Exercise 1. GitHub Overview

Estimated time

01:00

Overview

This exercise covers

- The procedure to create a new GitHub account
- The procedure to setup Git
- The procedure to stage and commit changes to local and online repositories
- The procedure to work with Git branches
- The procedure to push a branch to GitHub
- The procedure to issue pull requests and merge branch with the master
- The procedure to clone an existing repository and push changes to a fork you created

Introduction

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. GitHub is a website and a cloud-based service which helps developers store and manage their code, as well as track and control changes to their code.

Author: Upendra Rajan (uperajan@us.ibm.com/linkedin.com/in/upenr)

Requirements

A laptop running Windows 10 with Internet access.

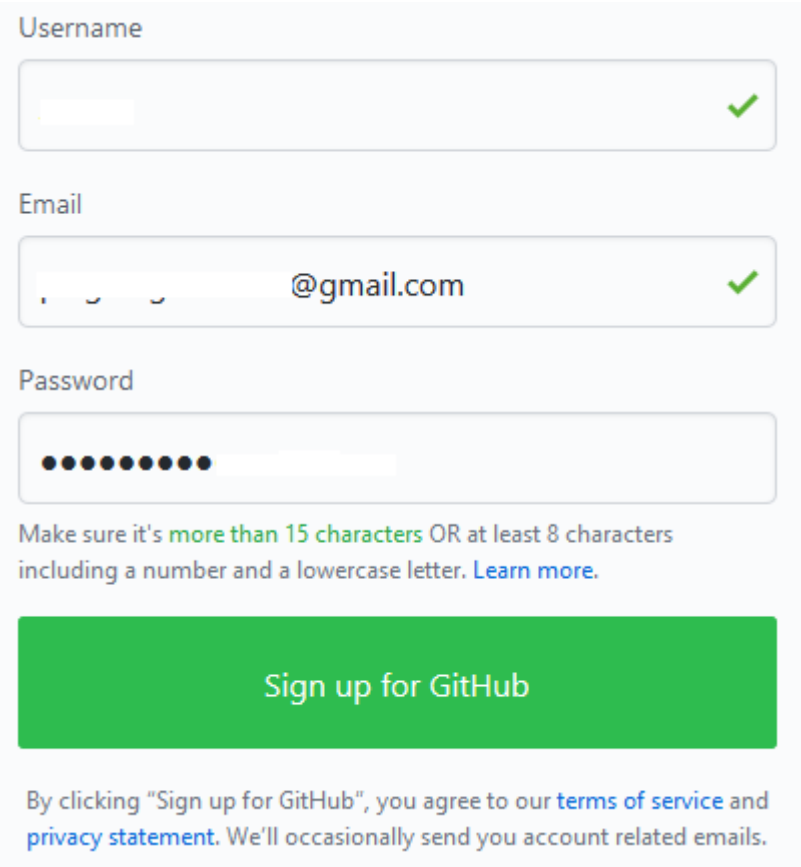
Exercise instructions

Section 1: Getting started with Git and GitHub

Information: Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. GitHub is a website and a cloud-based service which helps developers store and manage their code, as well as track and control changes to their code.

__ 1. Open the website https://github.com from Firefox.

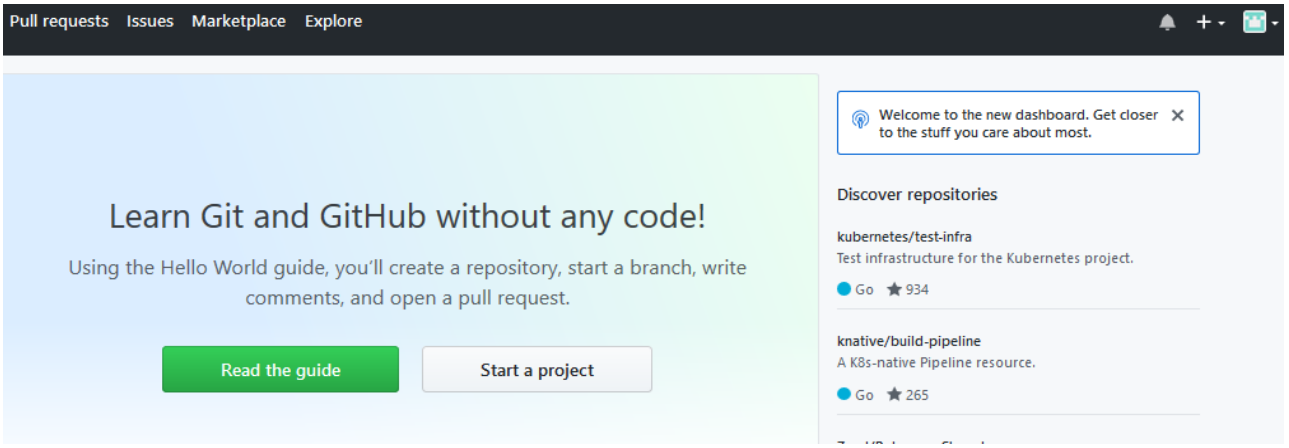
__ 2. Sign Up for a new account if you don't have one. If you have a verified account, skip to step 6.



