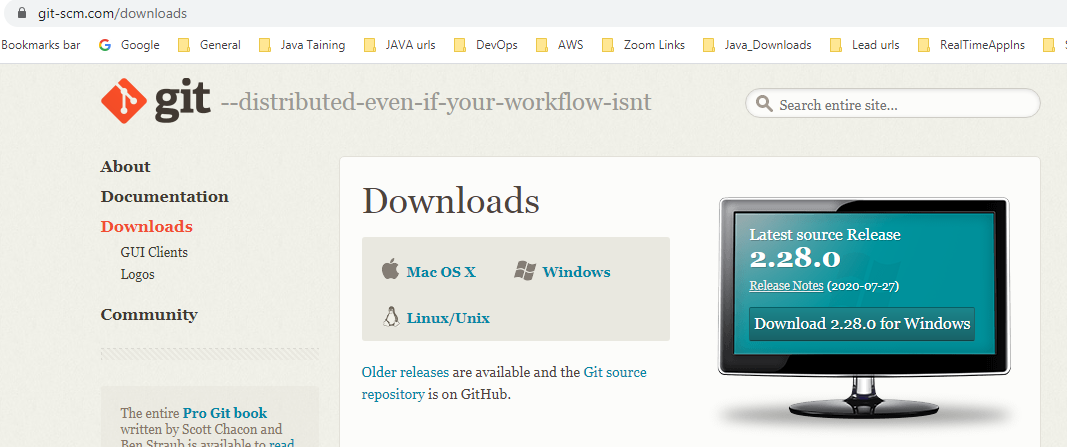
More features in GIT - Branching, taging..etc

GIT – 2 ways we can operate (CUI/GUI) .->Real time using CUI (cmd).

**SVN (**Centralized Repository Management**)**

**GIT (**Distributed Repository Management**)**

<https://git-scm.com/downloads>

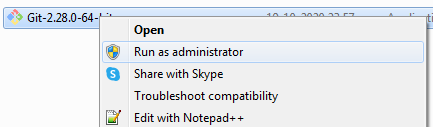


click Linux/Unix to see some cmnds.

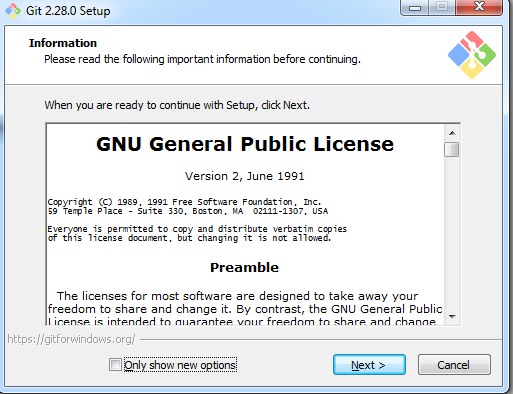
yum install git (install in cent os/RedHat/Fedora)

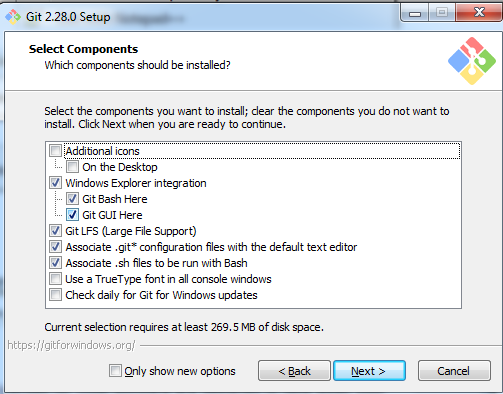
apt –get install git (installation in ubantu)

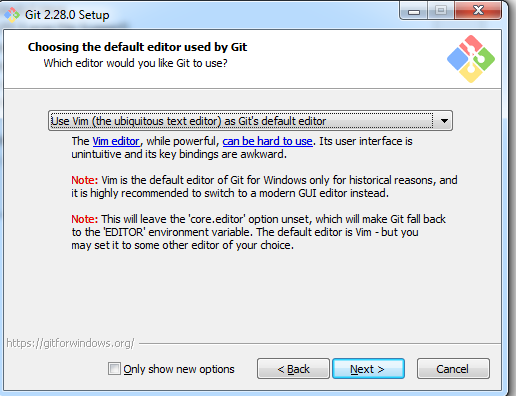
Install GIT:- click on windows desktop image you will get download git exe file.

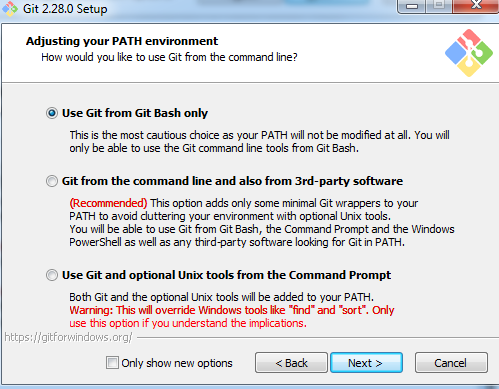


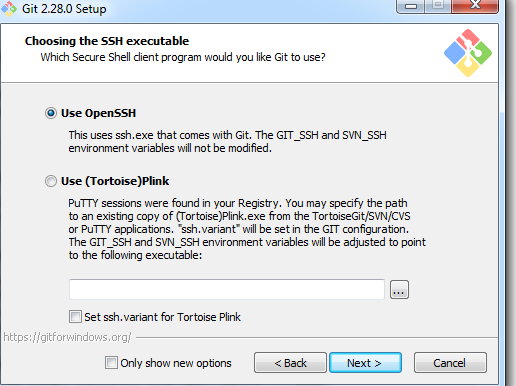
Below all are default settings no need to change any options.

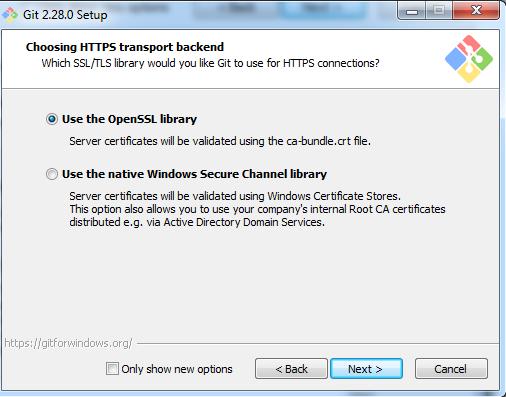


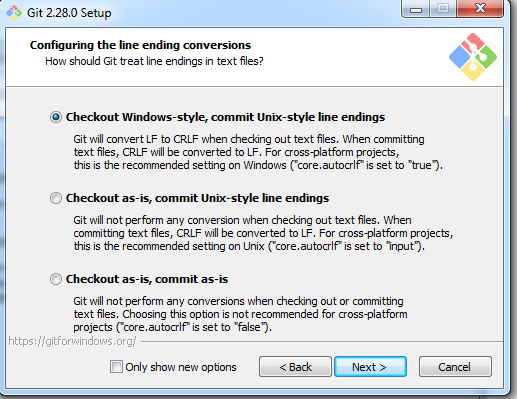


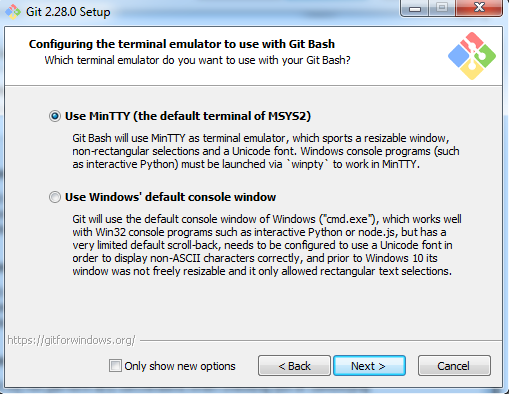


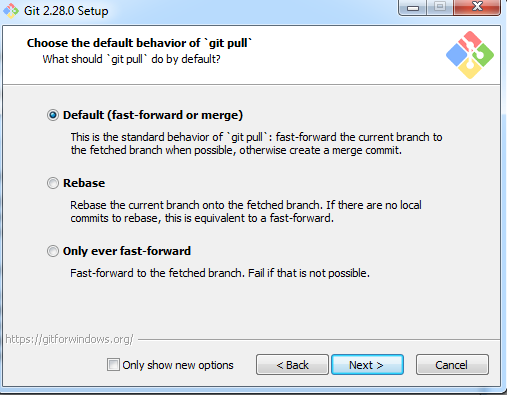


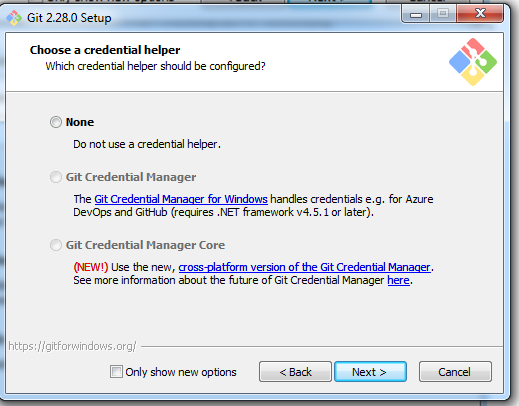


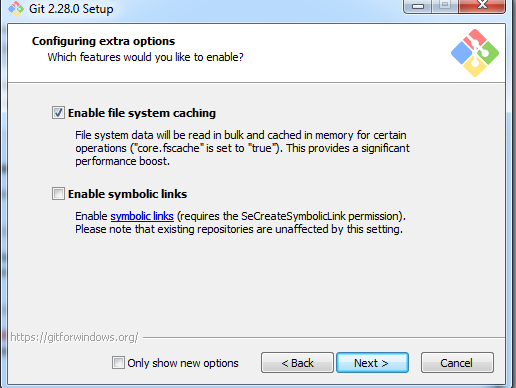


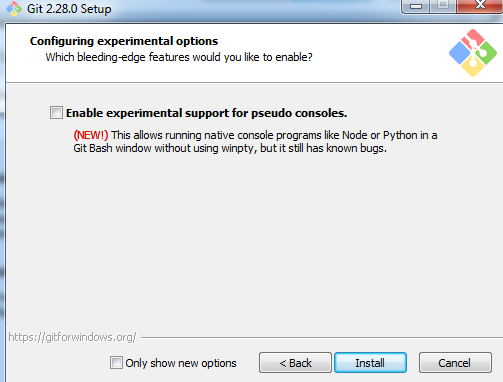


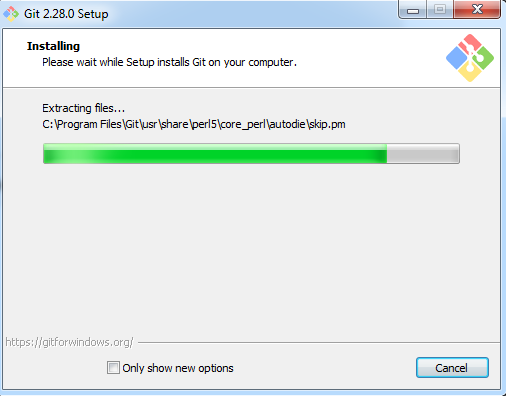


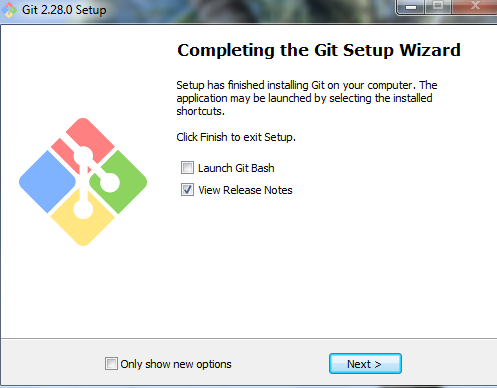








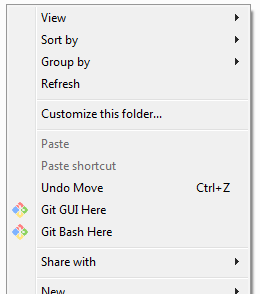




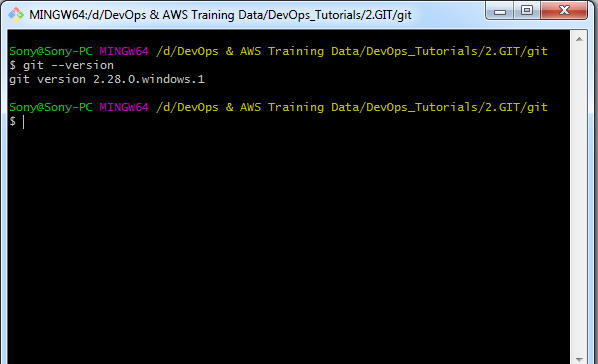
Installed successfully.

Create one git folder (any name).

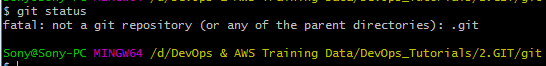
Open Git Bash



$ git –-version



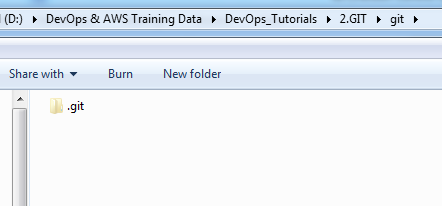
$ git status



$ git init -> initialize the git local repo



It Will create .git repo



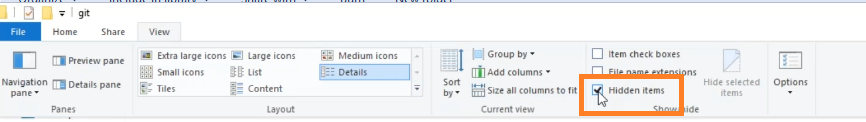
$ ls –a ->show hidden dir’s



Here master is branch will create automatically while installed in git local repo.

If not showing .get folder, enable hidden options.

In Windows10



**Configure email for user:-**

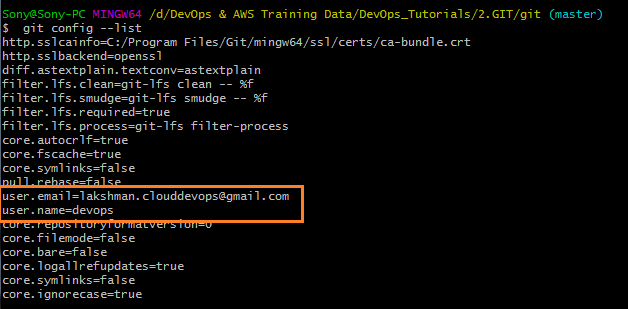
$ git config --global user.name "devops" -> devops is user setting



$ git config --global user.email [lakshman.cloud@gmail.com](mailto:lakshman.cloud@gmail.com) ->email setting

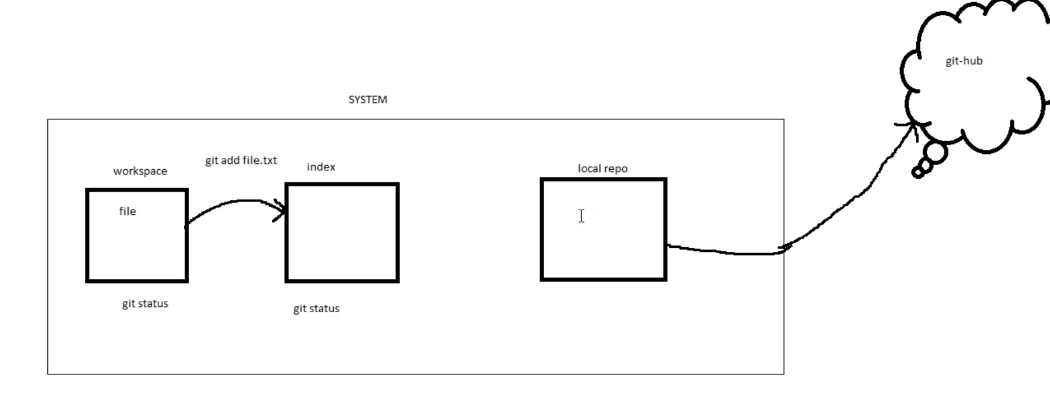


$ git config –-list -> Display list of config’s.



