

Project Day-1

Date:21-02-2023

- **Without Version Control System**

- No collaboration between the team.
- Once saved all changes are permanent.
- Since there is no backup maintained.

- **All these problems can be solved with the help of Version Control System.**

- Best Version Control Systems are : GitHub, GitLab, etc.

WorkFlow oF Git

1. **Remote Repository:** All the changes that Collaborators make just uploaded to Remote repository.
2. **Local Repository:** Users access all the files to the local Repository and then make changes. Once they have a set of changes, push changes to the remote repository.
3. **Working Copy:** it is a User Active Directory, the user modifies the existing file and creates the new file.
4. **Staging Area:** All the modified files commit a place. Once you make a changes then commit the changes before committing you just put them into a staging area. It is a place between working copy and remote repository.

The Commands that perform these Actions

CLONE: create copy of existing remote repository inside the local repository.

COMMIT: commits all the files in the staging area to the local repository.

PUSH: pushes all the changes made in the local repository to the remote repository.

FETCH: collects the changes made in the remote repository and copies them to the local Repository.

PULL: Similar to fetch.

Need to Generate SSH Key

You Can connect github without using supplying username and personal access Token at each visit.

Steps to be followed:

- Go to github account
- select settings
- Select ssh key and GPG keys
- Next select new ssh key
- Search in google Generating ssh key
- \$ ssh-keygen -t ed25519 -C "your_email@example.com" take this text
- Open git bash at anywhere
- Paste that text you copied
- Then enter enter
- One key will be generated and it showing one location

 MINGW64:/c/Users/User/Pictures

```
User@DESKTOP-MOBIJV6 MINGW64 ~/Pictures (master)
$ ssh-keygen -t ed25519 -C "priyankakankanawadi423@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/User/.ssh/id_ed25519):
/c/Users/User/.ssh/id_ed25519 already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Passphrases do not match. Try again.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Passphrases do not match. Try again.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/User/.ssh/id_ed25519
Your public key has been saved in /c/Users/User/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:J8TFWdGEVCQ06etPRmMyP8jY+P91xp+jKjasiThEq2I priyankakankanawadi423@gmail.com
The key's randomart image is:
+---[ED25519 256]---+
|          ..=BO+   |
|          . .o oo.  |
|          o  .      |
|          .  .  .    |
|          . . S . o.+ |
|          o  o =.B o  |
|          o  . o.+ =  |
|.E .. . . = ..o +=  |
|o ... oo o.o++o+    |
+-----[SHA256]-----+


User@DESKTOP-MOBIJV6 MINGW64 ~/Pictures (master)
$
```

- Go to the specified location and copy that generated key
- Go to github select new ssh key
- Title : anything you want Key: paste that copied key
- Click on add SSH key
- Confirm git access password

The screenshot shows the GitHub interface for user 'priyaadya'. The left sidebar contains navigation links: Public profile, Account, Appearance, Accessibility, Notifications, Access, Billing and plans, Emails, Password and authentication, Sessions, SSH and GPG keys (highlighted), Organizations, and Moderation. The main content area is titled 'SSH keys' and includes a 'New SSH key' button. Below the title, it states: 'This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.' Under the heading 'Authentication Keys', there is a single entry for a 'new Ssh key' with a SHA256 fingerprint, added on Feb 23, 2023, and marked as 'Never used'. A 'Delete' button is next to the entry. Below this, there is a link to a guide on generating SSH keys. The 'GPG keys' section below shows 'There are no GPG keys associated with your account.' and a link to learn how to generate a GPG key. A 'New GPG key' button is also present. In the bottom right corner, there is a 'Activate Windows' watermark.

Cloning from remote to local repository using ssh key

- Go to github account from which repository you want take that one copy ssh key from code
- Git Bash anywhere you want
- Use clone and paste that copied code

 MINGW64:/e

```
User@DESKTOP-MOBIJV6 MINGW64 /e
$ git clone git@github.com:priyaadya/Demo.git
Cloning into 'Demo'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 9 (delta 1), reused 8 (delta 0), pack-reused 0
Receiving objects: 100% (9/9), done.
Resolving deltas: 100% (1/1), done.

User@DESKTOP-MOBIJV6 MINGW64 /e
$ git status
fatal: not a git repository (or any of the parent directories): .git

User@DESKTOP-MOBIJV6 MINGW64 /e
$ 0
```

