GIT/GITHUB:-

WHAT IS N WHY needed?

-GIT Is a VCS(VERSION Control System) or SCM(Source Code Mgmt Tool)

-VCS is used for controlling the version of any system like file system.

-for eg:-if we create suryakant\_resume.docx which contains the skills A,B.By some time later in period of time he modifies n updates the suryakant\_resume.docx with skills contains B,C,D.

If after some time he needs the intial file which contains skills A,B in it.He cant retrieve it rght?

+THIS IS POSSIBLE TO RETRIEVE Old docx as well from very intial doc to second doc to latest doc also if we store all docx on Git or VCS systems.

+ if we are storing the files in local ,they can be deleted or can be unable to get back if system gets defected.

+if we are storing the files like suryakant\_resume.docx,suryakant\_resume\_V1.docx,suryakant\_resume\_V2.docx,..suryakant\_resume\_Vn.docx

It will lead to consume more space of storage device

-These are some of the problems with versioning of doc,which can be solved by using VCS like GIT etc.

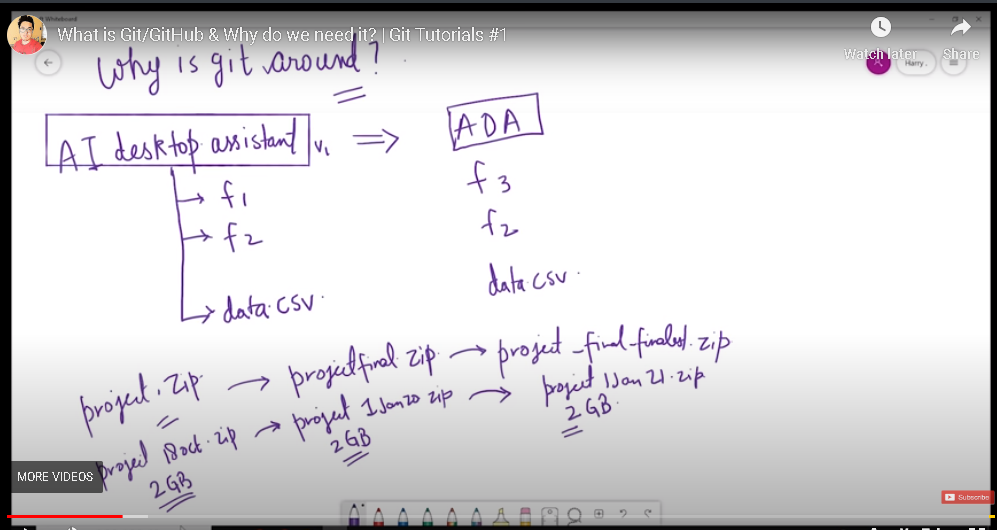
-GITHUB is a one of the hosting server thats hosts git repositary

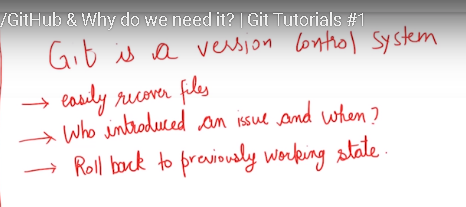
-other eg:-BITBUCKET,GITLAB etc.

-Git is developed by Linus Torwalds(Developer of Linux Kernal or Linux OS).

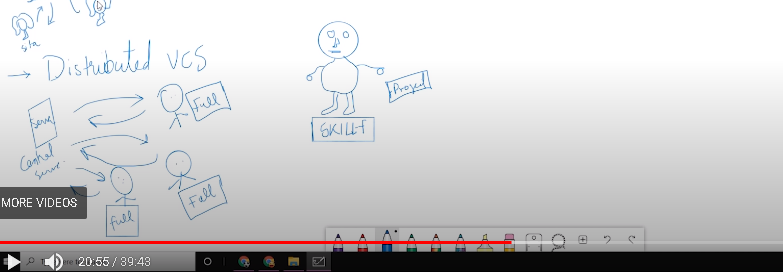
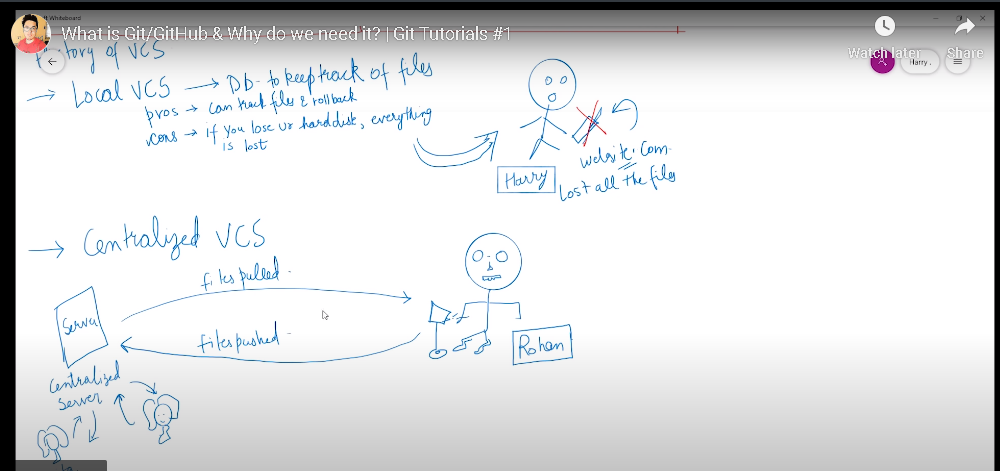
Video URL-<https://youtu.be/evknSAkUIvs>

WHY GIT IS AROUND:

