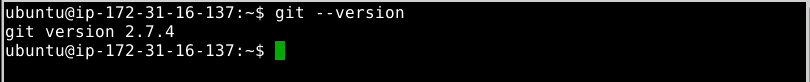
Lab 2.1:Deploy to GitHub via Git

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.

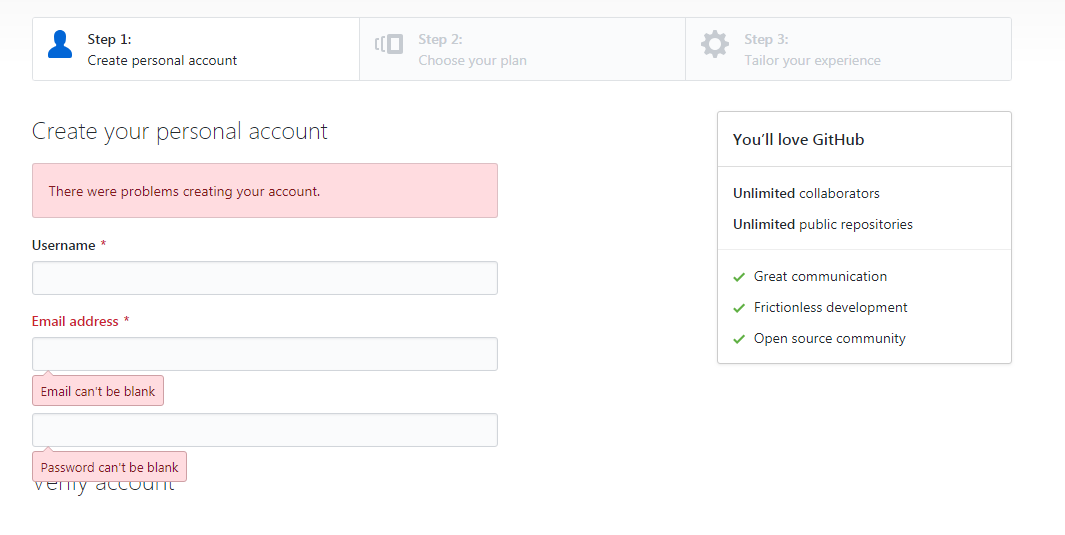


If git is not installed, then you can follow the below steps to install git

**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.

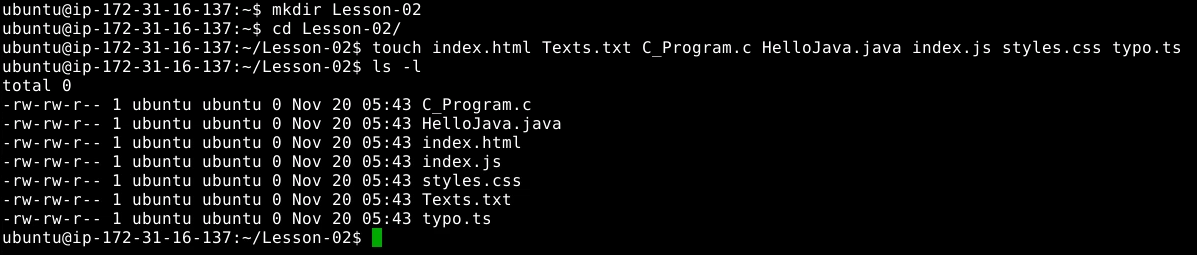
**Step 2.1.3 :** Login from Git local to connect remote GitHub.

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.



**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:



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.



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.