__ 3. Verify your account by following the prompt and select the 'Free' plan. Click **Continue**.

__ 4. Answer the questions in the 'Step 3: Tailor your experience' section and click **Submit**.

__ 5. Ensure that you are logged in.



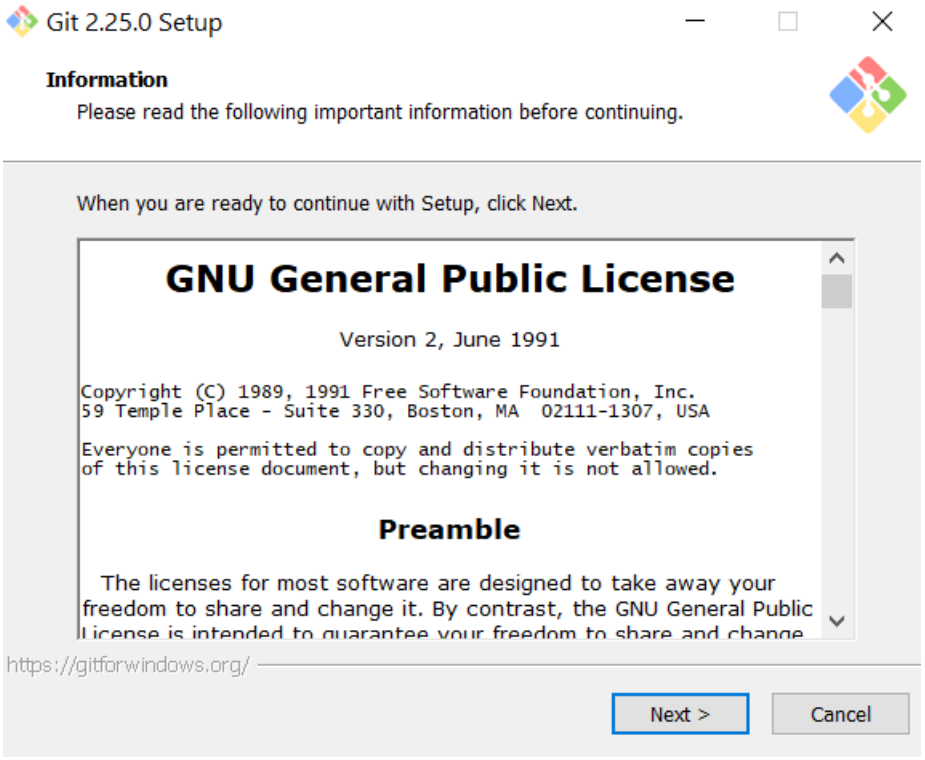
__ 6. Let's setup the Git client on your laptop. Get the installer using either of these two methods.

__ a. Get the installer '.exe' file from the link provided, or

__ b. From this link, find the right file for the operating system used in your access server: https://git-scm.com/downloads. Try to get version 2.25.0 of Git.