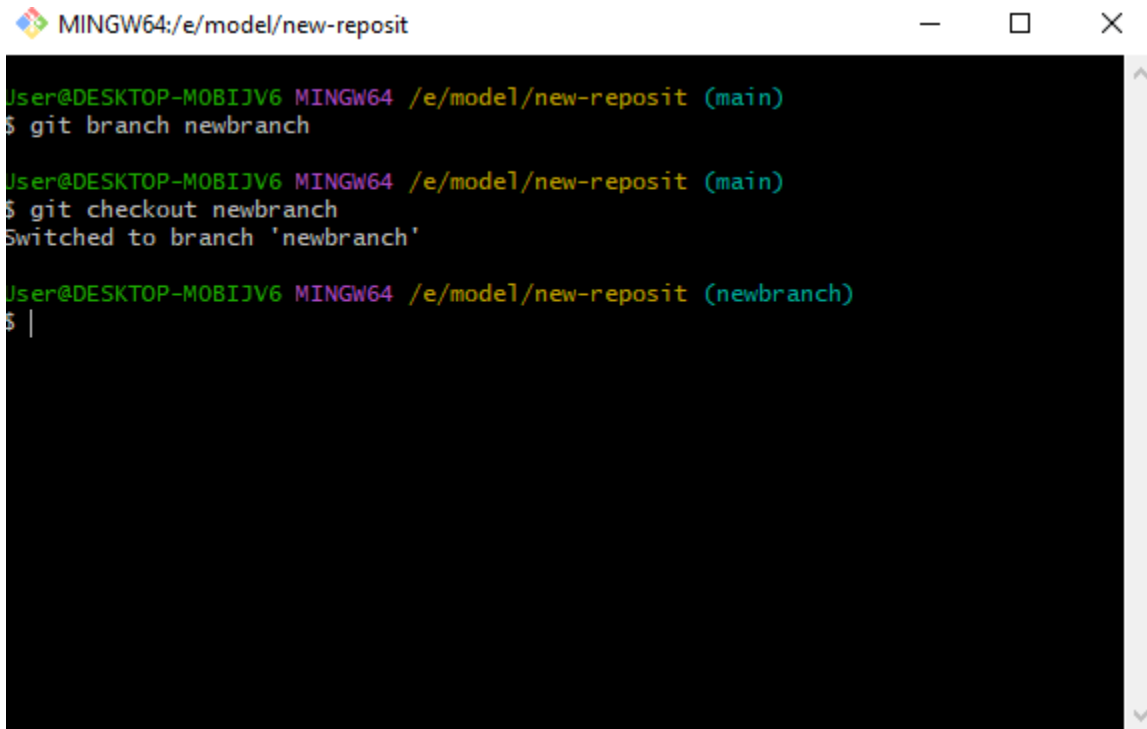
- After cloning you will get a remote repository in your local repository .If you want to add any new file from the local repository create a new file and git bash from where the file is located.
- Check git status
- Git add -A for adding files
- Git commit -m “message”
- Git push -u origin master

Project Day-3

Creating new branch, push and pull the request

Steps to be followed:

- Clone project
- Create a new branch in local repository
- Git branch branch name
- Git checkout branch name

A screenshot of a terminal window titled "MINGW64:/e/model/new-reposit". The window shows a series of Git commands being executed in a shell. The prompt is "User@DESKTOP-M0BIJV6 MINGW64 /e/model/new-reposit (main)". The first command is "\$ git branch newbranch". The second command is "\$ git checkout newbranch", followed by the output "Switched to branch 'newbranch'". The third command is "\$", and the prompt changes to "(newbranch)".

```
MINGW64:/e/model/new-reposit
User@DESKTOP-M0BIJV6 MINGW64 /e/model/new-reposit (main)
$ git branch newbranch

User@DESKTOP-M0BIJV6 MINGW64 /e/model/new-reposit (main)
$ git checkout newbranch
Switched to branch 'newbranch'

User@DESKTOP-M0BIJV6 MINGW64 /e/model/new-reposit (newbranch)
$ |
```