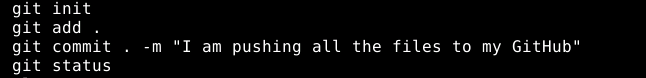
var texts = “I am your JavaScript Program”;

console.log(texts);

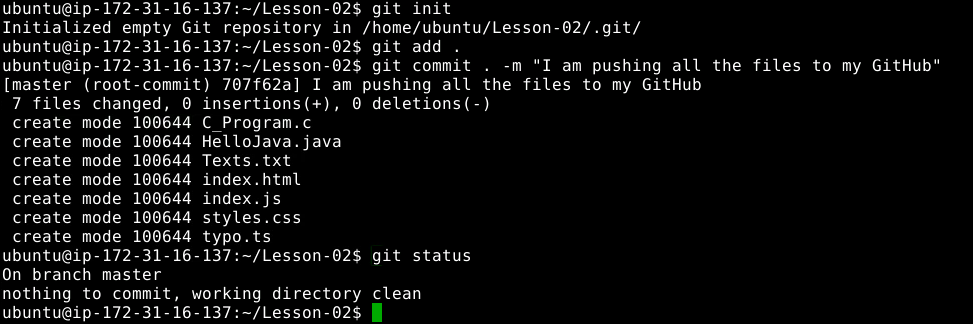
**Styles.css**, **Texts.txt,** and **typo.ts** files will not contain any codes or statements.

**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.

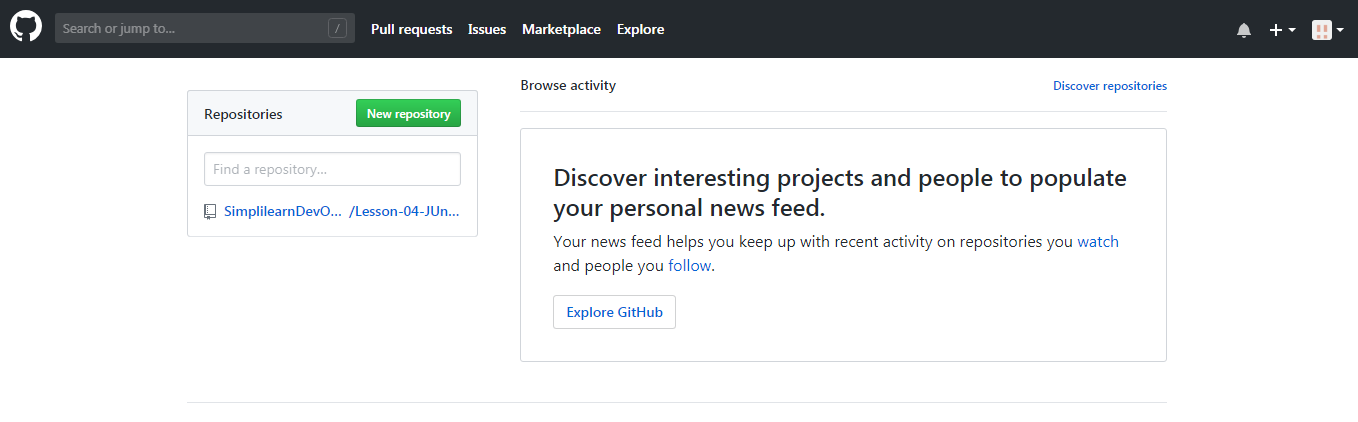


Please follow the below process for step-by-step confirmation of each command execution.