GIT Phases:-

1. Workspace area
2. Staging/Index area
3. Local area

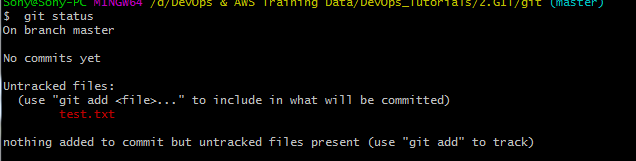


Github is here Distributed Repository (it’s a cloud/remote repository)

$ touch test.txt -> create one empty txt file



$ git status

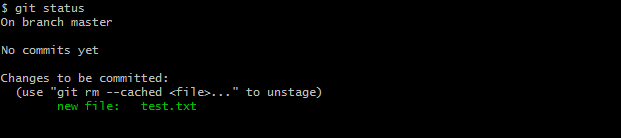


Untracked files: – it means ur files are in workspace area.(files are in red color)

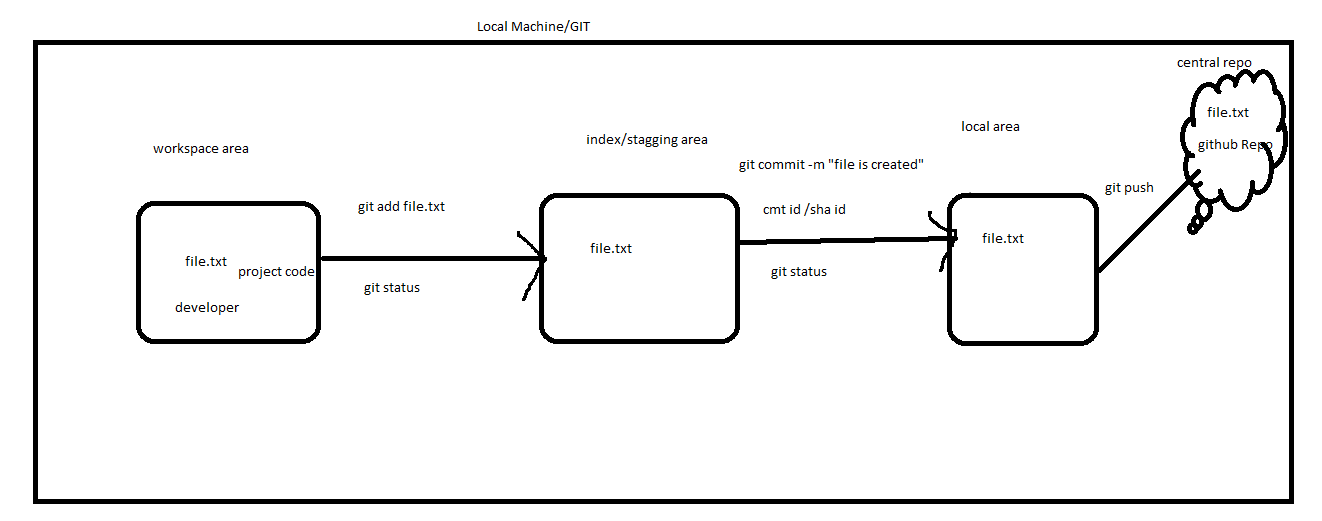
$ git add test.txt



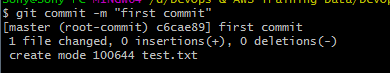
$ git status



Changes to be committed: -> it means ur files are moved into staging/index area.(files shows in green color).

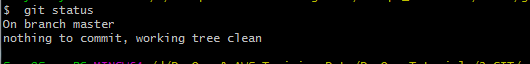


$ git commit -m "first commit" -> any label name provide.

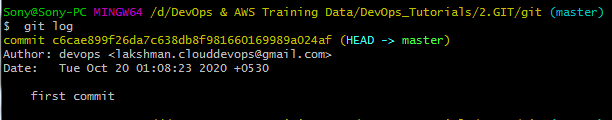


$ git status

Now file is not available to commit in staging area.



$ git log -> show log details



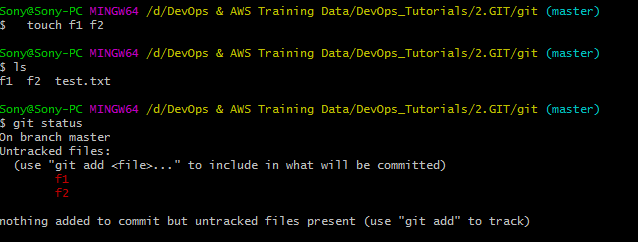
c6cae899f26da7c638db8f981660169989a024af -> commit id/sha code

**Now 2 files will add and commit:-**

$ touch f1 f2

$ ls

$ git status



$ git add .

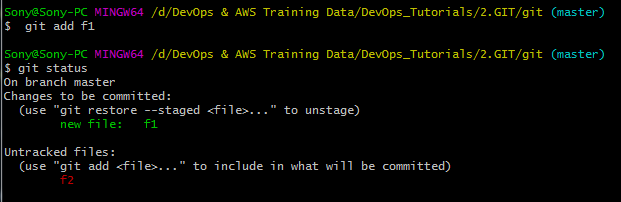
OR

$ git add \*

OR

$ git add –A

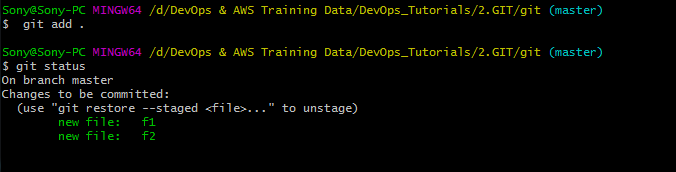
$ git add f1 f2 -> adding only one file



$ git add f2

OR

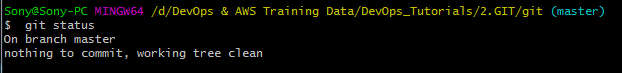
$ git add .



$ git commit -m "second commit" -> multiple files will move to local repo area



$ git status



$ git log

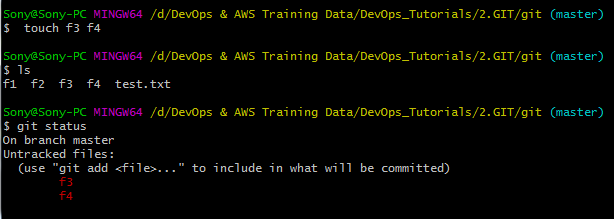


**Only Single file move to local repository:-**

$ touch f3 f4

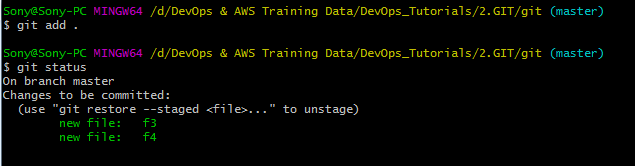
$ ls

$ git status



$ git add .

$ git status

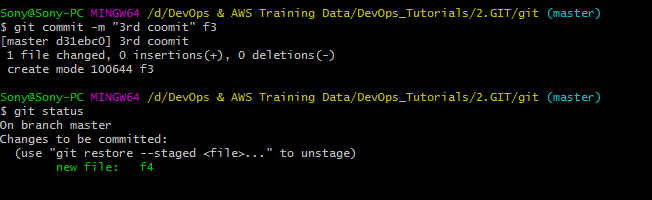


$ git commit -m "3rd commit" f3 -> for single file commit

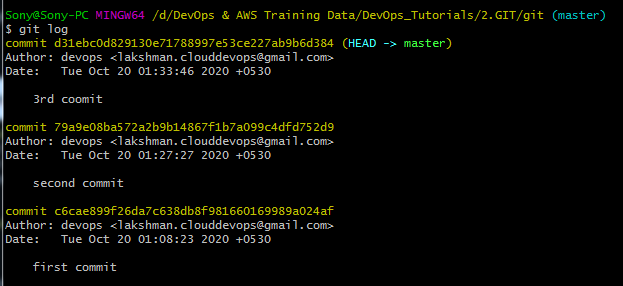
OR

$ git commit -m "3rd commit" -> for multiple files commit

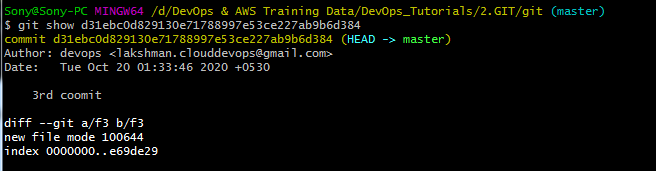
$ git status



$ git log -> for each commit it will show one commit id.



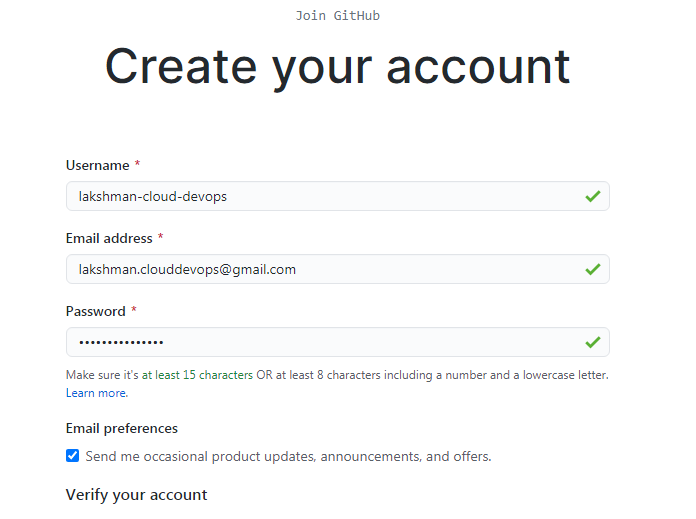
$ git show d31ebc0d829130e71788997e53ce227ab9b6d384 -> shows committed details .

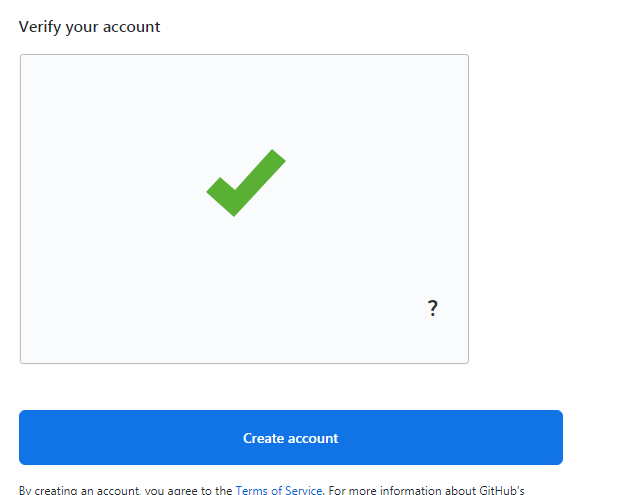


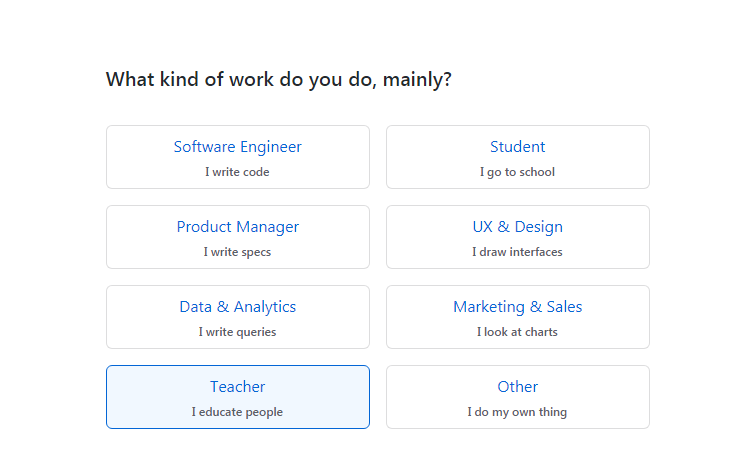
**How to create GitHub Repository:-**

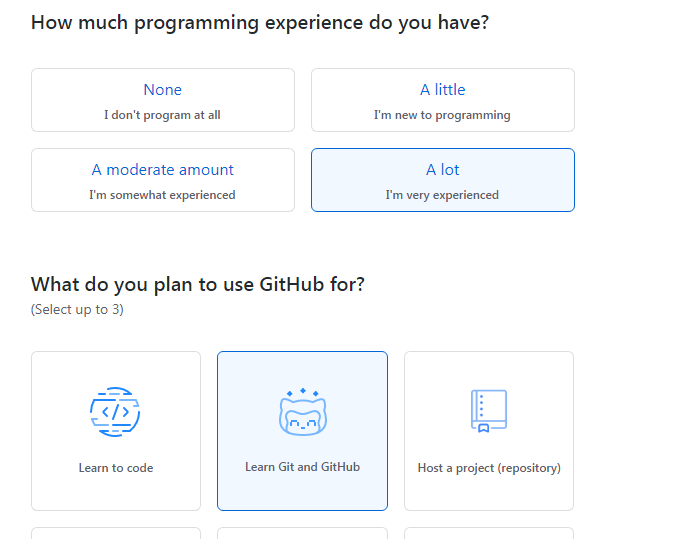
[**https://github.com/**](https://github.com/)