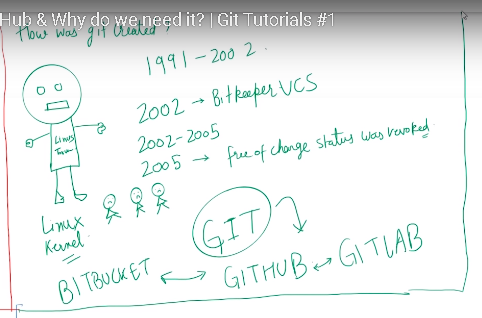




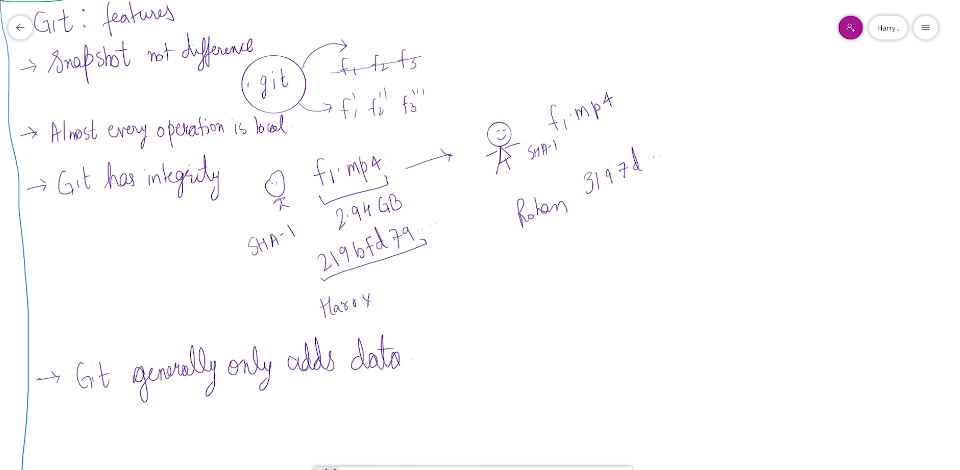
HISTORY OF VCS:-



HOW GIT CREATED:-

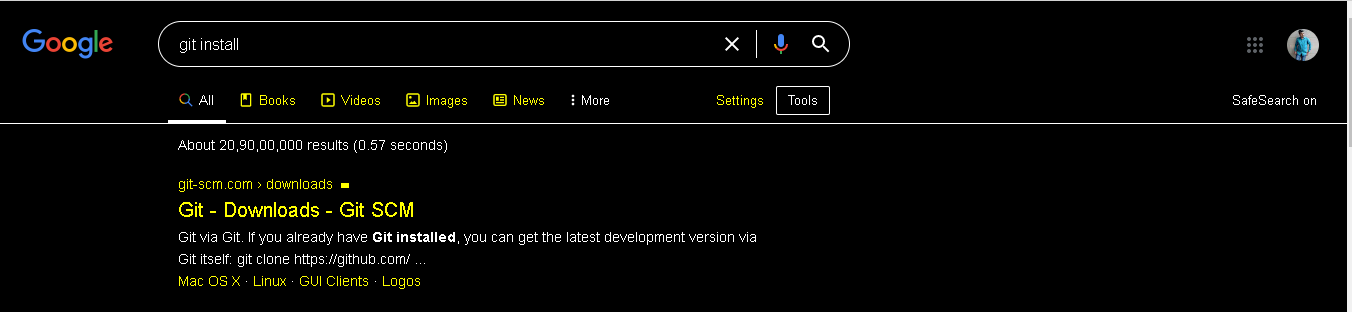


GIT FEATURES:-

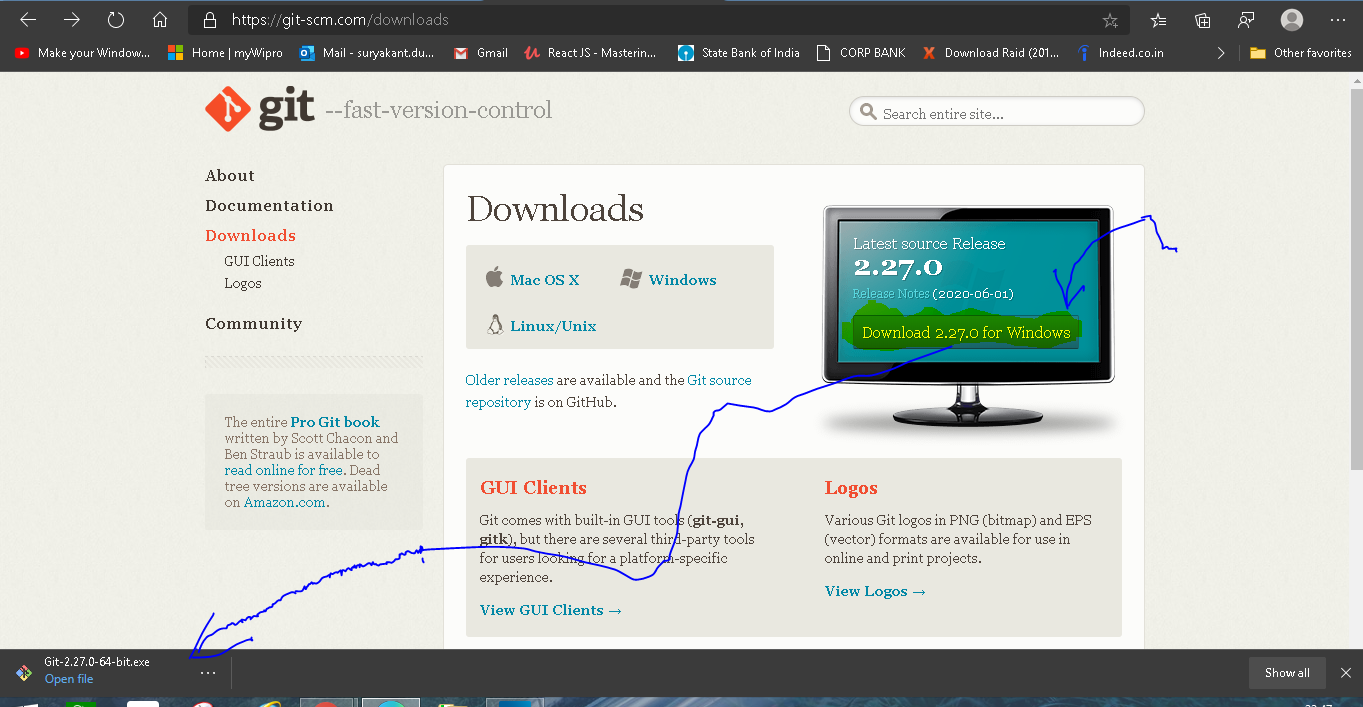


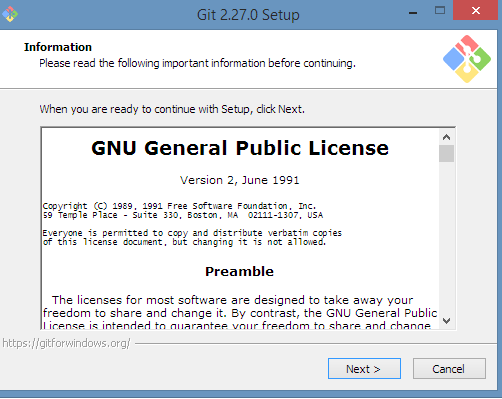
How to install GIT:-

Video url:-<https://youtu.be/qkbK31dMNfM>

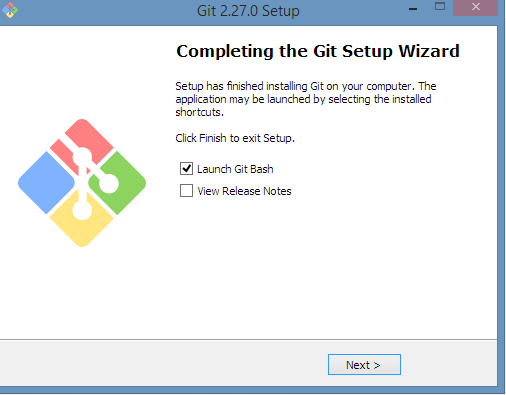
1)

2)



3)OPEN Downloaded file->

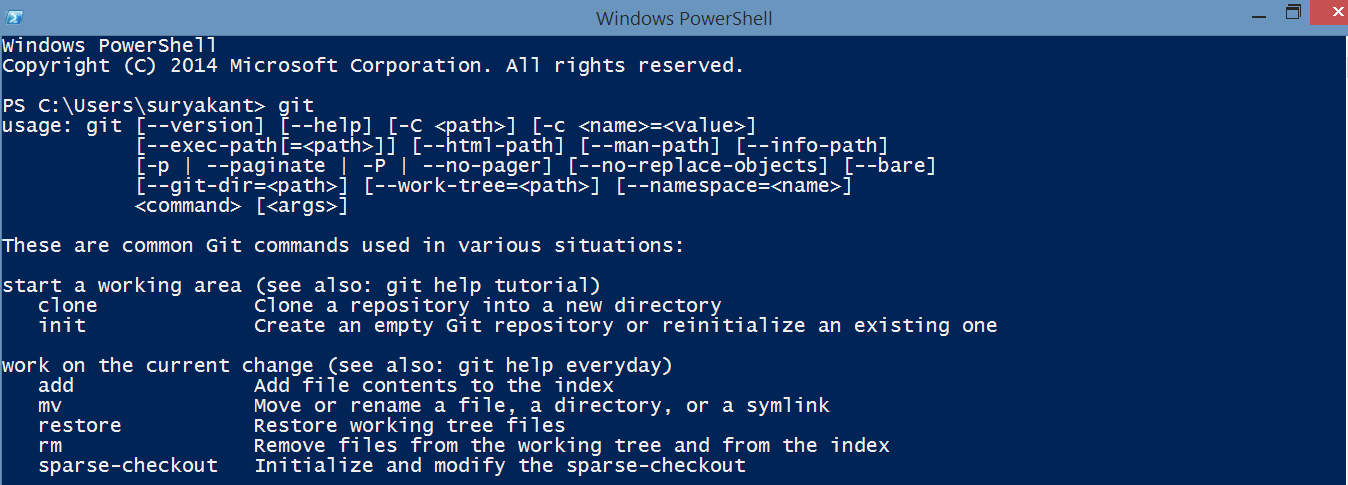
Click next->next->next....->until it gives complete installation window.



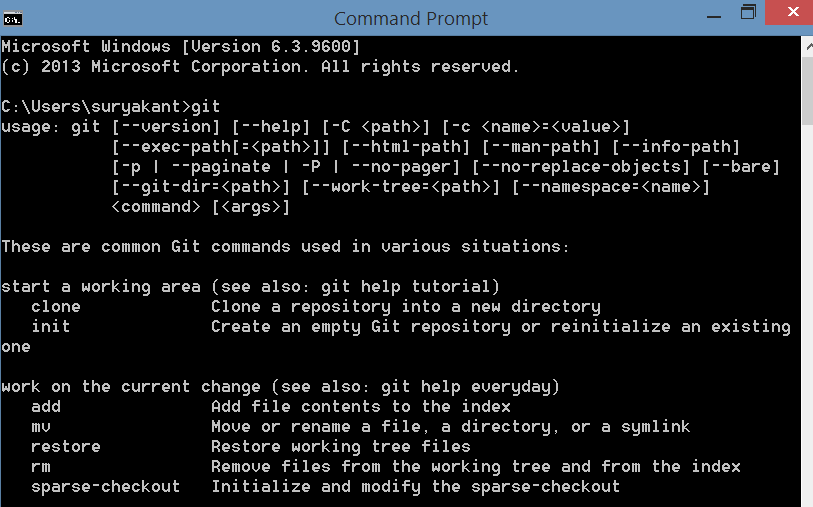
GIT BASH-IT IS A TERMINAL SAME AS WINDOWS CMD OR POWERSHELL,AS LINUX'S terminal,as mac os's terminal.it is command line tool used to fire linux commands and git commands like ls,pwd,git status etc.the look is like a linux

