Lab 2.1:Deploy to GitHub via Git

This section will guide you to:

- Install Git and set up your GitHub account
- Execute the most popular commands in Git
- Push all the files from local repository to GitHub

Step 2.1.1 : Install Git.

Git is already installed in your lab. You can check the version of git by executing the below command in the terminal.

```
ubuntu@ip-172-31-16-137:~$ git --version
git version 2.7.4
ubuntu@ip-172-31-16-137:~$
```

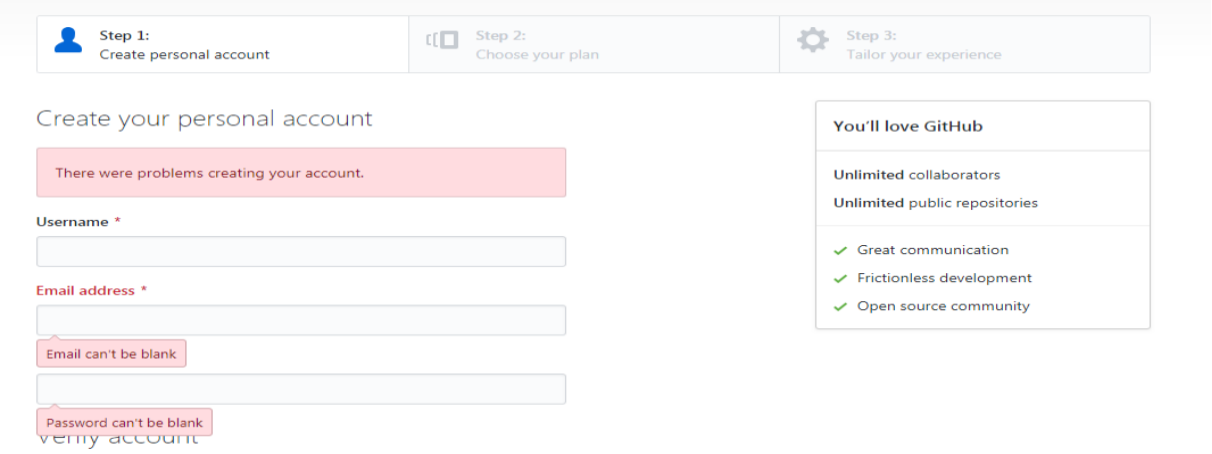
If git is not installed, then you can follow the below steps to install git

```
ubuntu@ip-172-31-16-137:~$ sudo apt-get install git
```

Step 2.1.2: Set up your GitHub account.

About GitHub: It is a web-based hosting service for version control using Git. It offers plans for public and private repositories. You can add multiple projects by creating multiple public repositories. In this section, you will only demonstrate on the public repository and its usage.

Navigate to https://github.com/ and click on **Sign up for GitHub**. Enter the details and click on **Create an account**.



In **Choose your personal plan**, Select **Free,** and click on **continue**. You can share basic information about yourself or you can **skip this step**.

You will receive an email to confirm your account. It is important to confirm your account before you use GitHub. Once confirmed, your GitHub account is set up successfully.

Step 2.1.3 : Login from Git local to connect remote GitHub.

Open the terminal in your lab and execute the below commands by replacing **your_Email_Id** with your registered email address in GitHub and **Your_Username** with your GitHub username.

```
ubuntu@ip-172-31-16-137:~$ #git config --global user.email "your_Email_Id"
ubuntu@ip-172-31-16-137:~$ #git config --global user.username "Your_Username"
```

Step 2.1.4 : Create multiple files and content in each file.

To create multiple files with different extensions and to create a folder to store all the files in one place, follow the steps shown below:

```
ubuntu@ip-172-31-16-137:~$ mkdir Lesson-02
ubuntu@ip-172-31-16-137:~$ cd Lesson-02/
ubuntu@ip-172-31-16-137:~/Lesson-02$ touch index.html Texts.txt C_Program.c HelloJava.java index.js styles.css typo.ts
ubuntu@ip-172-31-16-137:~/Lesson-02$ ls -l
total 0
-rw-rw-r-- 1 ubuntu ubuntu 0 Nov 20 05:43 C_Program.c
-rw-rw-r-- 1 ubuntu ubuntu 0 Nov 20 05:43 HelloJava.java
-rw-rw-r-- 1 ubuntu ubuntu 0 Nov 20 05:43 index.html
-rw-rw-r-- 1 ubuntu ubuntu 0 Nov 20 05:43 index.js
-rw-rw-r-- 1 ubuntu ubuntu 0 Nov 20 05:43 styles.css
-rw-rw-r-- 1 ubuntu ubuntu 0 Nov 20 05:43 Texts.txt
-rw-rw-r-- 1 ubuntu ubuntu 0 Nov 20 05:43 typo.ts
ubuntu@ip-172-31-16-137:~/Lesson-02$
```

You can use any of the text editors available in Linux, but prefer **vi** editor. To open the **vi** in editor mode, follow the below step. You can execute the same command for any file extension you have created in the above step.

```
ubuntu@ip-172-31-16-137:~/Lesson-02$ vi <Filename.Extension>
```

Execute the below command in the vi editor to save and return to the terminal.