__ 7. Open the Git installer and click 'Yes' if prompted for permissions.

__ 8. Click Next at the welcome screen.

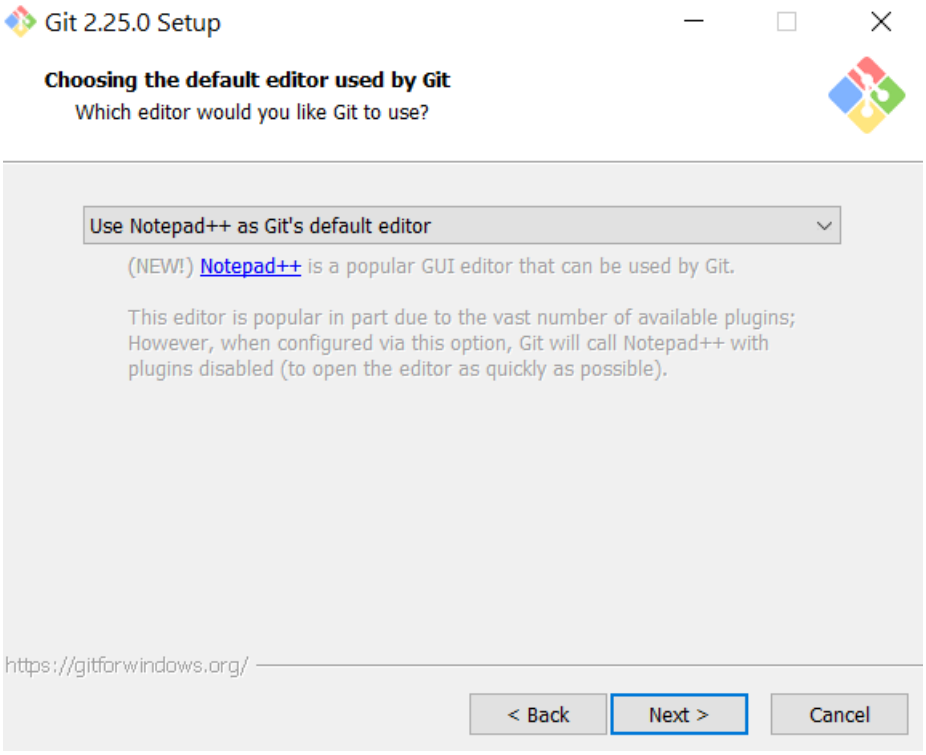


__ 9. Click Next to choose the default destination location.

__ 10. Click Next with the default components selected.

__ 11. Click Next with the default menu folder selected.

__ 12. From the drop-down menu, select ‘Use Notepad++ as Git’s default editor.’ If Notepad++ is not installed, click the link shown to download and install it. Click Next.



__ 13. Click Next with the option ‘Git from the command line and also from 3rd-party software’ selected.

__ 14. Click Next with ‘OpenSSL library’ selected.

__ 15. Click Next with ‘Checkout Windows-style, commit Unix-style endings’ option selected.

__ 16. Click Next with ‘Use MinTTY’ selected.

__ 17. Uncheck the ‘Enable Git Credential Manager’ option and click Install.

__ 18. Uncheck View Release Notes and click Next after the installation is complete.

__ 19. To execute the Git command from any directory, set the Windows path variable to these paths using the (Command Prompt) CLI.

```
setx path "%path%;C:/Program Files/Git/bin"
setx path "%path%;C:/Program Files/Git/cmd"
```

__ 20. Important: Close and open the command prompt window.

__ 21. Navigate to any other directory and issue the Chef command to verify the installation.

```
git --version
C:\Users\xxx>git --version
git version 2.20.1.windows.1
```

Section 2: Working with Git: Staging and committing

Information: This section explains some Git concepts and terminologies and shows how to use a few Git related commands.

__ 22. From the command prompt, navigate to Desktop and create a new directory and navigate to it. Here, **x** is your team number.

```
cd %userprofile%\desktop
mkdir gitsample-tx
cd gitsample-tx
```

__ 23. Create a new sample project and navigate to it. Here **x** is your team number.

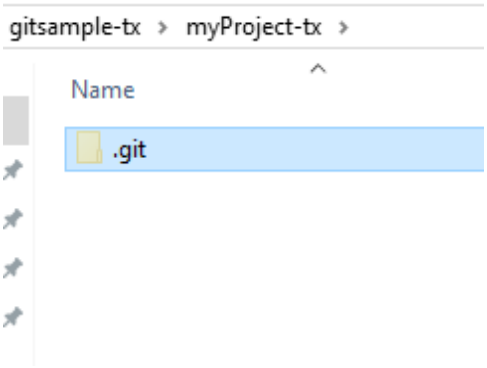
```
mkdir myProject-tx
cd myProject-tx
```

__ 24. Use the following command to initialize this as a Git repository.

```
git init
```

```
C:\Users\y580\Desktop\gitsample-tx\myProject-tx>git init
Initialized empty Git repository in
C:/Users/y580/Desktop/gitsample-tx/myProject-tx/.git/
```

__ 25. Navigate to your project directory and view the newly created '.git' directory. If this directory does not show up, it may be because it's hidden.



__ a. If you still want to verify it exists, open the Windows Command Prompt, navigate to the myProject-tx directory, and run the 'dir /aH' command.

__ 26. Right click and create a new text file in your 'myProject-tx' directory and call it **helloworld.txt**. Edit the content of this file and add a line such as "Hello world!", without the quotes.

__ 27. Git should be aware of this new fie. Enter the following command to view the status.

```
git status
```

```
C:\Users\y580\Desktop\gitsample-tx\myProject-tx>git status
On branch master
```

```
No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        helloworld.txt

nothing added to commit but untracked files present (use "git add" to track)
```

Information: The '**git commit**' command will record and save changes to the local repository. Once a change is made to your Git repository directory, you will have to add it to a staging area and then commit the changes. To add it to the staging area the '**git add**' command is used (as indicated in the output of the previous command).

The '**git add** .' adds all modified and new (untracked) files in the current directory and all subdirectories to the staging area (a.k.a. the index), thus preparing them to be included in the next **git** commit.

__ 28. Use the following command to add the new file to Git's staging area.

`git add helloworld.txt`

__ 29. Enter the following command to view the current status.

`git status`

```
C:\Users\y580\Desktop\gitsample-tx\myProject-tx>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   helloworld.txt
```

__ 30. Now that the file is indexed, let's commit our changes. The '-m' stands for a message to identify why this commit was made.

`git commit -m "Team X's first commit with the helloworld file"`

```
C:\Users\y580\Desktop\gitsample-tx\myProject-tx>git commit -m "Team X's first
commit with the helloworld file"
[master (root-commit) 173f32a] Team X's first commit with the helloworld file
 1 file changed, 1 insertion(+)
```

```
 create mode 100644 helloworld.txt
```

Section 3: Working with Git: Branching

Information: This section explains some Git branching concepts. You may want to create or try a new change to your code without changing the main project's code. To do this, you can branch the project to create your own branch and continue working on it.