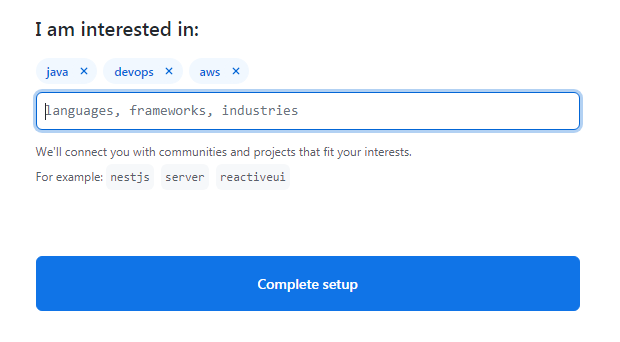
**Click SignUp**

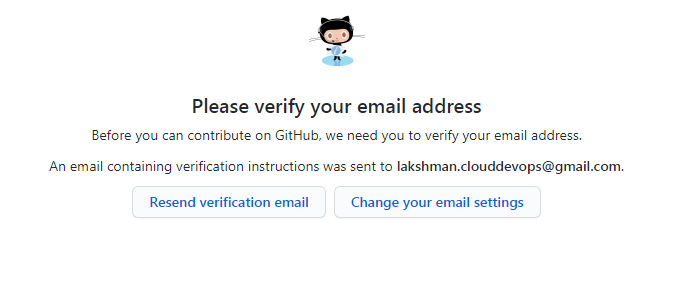
****

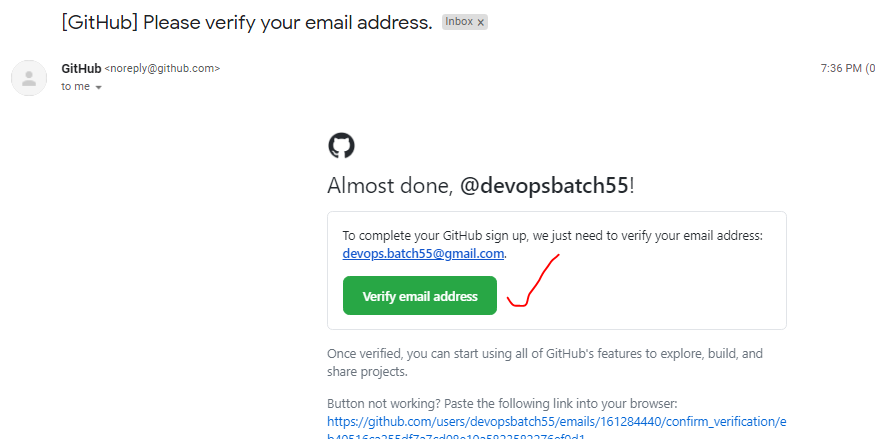
****

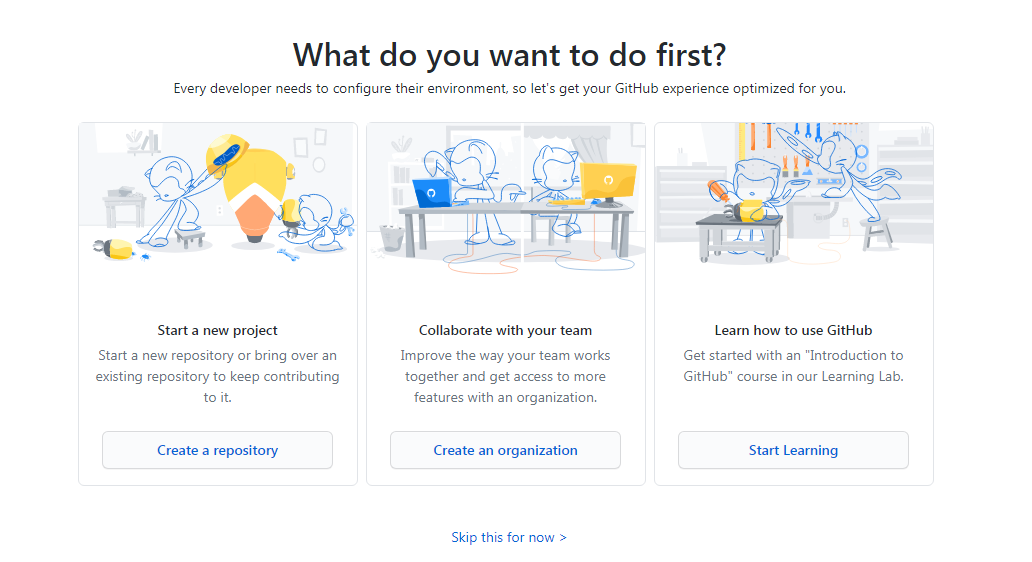
****

****

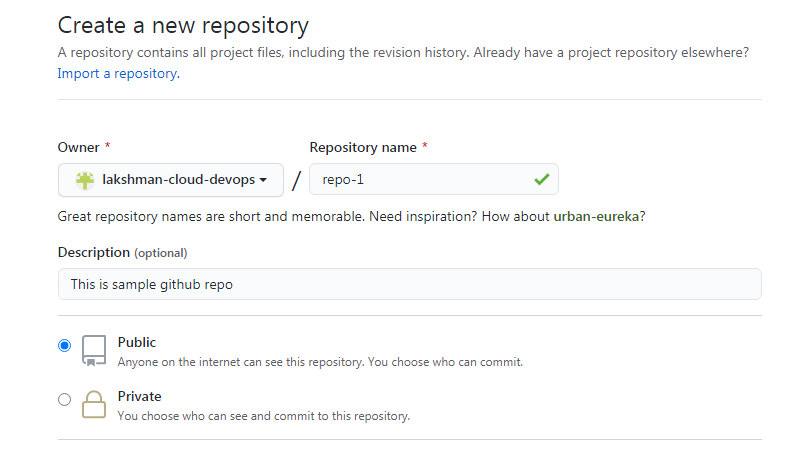
****

****

****

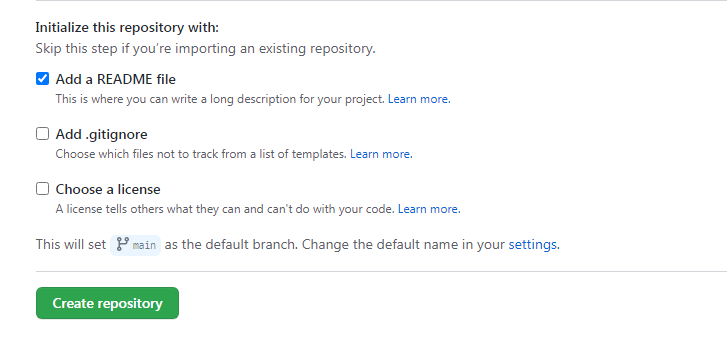
****

**Click Create a repository**

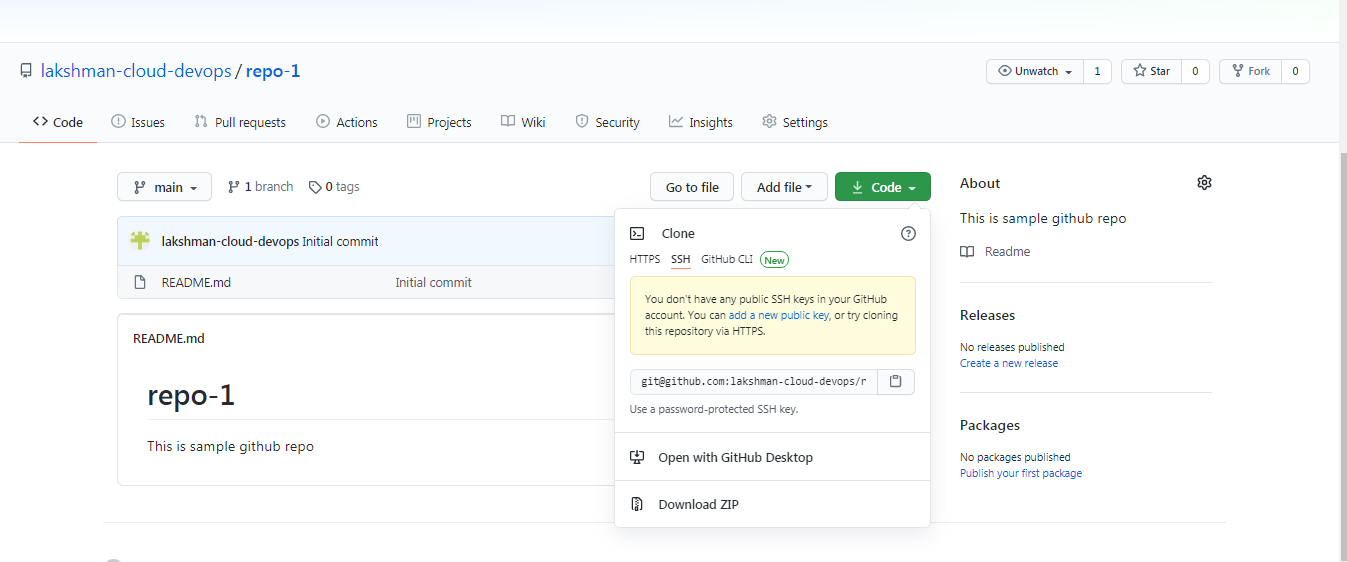
****

**If choose private repo –** it will chargeable 7 dollars.

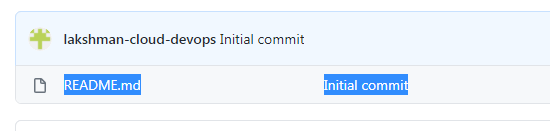
IMP Note :- **In Real time use Private repository only**.



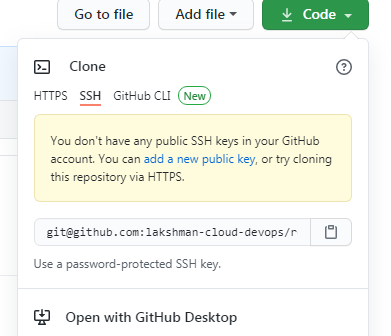
Repo-1 is created under my profile.



Automatically readme file is created.

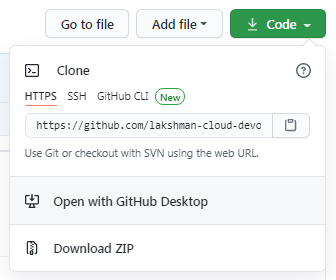


clone the code from github repo to local , click ssh/https and copy the github url([git@github.com:lakshman-cloud-devops/repo-1.git](mailto:git@github.com:lakshman-cloud-devops/repo-1.git)) and clone the code through git bash into ur local.



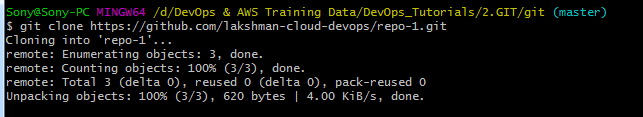
OR

Select HTTPS

****

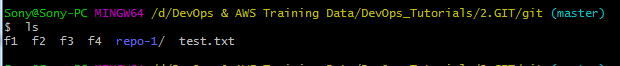
**Repo-1 repo code clone to local:-**

$ git **clone** https://github.com/lakshman-cloud-devops/repo-1.git



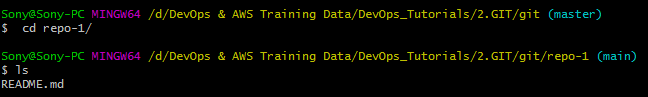
If private repo it will ask authentication pwd(for authentication they have configured SSH key in real time), this is public authentication pwd will not required.

$ ls

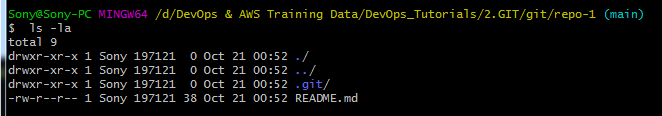


$ cd repo-1/

$ ls



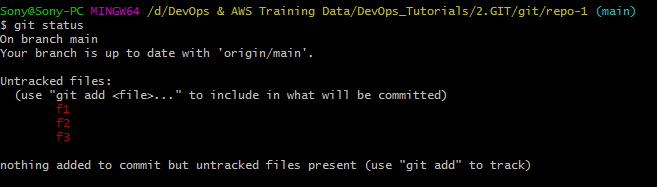
$ ls –la -> shows hidden files.



$ touch f1 f2 f3 -> created 3 files



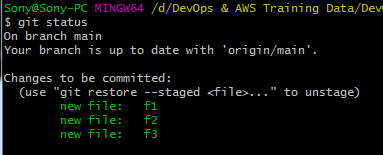
$ git status



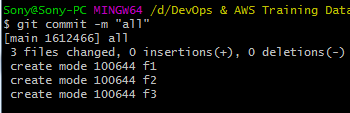
$ git add . -> added 3 files to workspace



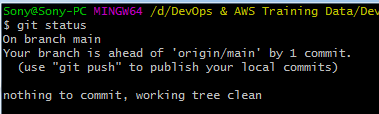
$ git status



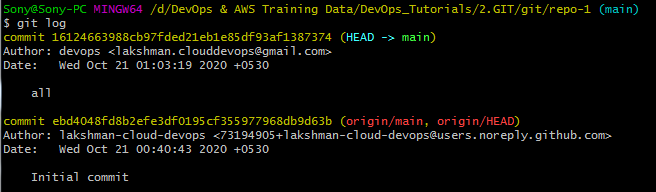
$ git commit -m "all" ->committed to staging/index



$ git status



$ git log

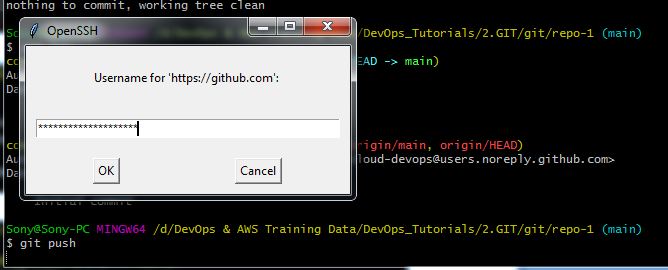


2nd commit id is created for readme file, while repo created.

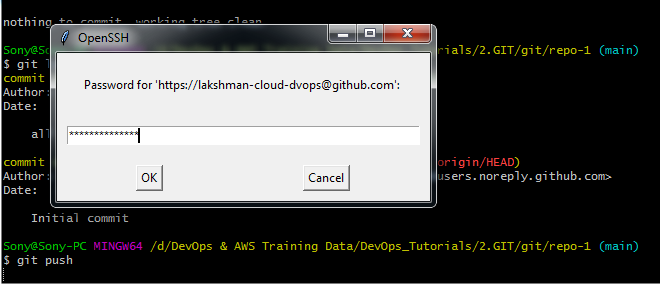
$ git push -> pushing code into github repo-1 from local.

Note : token as github pwd : lakshman.devops@123

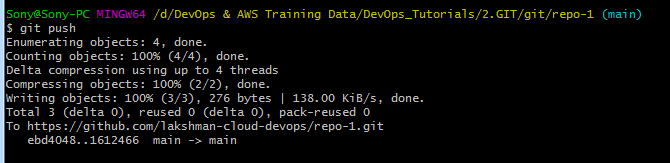
Github profile name : lakshman-cloud



Github pwd: lakshman.devops@123



Note: un/pwd – enter manually (don’t copy paste).



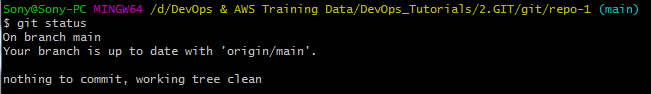
Its pushed to centralized repo(repo-1).

Note :- If your 2nd time creating repo and pushing it may saved credentials already So if you getting any permission/access denied error, follow below steps and try again git push.

**Goto control -> user accounts->credentials manager ->windows credentials-> github->remove->yes.**

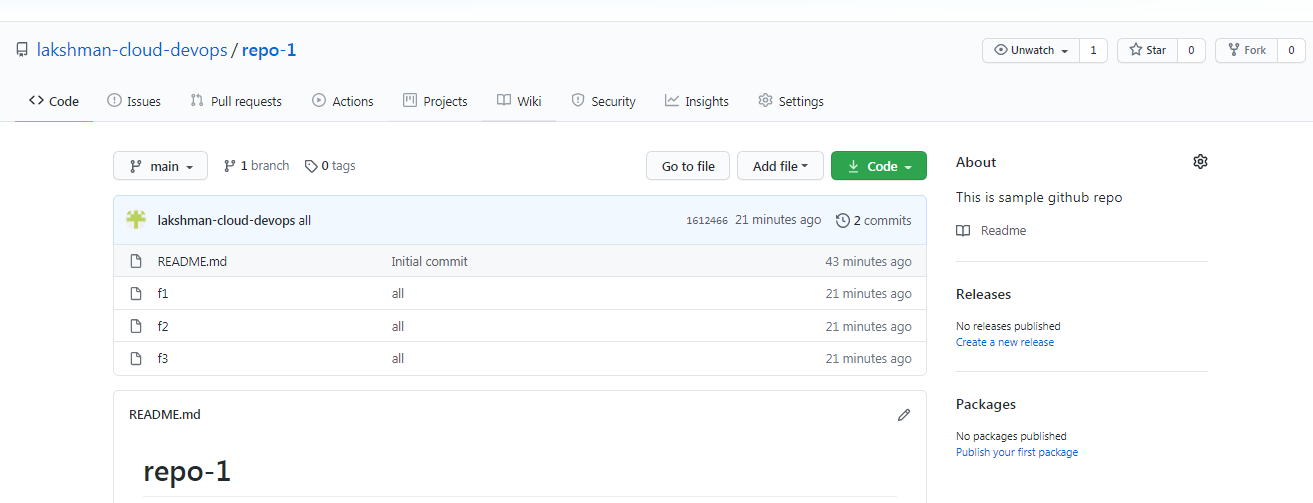
Verify status.

$ git status

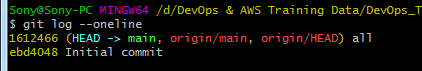


Here no changes to pushed the code.

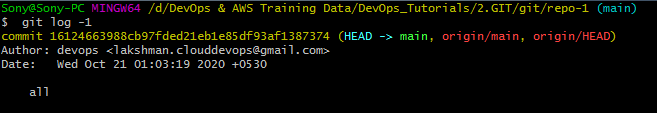
Refresh Github browser , 3 files will shows.



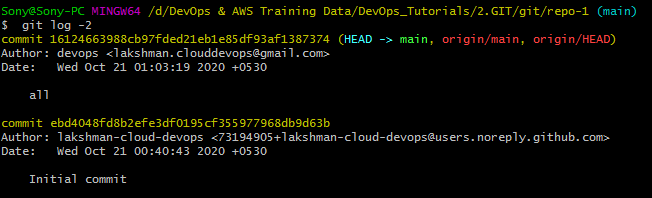
$ git log –-oneline -> last commit short details will show



$ git log -1 -> will show lastest one commit details



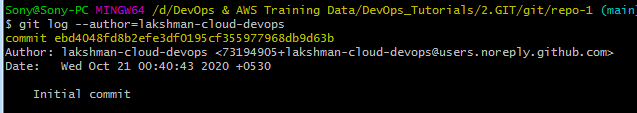
$ git log -2 -> will show lastest two commit details



$ git log --author=lakshman-cloud-devops -> will show user all commits.

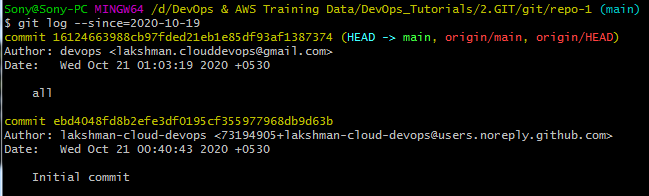
OR

$ git log --author=lakshman-cloud-devops -1 -> will show user lastest commit.



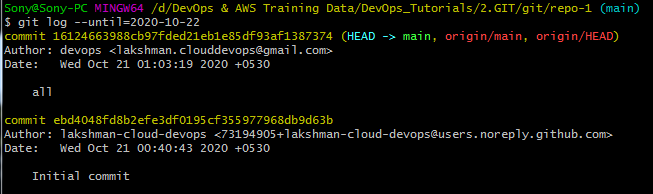
$ git log --since=yyyy-mm-dd

$ git log --since=2020-10-19 -> will show all commits **from date**.



$ git log --until=yyyy-mm-dd

$ git log --until=2020-10-22 -> will show all commits **upto** **to date**.

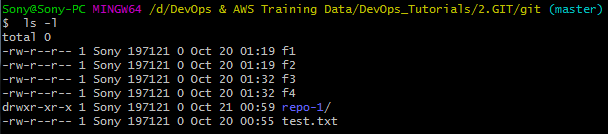


**Branching:-**

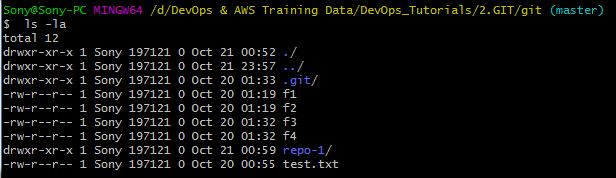
$ git branch -> shows default branch(master).



$ ls -l



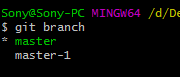
$ ls -la



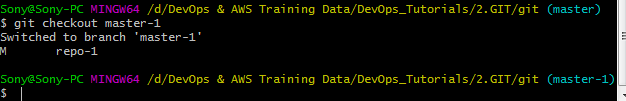
$ git branch master-1 -> create one branch (Ex: with User Story- US12345-Deployment Script)



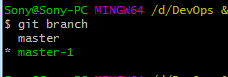
$ git branch -> shows all branch’s



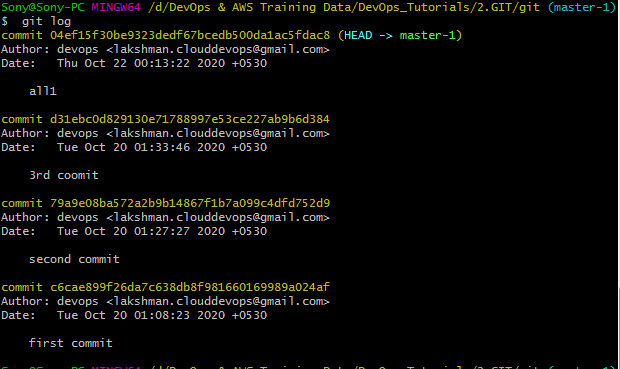
$ git checkout master-1 ->Move/change to another branch with copy of old branch code.

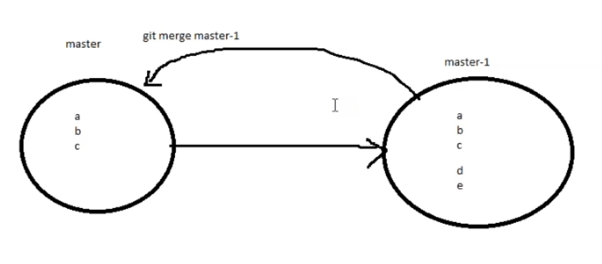


$ git branch -> master will become green color now.



