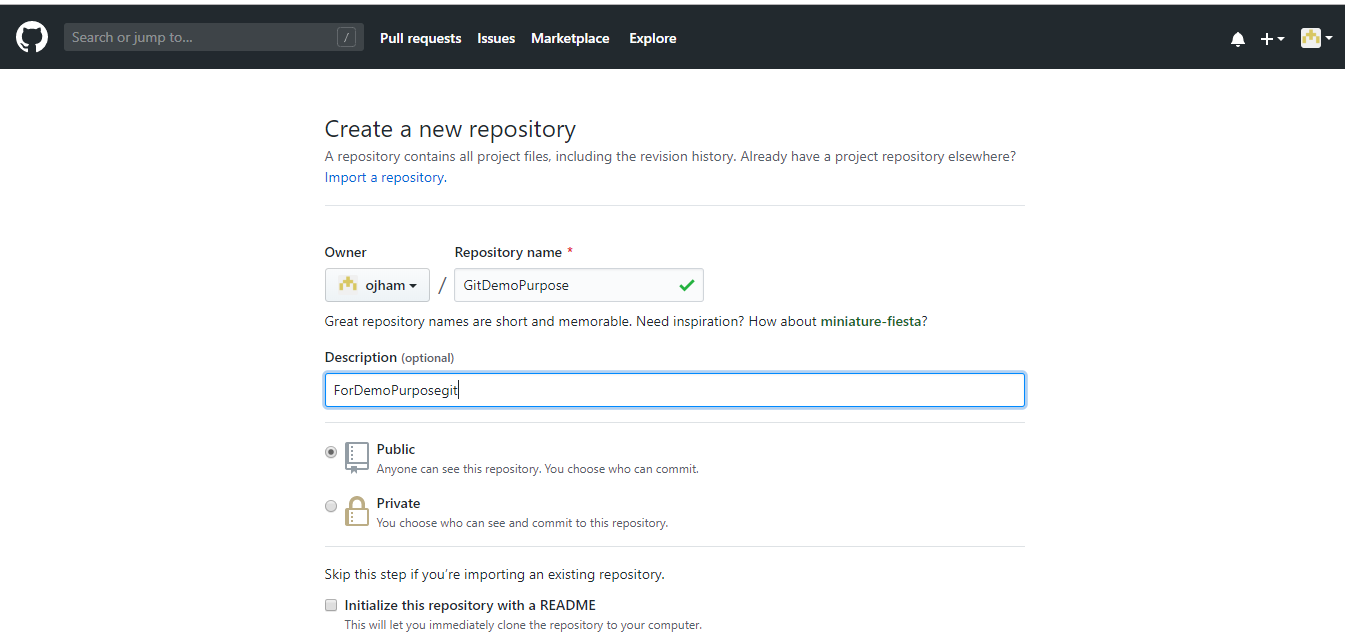
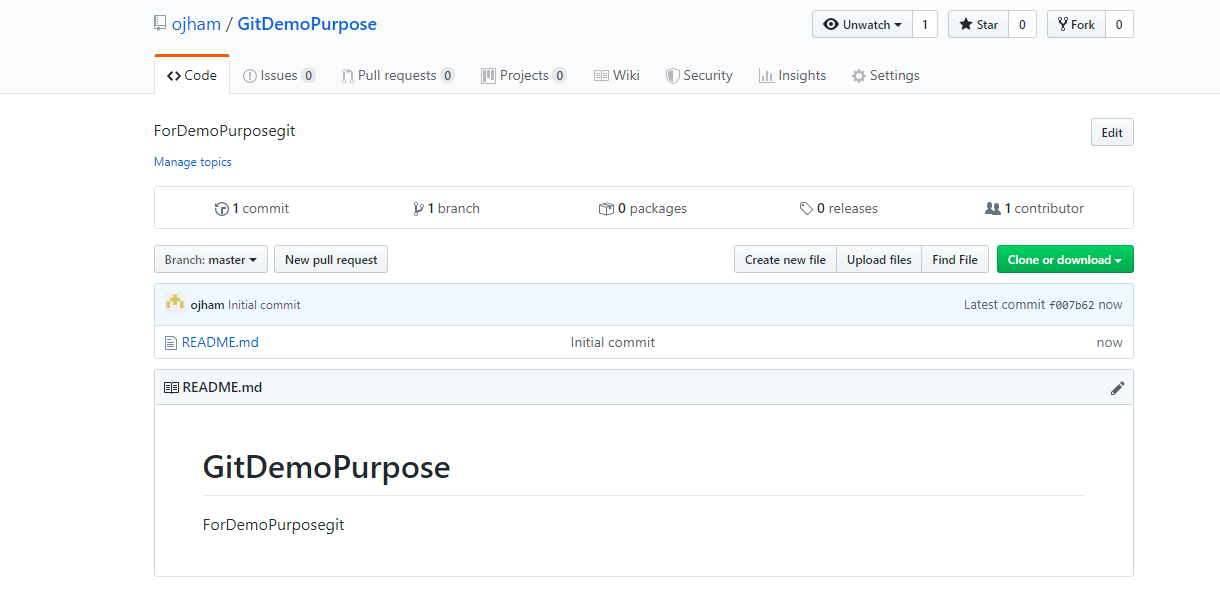
Gits commands with example:

* One Repository for one project should be created in the github.

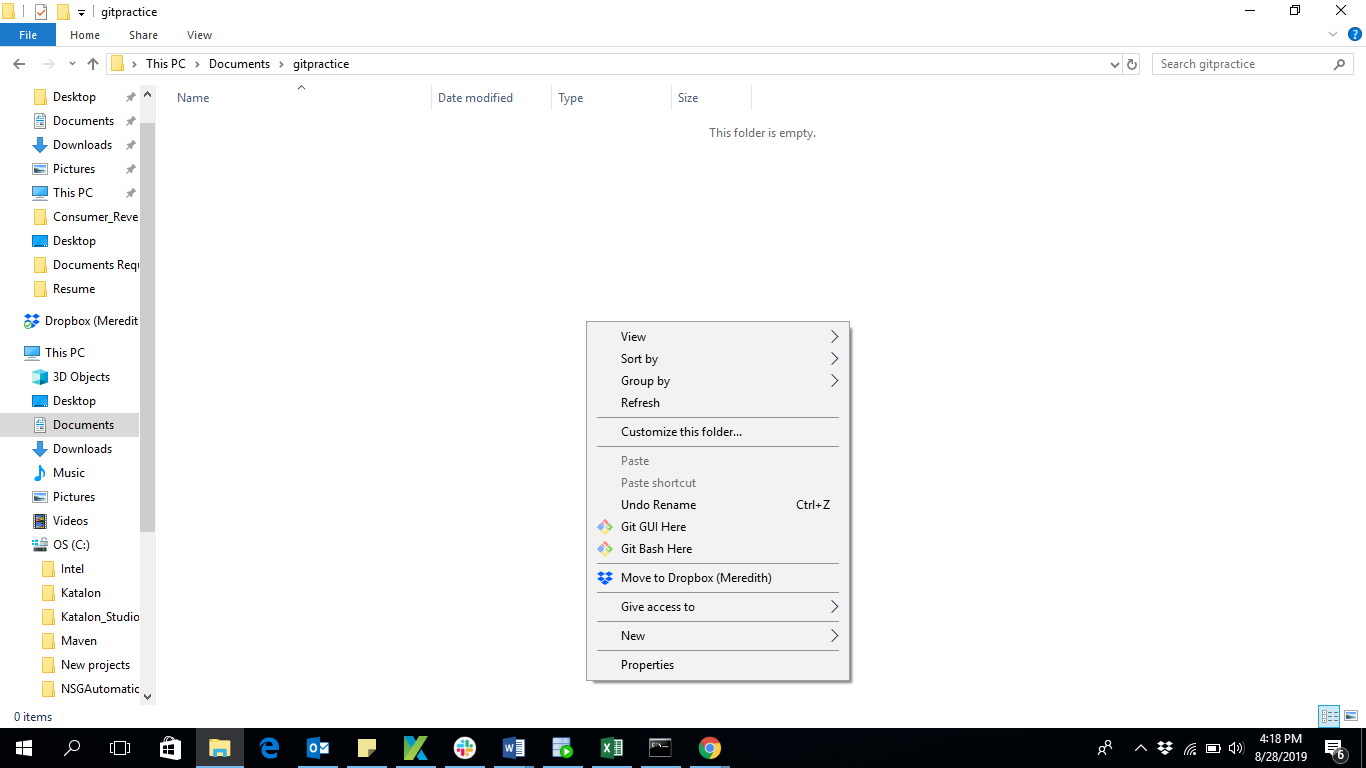
1. **Creating the new Central Repository in the git-hub.**

