11/18/2022

Nagaraju Lethadala

TATA Consultancy SerVICE

**GIT SUBMODULE**

**SUBMODULE**

It often happens that while working on one project, you need to use another project from within it. Perhaps it’s a library that a third party developed or that you’re developing separately and using in multiple parent projects. A common issue arises in these scenarios: you want to be able to treat the two projects as separate yet still be able to use one from within the other.

Git addresses this issue using submodules. Submodules allow you to keep a Git repository as a subdirectory of another Git repository. This lets you clone another repository into your project and keep your commits separate.

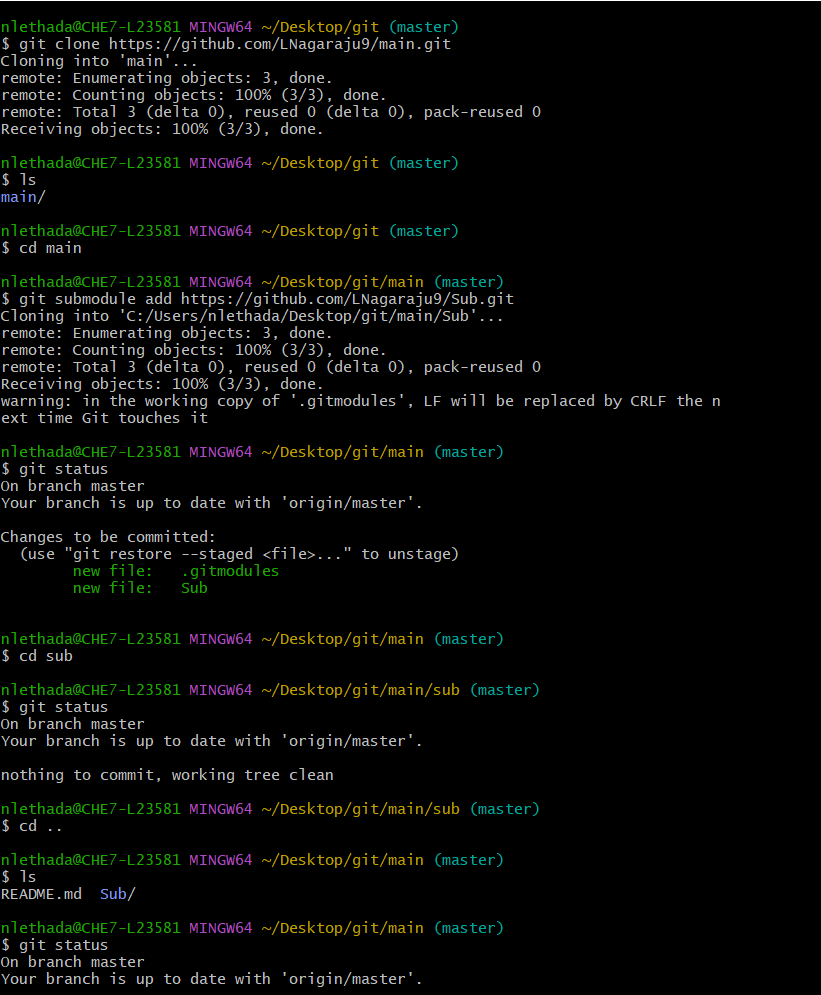
**1.Starting with Submodules**

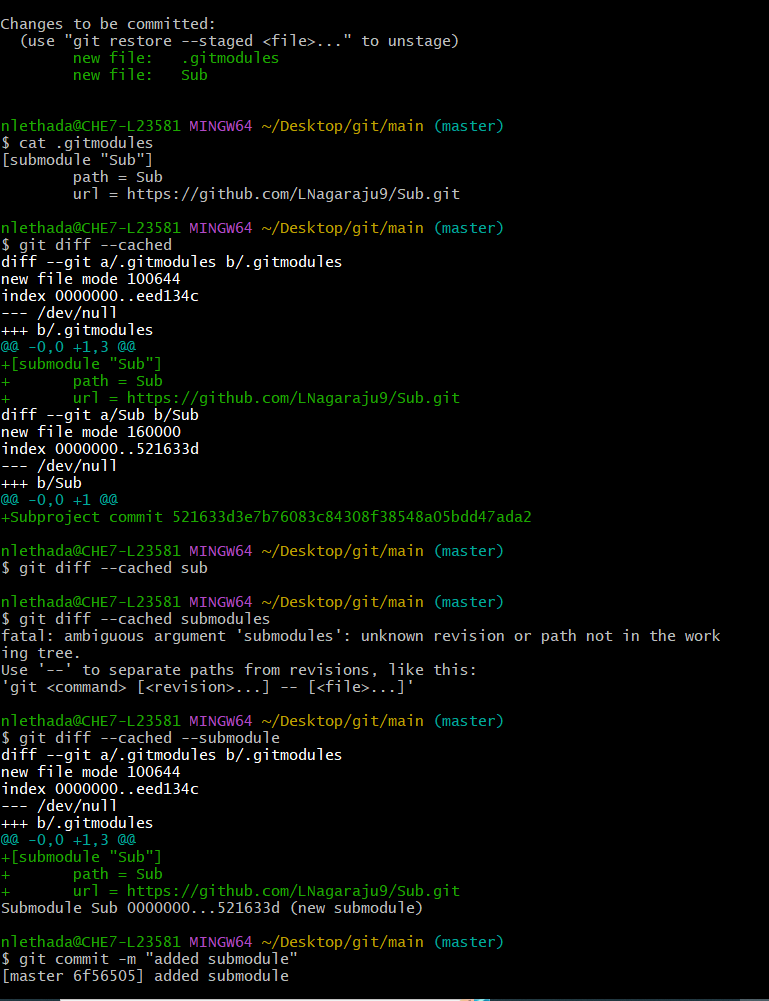
**1.1 Scenario 1:**

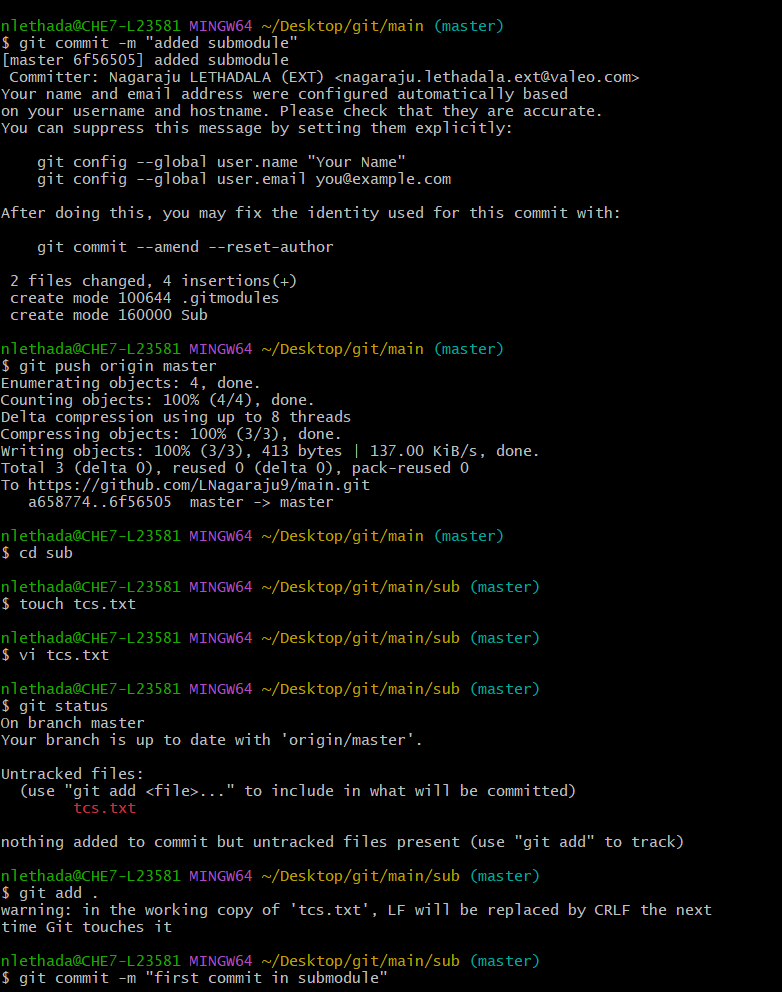
Let’s start by adding an existing Git repository as a submodule of the repository that we’re working on.