→ Make some changes in local repository and push

MINGW64:/e/model/new-reposit

```
User@DESKTOP-MOBIJV6 MINGW64 /e/model/new-reposit (newbranch)
$ git add -A
```

```
User@DESKTOP-MOBIJV6 MINGW64 /e/model/new-reposit (newbranch)
$ git commit -m "added"
[newbranch 5e68da4] added
1 file changed, 4 insertions(+)
```

```
User@DESKTOP-MOBIJV6 MINGW64 /e/model/new-reposit (newbranch)
$ git push
fatal: The current branch newbranch has no upstream branch.
To push the current branch and set the remote as upstream, use
```

```
git push --set-upstream origin newbranch
```

To have this happen automatically for branches without a tracking upstream, see 'push.autoSetupRemote' in 'git help config'.

```
User@DESKTOP-MOBIJV6 MINGW64 /e/model/new-reposit (newbranch)
$ git ^C
```

```
User@DESKTOP-MOBIJV6 MINGW64 /e/model/new-reposit (newbranch)
$ git git push --set-upstream origin newbranch
git: 'git' is not a git command. See 'git --help'.
```

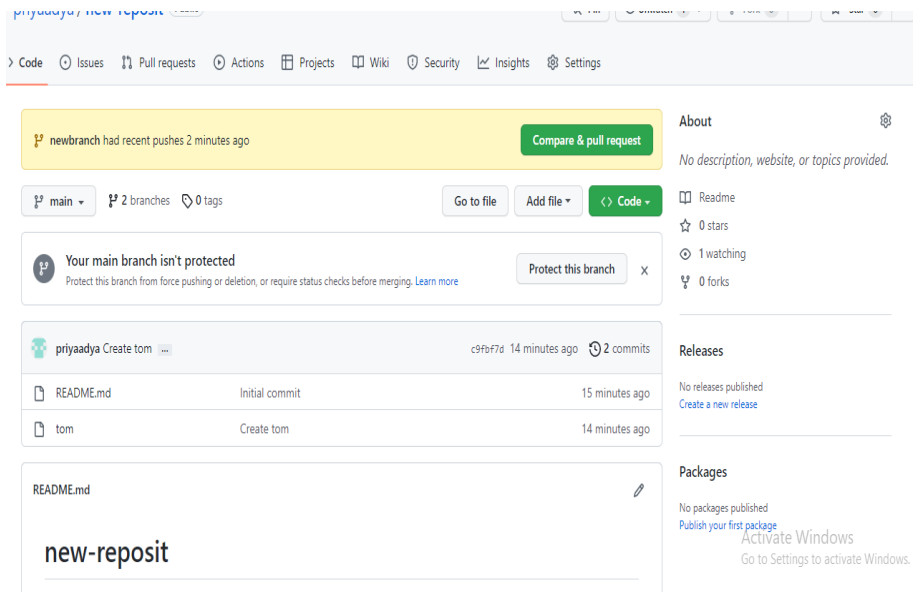
The most similar command is
init

```
User@DESKTOP-MOBIJV6 MINGW64 /e/model/new-reposit (newbranch)
$ ^C
```

```
User@DESKTOP-MOBIJV6 MINGW64 /e/model/new-reposit (newbranch)
$ git push --set-upstream origin newbranch
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 311 bytes | 155.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'newbranch' on GitHub by visiting:
remote:   https://github.com/priyaadya/new-reposit/pull/new/newbranch
remote:
To github.com:priyaadya/new-reposit.git
 * [new branch]      newbranch -> newbranch
branch 'newbranch' set up to track 'origin/newbranch'.
```

```
User@DESKTOP-MOBIJV6 MINGW64 /e/model/new-reposit (newbranch)
```

- Go to remote repo and compare and pull request
 - ◆ Create pull request
 - ◆ Merge pull request
 - Confirm merge



Importing project in eclipse

- Create one folder
- Open git bash
- Git clone https://github.com/Darshan008-pheonix/Emmployee_DB.git

```
User@DESKTOP-M0BIJY6 MINGW64 /e/project folder
$ git clone https://github.com/Darshan008-pheonix/Emmployee_DB.git
Cloning into 'Emmployee_DB'...
remote: Enumerating objects: 45, done.
remote: Counting objects: 100% (45/45), done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 45 (delta 3), reused 35 (delta 1), pack-reused 0
Receiving objects: 100% (45/45), 63.55 KiB | 9.08 MiB/s, done.
Resolving deltas: 100% (3/3), done.

User@DESKTOP-M0BIJY6 MINGW64 /e/project folder
$ |
```

- One file is imported in that folder

→ Go to eclipse => file =>open project from file system=>browse that respective folder from directory=>choose the project finish.