$ git log -> here will shows all branchs include master branchs also.



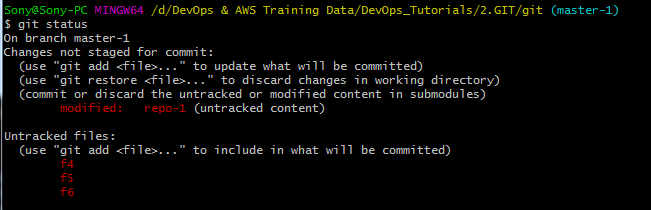


creating some files

$ touch f4 f5 f6



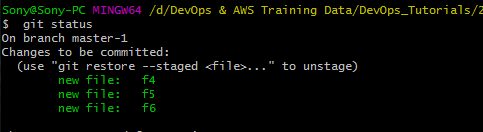
$ git status



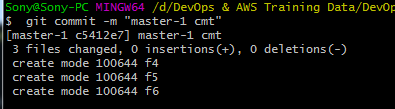
$ git add .



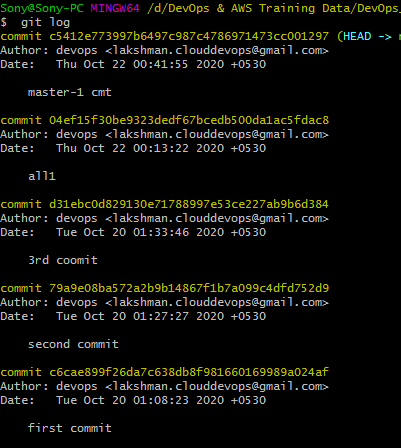
$ git status



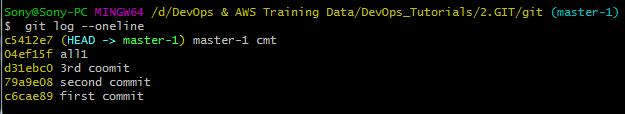
$ git commit -m "master-1 cmt"



$ git log



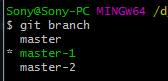
$ git log --oneline



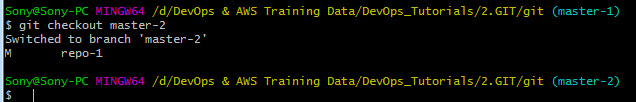
$ git branch master-2 ->created new branch



$ git branch



$ git checkout master-2 -> change branch



$ ls



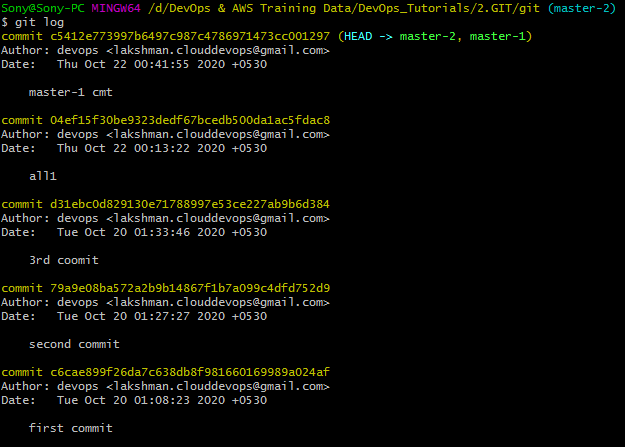
Note:-

master - 1 having f1,f2,f3 repo-1,test.txt

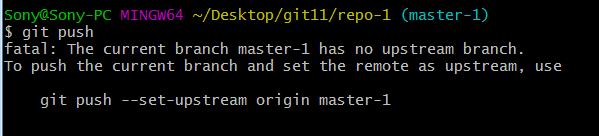
Master - 2 having new files f4,f5,f6.

Here both branchs files are merged with using checkout.

$ git log



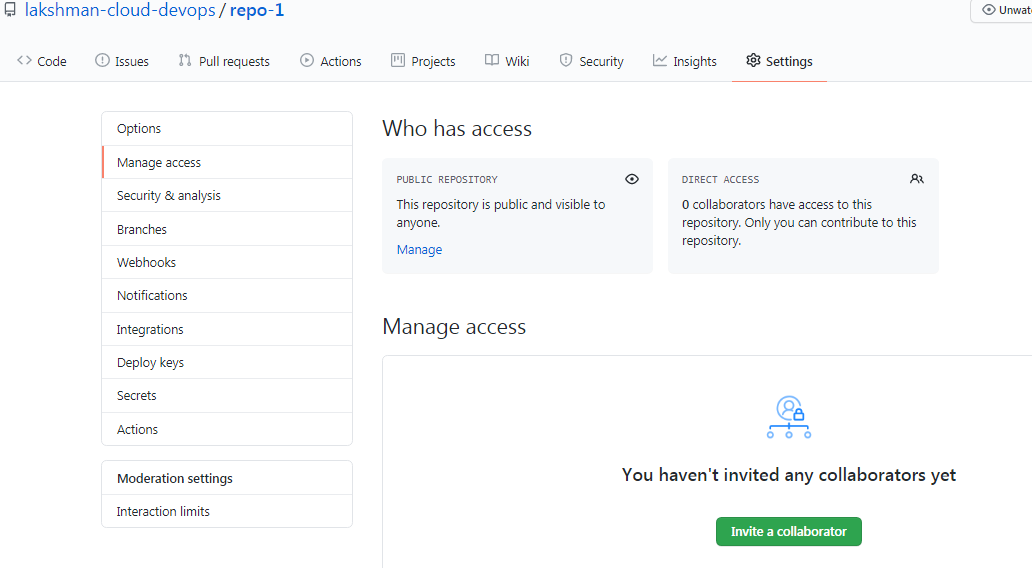
**Note: while pushing with new other branches we have to set upstream manually.**

****

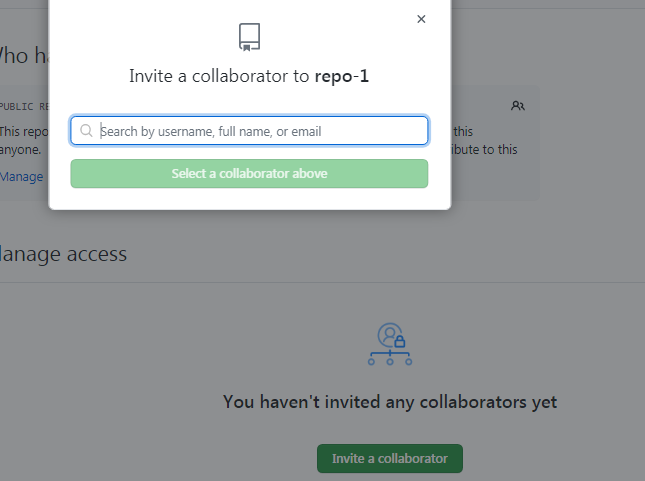
**Provide Permission for others to push the code into ur repository:-**

<https://github.com/lakshman-cloud-devops/repo-1>

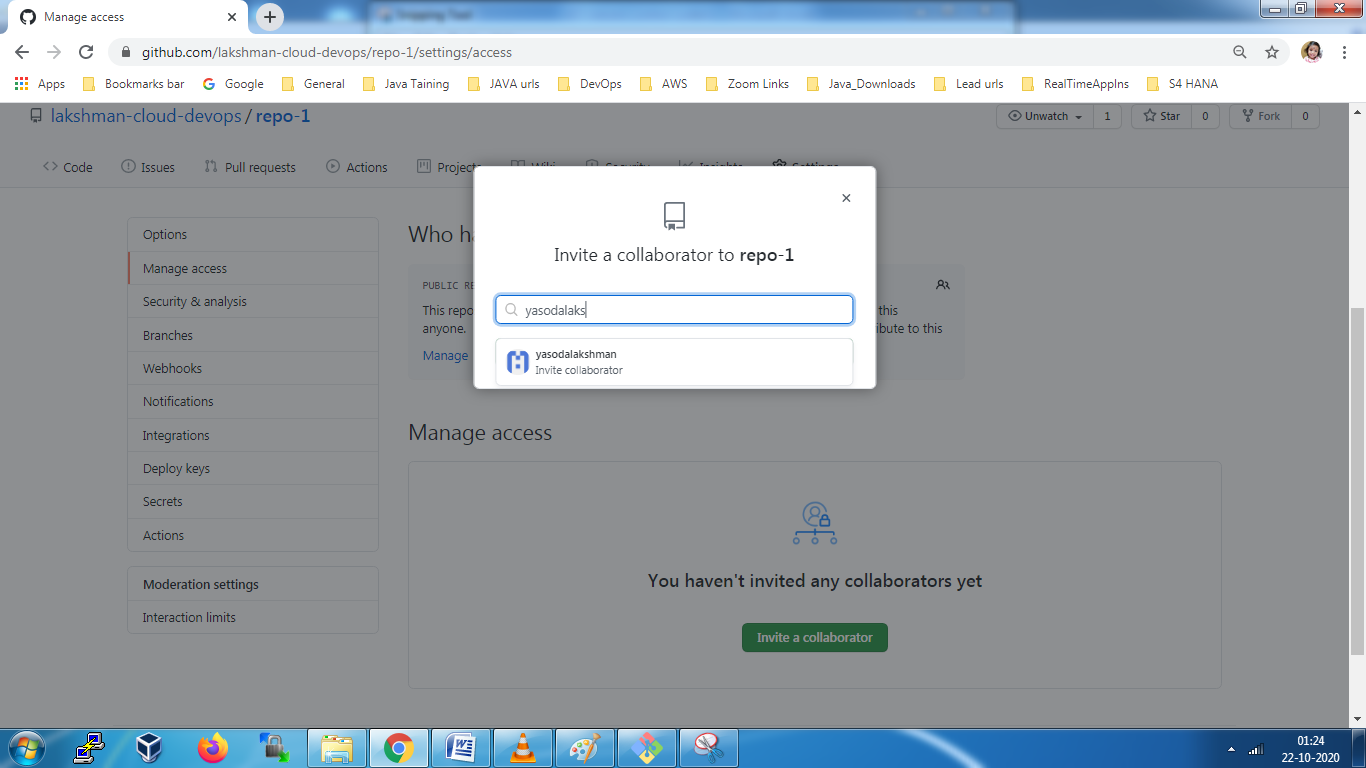
Goto Settings -> Manage Access ->provide ur name (it will as pwd for checking).

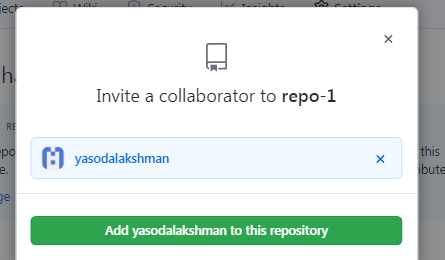


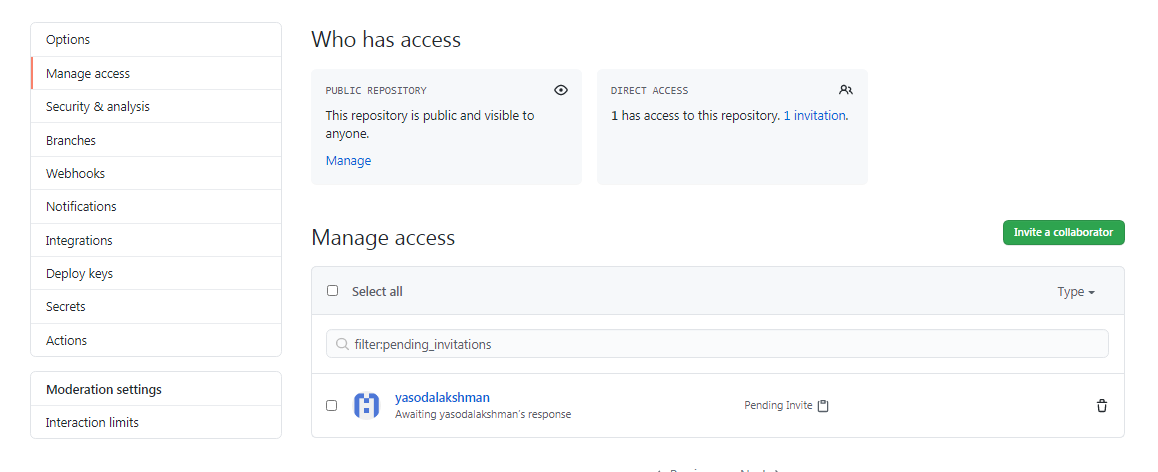
Click invite a collaborator.



Shows github accounts , select whom to provide push access.

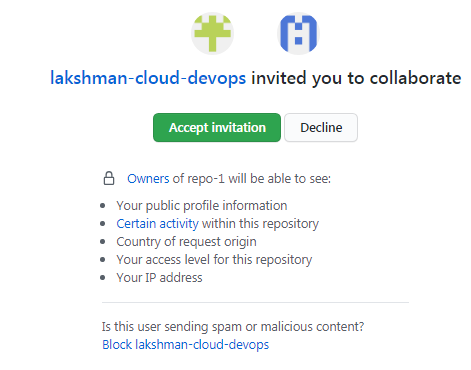






That user will get below invitation and accept the invitation.

Click mail accept link it will goto ur github ac and You will get below accept invitation .



After that you will get that repo(repo-1) into ur github.

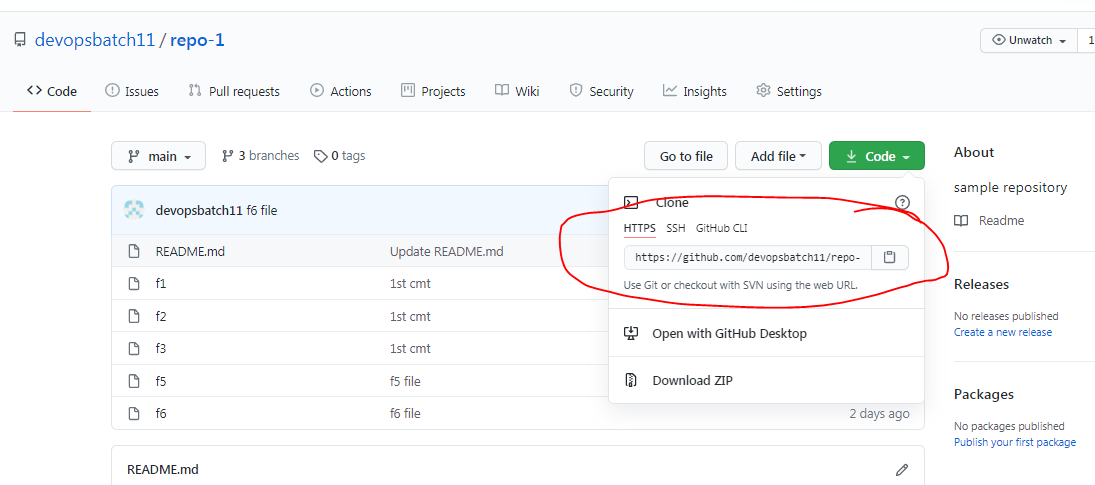


**Other users how to do code push to your repo/branch :- After got permission for my repo.**

create new folder in windows desktop-> Goto that folder

**right click->Open git bash**

git clone <https://github.com/lakshman-cloud-devops/repo-1.git>



cd repo-1

ls

git branch git\_access (create new branch)

git checkout git\_access

touch lakshman.txt

git add lakshman.txt

git commit –m “1st cmt”

git push

-Provide github credentials.->code will pushed to my repo-1.(refresh github repo, pushed file will shows).

Note:- if your getting fatal error- means you don’t have permission or code is not up to date.

If permission issue, take permission for that repo to push the code.

If code is not up to date, use “git pull” cmnd to get up to date code(from remote branch to local), after u can able push the code.

**We can create 3 files:-**

1. Touch



$ touch d1 d2 d3 -> touch used to initially empty file will create.





d1 d2 d3 files are created.

1. Cat

$ cat > d4 -> write what ever you want to store data into d4 file, after ctrl+d (quit and save file)



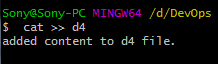


$ cat d4

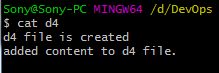


If you want add content to same file.

$ cat >> d4 -> >> used to add content for existing file.

 **ctrl+d**

$ cat d4



Content got added for old data in the file.

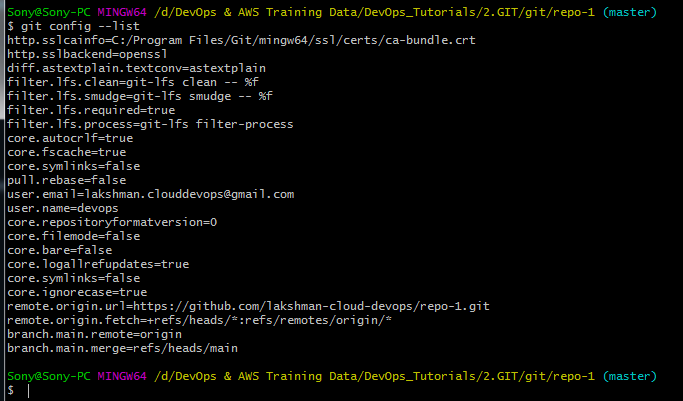
1. Vi editor

$ vi d5 -> write msg , esc:wq! Or esc:wq(! Is for force)(press i for insert mode, save file and quit) or esc:q! (only quit)

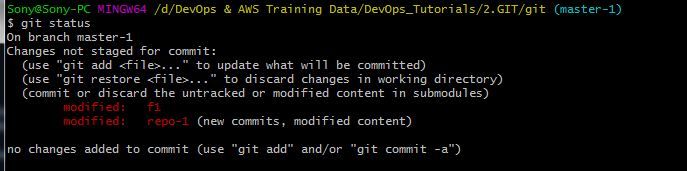


**Creating alias:-**

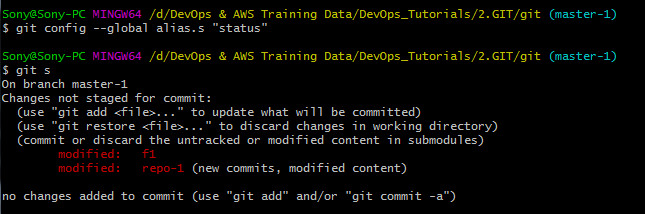
$ git config --list -> it will show list of config files.



