Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

* **Repository**: A software repository, or “repo” for short, is a storage location for software packages.
* Created private repository for practice
* **SSH Key**: A SSH key is an alternate way to identify yourself that doesn't require you to enter your username and password every time.
* Open git bash and enter this Command to Generate a SSH key

ssh-keygen -t rsa -C "username or email"   
Example: ssh-keygen -t rsa -C "tarak.suthar1999@gmail.com"

* **Cloning**: When we clone a repository, all the files are downloaded to the local machine but the remote git repository remains unchanged. Making changes and committing them on your local repository (cloned repository) will not affect the remote repository that you cloned in any way. These changes made on the local machine can be synced with the remote repository anytime the user wants.
* Successfully cloned the initially created repository to my local machine executing following commands.

$ mkdir antidote

$ git clone [git@github.com:taraksuthar1999/antidote.git](mailto:git@github.com:taraksuthar1999/antidote.git)

$ cd antidote

* **git add** is used to add files to the staging area. For example, the basic Git following command will index the temp.txt file:

$ git add index.php

* **git commit** will create a snapshot of the changes and save it to the git directory.

$ git commit –m “echo statement added”

* **git status** displays the list of changed files together with the files that are yet to be staged or committed.

$ git status

* **git push** is used to send local commits to the main branch of the remote repository. Here’s the basic code structure:

$ git push origin main

* **git fetch** You always need the latest update in your repo. Use below command to fetch latest updates

$ git fetch origin

* **git checkout** creates branches and helps you to navigate between them. For example, the following basic command creates a new branch and automatically switches you to it.

$ git checkout -b antidote-india origin/main

* To switch from one branch to another, simply use:

$ git checkout main

* **Git Merge** used to merge target branch to the main branch .

$ git merge --no-ff origin/antidote-india