Branches allow you to move back and forth between states of a project.

__ 31. Let's create a new branch for this project. ERROR: You may get an error in this step. To fix it, go to step 36 and come back to this step.

`git checkout -b myNewBranch`

```
C:\Users\y580\Desktop\gitsample-tx\myProject-tx>git checkout -b
myNewBranch
Switched to a new branch 'myNewBranch'
```

__ 32. Use the following command to view the branches of this project. The asterisk symbol next to the branch indicates which branch you are on right now.

`git branch`

```
C:\Users\y580\Desktop\gitsample-tx\myProject-tx>git branch
  master
* myNewBranch
```

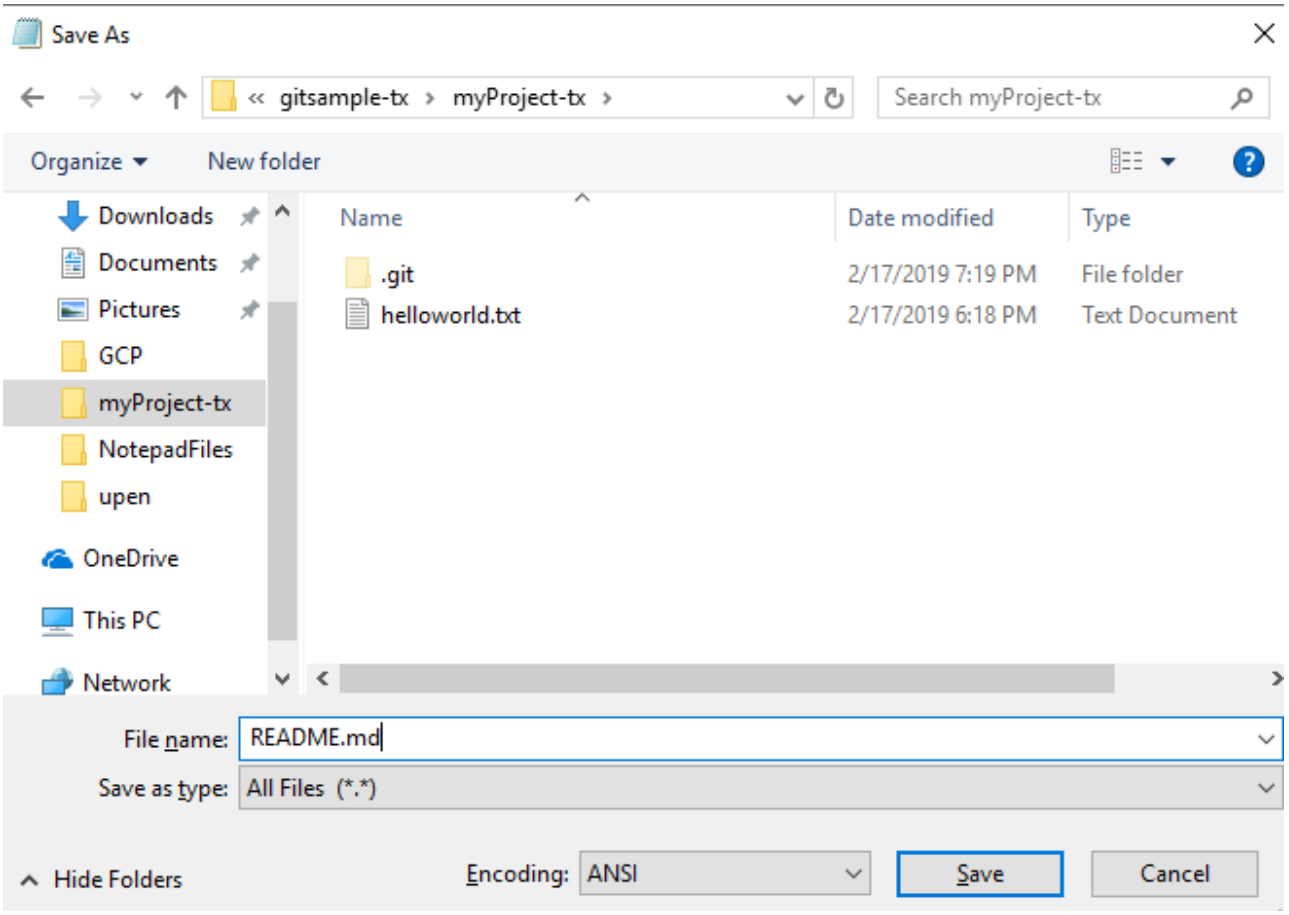
Information: You can switch between branches. If you switch the master branch and make commits there, the change will not be seen by this new branch and vice versa.