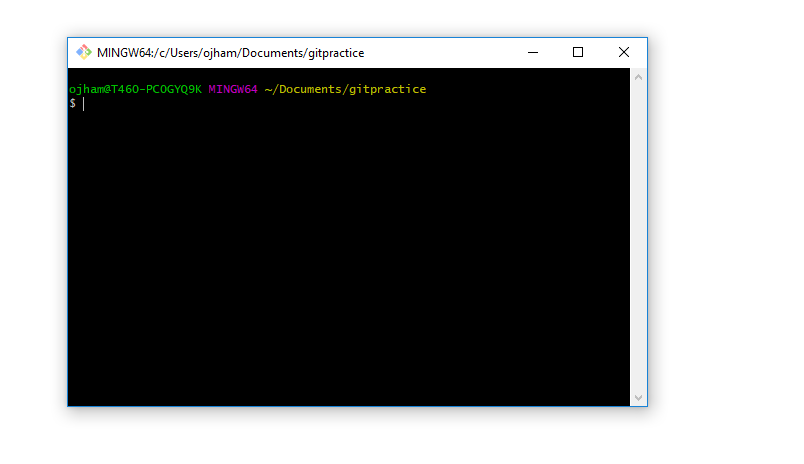


**gitignore- if you don’t want files to get pushed or pulled , those type of files should be included in gitignore.**



1. **Navigate to the folder where you want your local repository to be in your machine. Right click and select git bash here.**

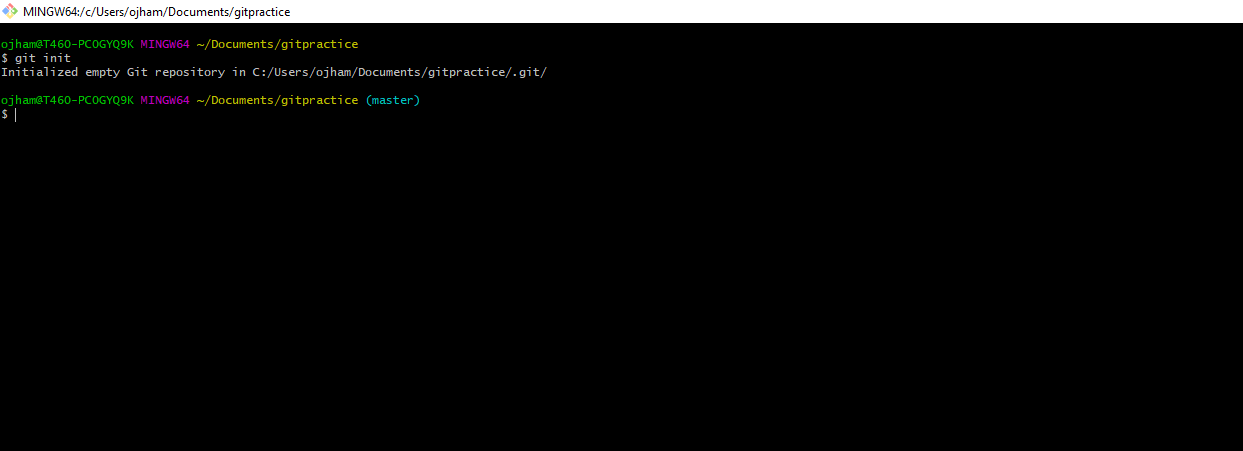




Commands

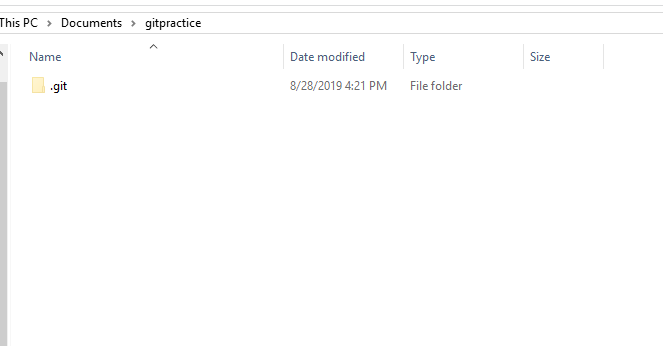
Command to create local repo:

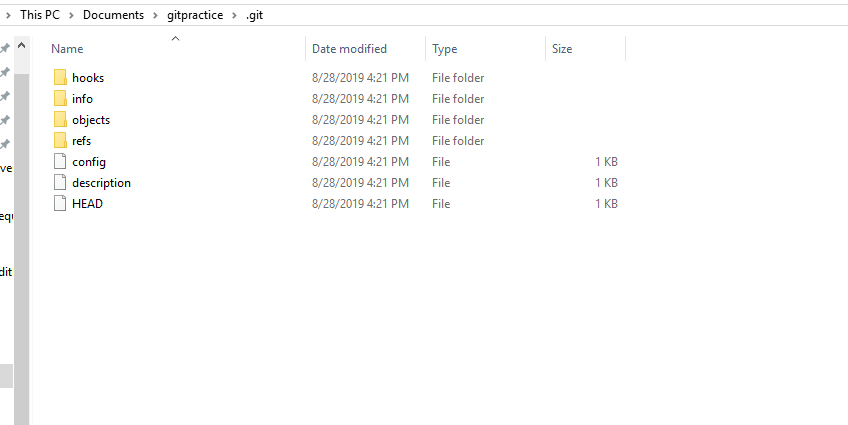
**$git init**



Local repository has been created on your local machine.