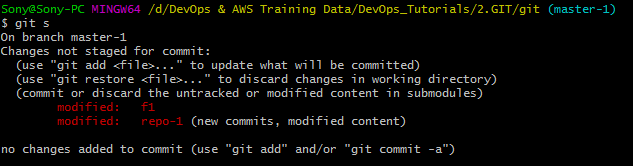
$ git status -> check status



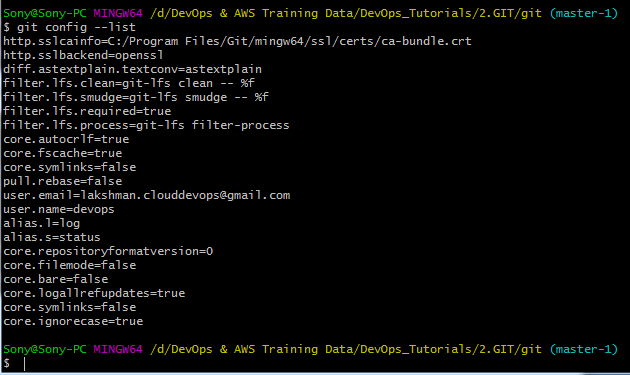
$ git config --global alias.s "status" -> here creating alias name s for git status cmnd.



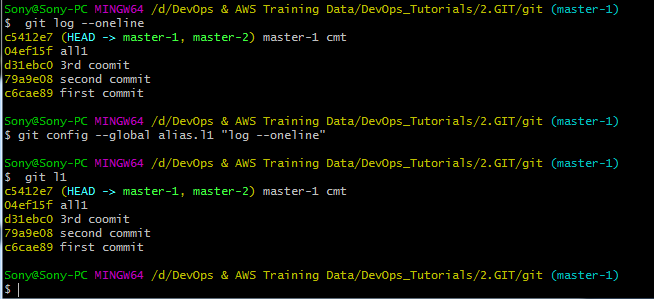
$ git s -> calling with alias name, am getting same output.



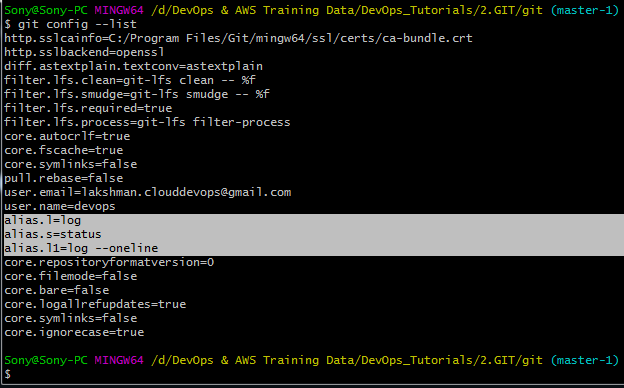
$ git config --list -> Will shows all created alias names.



Setting one more alias for log --oneline:-



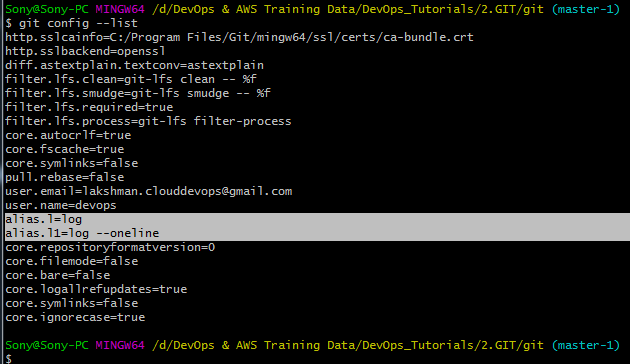
$ git config –list -> verify in config list



$ git config --global --unset alias.s -> remove from config.

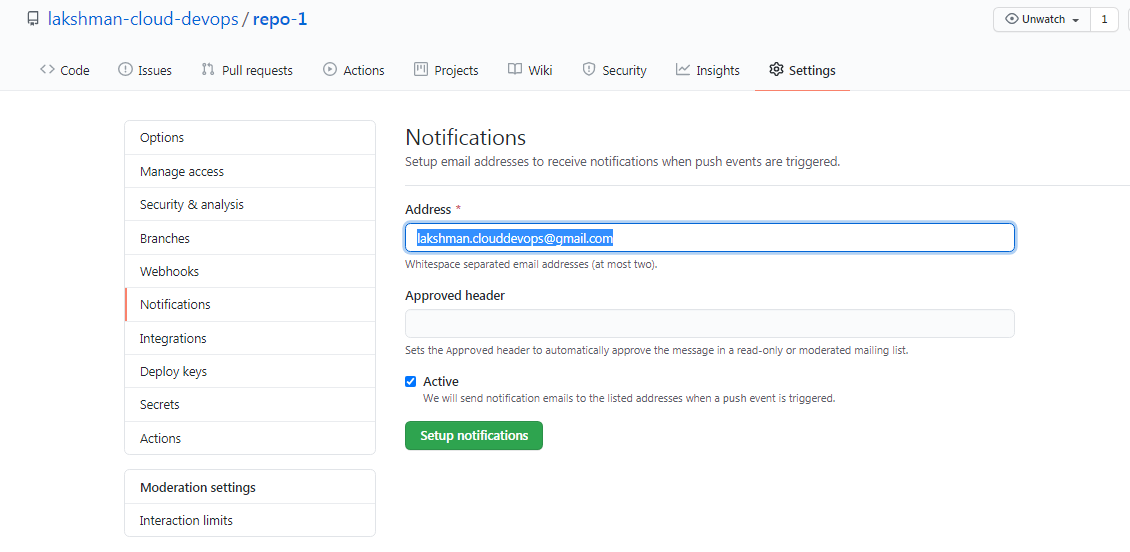


alias.s got removed in config.

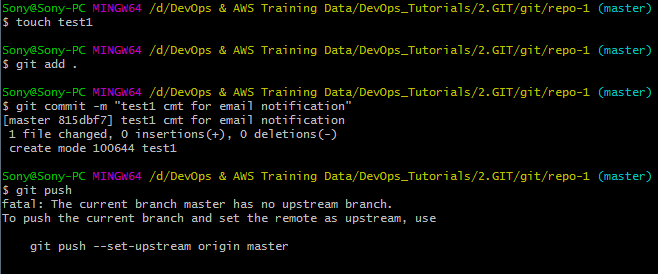


**Note:-** If you want remove user: $ git config --global --unset **user.name** and git config --global --unset **user.email** (devops user name and email will remove).

**Notification mail trigger, if any modification changes in github repository:-**



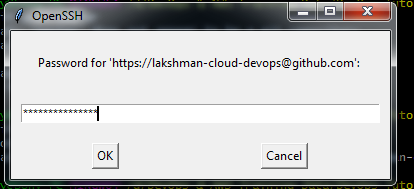
Create new file , add, commit



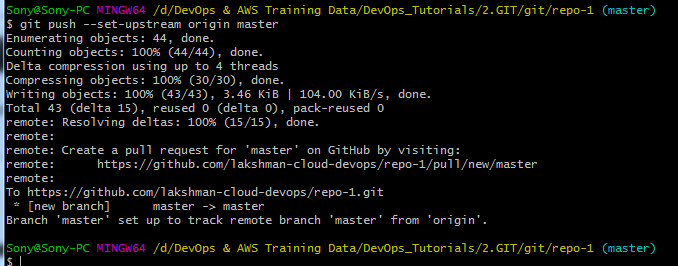
If not push use below cmnd.

$ git push --set-upstream origin master -> it will ask enter

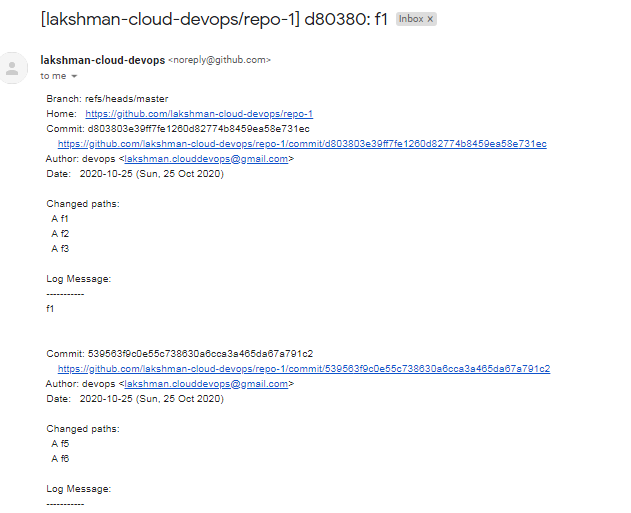
un/pwd : lakshman-cloud-dvops / clouddevops@123



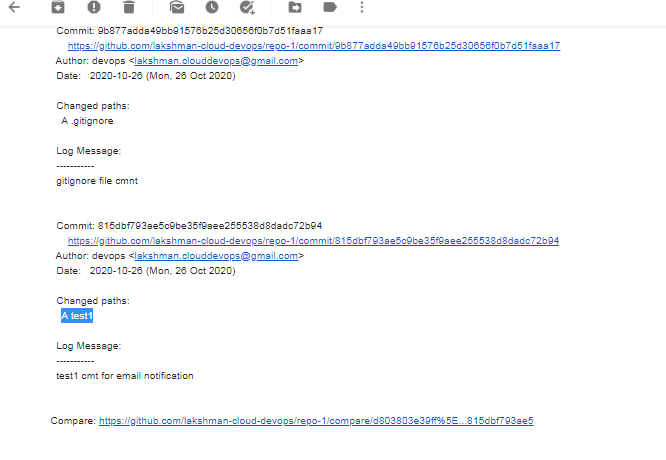
Committed successfully.



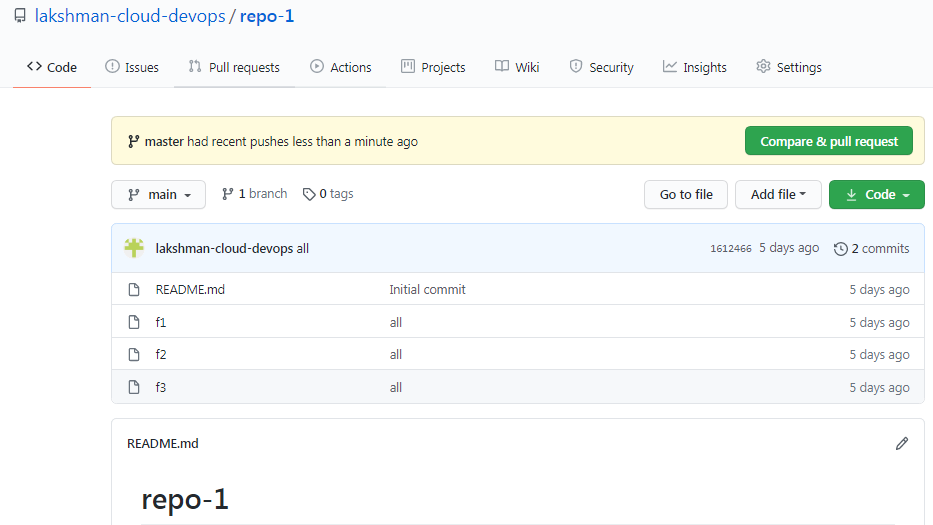
You will get one changes notification mail.



Test1 related changes

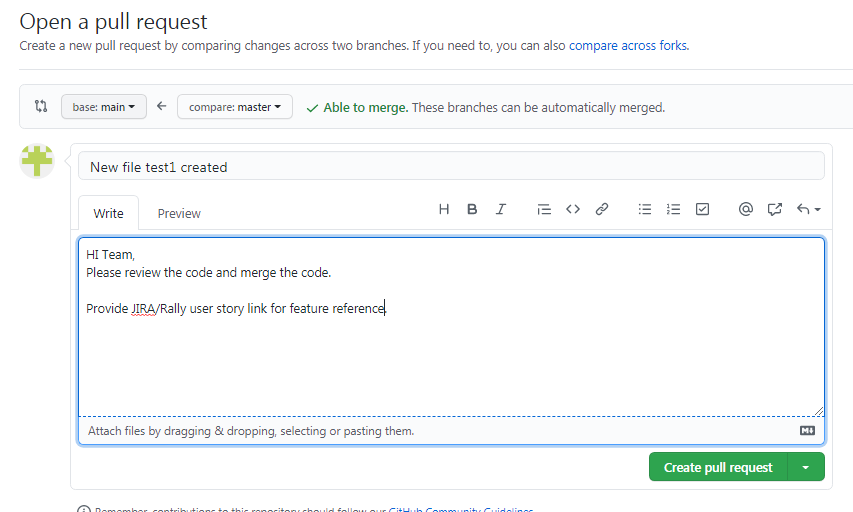


**For create pull request it will show (click compare & pull request).**

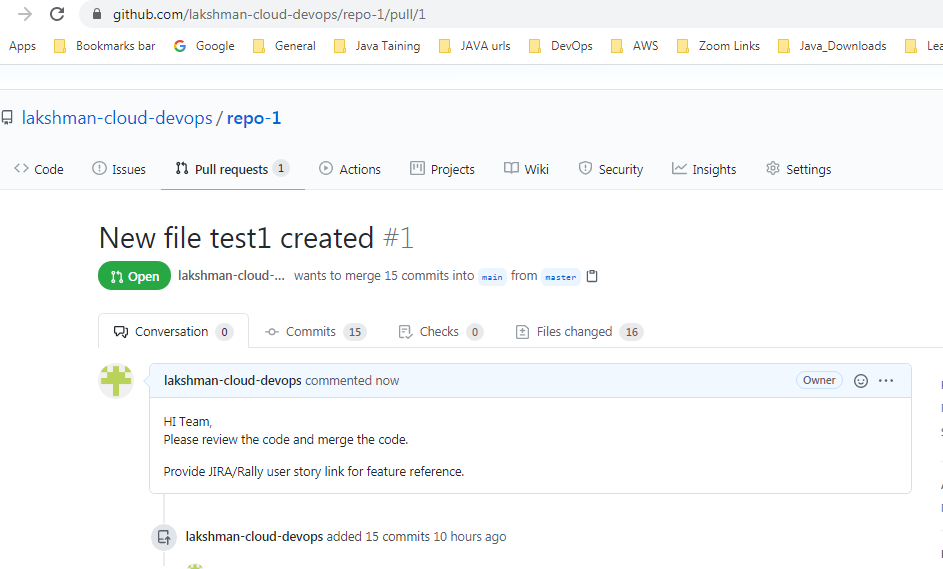


Note:- while pushing code, we need to create master branch(any branch) after code can push from master branch.

**Creating pull request in Real Time way:-**

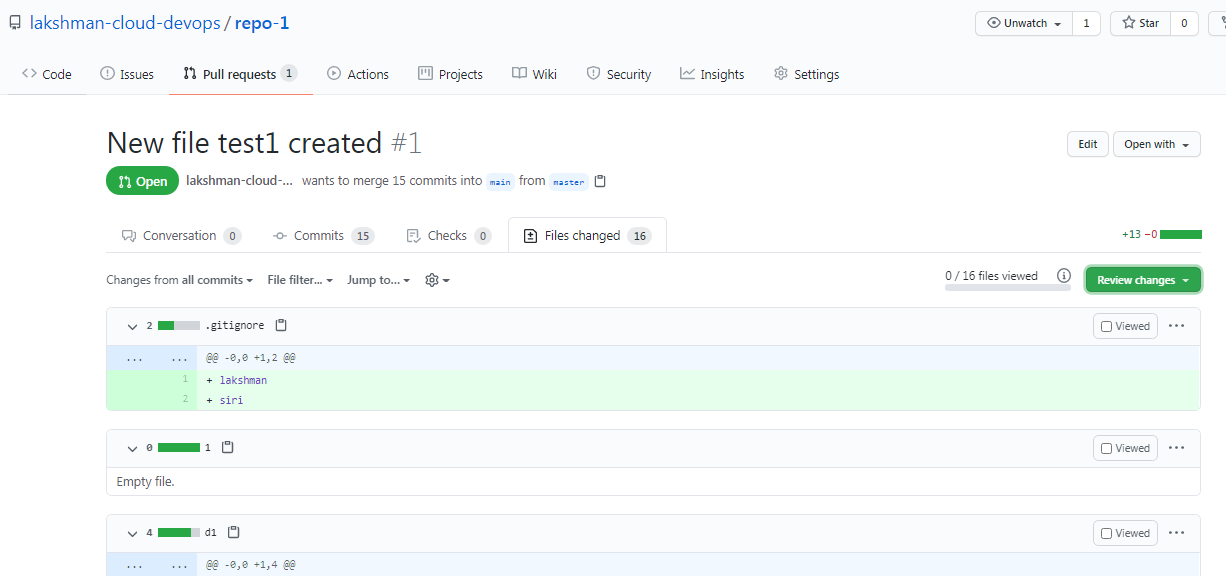


Click pull request

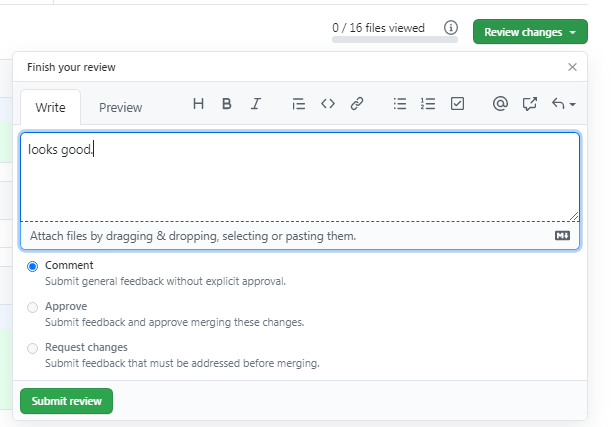


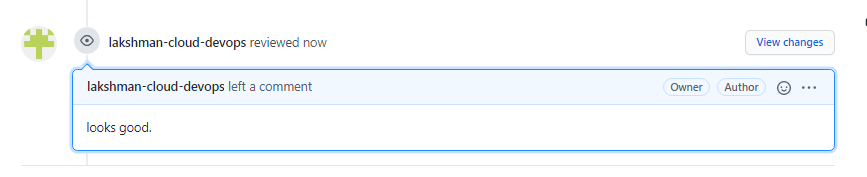
Pull request url : <https://github.com/lakshman-cloud-devops/repo-1/pull/1>

click on file changed -> it will show all file changes from previous file versions.