Esc + [shift]:wq

Add the below code in **C_Program.c** file.

```
#include<stdio.h>
int main(){
printf("Hello! I am C-Program. Thank you!");
return 0;
}
```

Add the below code in **HelloJava.java** file.

```
class HelloJava {
public static void main(String args[]){
System.out.Println(" I am your Java Program. Thank you! ");
        }
}
```

Add the below code in **index.html** file.

```
<html>
<head><title>HTML</title></head>
<body>
        <p> I am your HTML Page. Thank you! </p>
</body>
</html>
```

Add the below code in index.js file.

```
var texts = “I am your JavaScript Program”;
console.log(texts);
```

Styles.css, **Texts.txt,** and **typo.ts** files will not contain any codes or statements.

Step 2.1.5 : Initialize Git.

Since all the files are to be pushed,initialize a .git folder inside the directory by executing the below commands.

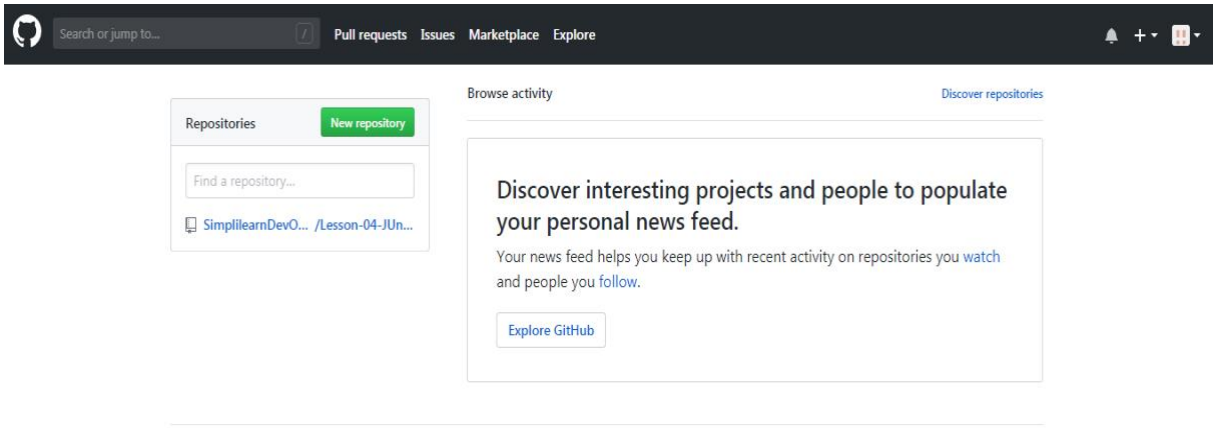
```
git init
git add .
git commit . -m "I am pushing all the files to my GitHub"
git status
```

Please follow the below process for step-by-step confirmation of each command execution.