. git folder will be residing in the folder where you created your local repository.





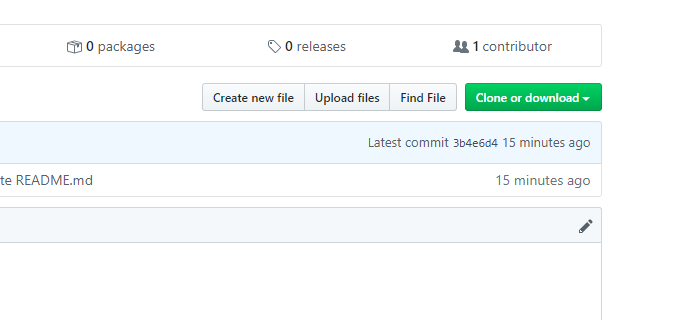
**Remote repository or central repository will be called as “Origin”.**

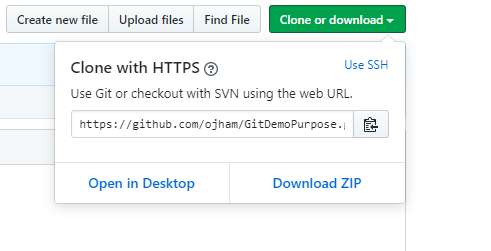
Link should be established between your central repository and local repository

So that we will come to know from which place we need to push or pull the code.

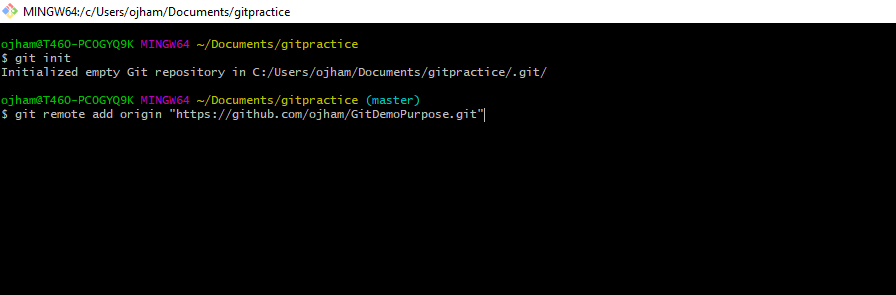
Git command to link origin and remote repository:

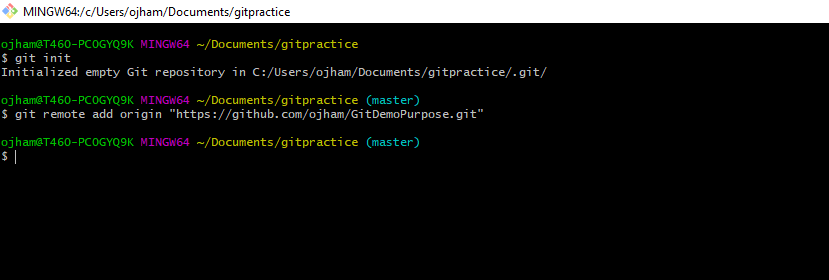
**$git remote add origin “link of your central repository”**





**Origin link-** [**https://github.com/ojham/GitDemoPurpose.git**](https://github.com/ojham/GitDemoPurpose.git)



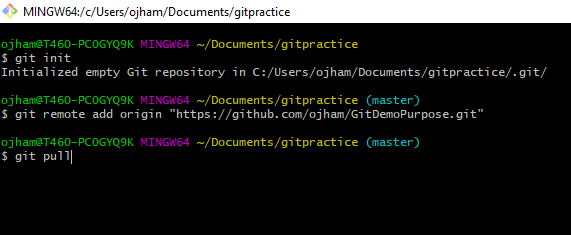


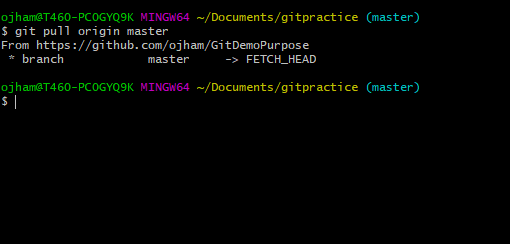
**Successfully linked.**

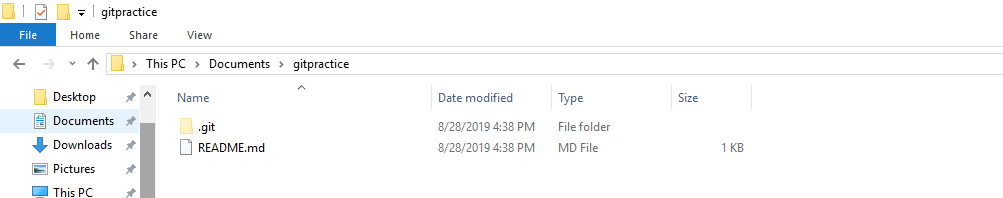
To pull the files from the central repository into your local repository:

**Git command:**

**$git pull branch name**







Used for updating your local repository with respect to central repository.

Files in the remote and local repository are in the compressed format.

But while working in the workspace it’s in the uncompressed format.

**Good practice:**

Before working updated your local repository with your central repository with the help of git pull. After getting all the updates work in your local repository

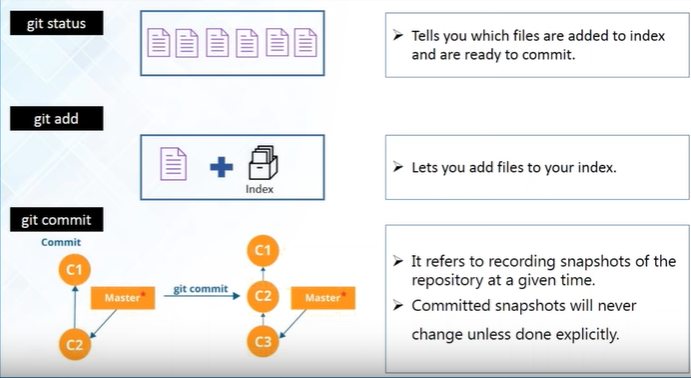
And once your code is working fine, then push your changes into your central repository.

There is layer between your workspace between local repository, that is called indexing. To commit your files into your local repository first we need to add the files in the index.

Workspace

Index

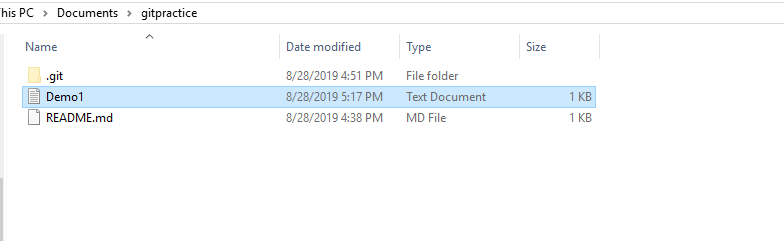
Local repository



**Version controls keeps the snapshot/object of our project /files.**

**So, every time when we commit it creates a new snapshot/object with the modified changes on the top of the old one.**

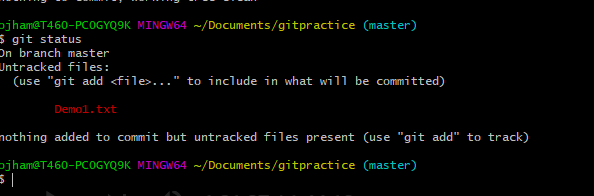
**Adding one new file into the local repository:**



Still this file is in the workspace not in your local repository. Because it has not been committed yet.