4)to check git installation:-

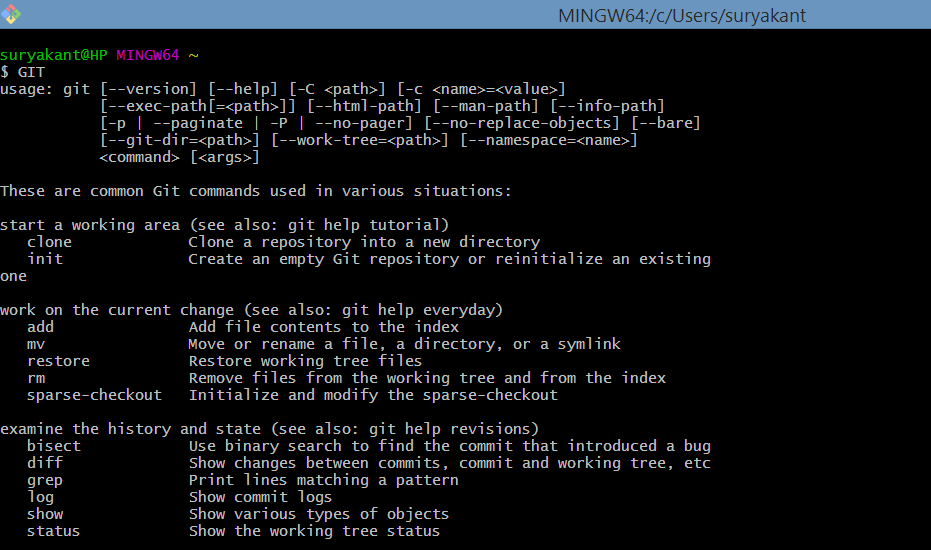
Using Windows POWERSHELL:-open windows powershell



Using Command Prompt:-open Command Prompt

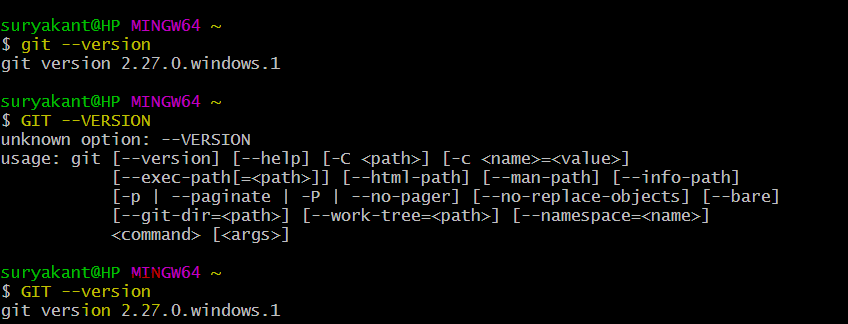


Using GIT BASH:-Open GITBASH



Note:-to increase the font size of letters in GIT BASH press **ctrl and +** and to reduce font size **ctrl and -** symbol.

To check git version in GIT bash OR WINDOWS POWERSHELL: **git --version**



Note:-in commands git can be capital GIT But actual commands like --version etc. should be as mentioned only with correct case and spelling.

Wrong command:- GIT --VERSION

Right command:- GIT --version or git --version

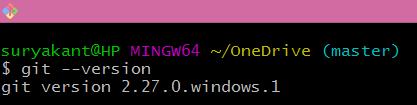
GIT INSTALLATION COMES WITH

* GIT BASH-COMMAND LINE TOOL FOR ACCESSING GIT FUNCTIONALITIES
* GIT GUI-GUI TOOL FOR ACCESSING GIT FUNCTIONALITIES
* GIT CMD-CMD TOOL

Git can be used not only with commands but also with GUI:-You can perform following commands

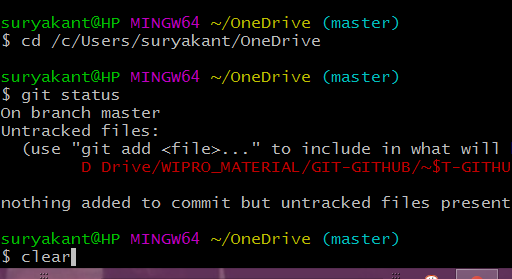
TERMINAL WORKS IN DIRECTORY or folder you can say

1. git --version:-enable user to check the version of git that is installed on the system.

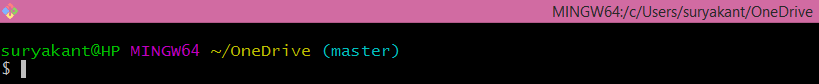


1. clear:-Clears the screen of git bash. Windows command to clear the CMD screen is cls

Before clear:

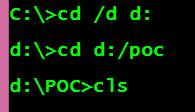


After Clear:-



In windows systems:

Before cls:



After cls:-



3)pwd=stands for present working directory.Gives the current working directory with path



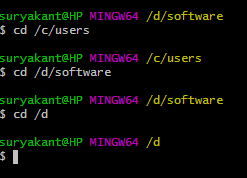
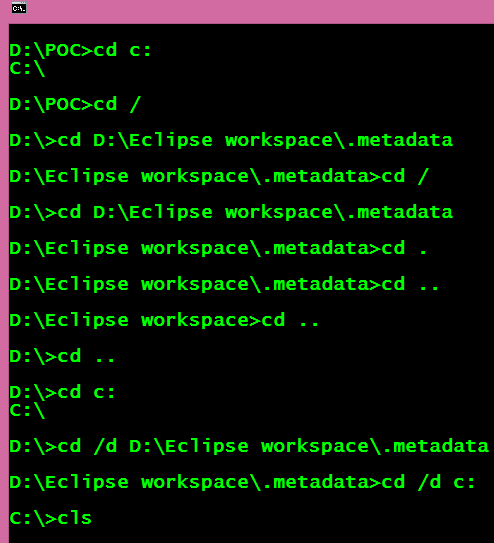
/c/users/suryakant This is the home directory for git.



4)cd= stands for change directory .used for changing the directory in which you want to work.

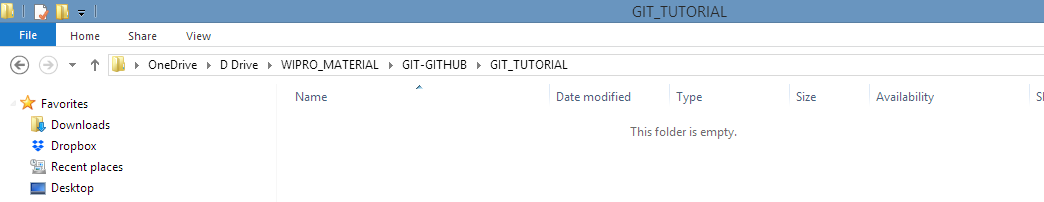
Syntax: cd /c/users,cd /e/installed\_files,cd /d/software,cd /b/backup,cd /a this is some directories where a,e,d,b,a are main directories n /users,/installed\_files,/software,/backup are subdireactories

Why /c,/d,/e,/a,/b becoz git supports linux style n not windows commands like cd /d c: or cd /d d:/poc

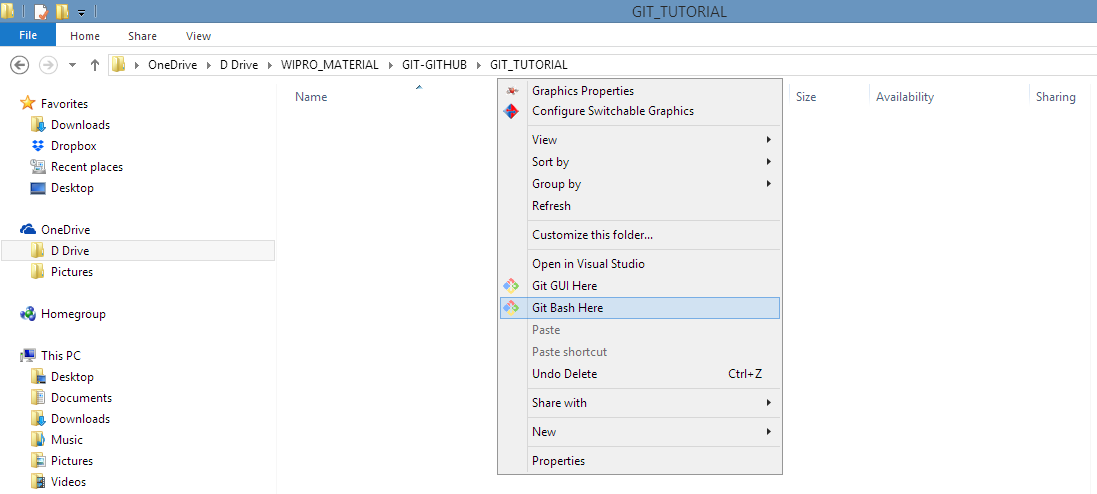
 

Either you can use cd command to move between directories or you can use one shortcut which is like:-