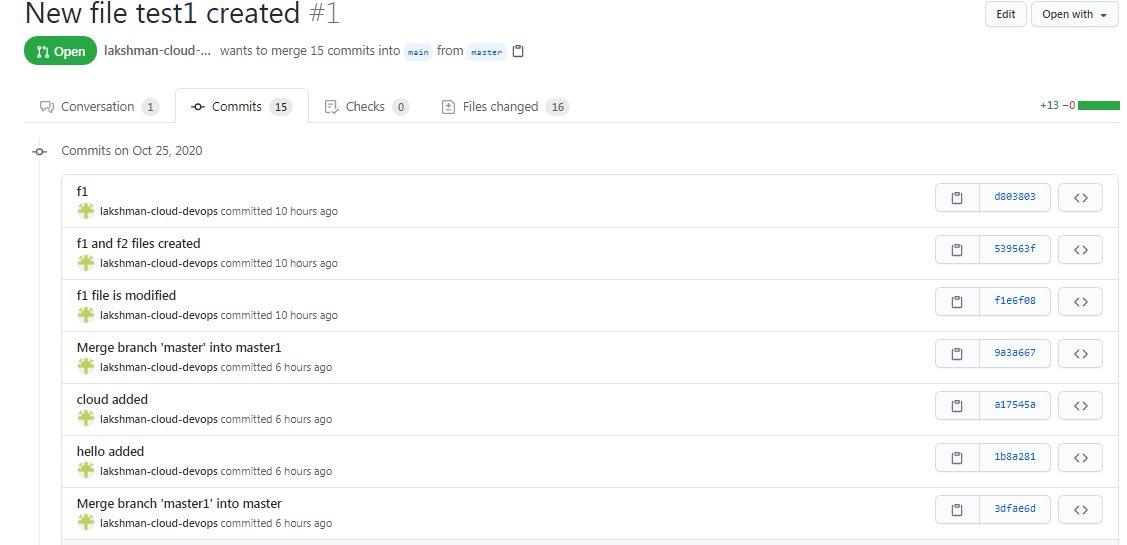


Code reviewer will review your code and comment the review.

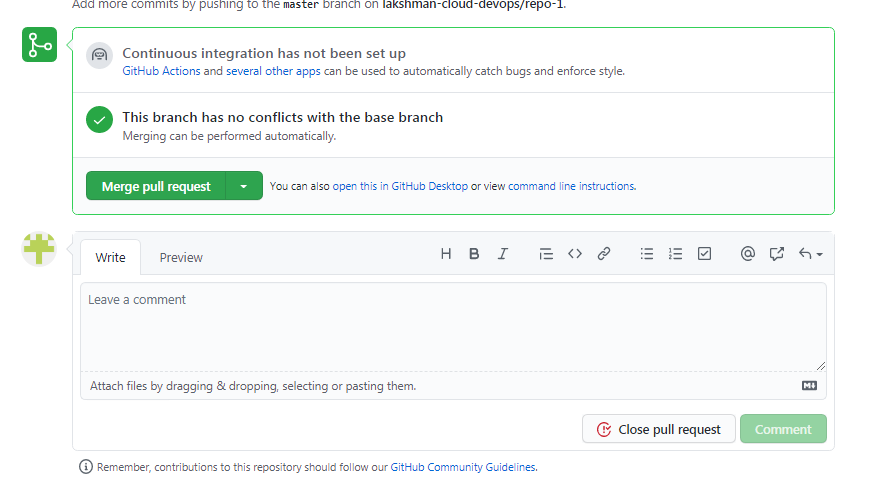


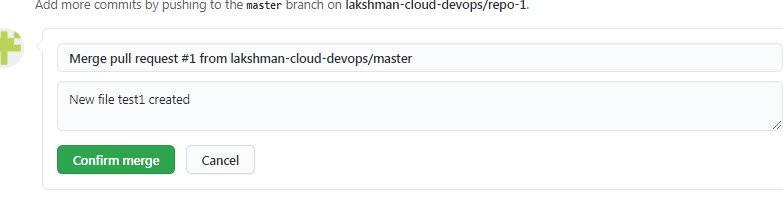


Shows all commits

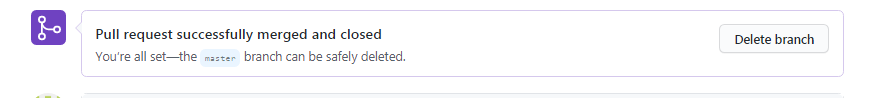


Finally merge the code into main/master repo.

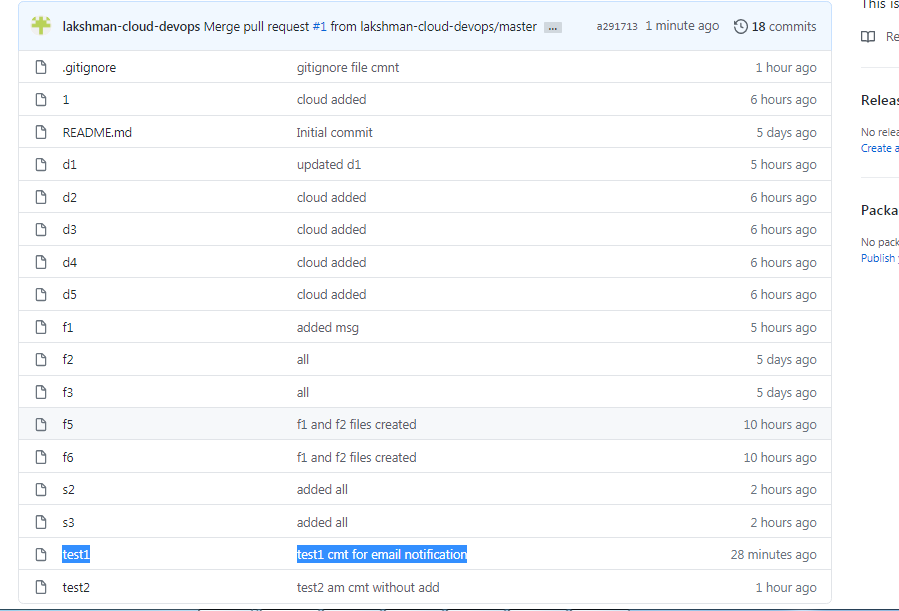




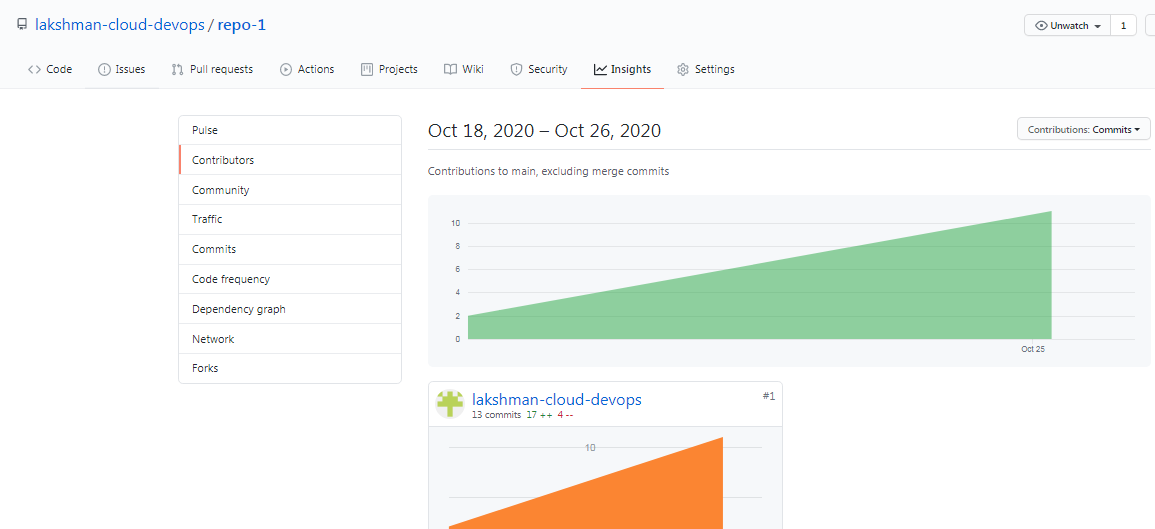
If you want you can delete the branch.



See all files are merged into repo.

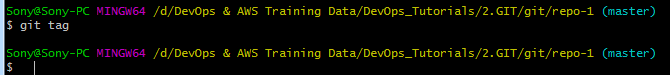


Contributors -> who are code pushing that info will show.

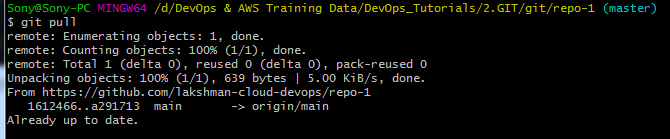


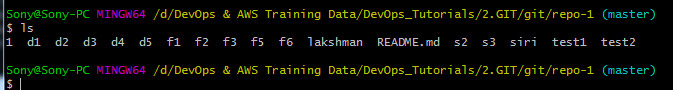
## [Releases](https://github.com/lakshman-cloud-devops/repo-1/releases) :- It will used to release the code versions for quarterly or product version wise.

$ git tag -> verify tags are available or not , it’s every release use the tags

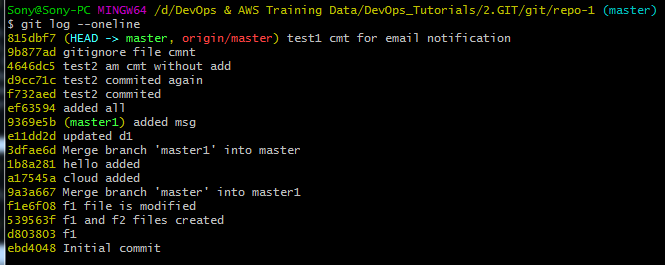


$ git pull -> it will used to pull the latest code from remote github to local repository.



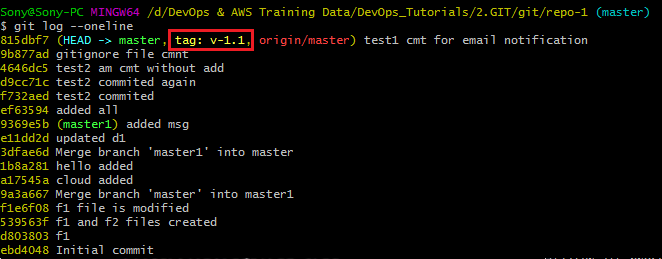


$ git log –oneline -> it will show all short cmt id’s.



Here we can use some cmt’s to create one tag.

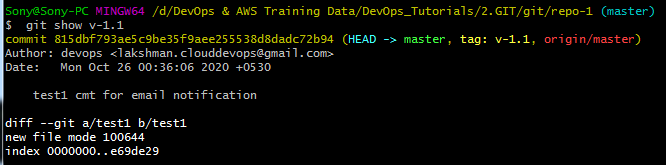
$ git tag v-1.1 -> new tag created.



$ git tag ->display tags

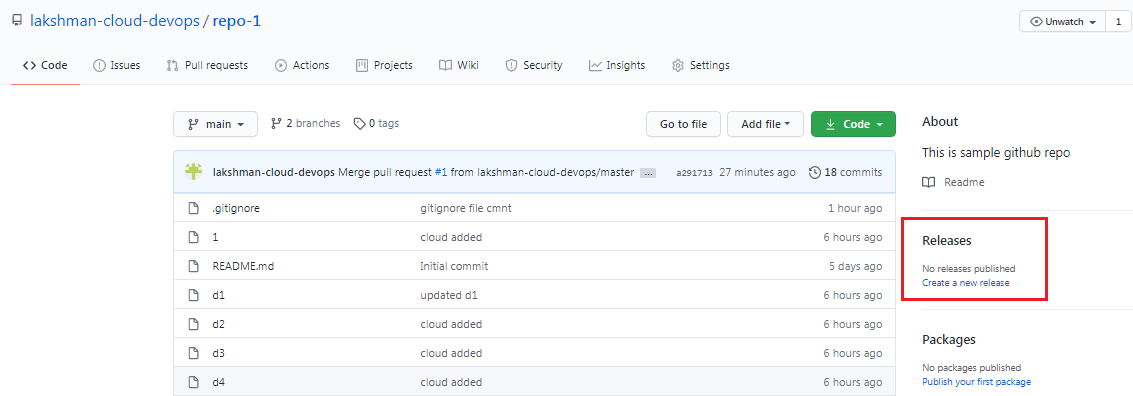


$ git show v-1.1 -> it will show that tag related cmt details.

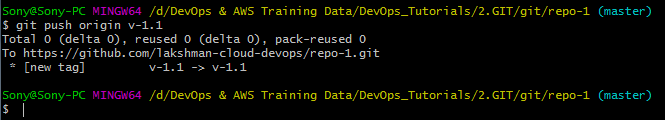


**How to push this tag to centralized repository:-**

As of now no releases.

****

$ git push origin v-1.1 -> push code to centralized repository



## It will ask un/pwd again.

## lakshman-cloud-devops / clouddevops@123

## 

## 

## You will trigger one more mail again for tag push.

## 

## Verify in Github tag got created.

## 

## 

## 

## If you want latest release related code, you can download zip code file or tar file (tar for linux).

$ git tag -d v-1.1 -> delete tag in only local repository(not github).



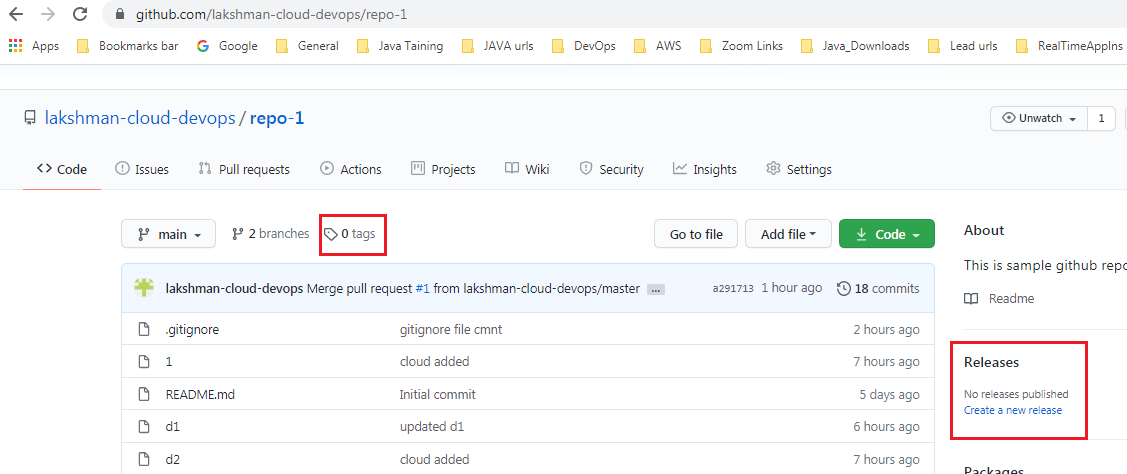
No tags showing.



