Lab Session 06

Objective:

Demonstrate the Git Commands and Clone of local repositories to remote repositories.

Theory:

Git:

Git is a DevOps tool for source code management—an open-source version control system (VCS) used to handle small to very large projects efficiently. Git is used to tracking changes in the source code, supporting non-linear development so that multiple developers can work together in near real-time.

Let us now look at the various steps in the Git installation on Windows.

Step 1:

Download the latest version of Git and choose the 64/32 bit version. After the file is downloaded, install it in the system. Once installed, select Launch the Git Bash, then click on finish. The Git Bash is now launched.

Step 2:

Check the Git version:

```
$ git --version
```

Step 3:

For any help, use the following command:

```
$ git help config
```

This command will lead you to a browser of config commands. Basically, the help the command provides a manual from the help page for the command just following it (here, it's config).

Another way to use the same command is as follows:

```
$ git config --help
```

Step 4:

Create a local directory using the following command:

```
$ mkdir test
$ cd test
```

Step 5:

The next step is to initialize the directory:

```
$ git init
```

Step 6:

Go to the folder where "test" is created and create a text document named "demo." Open "demo" and put any content, like "Hello Git". Save and close the file.

Step 7:

Enter the Git bash interface and type in the following command to check the status:

```
$ git status
```

Step 8:

Add the "demo" to the current directory using the following command:

```
$ git add demo.txt
```

Step 9:

Next, make a commit using the following command:

```
$ git commit -m "committing a text file"
```

Step 10:

Link the Git to a Github Account:

```
$ git config --global user.username
```

Note: simplilearn-github is the username on the Github account.

Step 11:

Open your Github account and create a new repository with the name "test_demo" and click on "Create repository." This is the remote repository. Next, copy the link of "test_demo."

Step 12:

Go back to Git bash and link the remote and local repository using the following command:

```
$ git remote add origin <link>
```

Here, <link> is the link copied in the previous step.

Step 13:

Push the local file onto the remote repository using the following command:

```
$ git push origin master
```

Step 14:

Move back to Github and click on "test_demo" and check if the local file "demo.txt" is pushed to this repository.

Additional Customization Options

1. This option enables users to add extra elements such as symbolic links for command lines. Nevertheless, one should always prefer default options for shortcuts or more.

2. There are some experimental options available such as pseudo control Support or Built in file system monitor concerning your installed Git version.

Lab Task:

How to Launch Git in Windows?

There are two methods to launch git in windows. One is launching git using a bash scripting shell with the help of the command line and another is launching git using a graphical user interface.

1. To launch git via bash scripting shell, First, open the window and search for git bash and open it.
2. To launch git via graphical user interface (GUI), similarly, first open the window and search for git GUI and click on the application icon and open it.

Configure GitHub Credentials

You can configure your local GitHub installation with credentials by using the following commands. Also, don't forget to add your own GitHub credentials for username and email address.

1. git config –global user.n
   ame "github_username"
2. git config –global user.e
   mail "email_address"