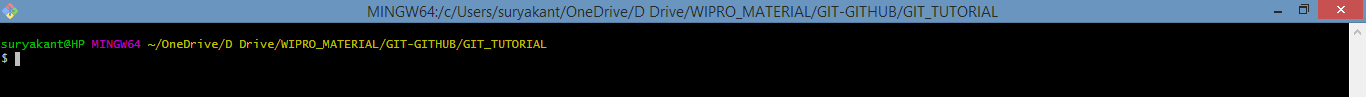
Go to that directory where you want to work with git in windows



After this right click on blank space

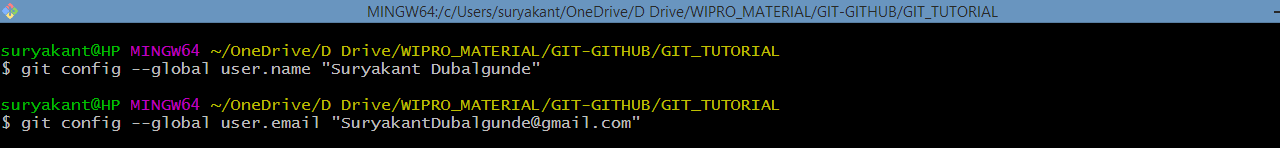


Will open an menu ->click git bash here or git gui here(better to become good coder open with git bash

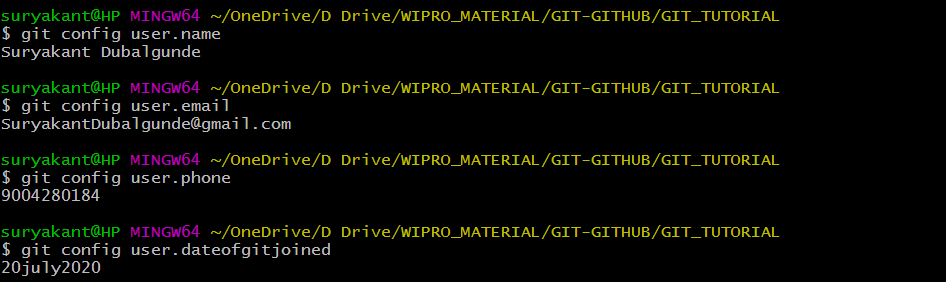


It will open git bash in that working directory where you want to work. After this you can fire the commands which you want on this directory.

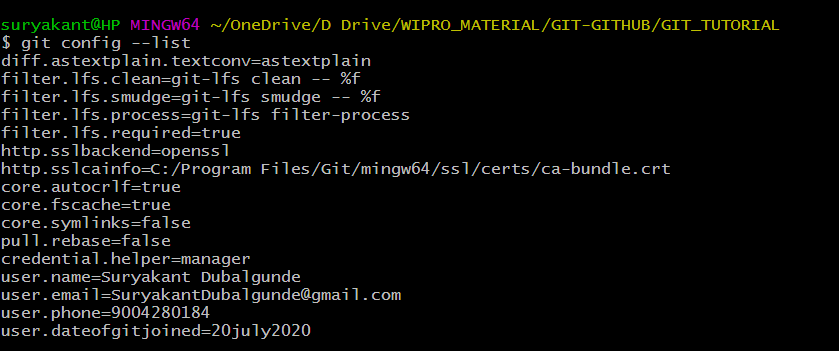
* How to configure your name and email id phone number in git ?so that anyone can know that this particular changes are committed by you and can contact with you via email or via phone



To view the username,email,phone number



To view all the git config configurations:-



5) mkdir foldername: creates folder

6)ls: list all the content of the pwd without hidden files .

ls -a : list all the content of the pwd with hidden files .

ls folder\_name :lists all content of the folder eg ls .git

7)vim filename.extension :opens an vim editor to edit the content of file.

8)cat filename.extension: shows conent of the file.

GIT THREE STAGE ARCHITECTURE OR How Git Works?

<https://youtu.be/Cdg2FGDMglk>

Git works in three stage Architecture which involves:-

1)Current Working Directory:-

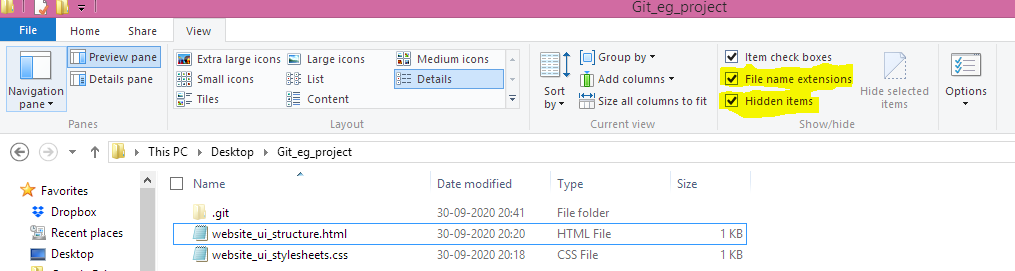
2)Staging Directory

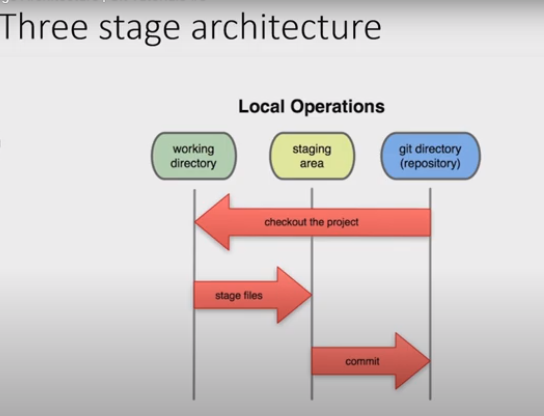
3)Git Repository or Git Directory:-it is a .git hidden folder created in a Current Working Directory which you want to store in git. In .git folder,we have all stored info about the version files n all.

.git folder is a hidden folder created in your current working directory which is hidden by default.

To unhide/see those .git folder in current working directory, perform following step:

* Go to current working directory path, click on view menu->in shw/hide panel,u will get file name extensions (to view the file names with extension like surya.txt) and hidden items checkbox,check them to allow to see .git folder + files with their extensions.





First of all, let us have a look at a two stage architecture Version Control System.

In a two stage Version Control System, we work on our working directory, commit some changes and then push to the repository. Then again, we can pull the changes from the repository (changes which might have done by our colleagues), commit some changes, and again push back the files to the repository.

Till now, it seems very good, but now imagine a situation, when you are working on a project consisting of say 4 files. Now, you have finished working on 2 files, but in other 2 files, you have encountered some bug. In such situation, if you commit changes to the repository, then the other two files containing bug will also be committed to the repository. This is a serious problem!

But git solves this problem by introducing a middle stage called “staging area” in this architecture. Let us have a look at three stage architecture now.

Thus, git allows us to first push our files to staging area, and then we can commit those staged files. Continuing our above example, we can only push those two bug-free complete files to staging area and then further commit our changes. The other two files in progress can remain in our working directory. They can be modified later and committed when you need to. Another important advantage of staging area is that files from staging area can again be pulled to our working directory, if we feel they need some more changes before committing them.

**It is to be noted that it is not compulsory in git to first push the files to staging area. The changes can even be directly committed, but it is always a good practice to first stage our files.**

Tracking first GIT Project:-

Video URL:-https://youtu.be/-e0RtzuNgS4

Let say,we have one project which we need to store with git technology for version control .

Path:-C:\Users\suryakant\Desktop\Git\_eg\_project

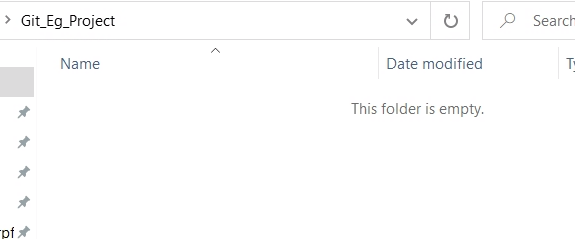
Assumption:-assumption we have made is like,there is no already existing .git repository created for Users folder,suryakant folder,Desktop folder

Steps we will follow is like:-

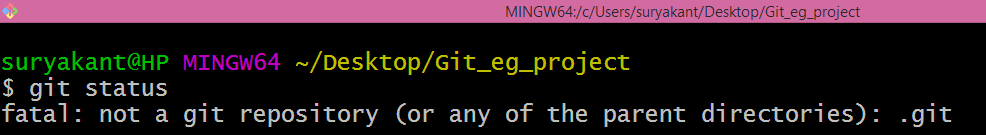
* Open windows path C:\Users\suryakant\Desktop\Git\_eg\_project
* When you will go into the git\_eg\_project right click->open in git bash(to open git in that directory)

Use following commands to track git operations:-

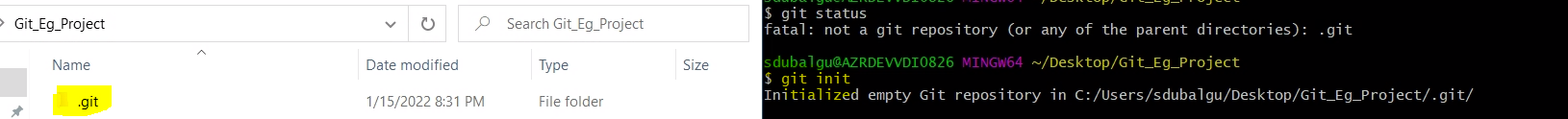
1. **Git status:**used to check the status of git directory/repository(like whether something is changed n untracked or somethings needs to be committed) created into the path.before starting the work in git you have to use git status command to check the status of git repositories.