__ 33. To switch back to the master branch, use the following command.

`git checkout master`

```
C:\Users\y580\Desktop\gitsample-tx\myProject-tx>git checkout master
Switched to branch 'master'
```

__ 34. Open Notepad or Notepad++ and create a new file with the content "**This is the content visible on the front page of the GitHub repository.**". Save it as 'README.md' in the myProject-tx directory. (The MD suffix should be added as shown below with 'All Files' as the file type)



__ 35. From the command line, enter the following commands to stage (or index) the README.md file and commit your changes.

```
git add README.md
```

```
git commit -m "Commit after creating README.md"
```

__ 36. Let's register this Git client with the user you created earlier on GitHub. Use the following commands.

```
git config --global user.name "Your_Name"
```

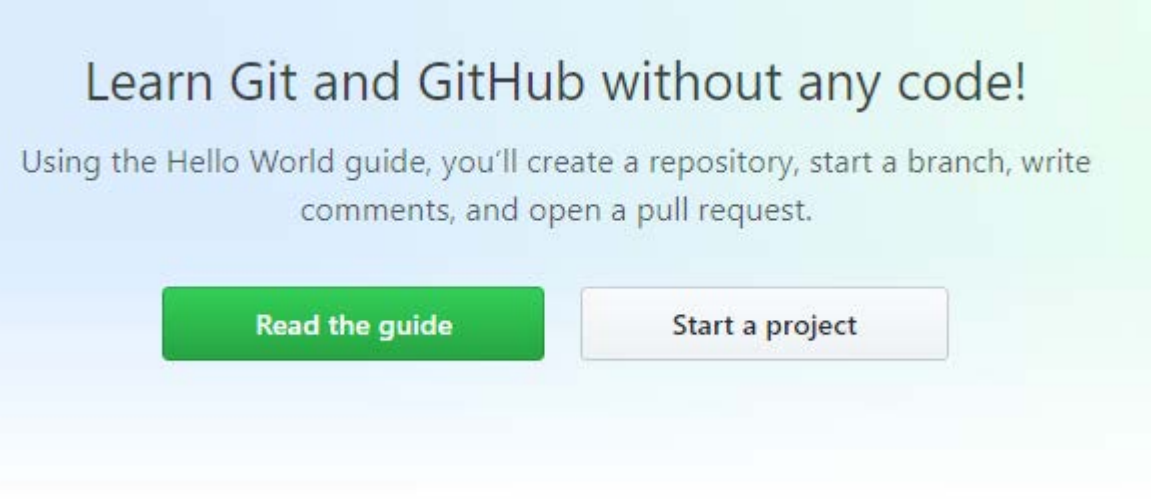
```
git config --global user.email "email_used_for_GitHub"
```

Section 4: Create remote repositories

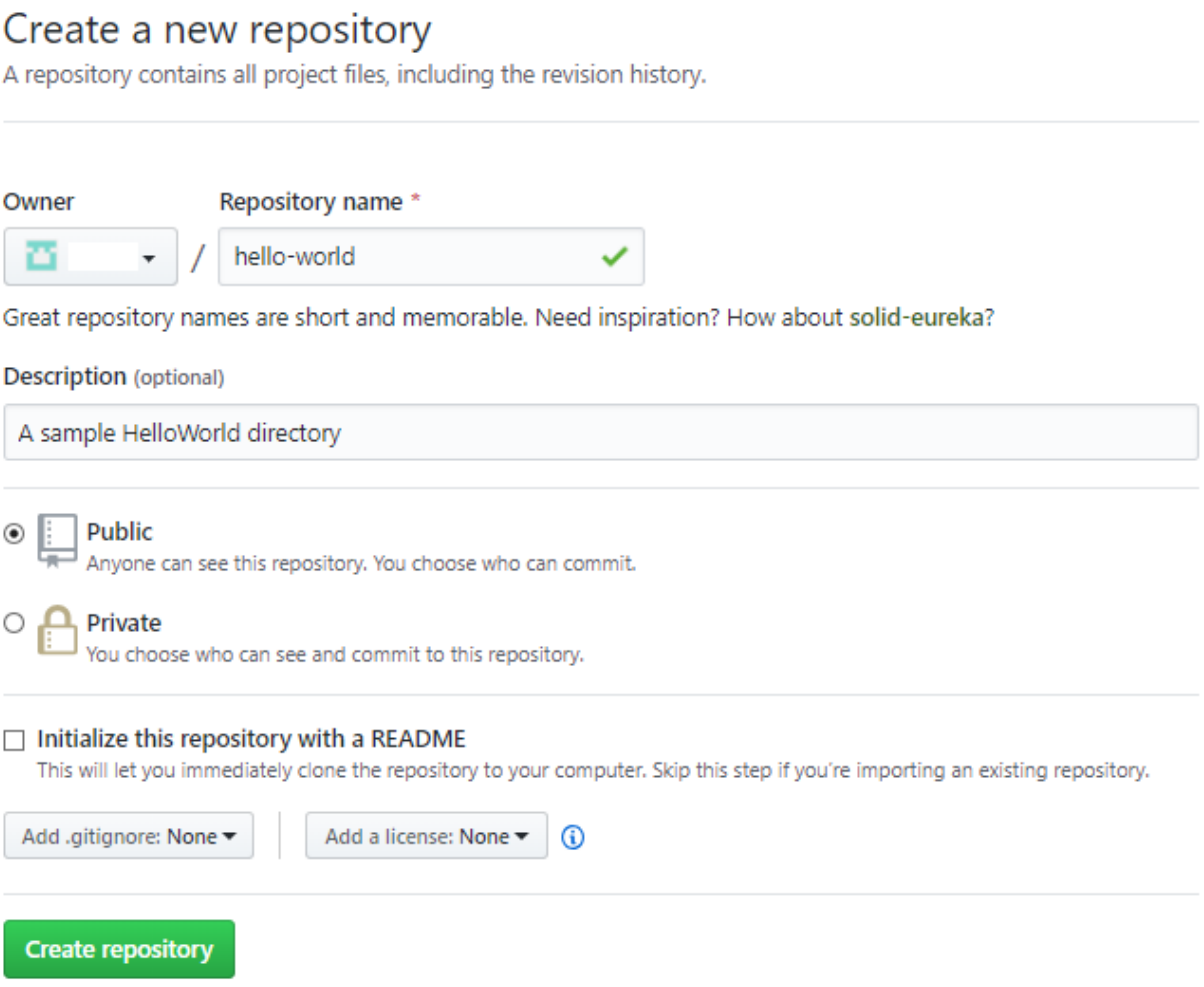
Information: This section explains how to register and make changes to a remote repository on GitHub.

