



Academic Year: 2023_24

Name: Jigar Siddhpura

SAPID: 60004200155

DIV: C/C2

Branch: Computer Engineering

SE - Experiment 9 - Configuration Management

Aim: Study of Configuration Management using GitHub

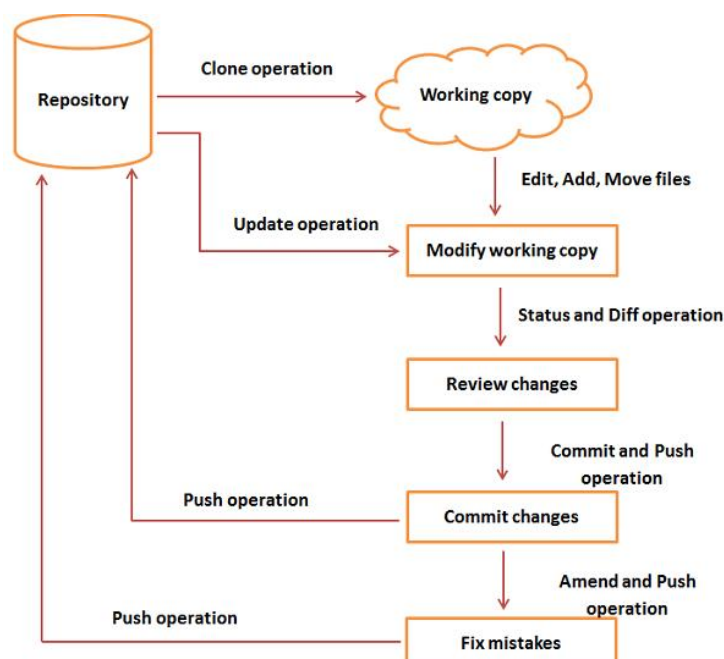
Theory:

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.





Academic Year: 2023_24
Git Life Cycle

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**.

```
[gituser@CentOS ~]$ pwd
/home/gituser

[gituser@CentOS ~]$ mkdir project.git

[gituser@CentOS ~]$ cd project.git/

[gituser@CentOS project.git]$ ls

[gituser@CentOS project.git]$ git --bare init
Initialized empty Git repository in /home/gituser-
m/project.git/

[gituser@CentOS project.git]$ ls
branches config description HEAD hooks info objects refs
```

2. Generate Public-Private RSA Key Pair

```
User1@CentOS ~]$ pwd
/home/user1

[user1@CentOS ~]$ ssh-keygen
```

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.



Academic Year: 2023_24

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.

```
[user1@CentOS ~]$ pwd
/home/user1

[user1@CentOS ~]$ mkdir user1_repo

[user1@CentOS ~]$ cd user1_repo/

[user1@CentOS user1_repo]$ git init
Initialized empty Git repository in
/home/user1/user1_repo/.git/

[user1@CentOS user1_repo]$ echo 'TODO: Add contents for
README' > README

[user1@CentOS user1_repo]$ git status -s
?? README

[user1@CentOS user1_repo]$ git add .

[user1@CentOS user1_repo]$ git status -s
A README

[user1@CentOS user1_repo]$ git commit -m 'Initial commit'
```

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

```
[user1@CentOS user1_repo]$ git log
```

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

```
[user2@CentOS project]$ git commit -m 'Implemented
my_strlen function'
```



Academic Year: 2023_24

Performance:

1. Perform all the commands using Git

```
C:\Users\Admin>cd ../../
C:\>cd .vscode
C:\.vscode>cd college
C:\.vscode\college>cd experiments
C:\.vscode\college\experiments>git clone https://github.com/Greninja28/SE-10.git
Cloning into 'SE-10'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
C:\.vscode\college\experiments>cd SE-10
C:\.vscode\college\experiments\SE-10>code .
C:\.vscode\college\experiments\SE-10>
```

2. Take screenshots of each of the command and respective output

```
PS C:\.vscode\college\experiments\SE-10> git add .
PS C:\.vscode\college\experiments\SE-10> git status -s
M start.py
PS C:\.vscode\college\experiments\SE-10> git commit -m 'file changed'
[main 8992e87] file changed
1 file changed, 1 insertion(+)
PS C:\.vscode\college\experiments\SE-10> git status -s
PS C:\.vscode\college\experiments\SE-10> git log
commit 8992e871e8f52ea791ea992f0dcd717ec04516fb (HEAD -> main)
Author: Sahej Jain <89766122+Greninja28@users.noreply.github.com>
Date: Wed May 3 09:20:21 2023 +0530

file changed

commit 2accd86c64e87864ffb442b5e68bb33432cb82bc (origin/main, origin/HEAD)
Author: Sahej Jain <89766122+Greninja28@users.noreply.github.com>
Date: Wed May 3 09:16:49 2023 +0530

Initail commit
```




Academic Year: 2023_24

3. Explore the commands for merging the documents and show the screenshots.

```
PS C:\.vscode\college\experiments\SE-10> git pull origin main
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused
Unpacking objects: 100% (3/3), 665 bytes | 41.00 KiB/s, done.
From https://github.com/Greninja28/SE-10
* branch          main      -> FETCH_HEAD
   8992e87..7ff546c  main      -> origin/main
Updating 8992e87..7ff546c
Fast-forward
 README.md | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
PS C:\.vscode\college\experiments\SE-10> git add .
PS C:\.vscode\college\experiments\SE-10> git commit -m 'Added a sum program'
[main f20de92] Added a sum program
 1 file changed, 3 insertions(+), 2 deletions(-)
PS C:\.vscode\college\experiments\SE-10> git push origin main
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), 343 bytes | 171.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Greninja28/SE-10.git
   7ff546c..f20de92  main -> main
PS C:\.vscode\college\experiments\SE-10>
```

Conclusion:

In conclusion, the study experiment on Git has provided valuable insights into the benefits of using Git for version control in software development. The experiment demonstrated that Git offers powerful tools and features to manage code changes, collaborate more efficiently, and ensure codebase consistency.

One of the significant advantages of Git is its branching and merging capabilities, which enable developers to work on multiple versions of a project concurrently without causing conflicts. This functionality is essential when working with large teams or when making significant changes to a codebase. Additionally, Git's revert and reset features make it easy to undo changes and recover previous versions of code, which can save developers time and effort in troubleshooting.

Furthermore, the study experiment confirmed that Git enhances collaboration and facilitates communication between team members by providing a centralized repository where all project files can be stored, accessed, and shared. This, in turn, reduces the likelihood of errors and inconsistencies in the codebase.