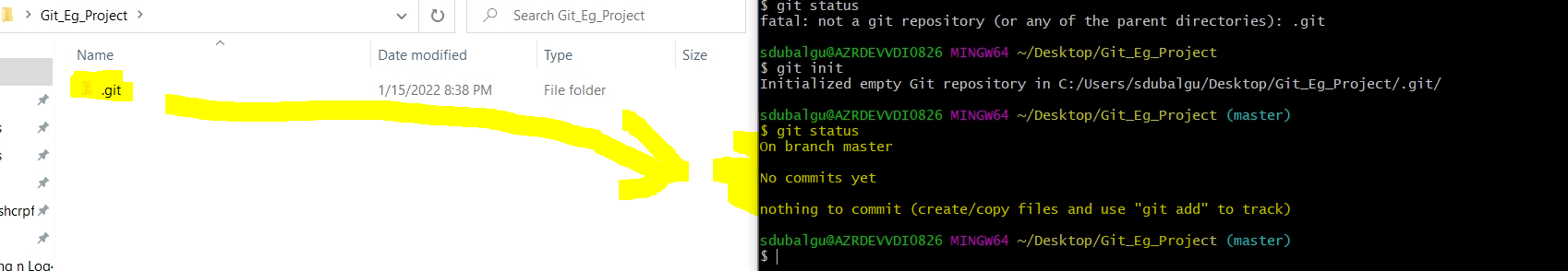


If no git repository created for that path or current working directory,then u will get something like this



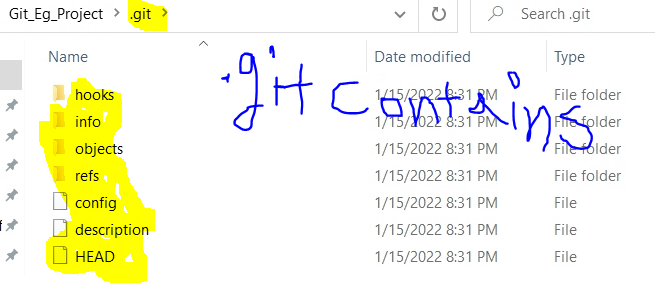
2. Git init:-if you dont have an .git folder(Which is hidden) in your current working directory or any parent directory or you will get above error then u use this command to initialize the git repository.

Once .git folder is created ,git will start tracking our files 

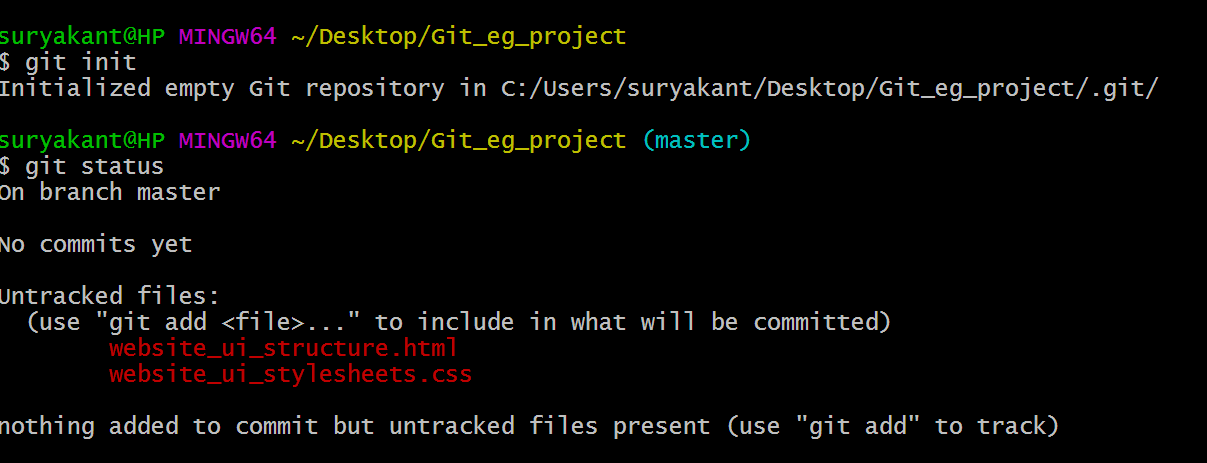
If we press git status then 

It will start tracking the files in current working directory.

.git file contains below files n folders:



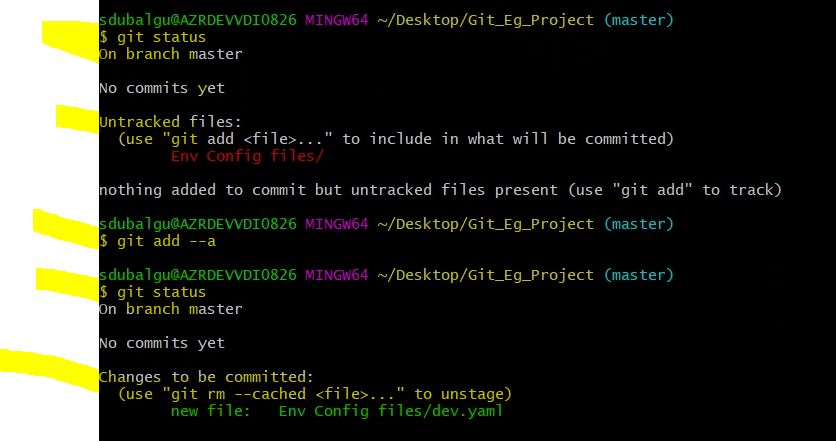
If some files are present in the git\_eg\_project then it will show untracked files in red which are not committed in git like below img:



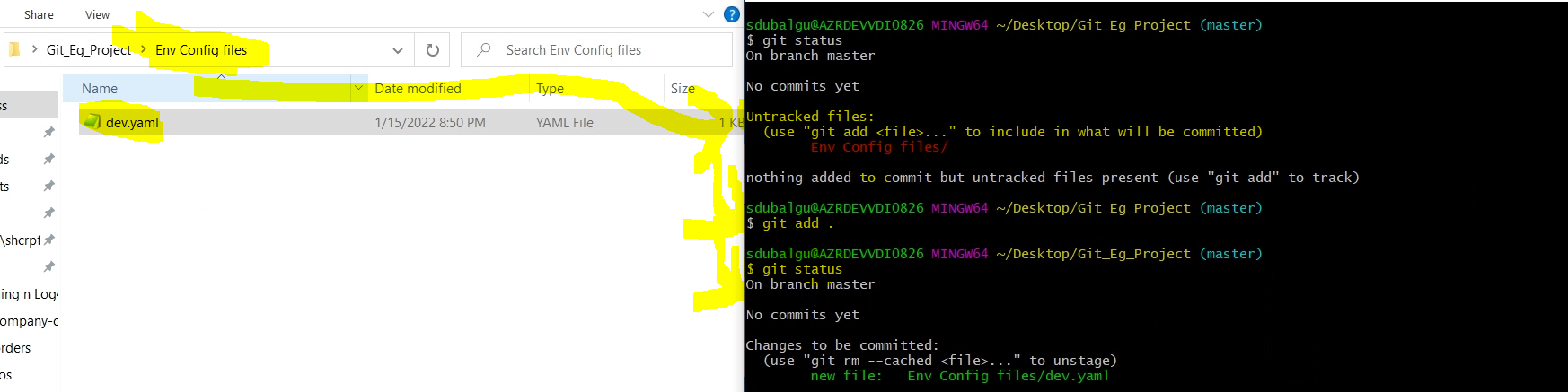
When you add some files or made any changes in the existing .git enabled folder,then to add this changes to staging area or directly commit or save into .git repo

3.git add --a or git add filename.extension or git add folder\_path or git add . :- When you add some files or made any changes in the existing .git enabled folder like add folder or update folder name or file name,then to add this changes to staging area or directly commit or save into .git repo git add command is used.it has various forms like below which can be used as per the requirement.git add this command is used to stage all or specific specified changes or untracked files in staging area.