Clone a GitHub Repository

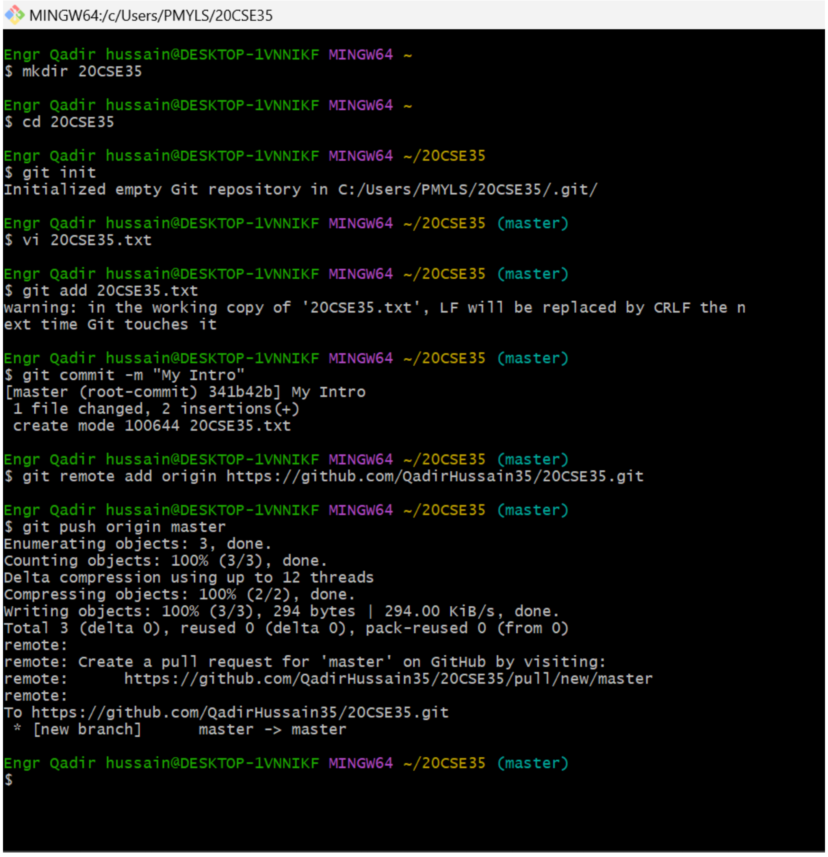
1. Initially you need to click the options repository on GitHub.
2. Then in the top right corner, click the option clone or download where a small drop-down box will appear having a URL for cloning over HTTPS.
3. Then enter into your Powershell windows and write clone URL as: git clone repository_url
4. On the other hand, you can clone a github repository with SSH URLs where first you need to generate an SSH key pair on your windows workstation as well as need to assign a public key to your GitHub account.

List Remote Repositories

1. Make a copy of the repository from GitHub for your working directory.
2. Ensure that the working directory should have the project name as "cd git_project" and replace the project name from the downloaded repository.

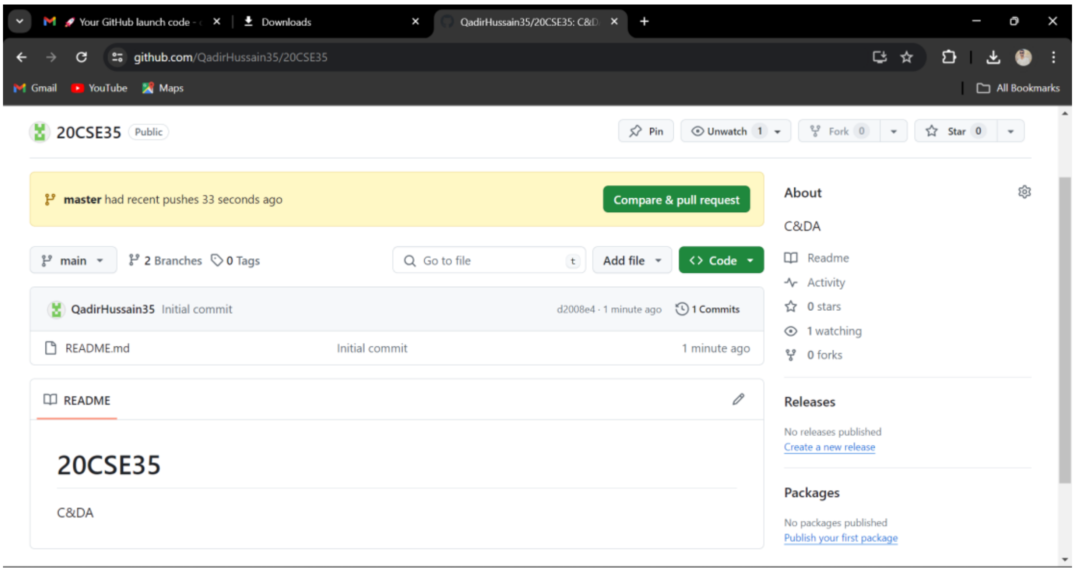
3. If the above option doesn't work, you can list the content using "ls command" for the current directory, especially to check your exact number of spellings.
4. Besides, you can list the remote repository in the sub-directory as "git remote -v".