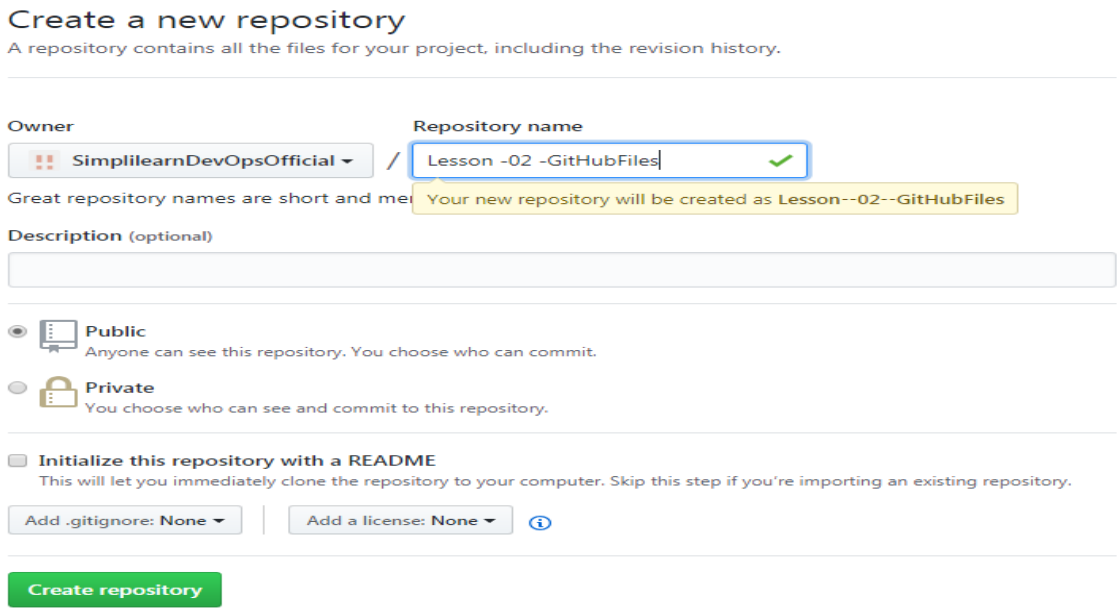
```
ubuntu@ip-172-31-16-137:~/Lesson-02$ git init
Initialized empty Git repository in /home/ubuntu/Lesson-02/.git/
ubuntu@ip-172-31-16-137:~/Lesson-02$ git add .
ubuntu@ip-172-31-16-137:~/Lesson-02$ git commit . -m "I am pushing all the files to my GitHub"
[master (root-commit) 707f62a] I am pushing all the files to my GitHub
 7 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 C_Program.c
 create mode 100644 HelloJava.java
 create mode 100644 Texts.txt
 create mode 100644 index.html
 create mode 100644 index.js
 create mode 100644 styles.css
 create mode 100644 typo.ts
ubuntu@ip-172-31-16-137:~/Lesson-02$ git status
On branch master
nothing to commit, working directory clean
ubuntu@ip-172-31-16-137:~/Lesson-02$
```

Step 2.1.6 : Create a repository in your GitHub account.

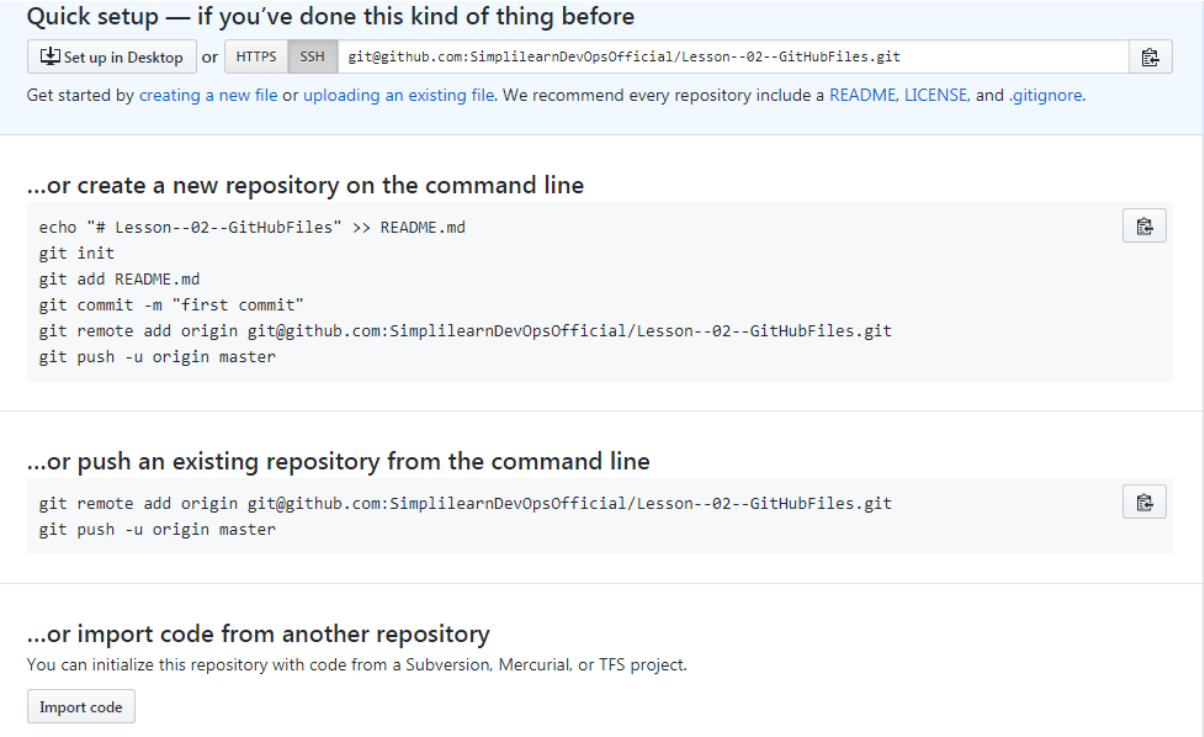
Go to the homepage of GitHub.com and click on **New Repository** as shown below.