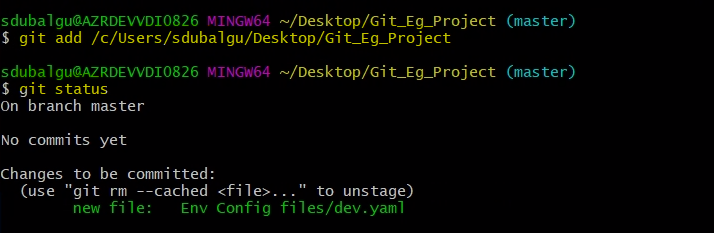
* git add --a :command is used to stage all changes or all untracked files in staging area.



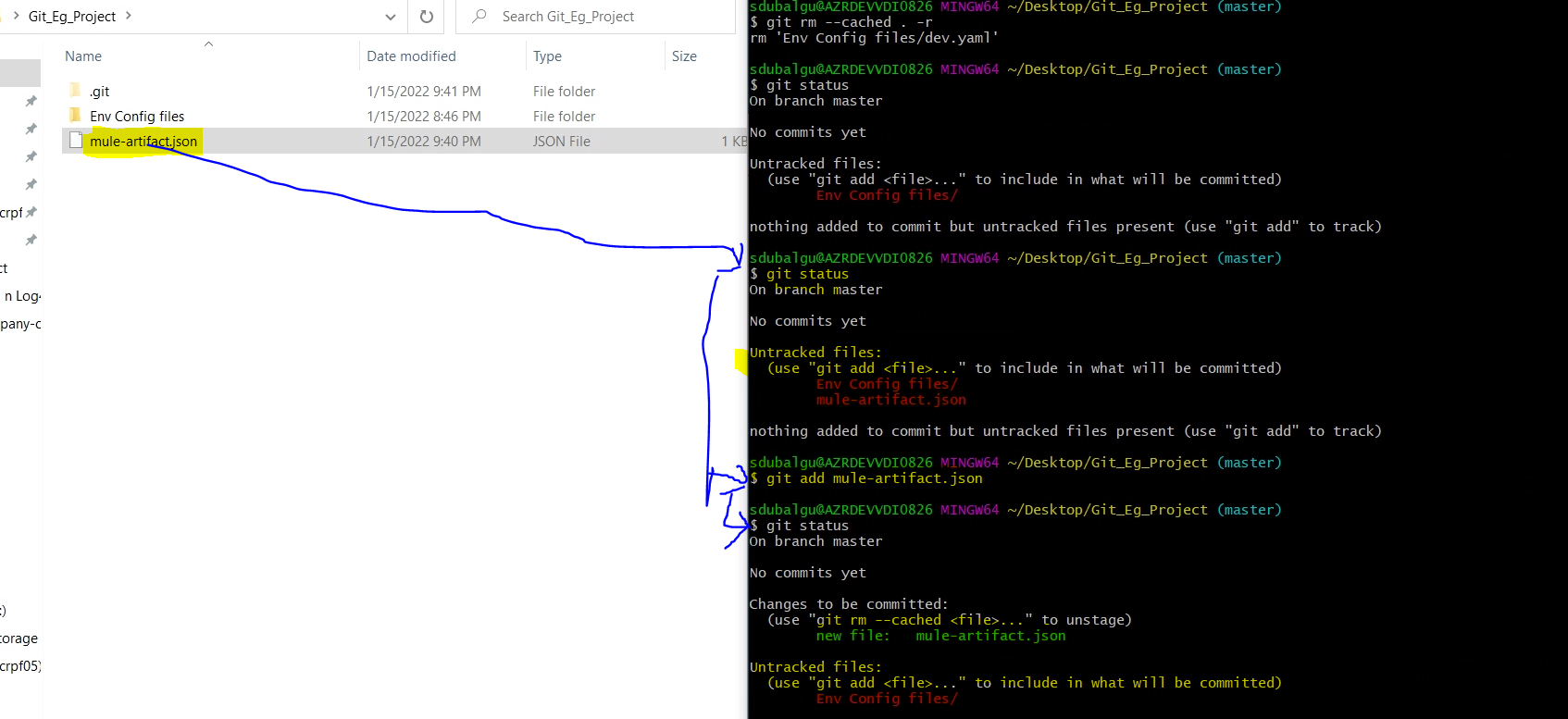
* git add . :this command is used to stage all changes or all untracked files in staging area.



* git add folder\_path :Let say we added an folder in .git enabled folder n in that we have created file called dev.yaml so if we use git add folder\_path then it will add that folder with all the files inside the folder in staging area



* git add filename.extension : Let say we added an file in .git enabled folder n so if we use git add filename.extension then it will add that file inside the staging area

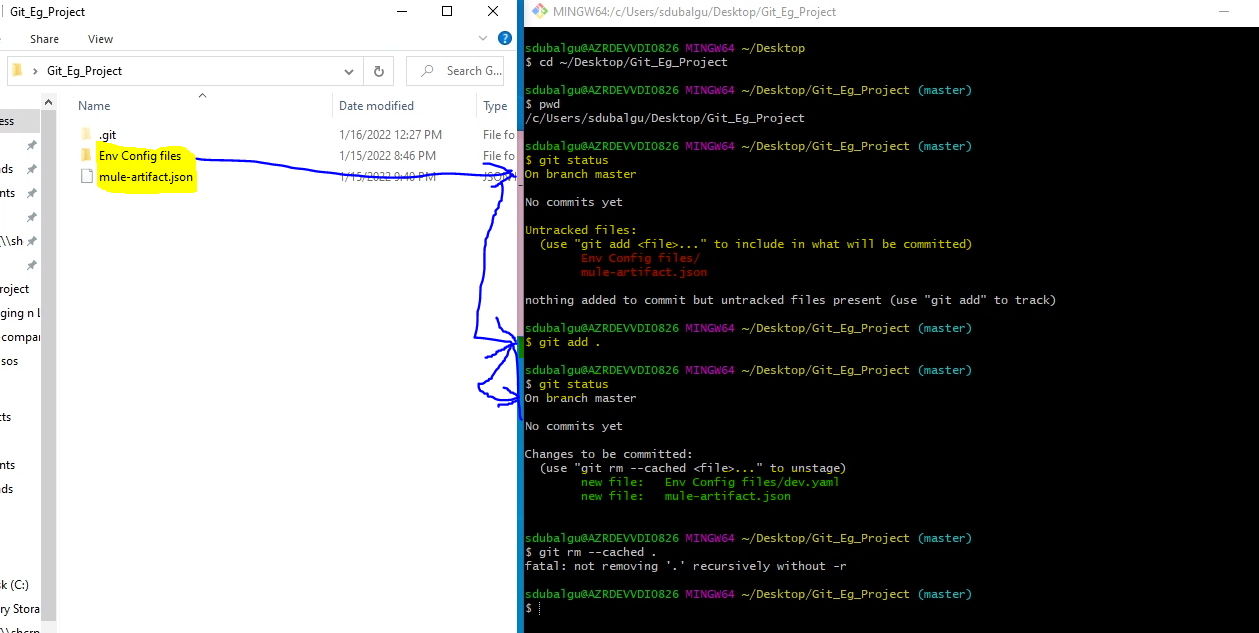


In the above screenshot,Here we have two changes which needs to be send in staging area 1) Env Config files/ -----this is a folder in which we have dev.yaml file created

2)mule-artifact.json ---------this is a json file

Since we used git add mule-artifact.json so what git system did is ,it moved mule-artifact.json file in staging area n kept Env config files/ folder in untracked files section

Basic Example of git add with 2 set of changes 1) is mule-artifact.json file 2)Env Config files/ folder



4.git restore --staged . or git restore –staged file.ext: unstages the file from staging area to modified files

5.rm -rf file.ext:to delete the file from the git repo

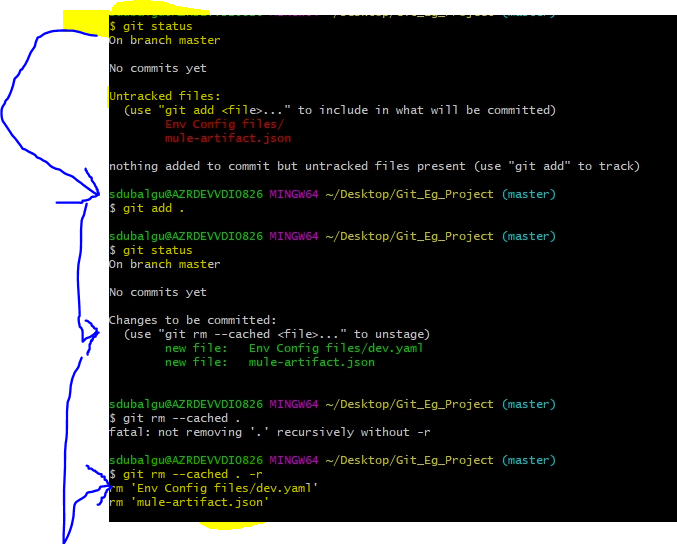
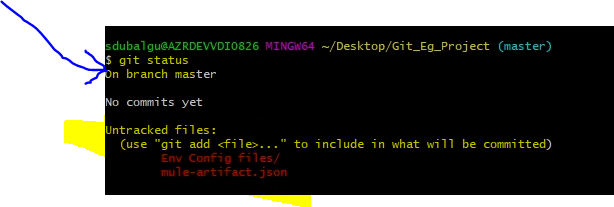
6. git remote add origin url\_of\_git\_repo: connects ur local sys repo to git\_repo here origin is alias name for the url of git

7. git push origin branch\_name: it pushes the changes to the git repo called origin(url\_of\_git\_repo)

4. git rm --cached -r . or git rm --cached -r filename.extension or git rm --cached -r folder\_path : this command is used to remove changes from staging area to untracked files.