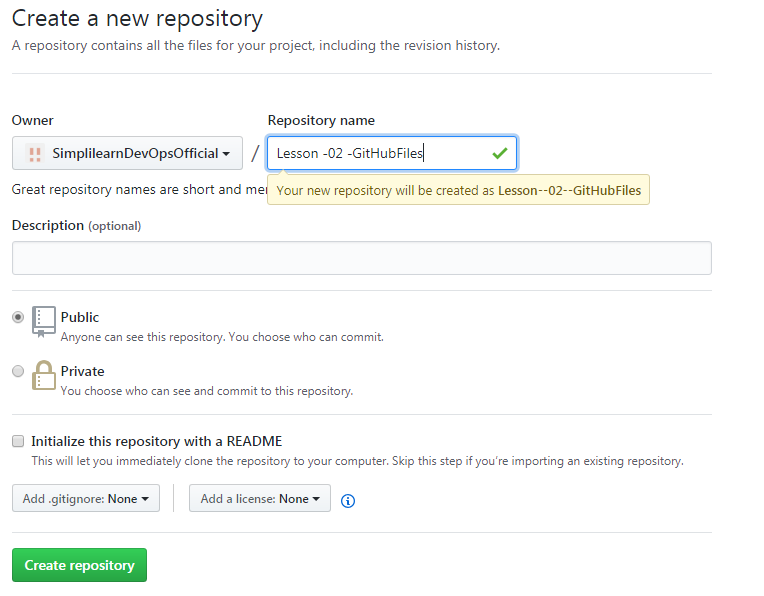


**Step 2.1.6 :** Create a repository in your GitHub account.

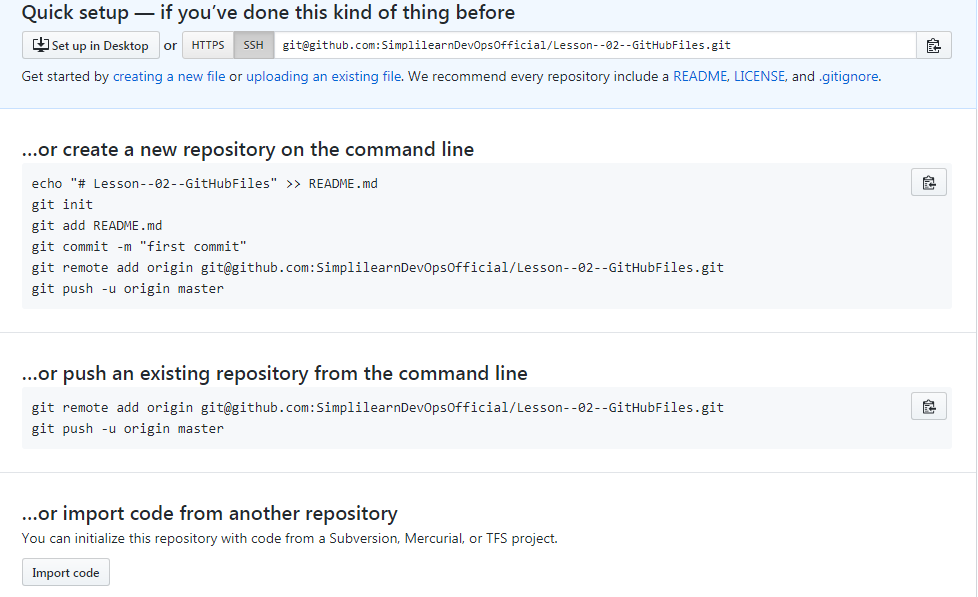
Go to the homepage of GitHub.com and click on **New Repository** as shown below.



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.

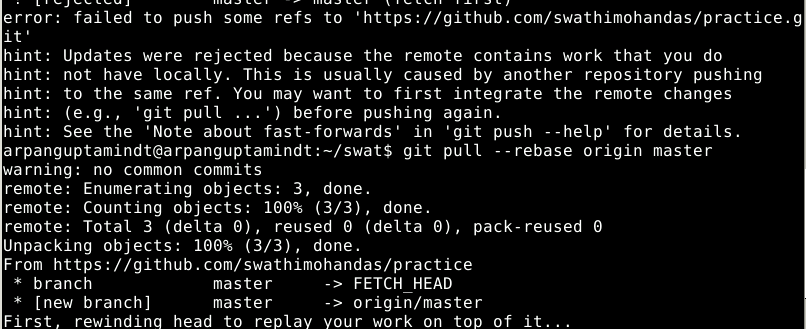


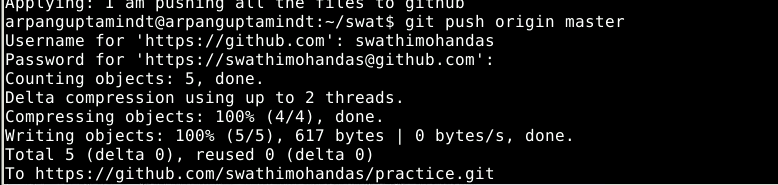
git remote add origin git@github.com:SimplilearnDevOpsOfficial/Lesson--02--GitHubFiles.git