Git command to know which files are added in the index:

**$git status**

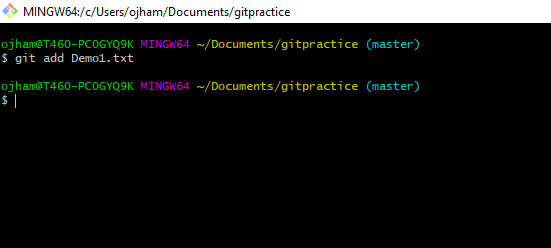


Untracked files are the one which are not added in the index.

First, we need to add the files in the index

Git Command to add files in the index:

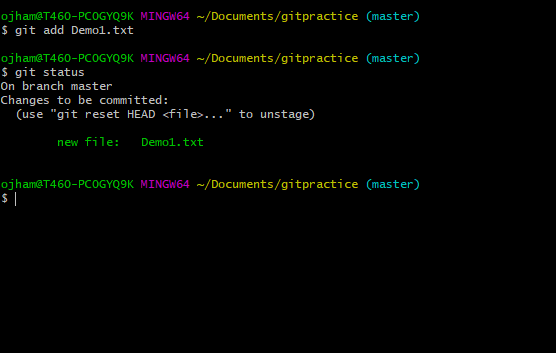
**$git add filename**



To check if file has been added to the index;

Git command to check status:

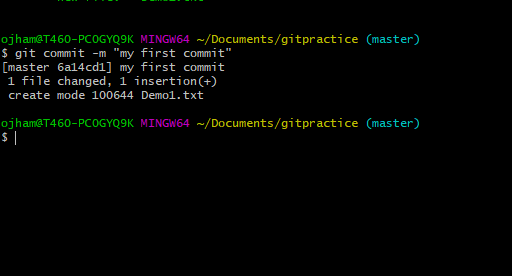
**$git status**

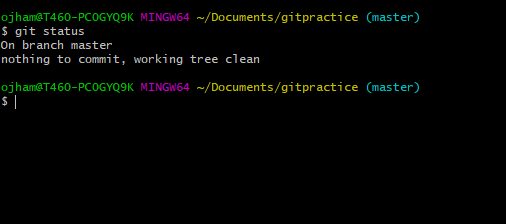


To commit the changes in the local repository.

Git command to the commit the changes into local repository.

**$git commit – m “message”**

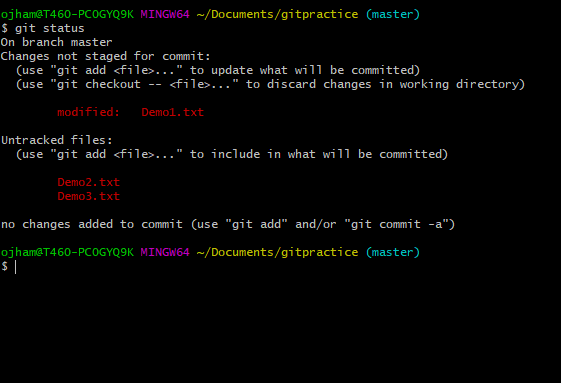




When no files are there in the index.

**Committing multiples files at a time:**

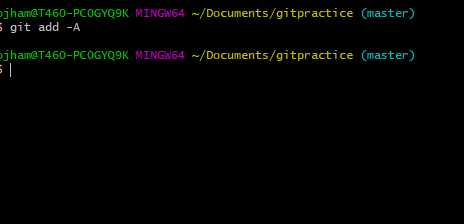
**Added multiple files and modified the previous file.**

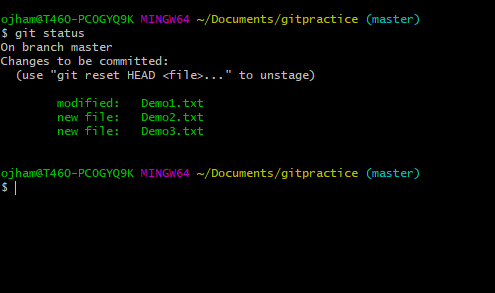


To Add multiple files into the index at once:

**Git Command:**

**$ git add -A**

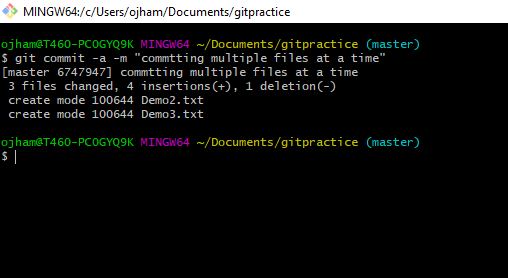




Committing multiple files at a time:

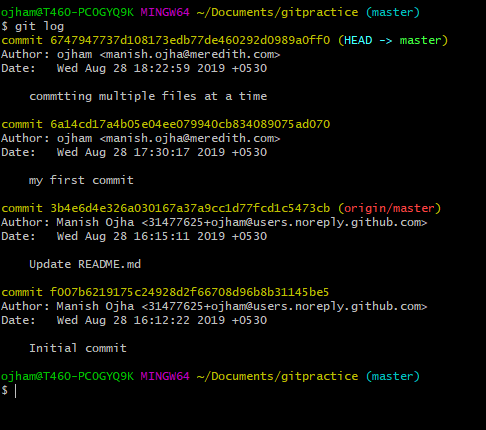
**Git command :**

**$git commit - a – m”Message”.**



**Command to check how git stores all the commit:**

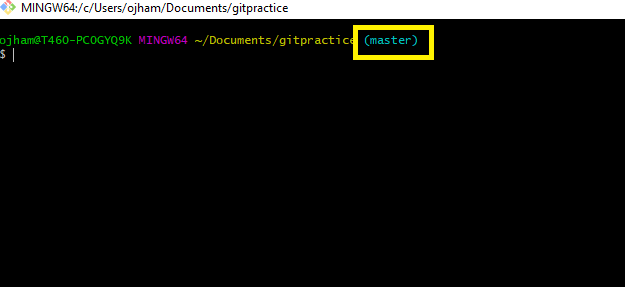
**$git log**



It shows all the commit done so far.

We cannot commit without providing the message.

