SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

Academic Year: 2023_24

Experiment No. 9

Team Members :

Riddhi Shah-60004210161

Dhruvin Chawda-60004210159

Aim: Study of Configuration Management using GitHub

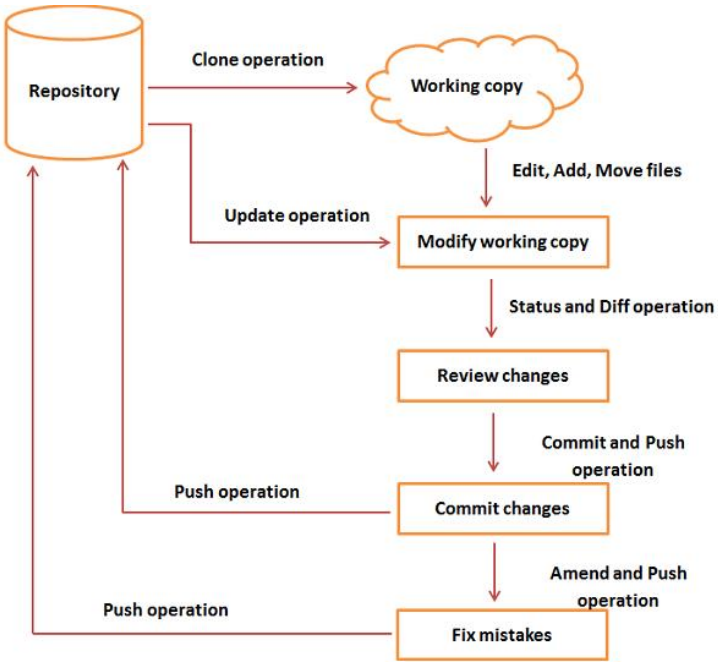
Theory:

Git is a distributed revision control and source code management system with an emphasis on speed. Git was initially designed and developed by Linus Torvalds for Linux kernel development. Git is a free software distributed under the terms of the GNU General Public License version 2.

Git Life Cycle

General workflow is as follows –

1. Clone the Git repository as a working copy.
2. Modify the working copy by adding/editing files.
3. If necessary, update the working copy by taking other developer's changes.
4. Review the changes before commit.
5. Commit changes. If everything is fine, then push the changes to the repository.
6. After committing, if something is wrong, then correct the last commit and push the changes to the repository.



Git Life Cycle

Implementation :

1. Creating Git Repository

   Initialize a new repository by using **init** command followed by **--bare** option. It initializes the repository without a working directory. By convention, the bare repository must be named as **.git**.

```
Acer@BlackBeast MINGW64 ~/Desktop/TE/sem6/SE (master)
$ pwd
/c/Users/Acer/Desktop/TE/sem6/SE

Acer@BlackBeast MINGW64 ~/Desktop/TE/sem6/SE (master)
$ mkdir project.git

Acer@BlackBeast MINGW64 ~/Desktop/TE/sem6/SE (master)
$ cd project.git/

Acer@BlackBeast MINGW64 ~/Desktop/TE/sem6/SE/project.git (master)
$ git --bare init
Initialized empty Git repository in C:/Users/Acer/Desktop/TE/sem6/SE/project.git/

Acer@BlackBeast MINGW64 ~/Desktop/TE/sem6/SE/project.git (BARE:master)
$ ls
HEAD  config  description  hooks/  info/  objects/  refs/

Acer@BlackBeast MINGW64 ~/Desktop/TE/sem6/SE/project.git (BARE:master)
$
```

2. Generate Public-Private RSA Key Pair

```
Acer@BlackBeast MINGW64 ~/Desktop/TE/sem6/SE/project.git (BARE:master)
$ pwd
/c/Users/Acer/Desktop/TE/sem6/SE/project.git

Acer@BlackBeast MINGW64 ~/Desktop/TE/sem6/SE/project.git (BARE:master)
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Acer/.ssh/id_rsa): s
Created directory '/c/Users/Acer/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Acer/.ssh/id_rsa
Your public key has been saved in /c/Users/Acer/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:1Hr1XZSTCkwoOgQ22/znLYfoZBircGOQ/xDrF4iryA4 Acer@BlackBeast
The key's randomart image is:
+---[RSA 3072]----+
|     ++.  +.    +|
|    ..=o o o   = |
|     ..oo . o . o|
|    .  ... . o ..|
|   o....S...  . .|
|   .o.o.+.+ o    |
|E  ..B o.+ + o   |
|+  .= =.+   o    |
|o+.  o.. .       |
+----[SHA256]-----+

Acer@BlackBeast MINGW64 ~/Desktop/TE/sem6/SE/project.git (BARE:master)
$
```

3. Adding keys to authorized keys

   Suppose there are two developers working on a project. Both users have generated public keys.