git push -u origin master

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

If we give wrong credentials then we may face errors while doing it again for that we need to follow below steps.





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

**Creation of SSH Key and addiile doing ng it to GitHub.**

Switch the current directory to ssh by executing below command. 

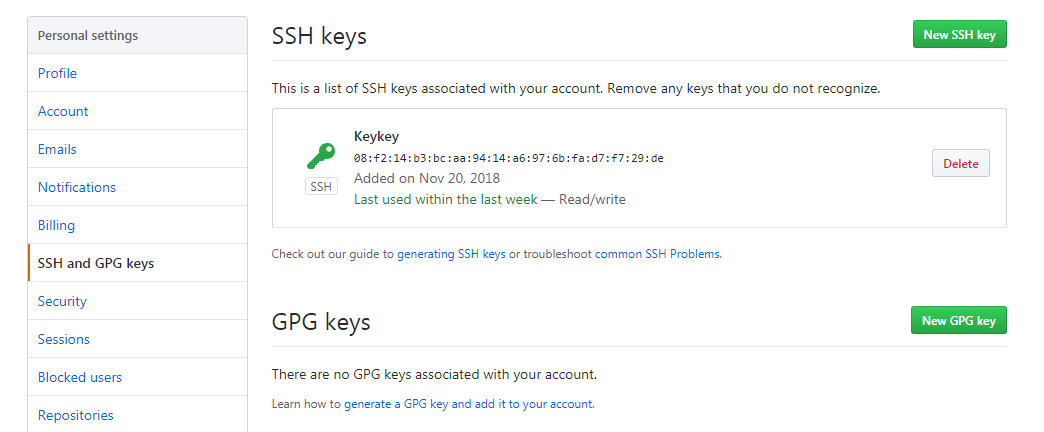
cd ~/.ssh

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.

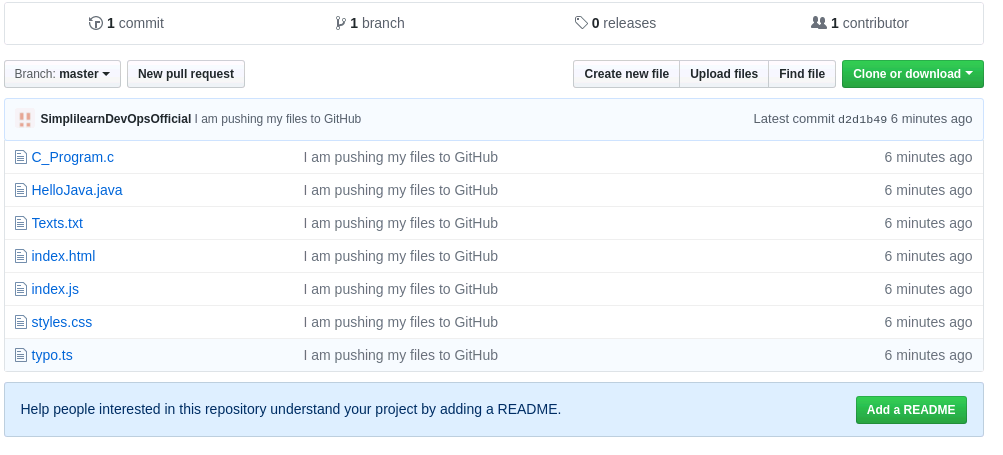


git remote add origin git@github.com:SimplilearnDevOpsOfficial/Lesson--02--GitHubFiles.git

git push -u origin master

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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

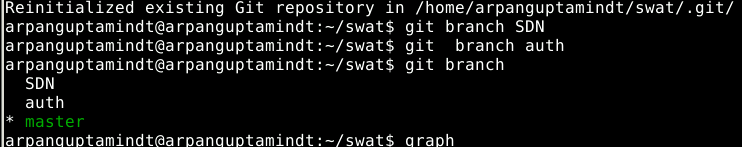


The By default branch is named as master branch.