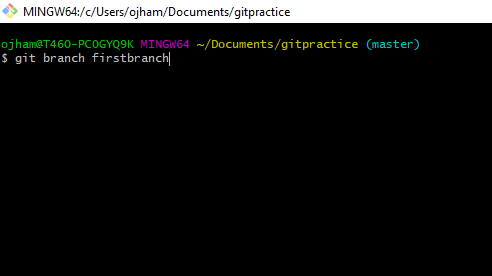
**Parallel development- Branching:**

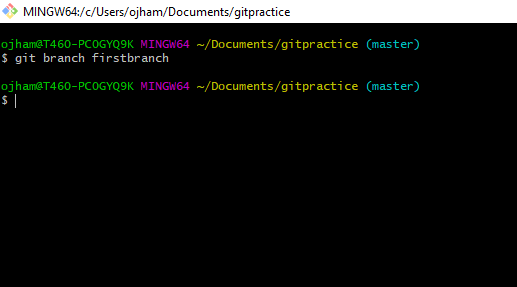


Highlighted part shows we are in the master branch.

Command to create new Branch:

**$git branch “branchname”.**



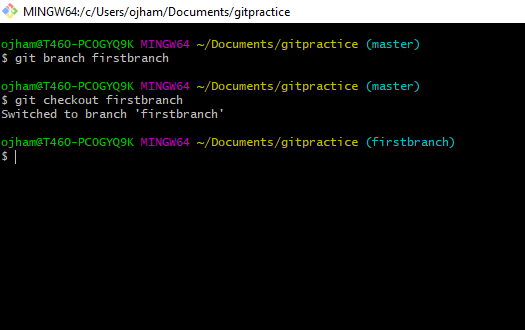


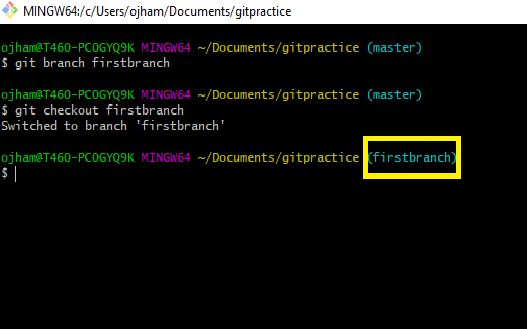
First branch will contain all the files present in the master branch since its

originated from the master branch.

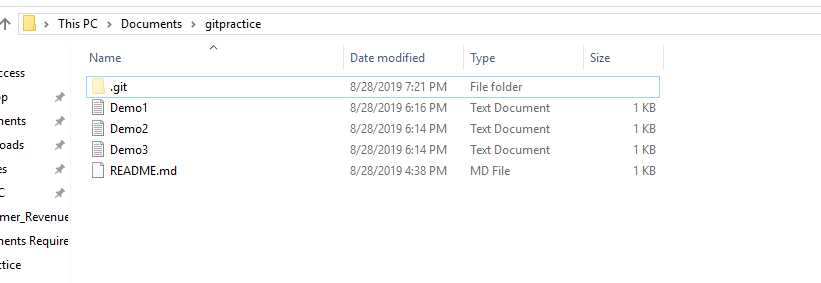
Command to switch to any branch: It called checking out in git.

**$git checkout “branch name”.**

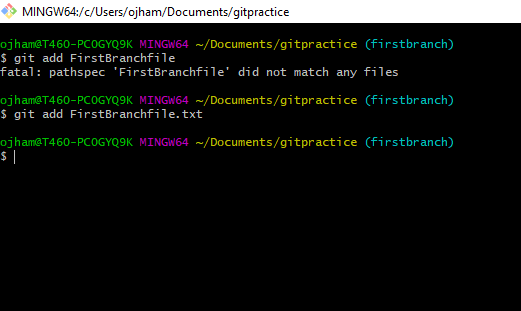


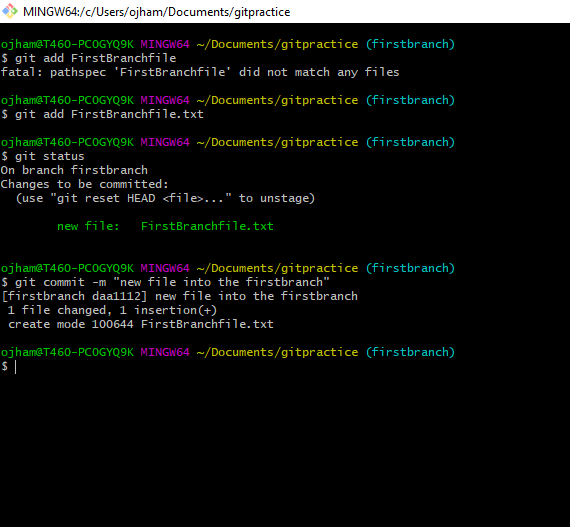
****

Now we have switched to the first branch.

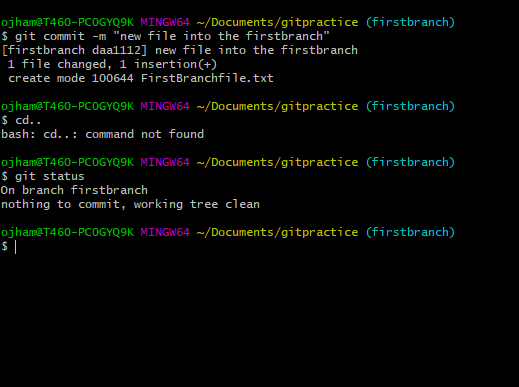


**Added a new file into the branch:**



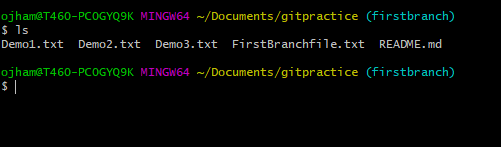


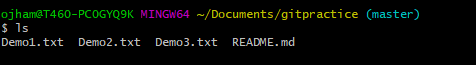
**Committed the new file into the local repository in the firstbranch.**



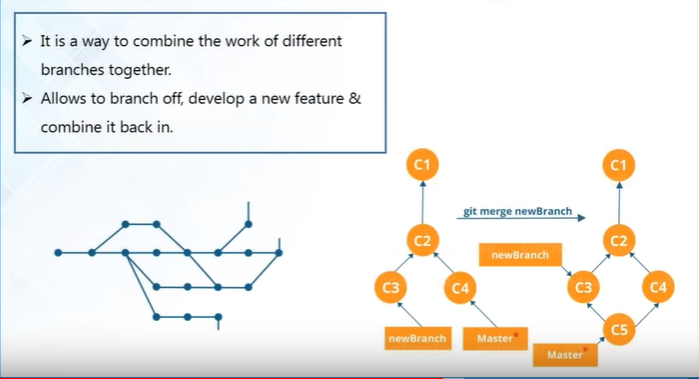
**To list out the files in any branch:**

**$ls**





**Merging:**



If you are creating some new feature in the branch and you want to add it in the master branch, then merging option is required.

Its way of combining work of different branches together.

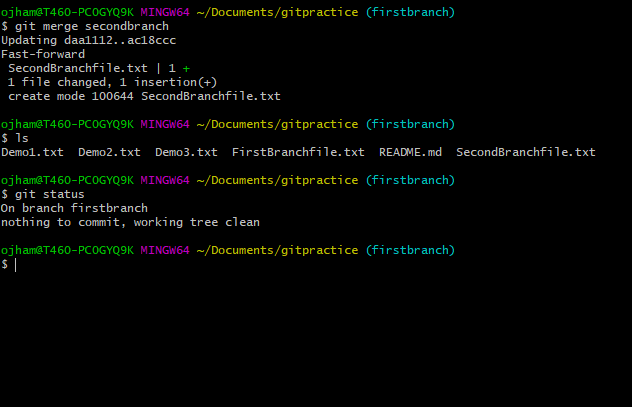
Master branch will always contain your production quality code.