-r is used to mention recursive mode i.e. backtrack the folders .let say we have folder in which 6 files are there so to remove all 6 files with folder then we have to use -r command

* git rm --cached -r . : this command is used to remove all changes from staging area to untracked files.

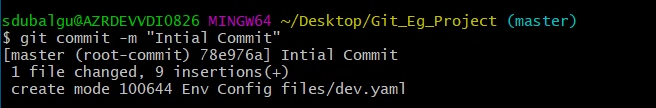
* git rm --cached -r filename.extension : this command is used to remove specified file changes from staging area to untracked files.
* git rm --cached -r folder\_path : this command is used to remove specified folder changes from staging area to untracked files.

5. git commit or git commit -m “comment to be written” : this command is used to save or commit the changes from staging area to .git repository .

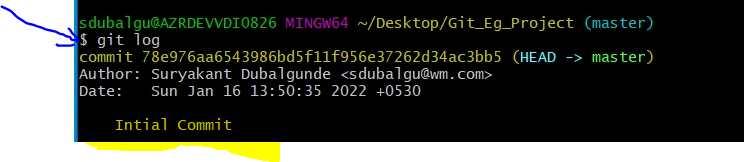
* git commit : this command will be used to save or commit the changes from staging area to .git repository. This command will open an vim editor to enter the comment for saving the changes.

:wq is command used to get out of the vim editor window.

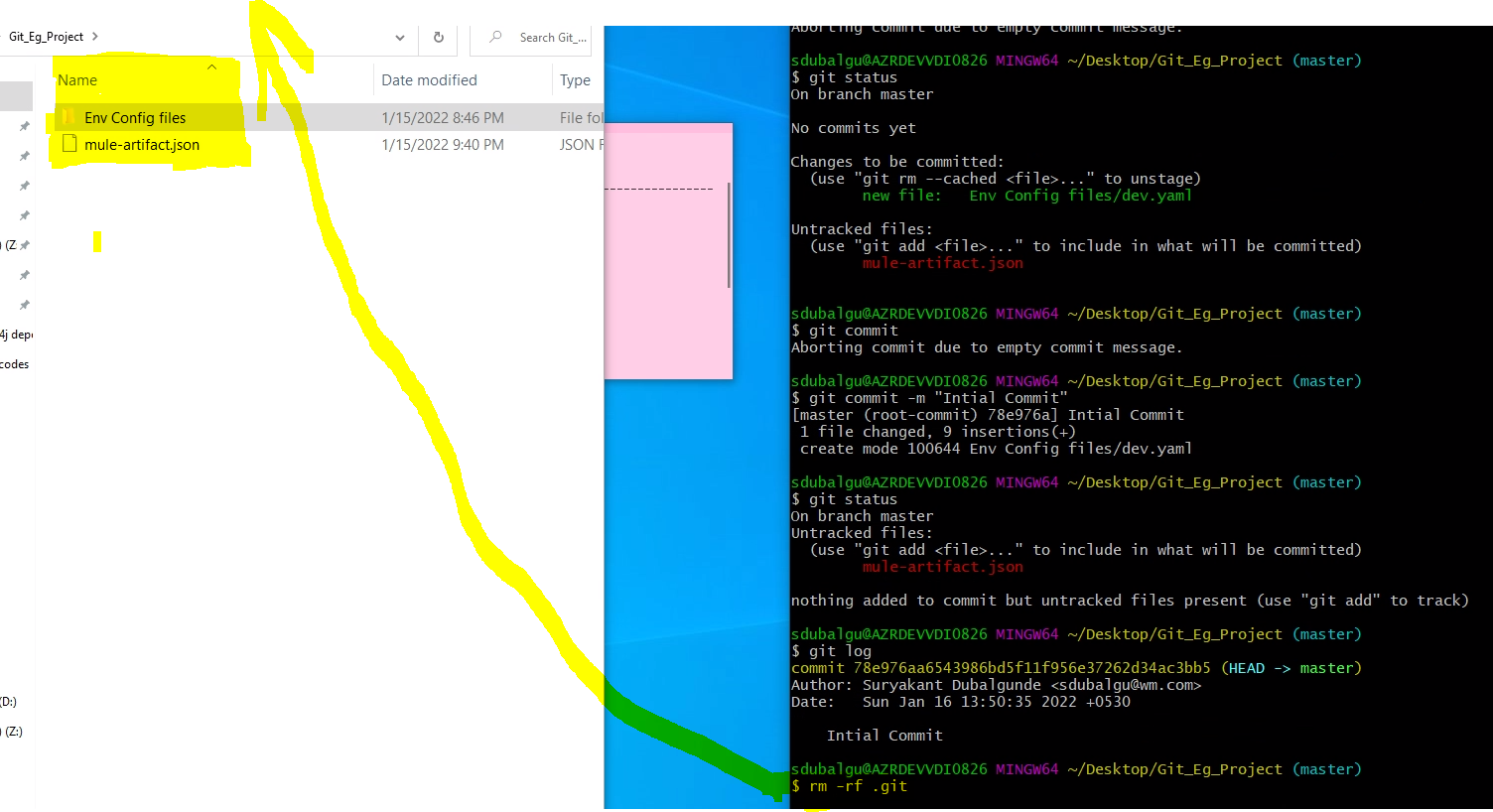
* git commit -m “comment to be written” : this command will be used to save or commit the changes from staging area to .git repository.



6. git log: this command is used to get the list of all commits that have been done

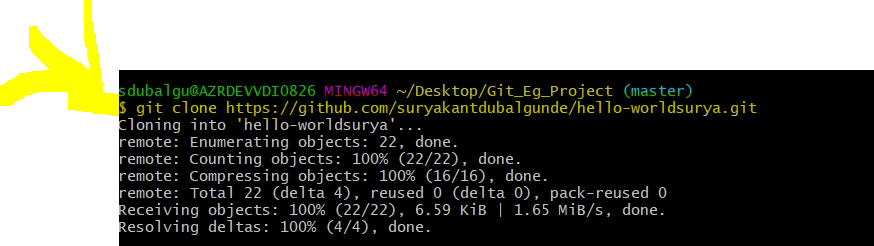


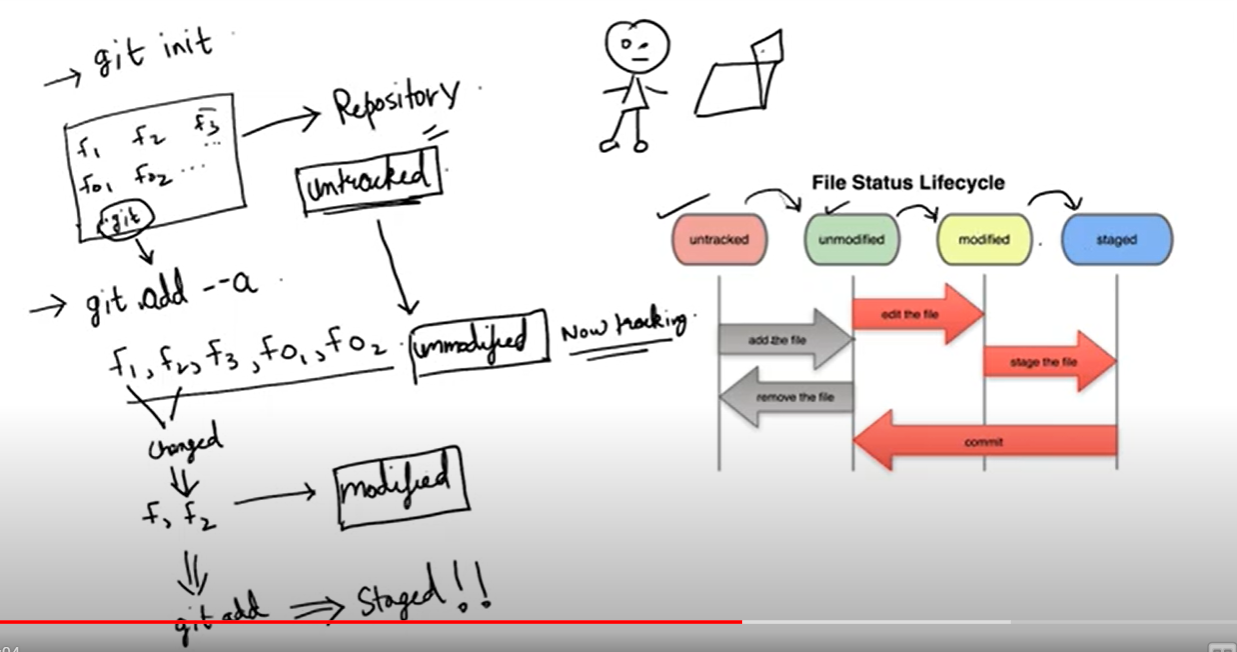
7. rm -rf .git or rm -rf filename.extension or rm -rf folder\_path: this command is used to remove the specified item like .git repo or filename.extension or folder 