   Both add their public key to the server by using ssh-copy-id command as given below

```
[user1@CentOS ~]$ pwd

/home/user1



[user2@CentOS ~]$ ssh-copy-id -i ~/.ssh/id_rsa.pub gituser@git.server.com
```

4. Push changes to the repository

   We have created a bare repository on the server and allowed access for two users. Both users can push their changes to the repository by adding it as a remote.

   Git init command creates **.git** directory to store metadata about the repository every time it reads the configuration from the **.git/config** file.

   User1 creates a new directory, adds README file, and commits his change as initial commit. After commit, he verifies the commit message by running the **git log** command.

```
Acer@BlackBeast MINGW64 ~/Desktop/TE/sem6/SE (master)
$ pwd
/c/Users/Acer/Desktop/TE/sem6/SE

Acer@BlackBeast MINGW64 ~/Desktop/TE/sem6/SE (master)
$ mkdir test_repo

Acer@BlackBeast MINGW64 ~/Desktop/TE/sem6/SE (master)
$ cd test_repo

Acer@BlackBeast MINGW64 ~/Desktop/TE/sem6/SE/test_repo (master)
$ git init .
Initialized empty Git repository in C:/Users/Acer/Desktop/TE/sem6/SE/test_repo/.git/

Acer@BlackBeast MINGW64 ~/Desktop/TE/sem6/SE/test_repo (master)
$ echo 'TODO: Adding contents in the readme ^_0' > README

Acer@BlackBeast MINGW64 ~/Desktop/TE/sem6/SE/test_repo (master)
$ git status -s
?? README

Acer@BlackBeast MINGW64 ~/Desktop/TE/sem6/SE/test_repo (master)
$ git add .
warning: in the working copy of 'README', LF will be replaced by CRLF the next time Git touches it

Acer@BlackBeast MINGW64 ~/Desktop/TE/sem6/SE/test_repo (master)
$ git status -s
A  README

Acer@BlackBeast MINGW64 ~/Desktop/TE/sem6/SE/test_repo (master)
$ git commit -m "first commit :)"
[master (root-commit) e268b88] first commit :)
 1 file changed, 1 insertion(+)
 create mode 100644 README
```

5. Checking log message by executing the git log command.

```
Acer@BlackBeast MINGW64 ~/Desktop/TE/sem6/SE/test_repo (master)
$ git log
commit e268b880f0b3bf24e972e226ae774693966a719c (HEAD -> master)
Author: Dhruvin3103 <dcchawda2003@gmail.com>
Date:   Fri Apr 5 13:17:36 2024 +0530

    first commit :)

Acer@BlackBeast MINGW64 ~/Desktop/TE/sem6/SE/test_repo (master)
$
```

6. Commit changes

To commit the changes, he used the git commit command followed by –m option. If we omit –m option. Git will open a text editor where we can write multiline commit message

```
Acer@BlackBeast MINGW64 ~/Desktop/TE/sem6/SE/test_repo (master)
$ git add .

Acer@BlackBeast MINGW64 ~/Desktop/TE/sem6/SE/test_repo (master)
$ git commit -m "<----adding files---->"
[master 2db72d8] <----adding files---->
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Test_cases.xlsx
 create mode 100644 test_scenario.xlsx

Acer@BlackBeast MINGW64 ~/Desktop/TE/sem6/SE/test_repo (master)
$ git log
commit 2db72d8be960e0d022a18de045c597bf39d6ce23 (HEAD -> master)
Author: Dhruvin3103 <dcchawda2003@gmail.com>
Date:   Fri Apr 5 13:23:14 2024 +0530

    <----adding files---->

commit e268b880f0b3bf24e972e226ae774693966a719c
Author: Dhruvin3103 <dcchawda2003@gmail.com>
Date:   Fri Apr 5 13:17:36 2024 +0530

    first commit :)

Acer@BlackBeast MINGW64 ~/Desktop/TE/sem6/SE/test_repo (master)
$
```

Conclusion:

Git is a version control system that can keep several versions on a local workstation or integrate with a remote file management system. To perform commands, we used Git bash cmd and saved the local files to the GitHub files management server. We also saw the git log command, which stores the git activity log.