This code is for pushing local files on github.



```
Engr Qadir hussain@DESKTOP-1VNNIKF MINGW64 ~
$ mkdir 20CSE35

Engr Qadir hussain@DESKTOP-1VNNIKF MINGW64 ~
$ cd 20CSE35

Engr Qadir hussain@DESKTOP-1VNNIKF MINGW64 ~/20CSE35
$ git init
Initialized empty Git repository in C:/Users/PMYLS/20CSE35/.git/

Engr Qadir hussain@DESKTOP-1VNNIKF MINGW64 ~/20CSE35 (master)
$ vi 20CSE35.txt

Engr Qadir hussain@DESKTOP-1VNNIKF MINGW64 ~/20CSE35 (master)
$ git add 20CSE35.txt
warning: in the working copy of '20CSE35.txt', LF will be replaced by CRLF the n
ext time Git touches it

Engr Qadir hussain@DESKTOP-1VNNIKF MINGW64 ~/20CSE35 (master)
$ git commit -m "My Intro"
[master (root-commit) 341b42b] My Intro
 1 file changed, 2 insertions(+)
 create mode 100644 20CSE35.txt

Engr Qadir hussain@DESKTOP-1VNNIKF MINGW64 ~/20CSE35 (master)
$ git remote add origin https://github.com/QadirHussain35/20CSE35.git

Engr Qadir hussain@DESKTOP-1VNNIKF MINGW64 ~/20CSE35 (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 294 bytes | 294.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/QadirHussain35/20CSE35/pull/new/master
remote:
To https://github.com/QadirHussain35/20CSE35.git
 * [new branch]      master -> master

Engr Qadir hussain@DESKTOP-1VNNIKF MINGW64 ~/20CSE35 (master)
$
```

This is the local file pushed in github.



master had recent pushes 33 seconds ago

QadirHussain35 Initial commit	d2008e4 · 1 minute ago	1 Commits
README.md	Initial commit	1 minute ago

20CSE35

C&DA

About: C&DA

No releases published

No packages published

Department of Computer Systems Engineering and Sciences
B.E (Computer Systems)
Semester: 7th
Course: Cloud and Distributed Computing
Course Code: CS-416
Credit Hours: 3+1
Lab Credit Hours: 01

Lab CLO: Demonstrate the basic service delivery models of cloud and distributed computing: Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a service (IaaS)
Bloom's Taxonomy Level: C3
Linked PLO: 05 (Modern Tool Usage)
Max Marks: 20

Marks Distribution

Semester Work: 10
Final Lab Examination: 10

Lab Examination Rubrics						
Attribute	**Attainment Extent (Marks)**					**Marks Obtained (out of 2)**
	0	**0.5**	**1**	**1.5**	**2**	
Software Installation and configuration (Knowledge about how to access Cloud /Distributed software)	Not able to recognize cloud/Distributed computing-based software	Knows a little bit about software but unable to access	Knows about software but cannot access it without detailed help	Knows about cloud based /Distribution computing software and able to access it with minor help	Able to recognize cloud based /distributed computing-based software and access it independently	
Use of Software Knowledge and environment setting (Knowledge about using Cloud based /distributed computing-based software and its environment)	Not able to work on cloud based /distributed computing-based software	Very limited work on Cloud based /distributed computing-based software with complete dependance on external help	Able to use Cloud based /distributed computing-based software with major external support	Able to use Cloud based /distributed computing-based software components with minor errors	Able to use Cloud based /distributed computing-based software properly with no errors	
Procedural Skills	Not able to follow procedure	Rarely follows	Follows procedure	Follows procedures	Follow correct procedure as	

(Procedural skills of students using programming codes for the lab/or algorithm given is assessed)	given to perform lab	procedure and mainly asks others	on large parts with some errors	as given with minimal errors	provided independently	
Accuracy of Completion of task (Student assessment of completing the given task as per requirement is assessed)	Completely wrong or fundamental level errors indicating student is completely unaware about completion of task	Very few things of task as per requirement correct	Some tasks completed correctly but major portion inaccurate	Major parts of task completely accurate with minor mistakes	Accurate completion of tasks with no errors	
Responding to questions about Lab work (Student will be assessed for having confidence while answering questions asked about cloud based/distributed computing-based software and given task)	Completely confused and not able to respond	Confused but able to respond a few questions	Middle level confidence and able to respond	Good confidence except minor level confusion and ability to respond	Completely confident and responding well to questions asked	
CLO Score in Lab Examination (Out of 10)						
Notes: 1. There shall be a 10 marks final examination on Rubrics 2. Students not getting 50% KPI shall be given additional chance to repeat and improve in CQI						