Git init command can be used to create the .git file n start the track changes again.

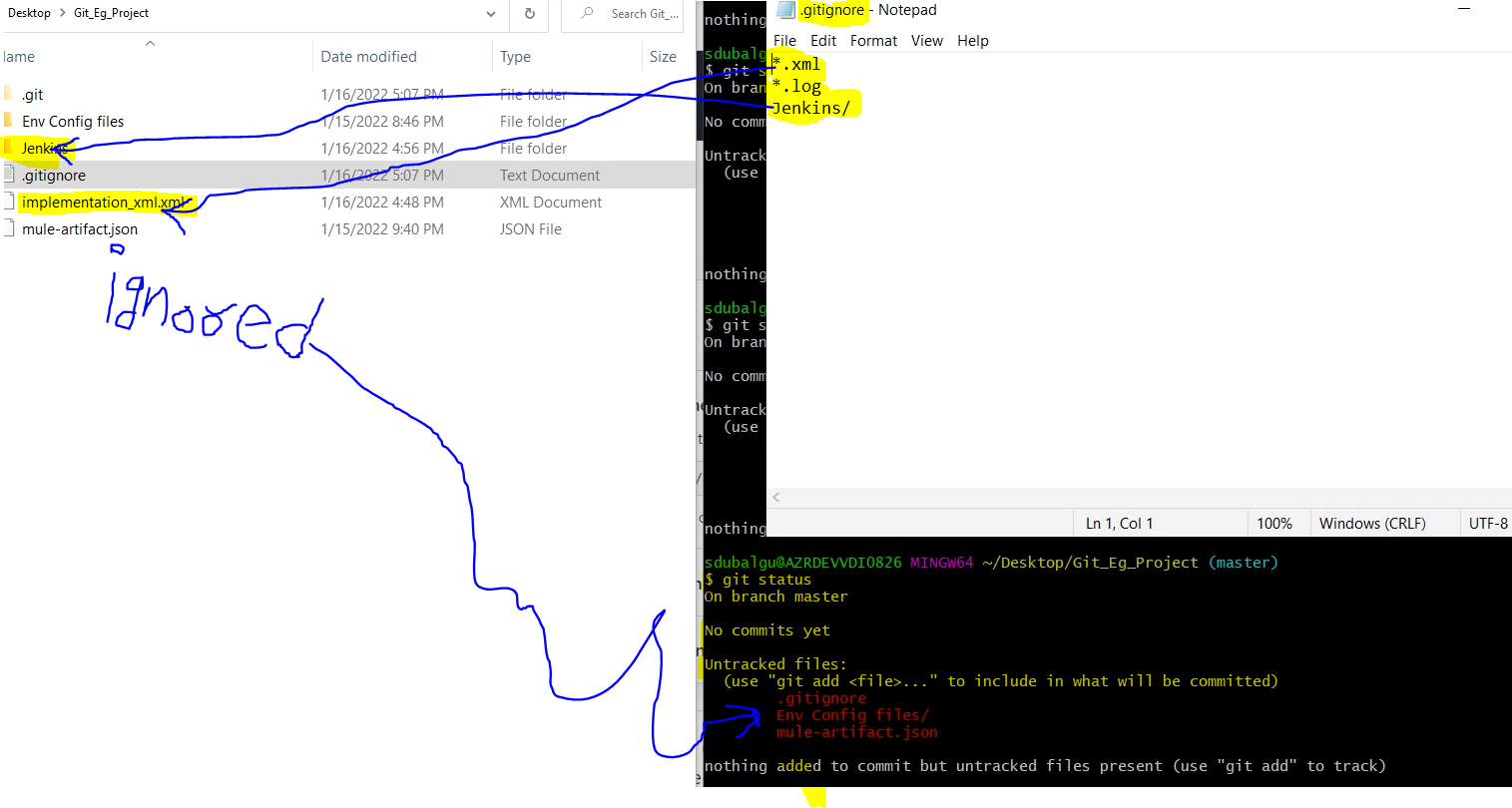
8. git clone github\_url :this command is used to copy the code from git repositories.

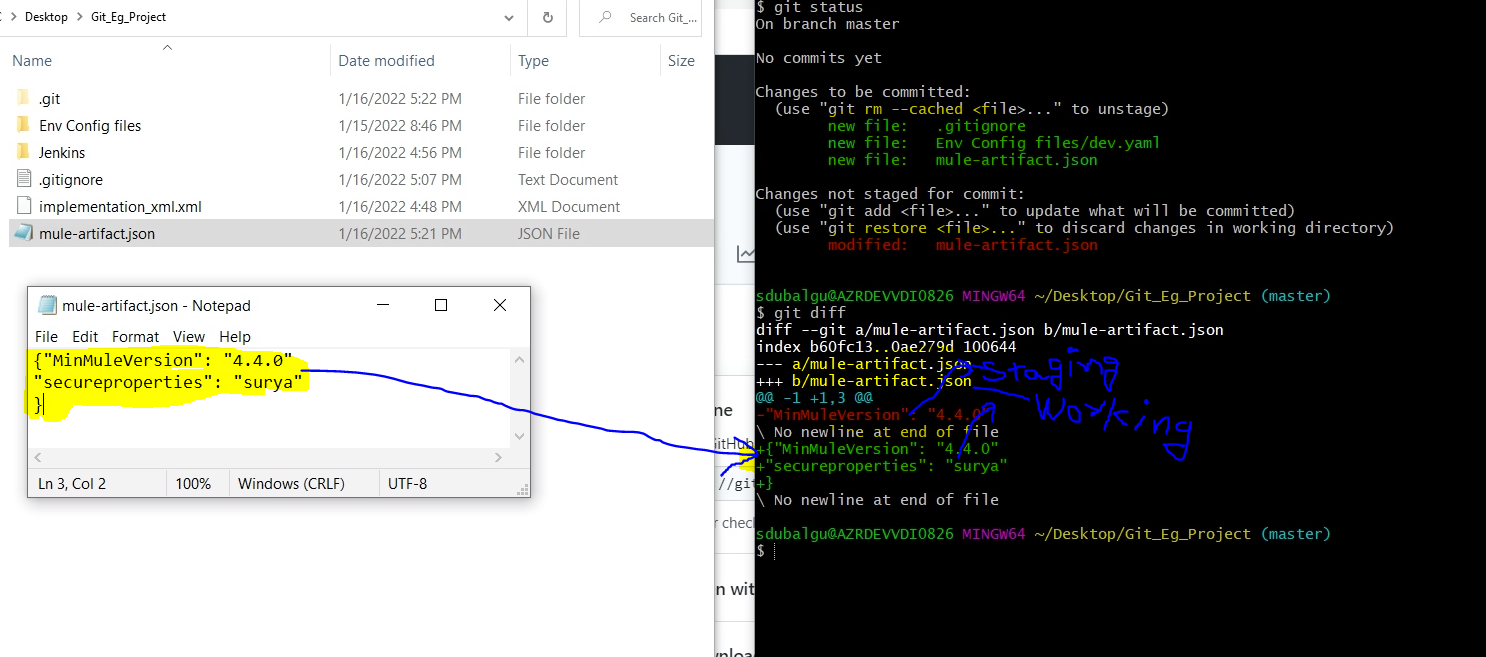




9.touch filename.extension\_tobecreated or touch .gitignore :touch command is used to create a blank file in .git enabled folder.

.gitignore is a file which is used to specify which files needs to be untracked from git repositories.

 10.git diff : compares the working directory change n staging area change



11. git commit -a -m “comments to be added” :