**Before merging we should make sure that we are checkout in the destination branch.**

Git command :

**$git merge branch**

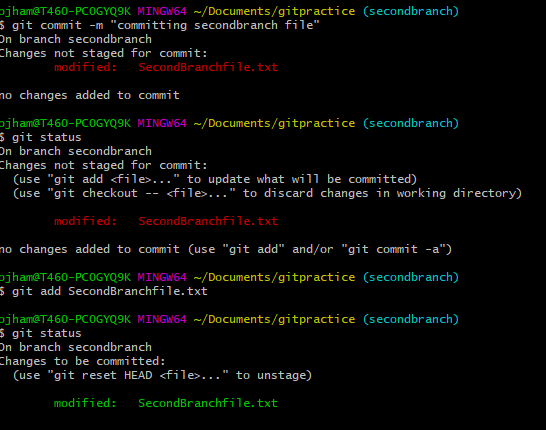
**Note: branch- should be the branch name you want to merge in the checkout branch.**



Here the first branch is the destination branch and second branch is merged in the first branch.

If we want to do some changes in some file in the second branch, we can do that and it will not effect the first branch since we have not merged the changes.

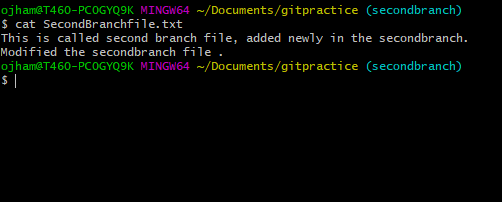
Again, checkout the second branch and modify the file:

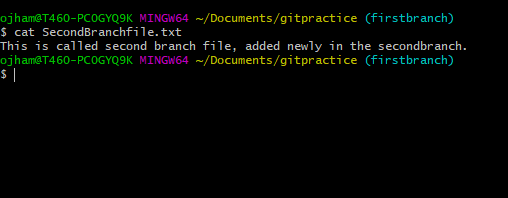


Note: if file is already added in the local repo /or being tracked , then no need of

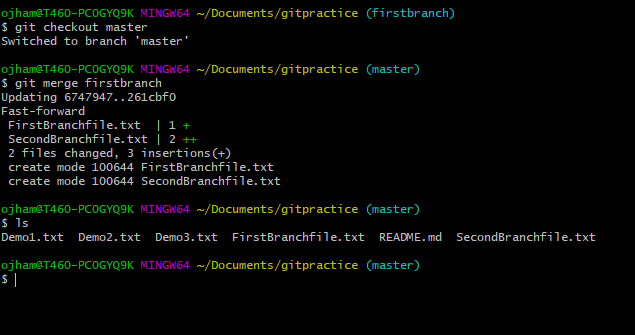
git add command directly we can go for **git commit- a** command.

Purpose for the cat command:

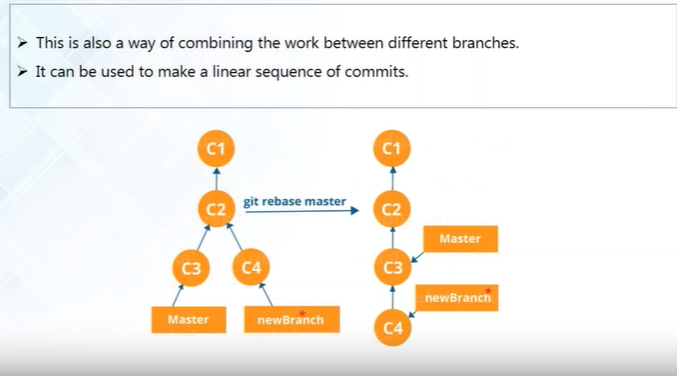




Merged the first branch into the local repo the master branch:



Re Basing:



**Difference between git pull and git fetch:**

**$git pull** -will pull all the new files and changed files from the central repository and place them into master branch or connect then with the master branch.

**$git fetch**- will also fetch the new files and changed files not into my workflow.

It also shows different branches from the remote repository (new branches which have been added or files modified in other branch.

Fetch command will not do any changes in the local repo.

Git Command for rebase

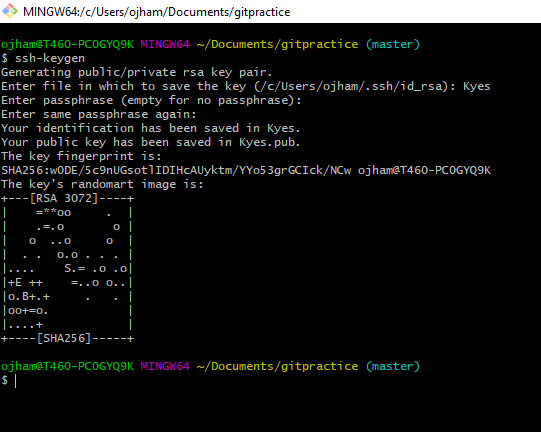
**$git rebase branch**

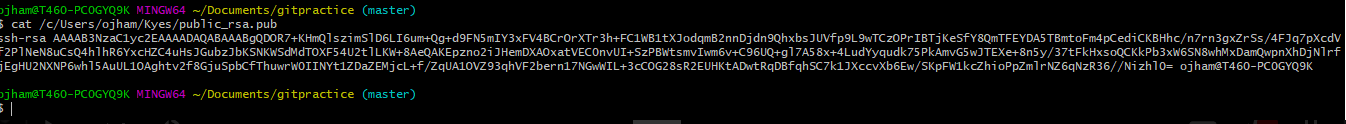
Note – branch is branch name.

To Push the code into the central repository we need establish the connection between the local repository and remote repository. We establish this connection using SSH public key.