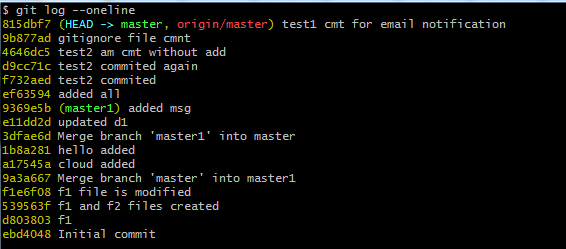
$ git push origin -d v-1.1 -> delete tag/release in centralized repository(github)



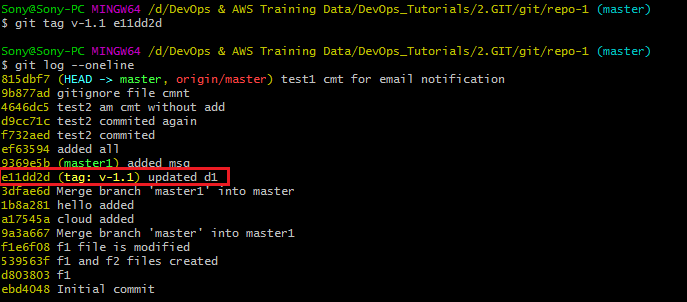
Removed tag/release in centralized repository(github)



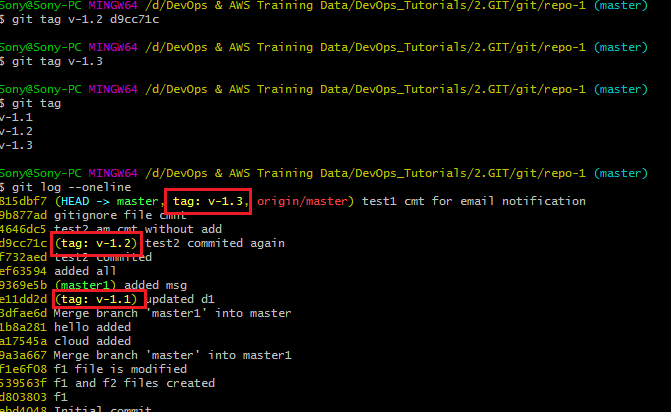
**How to create tag for specific cmt id’s:-**



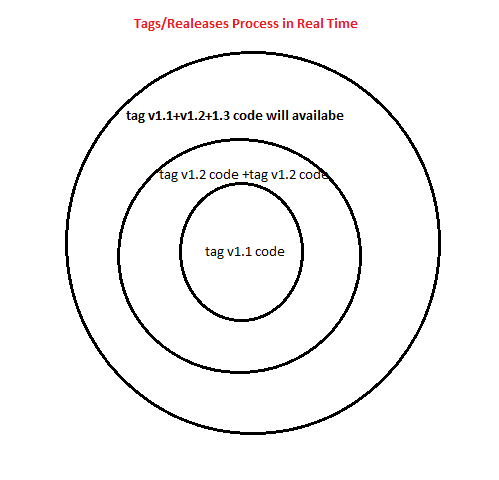
$ git tag v-1.1 e11dd2d -> upto this cmt id(e11dd2d) tag is created.



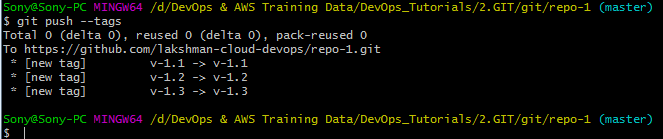
## Now creating another 2 tags .

****

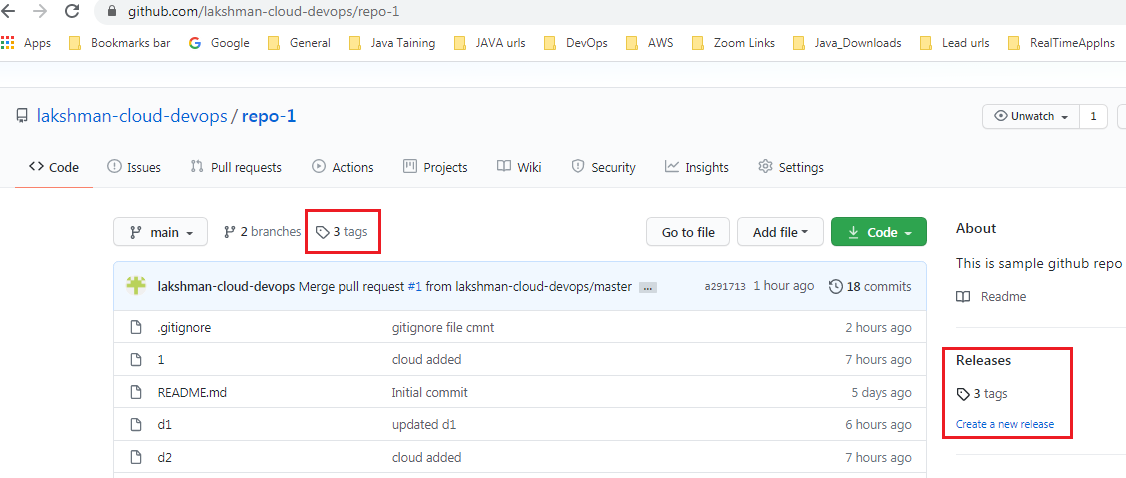
**Note :** $ git tag v-1.3 -> if not given cmt id , it will take with HEAD.

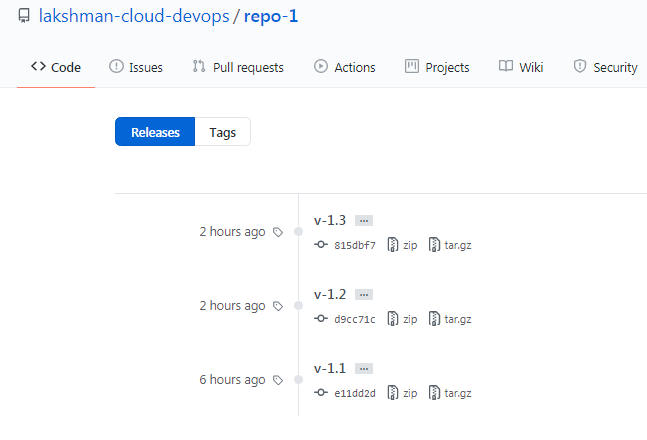


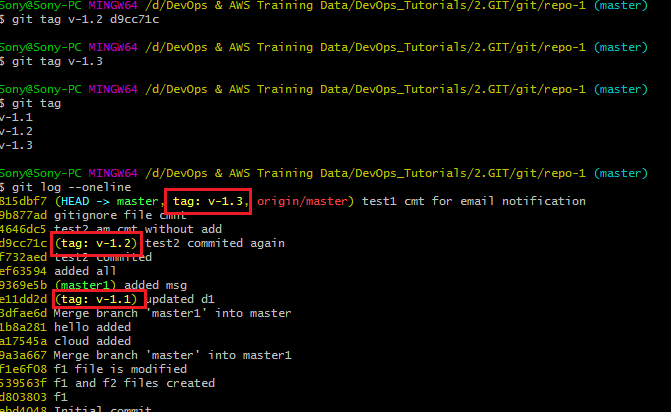
$ git push --tags -> all tags at a time will push to github.(enter un/pwd)



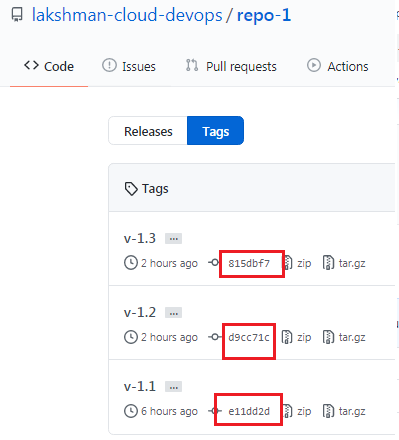
Verify in github







Compare cmt id’s.

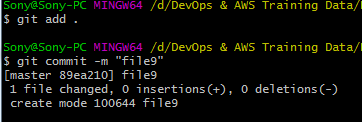


**How to change cmt label msg for latest cmt:-** only possible for latest cmt, but not for old cmts.

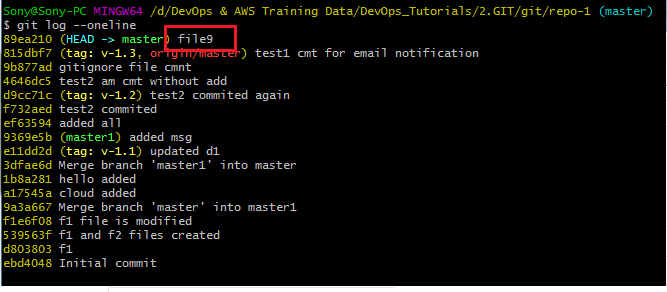
Created file9 file



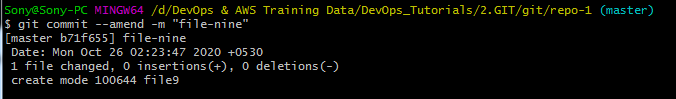
Added and committed.



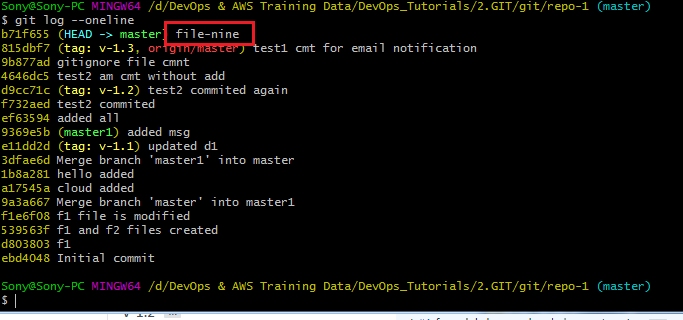
Need to cmt



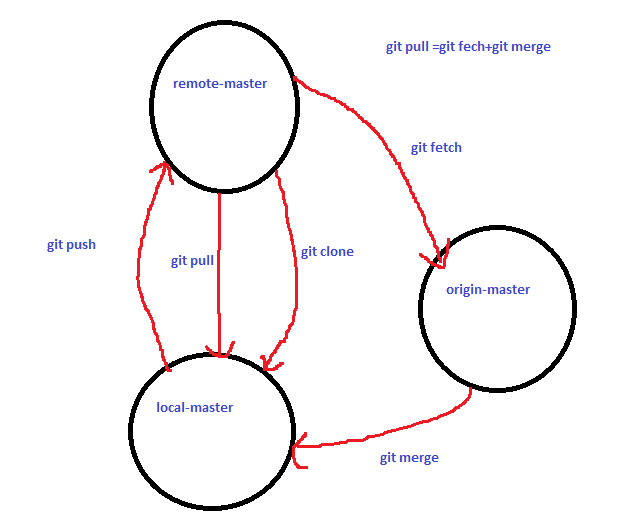
$ git commit --amend -m "file-nine" -> amend used to provide new cmt label for latest cmt.



Verify log -> changed label name.

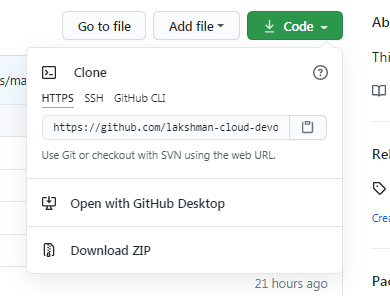


**Real Time GIT Process:-**

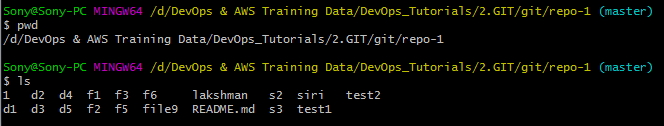


<https://github.com/lakshman-cloud-devops/repo-1>

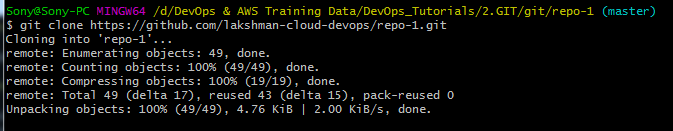
get clone https/ssh url from github



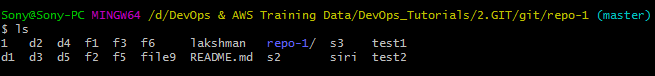
List of directory



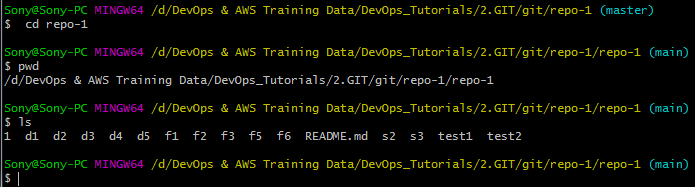
$ git clone <https://github.com/lakshman-cloud-devops/repo-1.git> -> clone the code.



ls



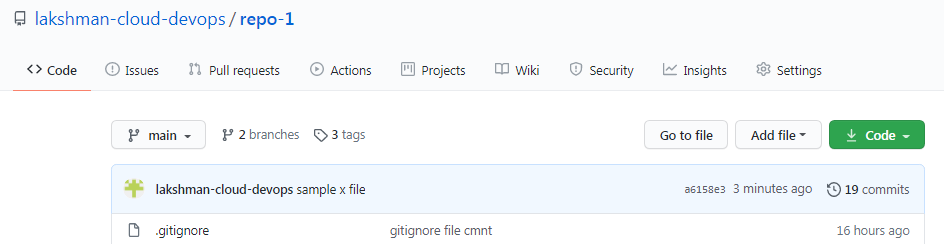
All repo-1 code files came to local master



Created one file and add,commit,push. (push –enter un/pwd)



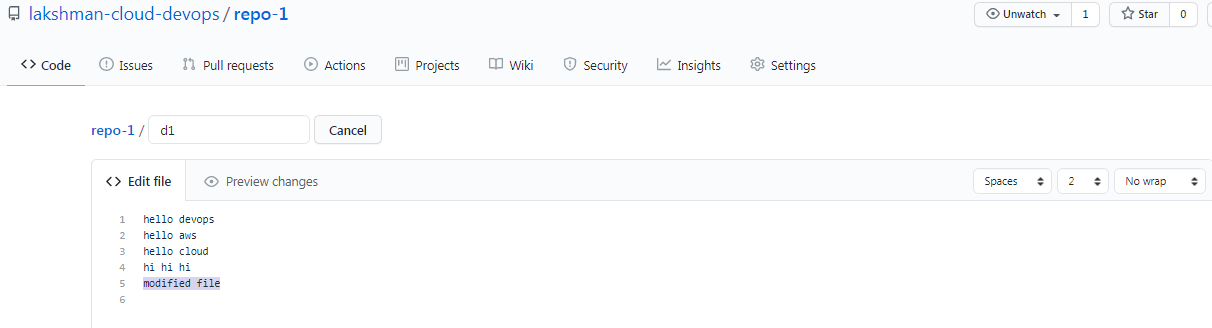
File received



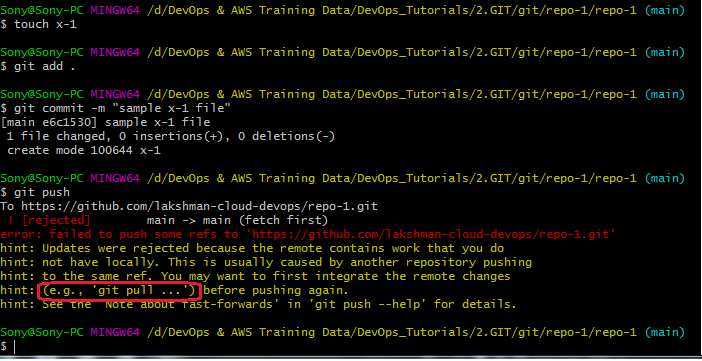
Modified one file in central repository , click commit changes.

OR

Any other developer pushed the code/modified the code.

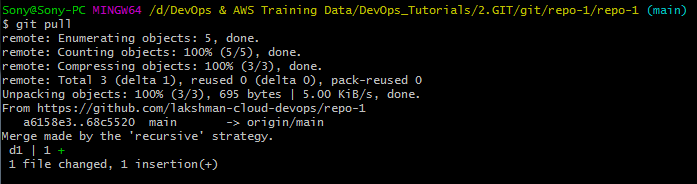


After creating new file again from local and add,commit,push

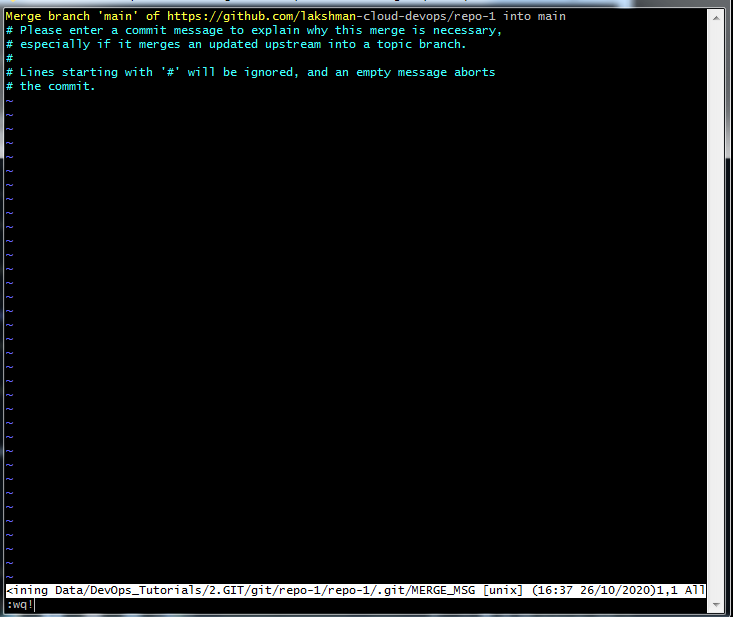


Note :-while push will get fatal error .(ur code is not up to date from github(remote branch))

$ git pull -> get latest code merge into local master (enter un/pwd)

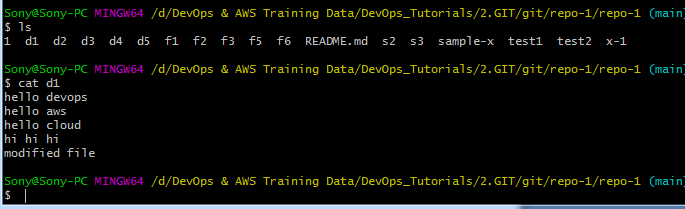


Here merging central to local branch.