__ 37. You will have to register your email account with GitHub before you can create an online repository or start a project. Log in to your email account, find the verification email and log in from the link in the email. This should verify your account.

__ 38. After your account is verified, click the 'Start a project' button.



__ 39. Enter the repository name as 'hello-world'. Make it a public repository. A sample screenshot is shown below.



__ 40. Click the 'Create repository' button.

__ 41. Go back to your command prompt and enter the following commands to push your existing repository to GitHub.

`git remote add origin https://github.com/<USERNAME>/hello-world.git`

`git push -u origin master`

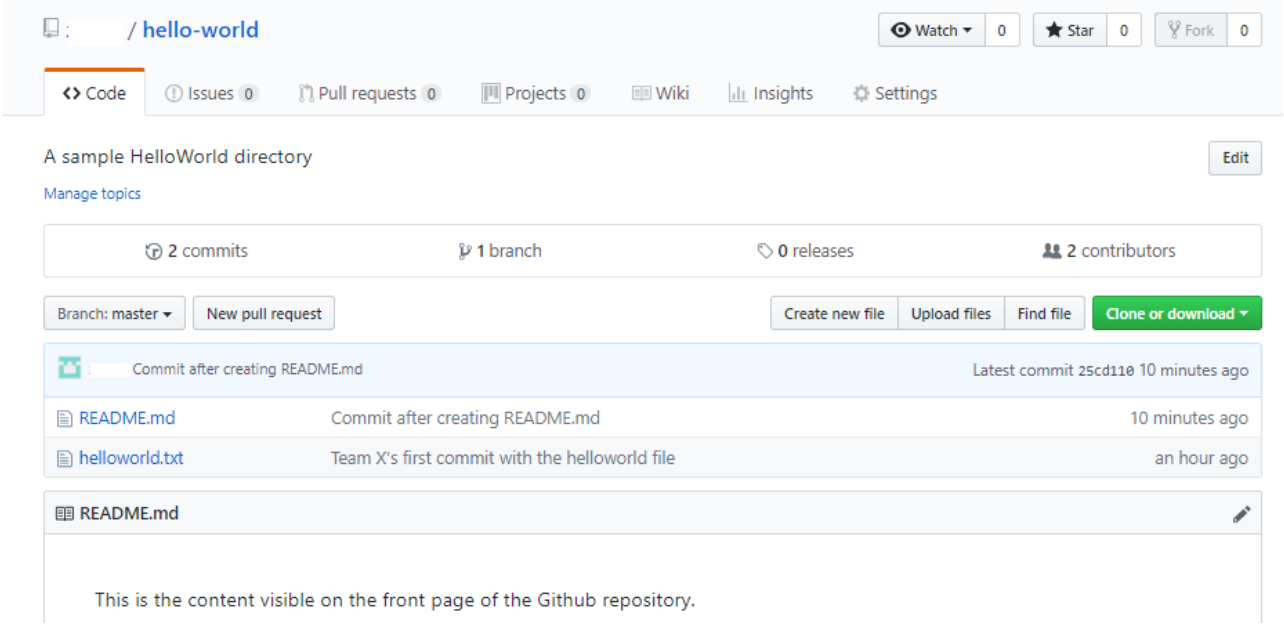
__ 42. Enter the username and password combination you used while creating this account when prompted. A sample output is shown below.

```
C:\Users\y580\Desktop\gitsample-tx\myProject-tx>git push -u origin master
Username for 'https://GitHub.com': username
Password for 'https://username@GitHub.com':
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 581 bytes | 145.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To https://GitHub.com/username/hello-world.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

Information: The 'origin' keyword can be thought as an alias for the remote repository's URL.

The ‘git push <remote> <branch>’ command will push the specified branch to <remote>, along with all of the necessary commits and internal objects. The '-u' paramter is used to indicate that for every branch that is up to date or successfully pushed, an upstream (tracking) reference is to be added.

__ 43. Refresh the GitHub repository page and you should be able to see the two files you created before.



Section 5: Push branch to GitHub

Information: This section explains how to push your branch to GitHub. It's very likely that you will write code in your own branch. You can push your branch to GitHub and it can then be merged with the master branch later.

__ 44. From the command prompt window, shift to the new branch you created before.