Enter the name as "**Lesson -02- GitHubFiles**" and click on **Create repository**.



You will be redirected to a quick guide page and you will be navigated automatically inside the directory you have created.



Since a repository is already created, **"…or create a new repository on the command line**" should be skipped. Click on **SSH** to change the instructions from **HTTPS** to **SSH**.

Copy the git remote add origin <URL_of_Your_GitHub_Repository> and execute it in the terminal.

```
git remote add origin git@github.com:SimplilearnDevOpsOfficial/Lesson--02--GitHubFiles.git
git push -u origin master
```

****

If you're unable to push the files to your Github.com account, then follow the below steps:

Creation of SSH Key and adding it to GitHub.

Switch the current directory to ssh by executing below command.

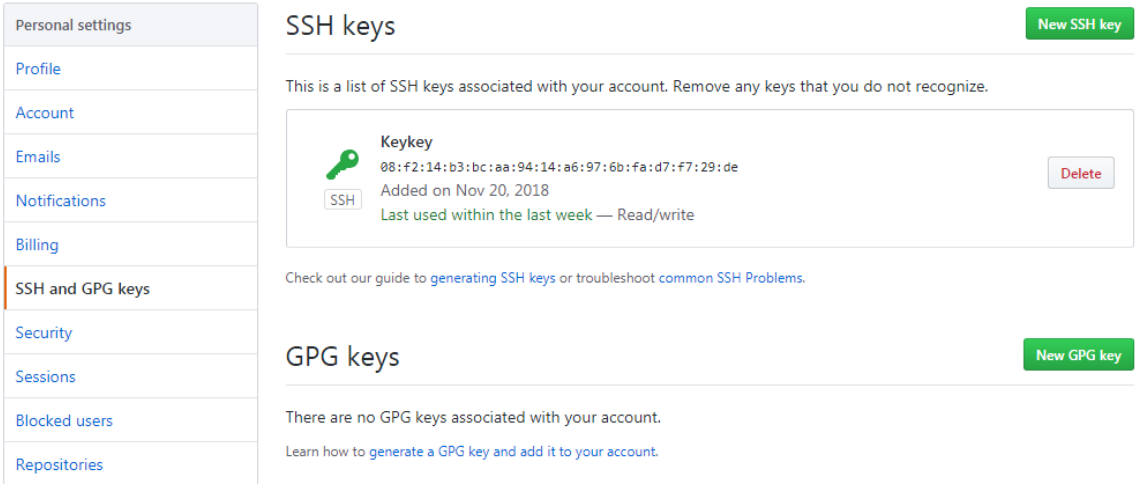
```
cd ~/.ssh
```

Generate an RSA key for the registered email Id. (An example is available below)

```
ssh-key gen -t rsa -C “rakesh.deshpande@simplilearn.net”
```

```
gedit id_rsa.pub
```

Copy the entire key from the clipboard. Choose ***Your avatar*** **> settings** > **SSH & GPG Keys** and click on **New SSH key** and paste the key and **save** it.



In the terminal, execute **ssh-add** to save the key and link it with local git.

Copy the git remote add origin <URL_of_Your_GitHub_Repository> and execute it in the terminal.

```
git remote add origin git@github.com:SimplilearnDevOpsOfficial/Lesson--02--GitHubFiles.git
git push -u origin master
```

****

Reload your GitHub.com account to confirm the output shown below.

1 commit | **1** branch | **0** releases | **1** contributor

Branch: **master** | **New pull request** | **Create new file** | **Upload files** | **Find file** | **Clone or download**

SimplilearnDevOpsOfficial I am pushing my files to GitHub — Latest commit `d2d1b49` 6 minutes ago

C_Program.c	I am pushing my files to GitHub	6 minutes ago
HelloJava.java	I am pushing my files to GitHub	6 minutes ago
Texts.txt	I am pushing my files to GitHub	6 minutes ago
index.html	I am pushing my files to GitHub	6 minutes ago
index.js	I am pushing my files to GitHub	6 minutes ago
styles.css	I am pushing my files to GitHub	6 minutes ago
typo.ts	I am pushing my files to GitHub	6 minutes ago

Help people interested in this repository understand your project by adding a README. **Add a README**