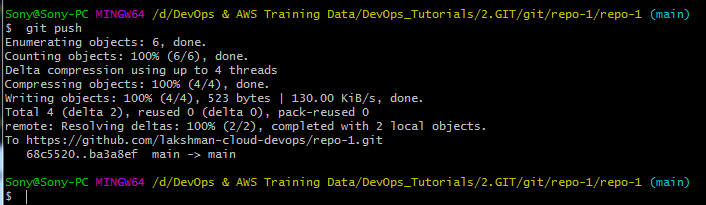


Esc:wq!

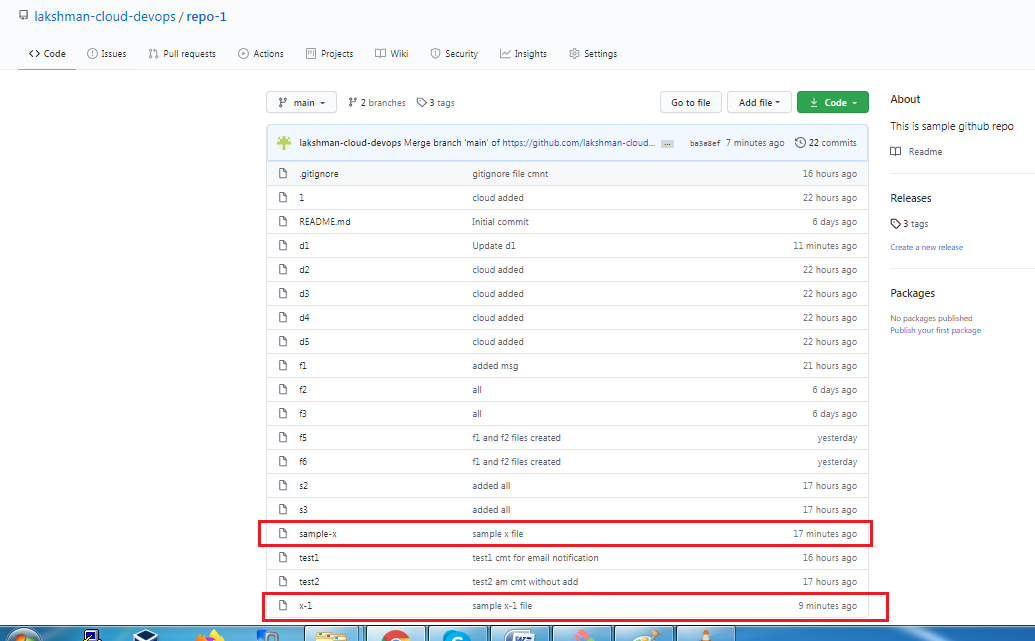
Verify code pulled or not



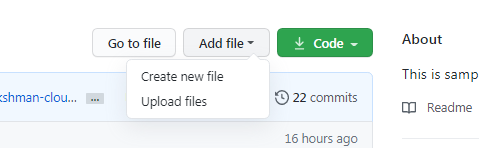
$ git push -> now push code to central repo.

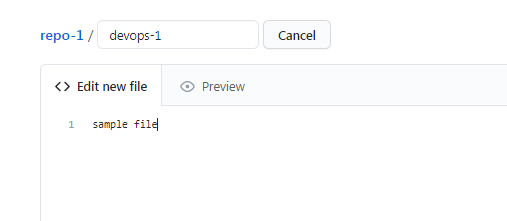


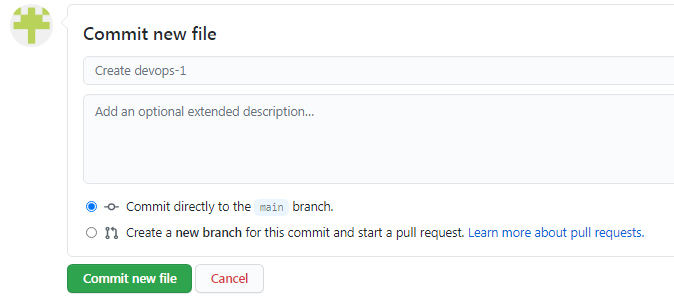
Git pushed to github repo.



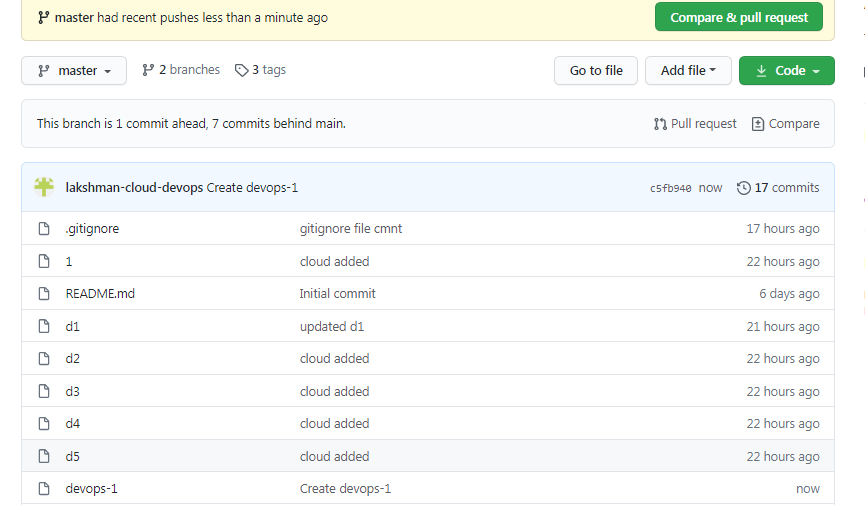
Creating new file in central repo or consider one x developer pushed code.





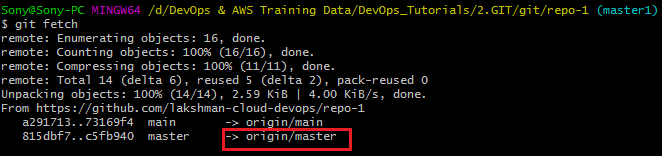


Devops-1 file created

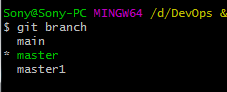




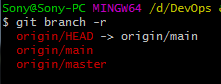
$ git fetch -> code fetch from central to origin-master.



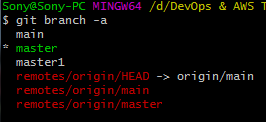
$ git branch -> shows local branches



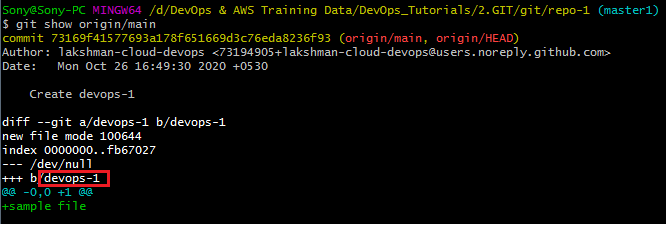
$ git branch –r -> shows remote origin branches



$ git branch –a -> shows local and remote branches



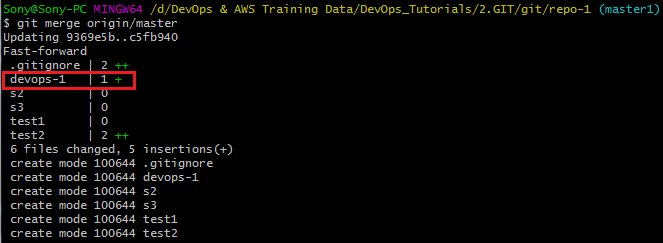
$ git show origin/main -> show the branch details with pending file .



Now that devops-1 file is not there in local ls.

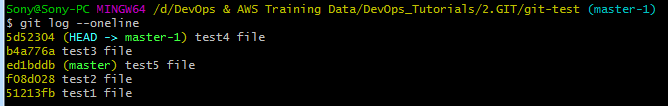


$ git merge origin/master -> merge origin/master branch code files move to local branch.

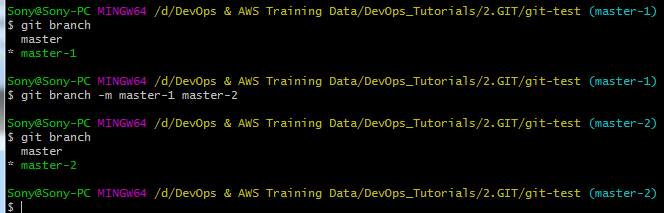


Devops-1 file and some other files merged to local branch

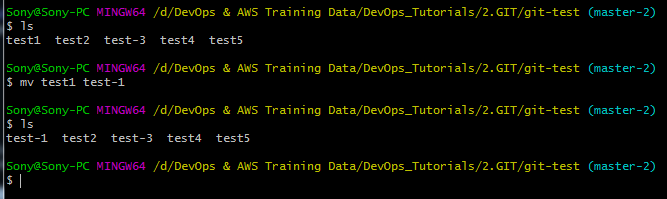




**Rename branch name:-**

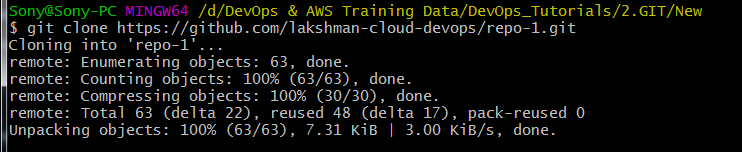


**Rename file name:-**

****

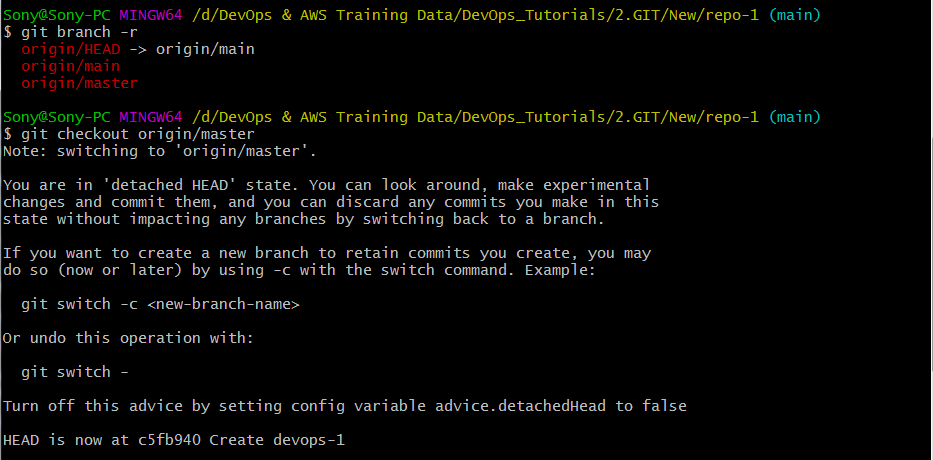
**How to switch remote branches:-**

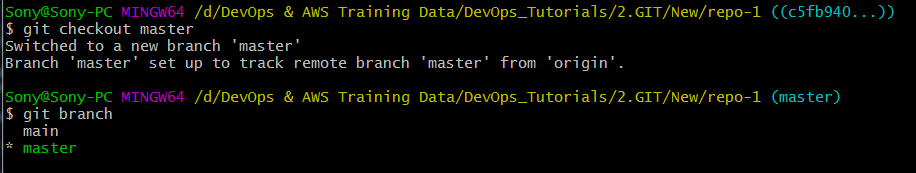
**Goto any new folder**

****

****

**How to check other remote branches.**

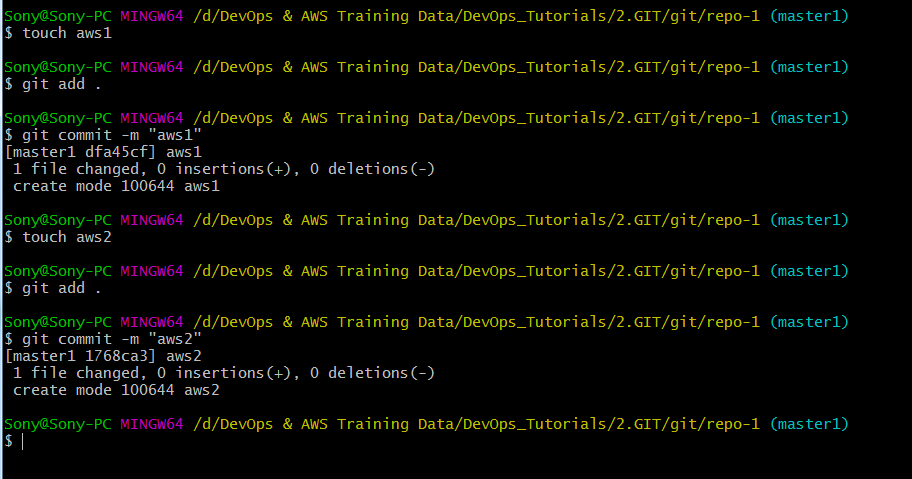
****

****

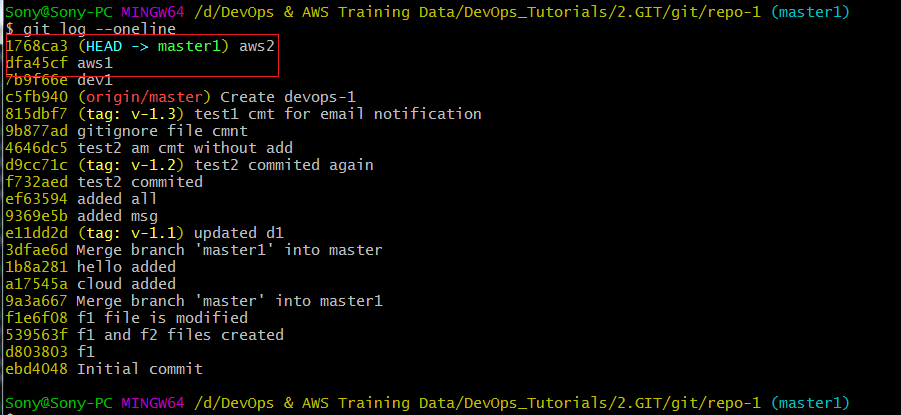
**Main and master 2 branches came to local.**

**How to merge only specific cmt code from one branch to other branch:-**

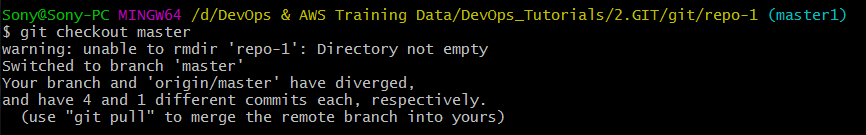
**Created 2 files and add,cmt done.**

****

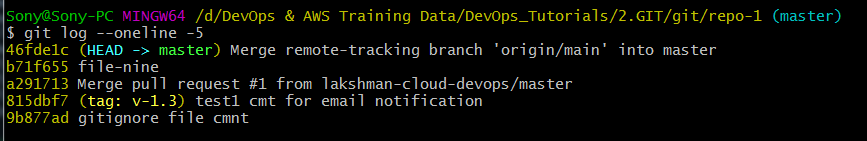
**2 cmt ids generated.**

****

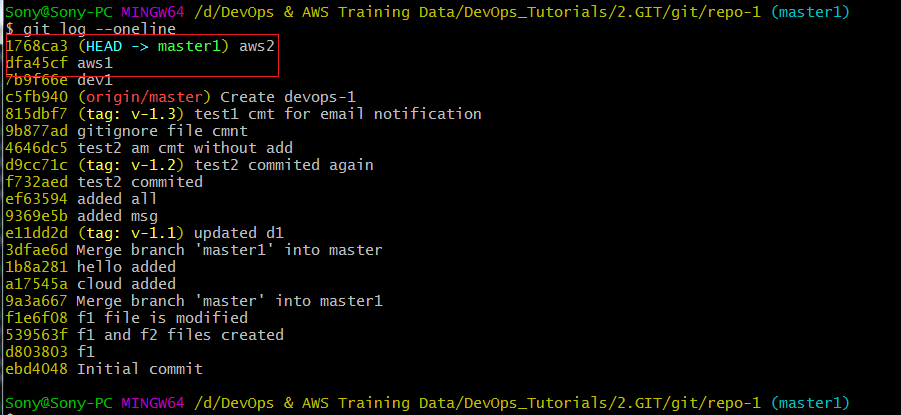
**Checkout to master**

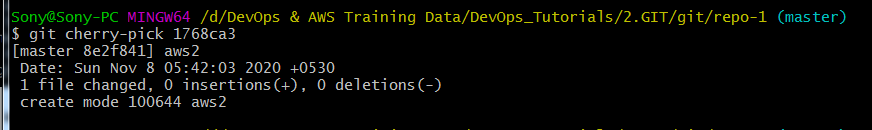
****

**Verify latest 5 logs**

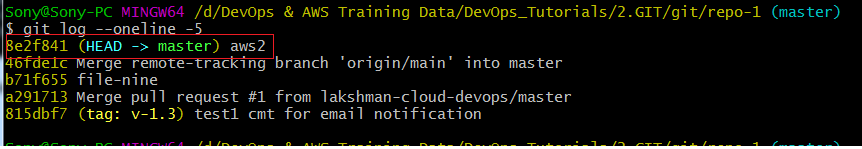
****

**Merge specific cmt id to master branch**

****

****

**Verify latest 5 logs-> aws2 code merged to master with new cmt id.**

****

**================================GIT END============================================**