```
git checkout myNewBranch
```

__ 45. Go to the myProject-tx directory. The README.md file should not be there, as it was created only by the master branch.

__ 46. Shift to the master branch and make sure that the README.md file appears.

```
git checkout master
```

__ 47. You decide to start off with the same set of files the master has. Do the following to get the master files into the branch.

```
git checkout myNewBranch
git rebase master
```

```
C:\Users\y580\Desktop\gitsample-tx\myProject-tx>git checkout myNewBranch
Switched to branch 'myNewBranch'
C:\Users\y580\Desktop\gitsample-tx\myProject-tx>git rebase master
First, rewinding head to replay your work on top of it...
Fast-forwarded myNewBranch to master.
```

__ 48. We're now in our new branch. Open the HelloWorld.txt file using your text editor and change its contents to "Hello World. I have added a new feature through my branch." Save and exit the file.

__ 49. Verify that the change is identified by Git.

```
git status
```

```
C:\Users\y580\Desktop\gitsample-tx\myProject-tx>git status
On branch myNewBranch
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   helloworld.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

__ 50. Commit the changes. (The dot is used to add all the files in the current directory and the index.)

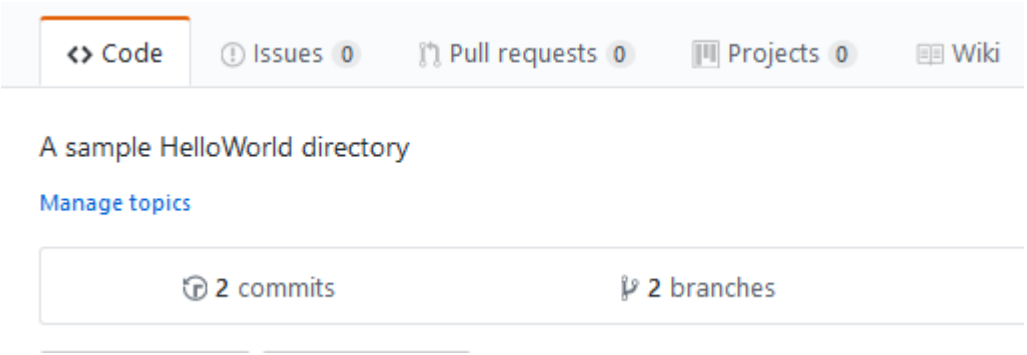
```
git add .
```

```
git commit -m "Commit after adding a new feature."
```

__ 51. Now let's push this new branch to GitHub. Enter the username and password combination when prompted.

```
git push origin myNewBranch
```

__ 52. Go back to your GitHub page and refresh. You should be able to see '2 branches' now.

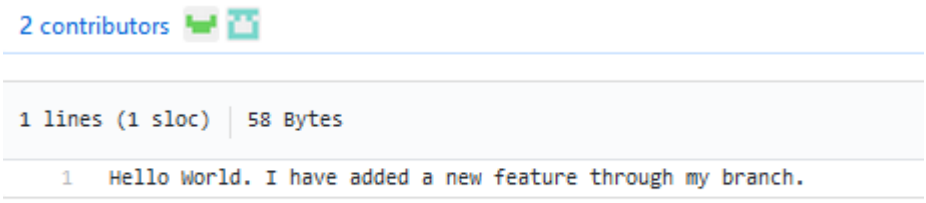


__ 53. Click the '2 branches' link and view the newly created branch.

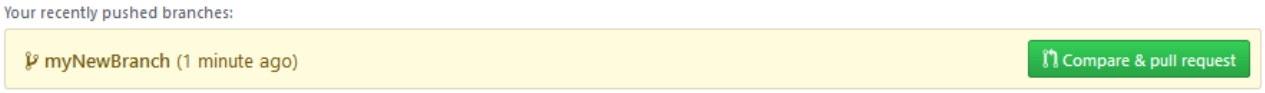
Section 6: Merge branch to master on GitHub and pull code

Information: This section explains how to issue a Pull Request and merge your branch to the master.