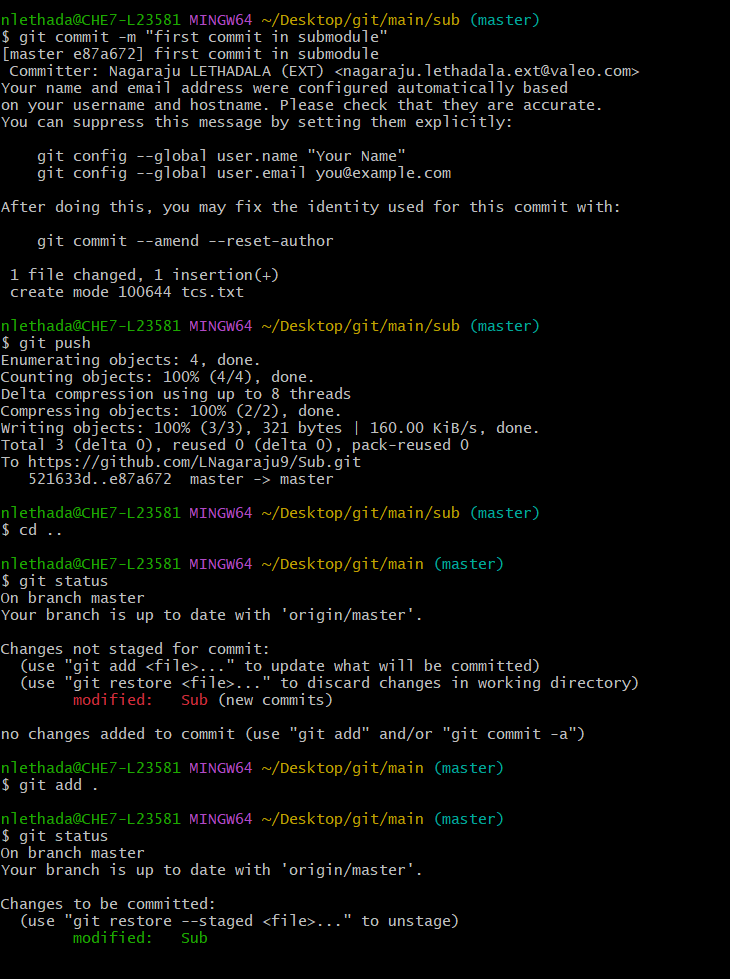
* Go to the directory of the repository on which you want to add the submodule
* First clone main project into local repository uses this below command
* **Git clone <main repo url>**
* Ex: git clone <https://github>.com/Lnagaraju9/main.git
* Use the command
* **Git submodule add <submodule url>**
* Ex: **git submodule add** [**https://github.com/Lnagaraju9/Sub.git**](https://github.com/LNagaraju9/Sub.git)
* Submodule folder name is **Sub**
* After adding the submodule two file will add in local repository those are **“.gitmodule**”, Submodule folder name is **Sub**
* Then you check the patch of the submodule by using below command
* **Cat <Submodule folder name>**
* Now you can also check the difference between main repository and sub repository by using below two commands
  + **Git diff –cached**
  + **Git diff –cached –submodule**
* Next you can commit this file staging area to local repository by using this command **git commit -m “message”**
* Now you are to push this commit local repository to remote repository by using this command **git push -u origin master**
* Now you see submodule is added in remote
* After certain changes or creation of new file in submodule then you add to staging area and commit then you push these changes to remote now these changes reflected on remote repository
* Now you want add these changes into main repository so now you move to the main directory in working area then you add these changes to staging area as well as local repository by using above mentioned command then you push changes into remote now that code or file will also available in main repository.