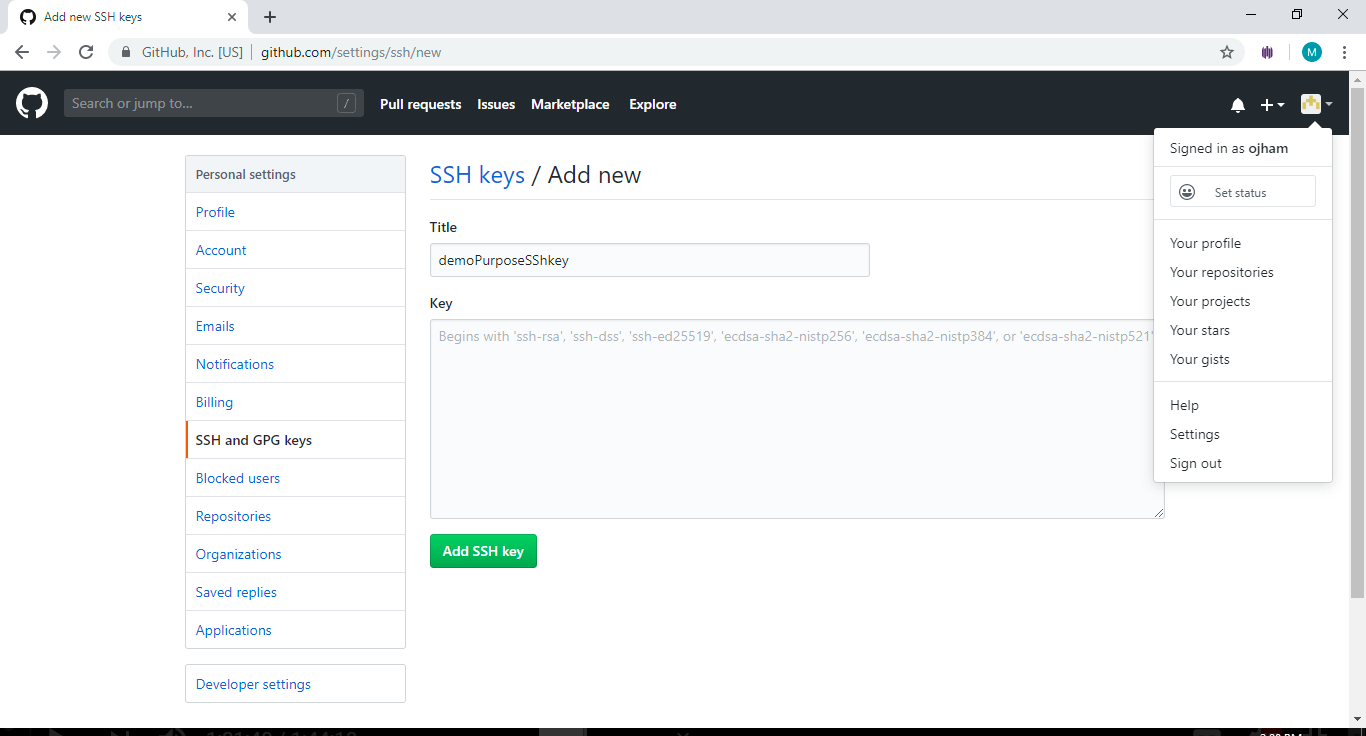
**Command for SSh key:**

**$ssh-keygen**

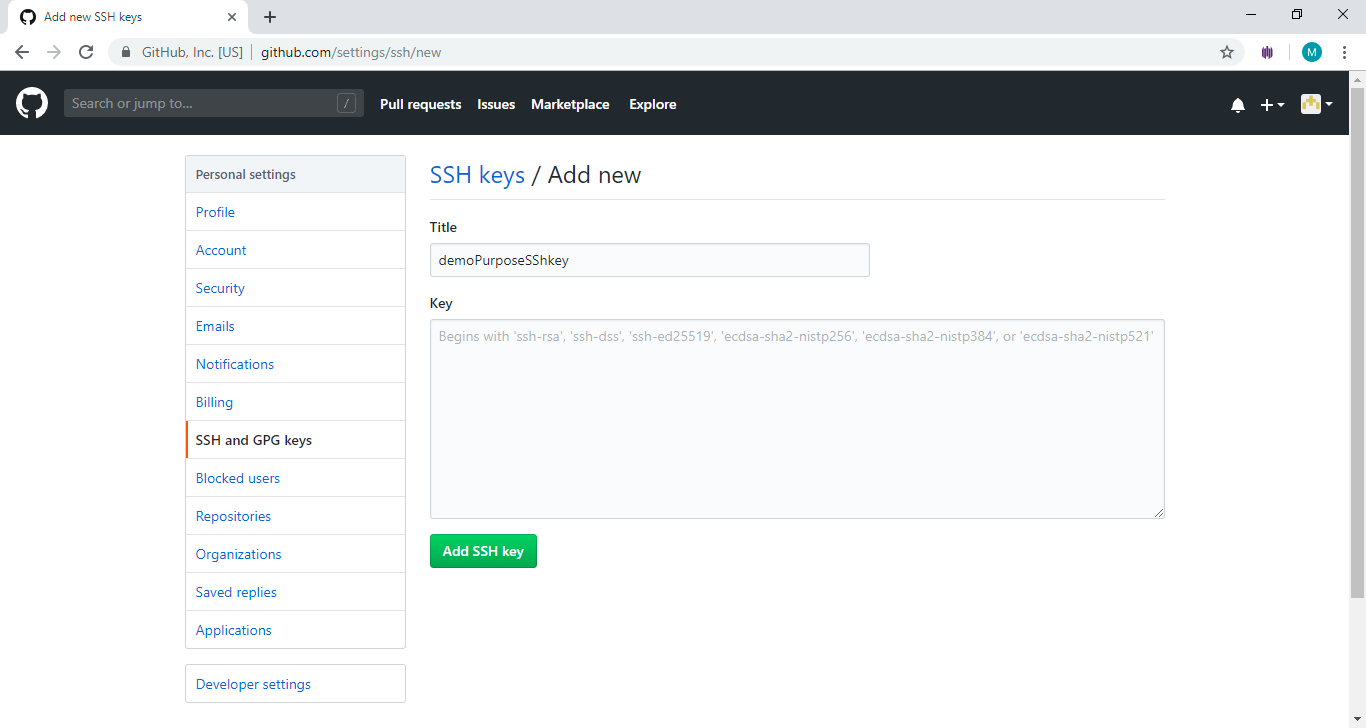


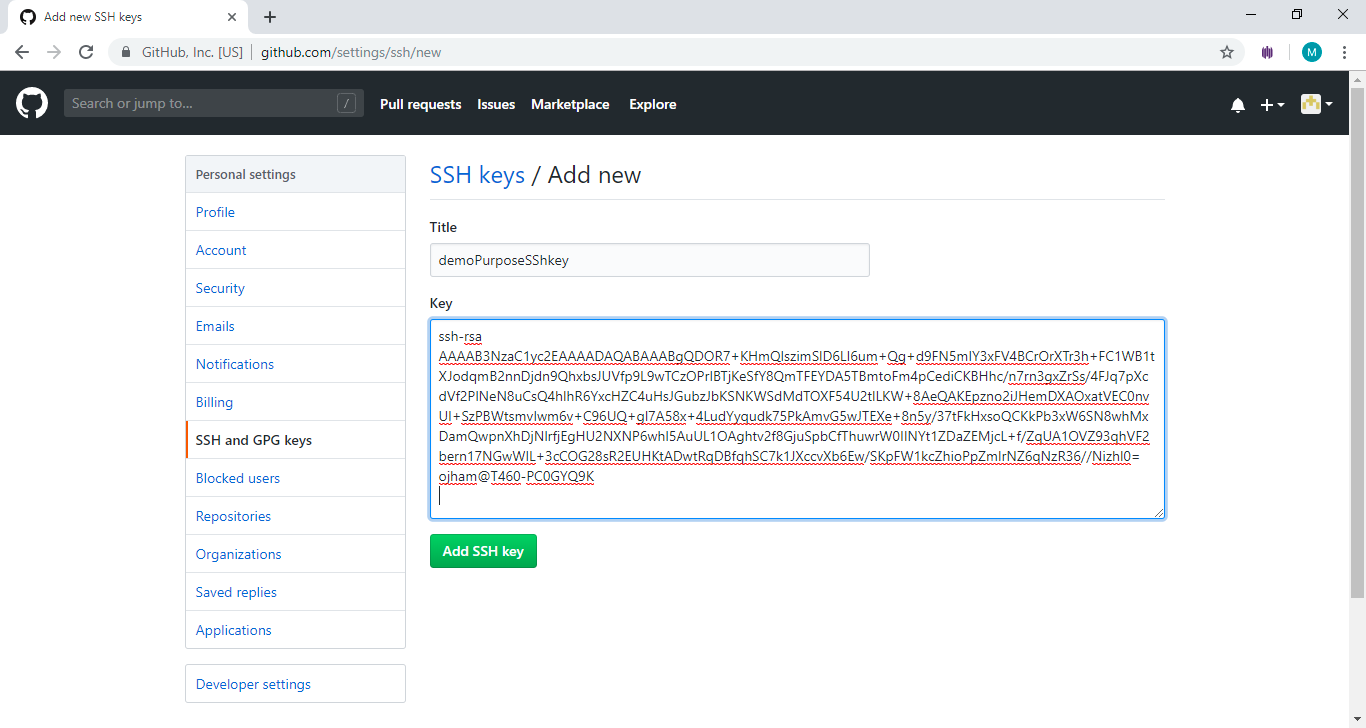


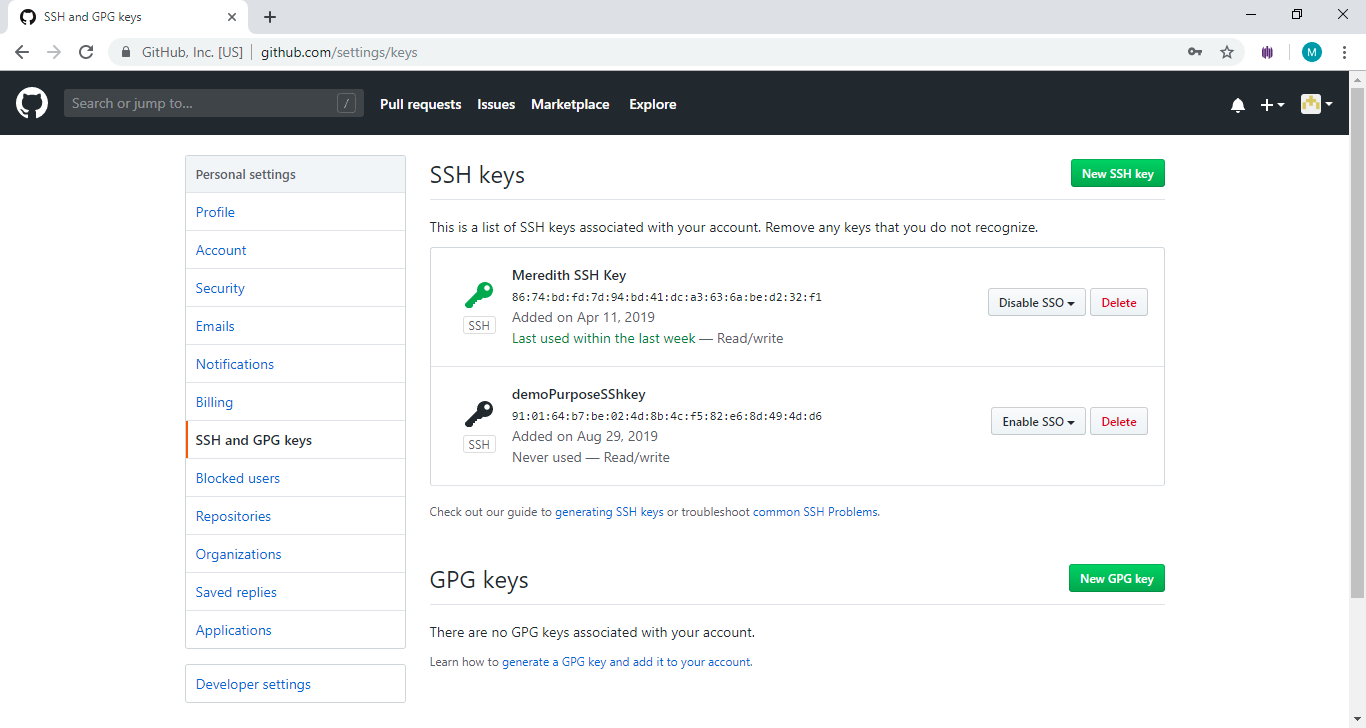
**Add the ssh key into your github account.**



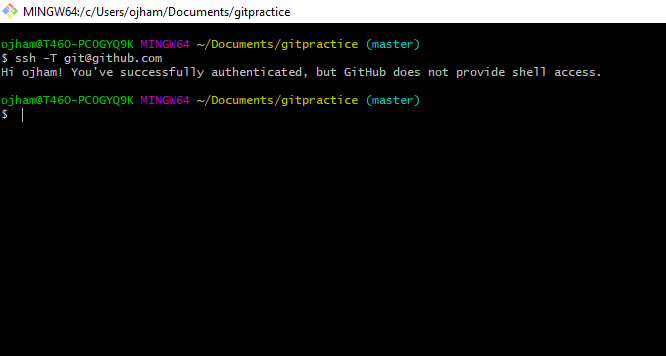
**Click on settings and go to SSH and GPG keys**





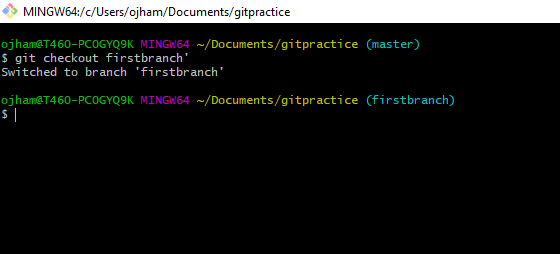


**To authenticate your account**



**To push the code into the central repository**

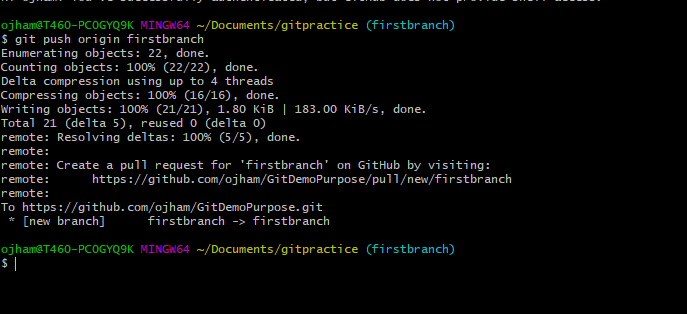
**If you want to push particular branch then checkout that branch.**



Command to push the changes into the central repository branch

**$git push origin branchname**

**Branch name- branch in which we are checked out.**



**Saving.**