Branches are created for various environements i.e for production for development for testing and for bug fixing.

For creation of the branch we will be using the git branch commands

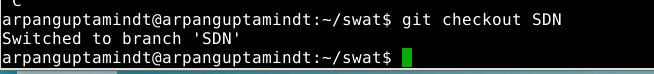
Head will define to which branch it is pointing towards.

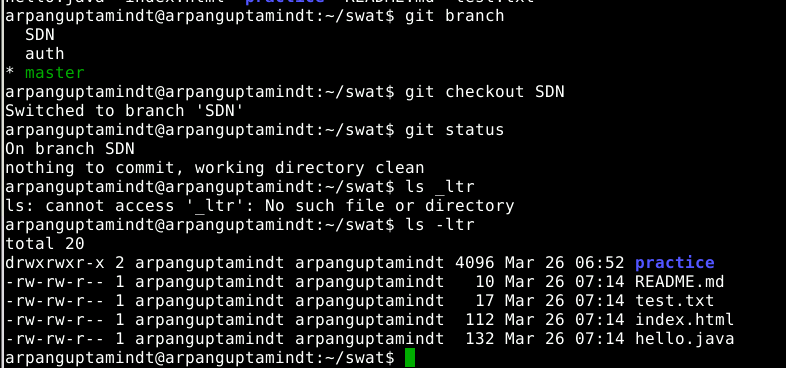


To point the head towards other branch we will do the following

Git checkout SDN

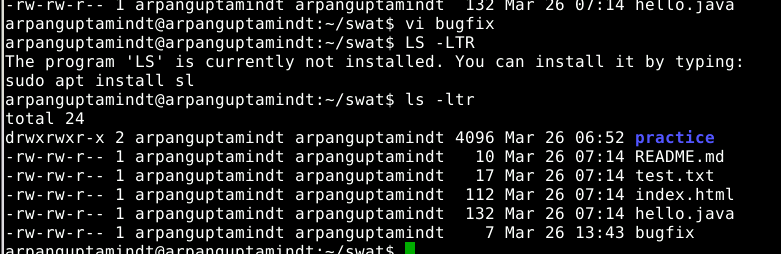
This will point head towards SDN