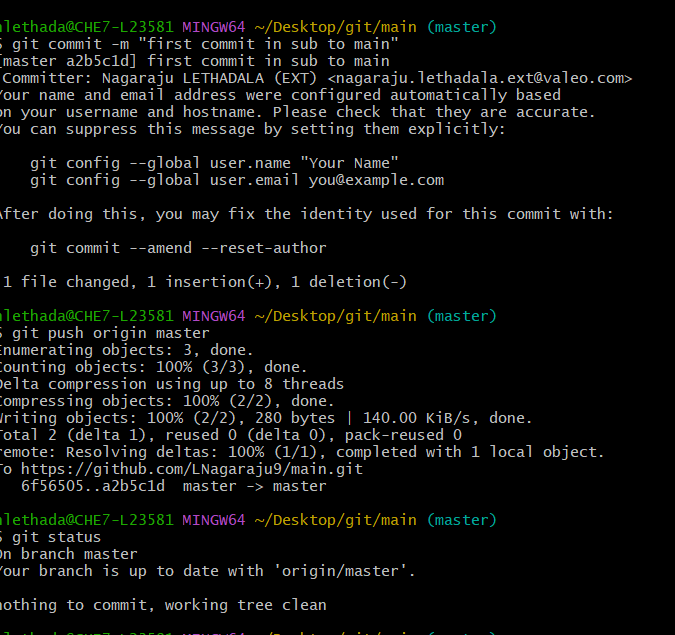
Example:





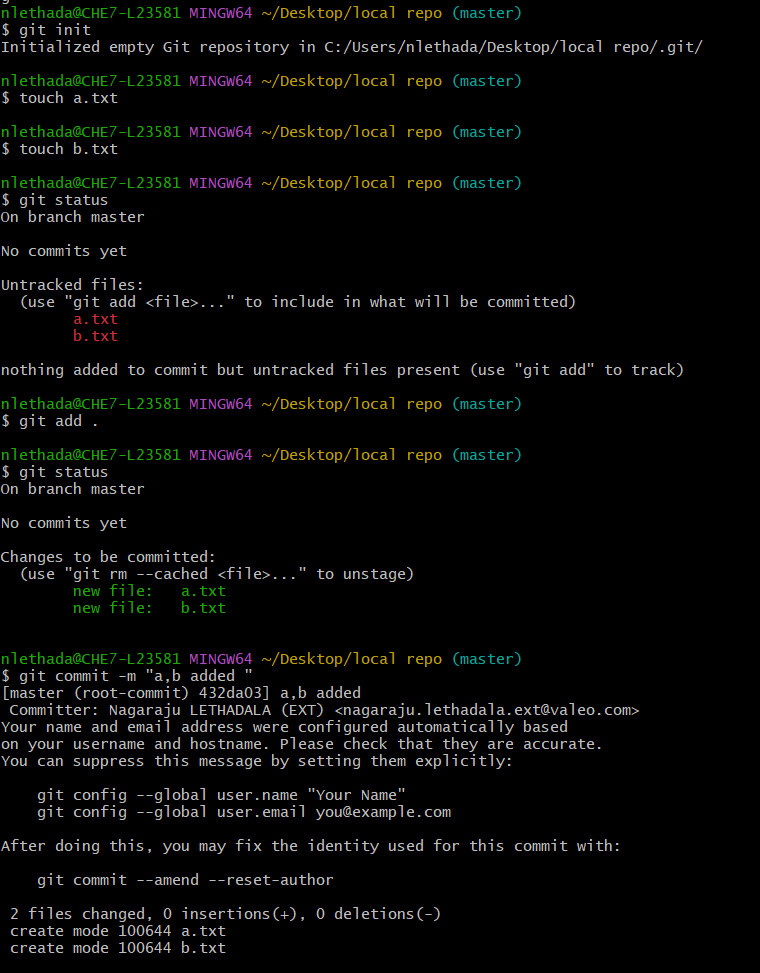


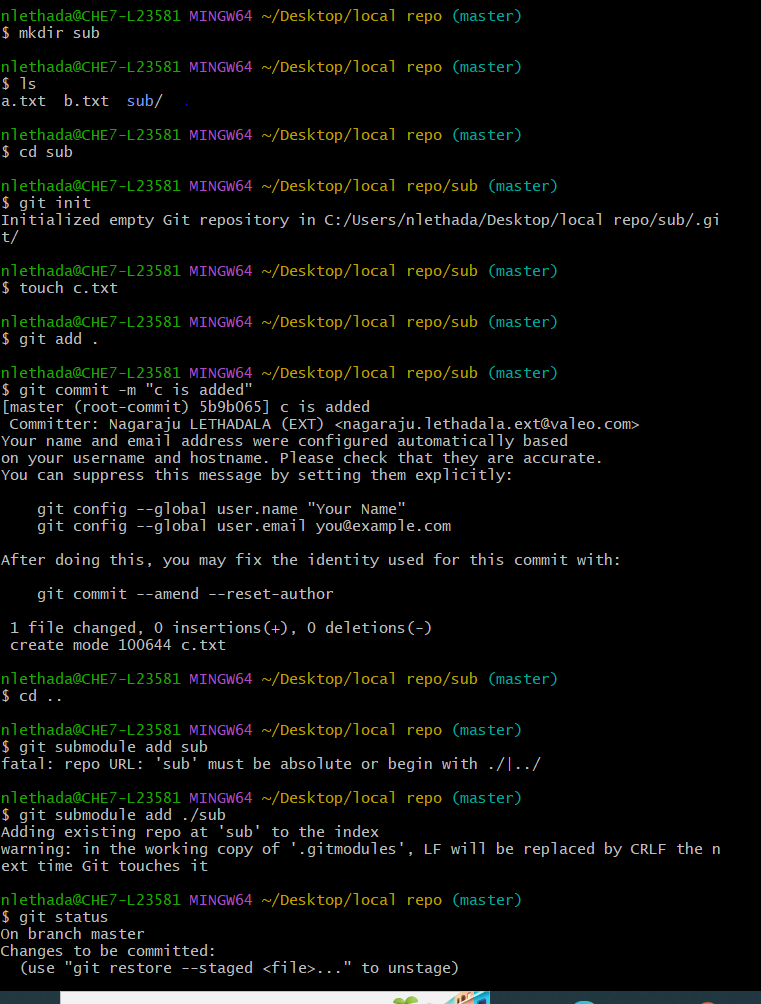
****

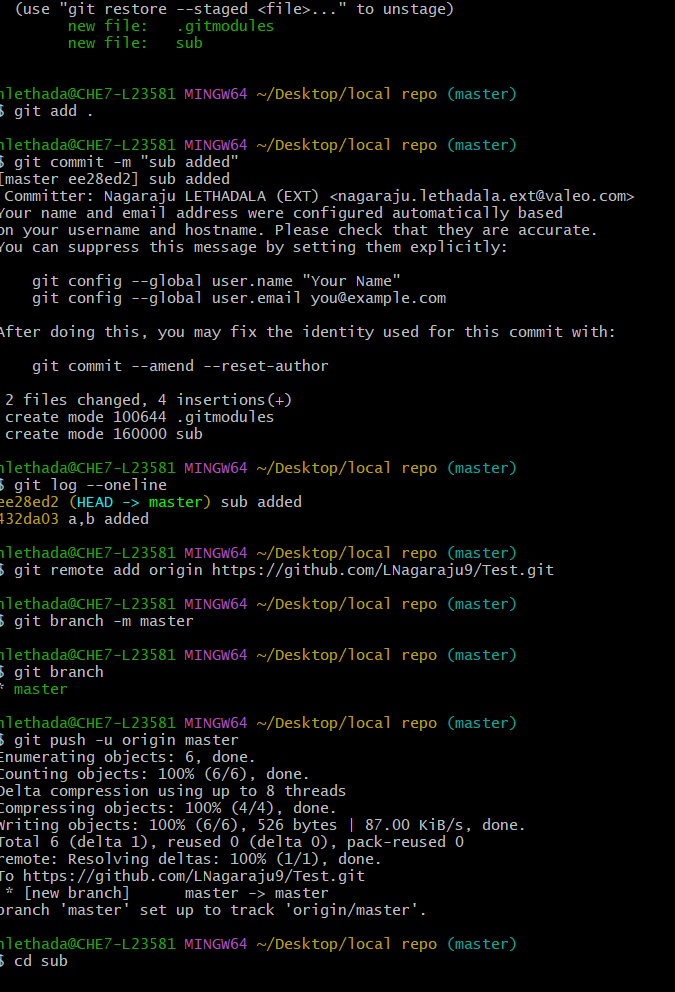
****

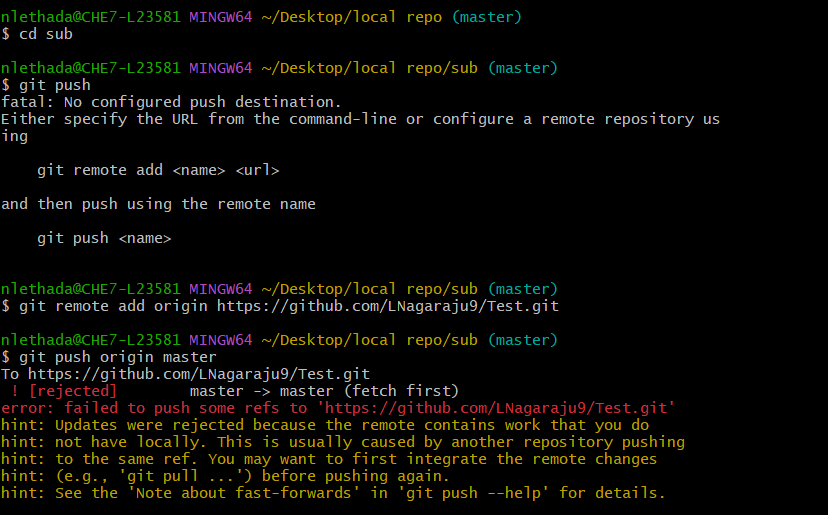
**1.2 Scenario 2:**

* Go to where you are created a folder structure in you are system open git bash
* First you create a main folder by using **mkdir <folder name>**
* Then you go to inside the main folder by using **cd <folder name>**
* Create a new two text files inside the folder by using **touch <file name>**
* Move this file working directory to staging area by using **git add <file name>**
* And commit these two files into staging area to local repository **git commit -m “meassge”**
* And you create a new subfolder by using **mkdir <subfolder name>**
* Go inside the subfolder then create a new text file by using **touch <file name>**
* Move this file working directory to staging area by using **git add <file name>**
* And commit this file into staging area to local repository **git commit -m “meassge”**
* Now go to main folder **cd ..**
* Now you add the main folder to subfolder **git submodule add ./<subfolder name>**
* Then adding this folder structure to remote repository by using **git remote add origin <url>**
  + Ex: git remote add origin <https://github.com/LNagaraju9/Eclipse.git>
* Now push to local repository to remote repository by using **git push -u origin master**
* Now you see above created folder structure but you notice one thing subfolder not working because you are not added submodule url inside the subfolder so you try to push submodule changes into remote it gives error.
* Now you add the submodule ur inside subfolder by using this command **git submodule add <ur>**
* Now you change some files in subfolder then push to remote it works fine and also this file or code will useful for main folder as well as remote main repository by using related commands in git.