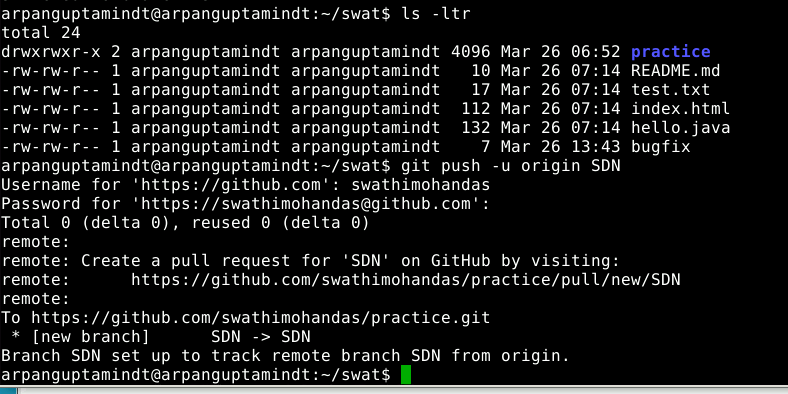


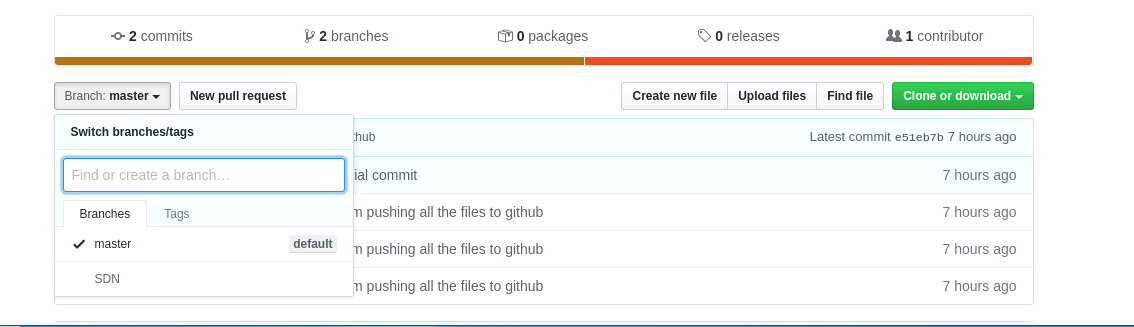
When we create the brach and checkout to that branch all the contents of the master branch will also move by default during the fisrt time

Now we will create one new file in the SDN branch as follows



Now to push the branch to the github repository

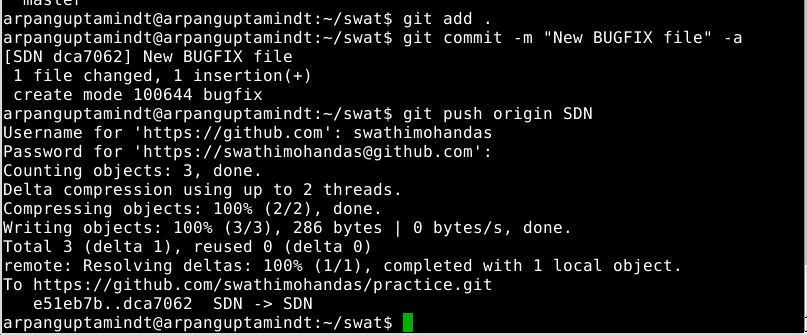


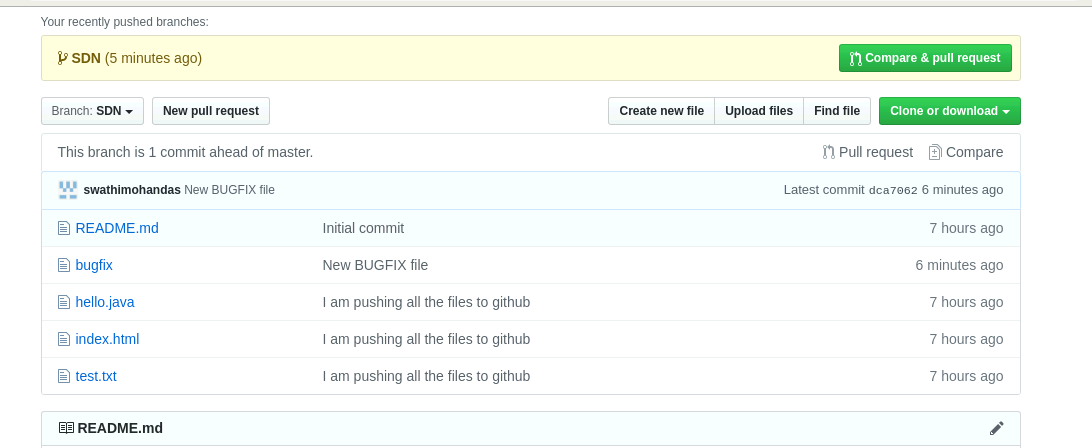


Now you could see the branch is getting reflected in the github respository as well.