****

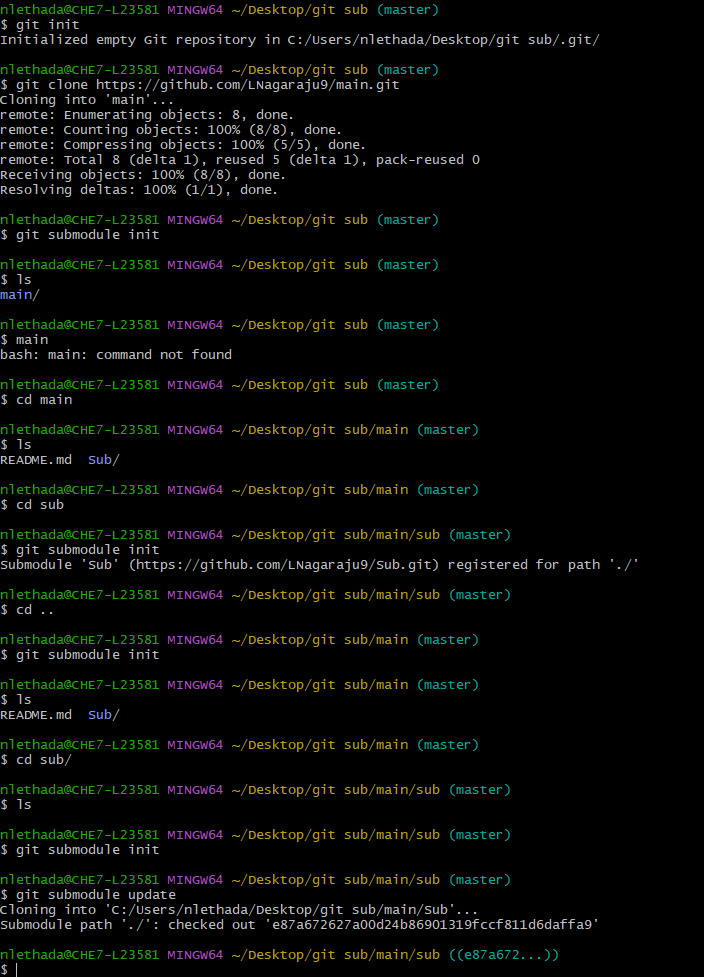
****



**2.Cloning a Project with Submodules:**

**2.1 Scenario 1:**

* Now you can directly clone whole main repository and sub repository
* Go to where you create this folder structure open git bash
* Start the cloning process by using git command **git clone <url>**
  + EX:**git clone** [**https://github.com/LNagaraju9/main.git**](https://github.com/LNagaraju9/main.git)
* Now you see the folder structure in local repository
* But in the subfolder files not files not display
* So now you can fallow below steps
* Go to subfolder in that use the **cd <subfolder name>**
* give the command as **git submodule init**
* And **git submodule update**
* Now you can check the submodule files in local repository



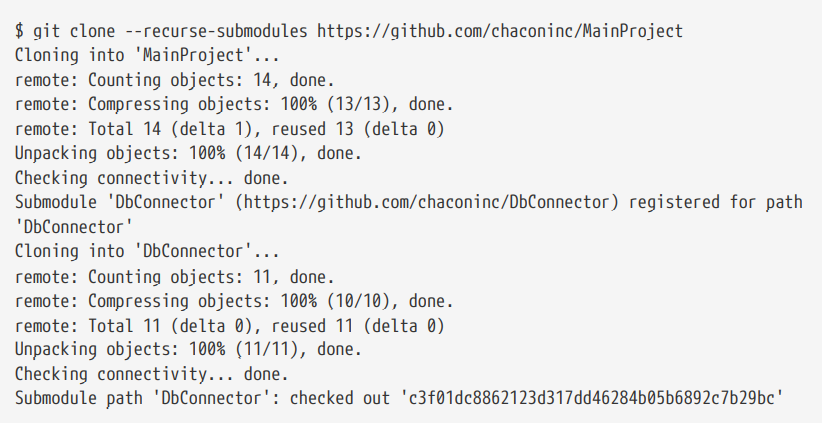
**2.2 Scenario 2:**

* Now you can directly clone whole main repository and sub repository
* Go to where you create this folder structure open git bash
* Start the cloning process by using git command

**git clone –recurse-submodules <url>**

EX:**git clone –recurse-submodules** [**https://github.com/LNagaraju9/main.git**](https://github.com/LNagaraju9/main.git)

* Now you see the folder structure in local repository
* Now you can check the submodule files in local repository

****

**Working on a Project with Submodules:**

Now we have a copy of a project with submodules in it and will collaborate with our teammates on both the main project and the submodule project

**Pulling in Upstream Changes from the Submodule Remote**

Now you can work on submodule

Any of the people or team or you belong to that submodule now we can easily that data into our local repository by using below commands

First you change in submodule in remote repository

Then you fetch details into local repository by using command **git fetch**

Use another command for getting the files in local repository **git merge origin/master**

Now you check files will visible in local repository