But the same files will be refected in both the branch .

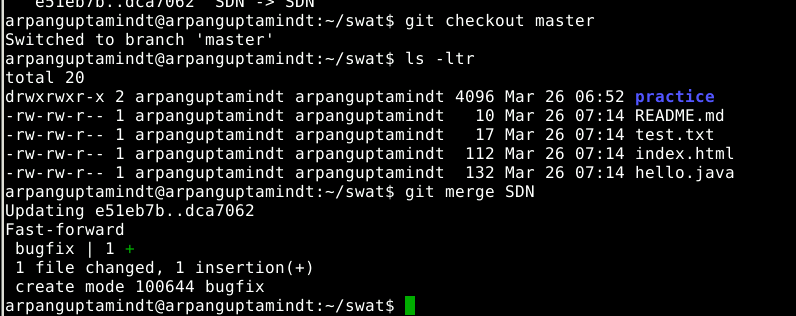
To push the new files created we need to exectute below commands.

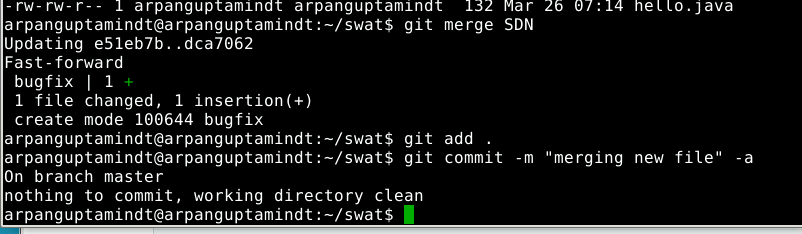


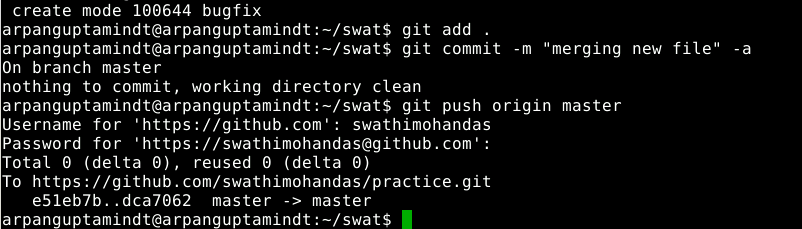


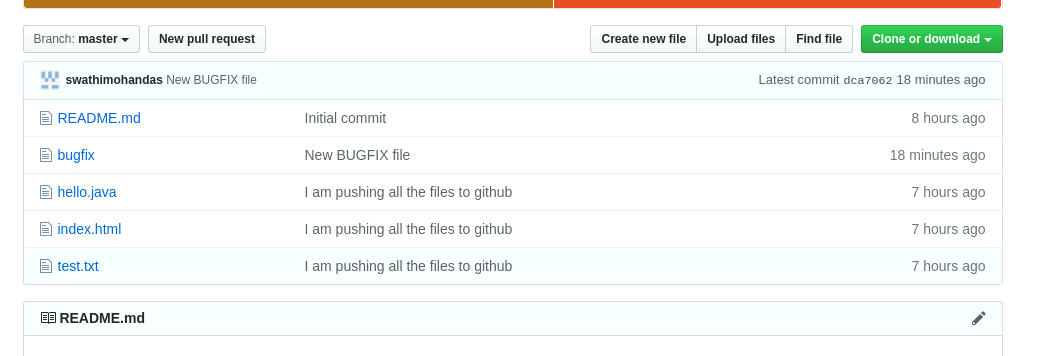
We could see the new files in this SDN branch

Now once the testing is done we will be marching the new file to the master branch









Now to delete the branch

Do the following

git branch –d SDN

then execute below command to check whether the branch is delete or not

git branch

To delete the branch from the github repository

git push origin –d SDN

How to get remote branch in our local

For example create a branch in the git repository.