__ 54. Click the newest branch and view its files. Click the HelloWorld.txt file and view its contents.

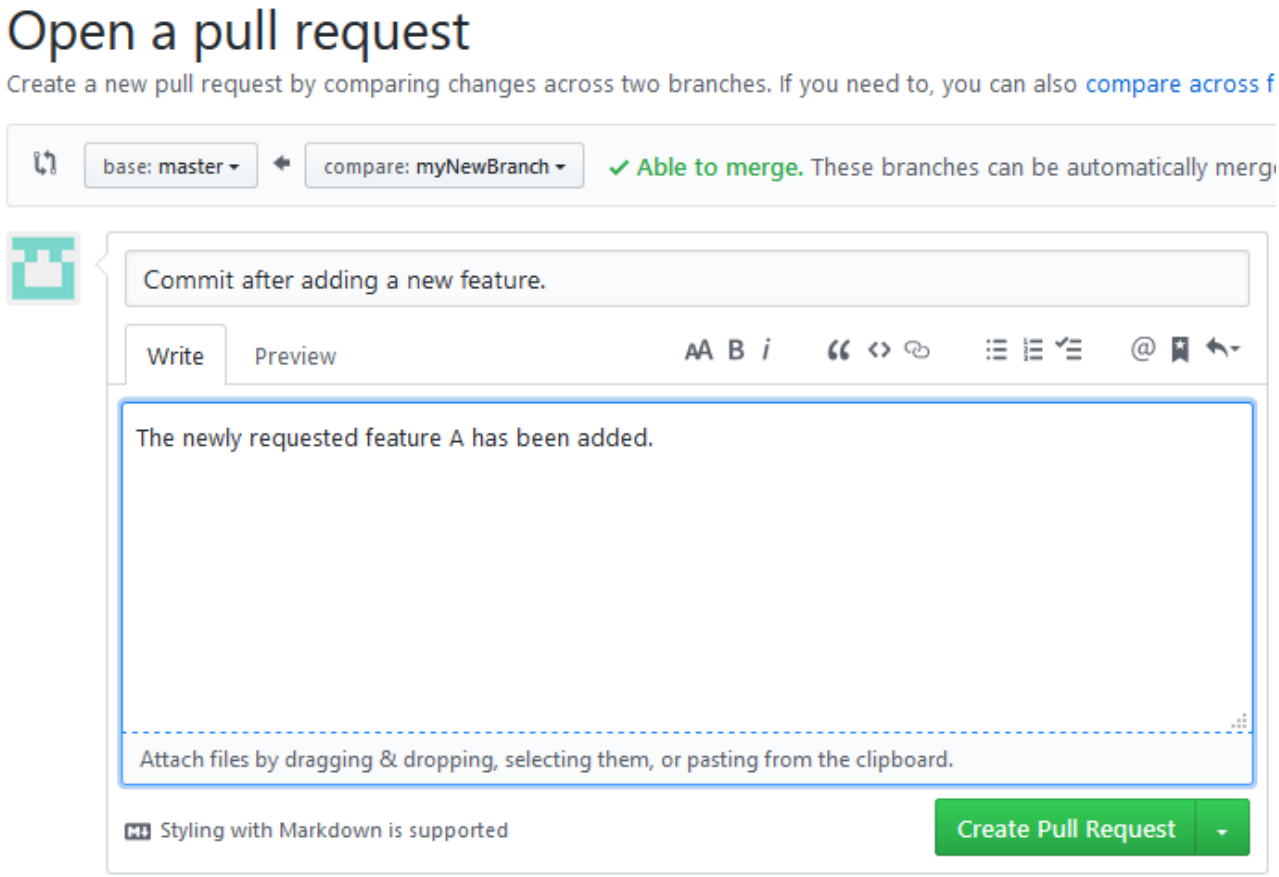


__ 55. Go back one level to the ‘myNewBranch’ branch, and click ‘Compare & pull request’.

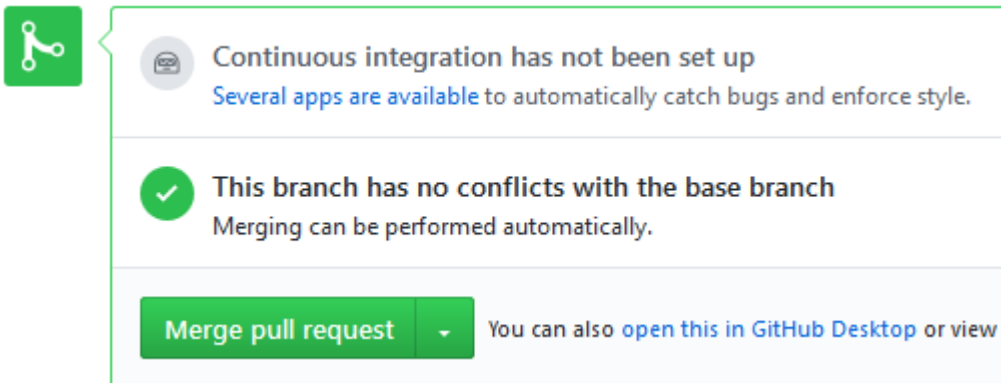


__ 56. Scroll down and view the changes made.

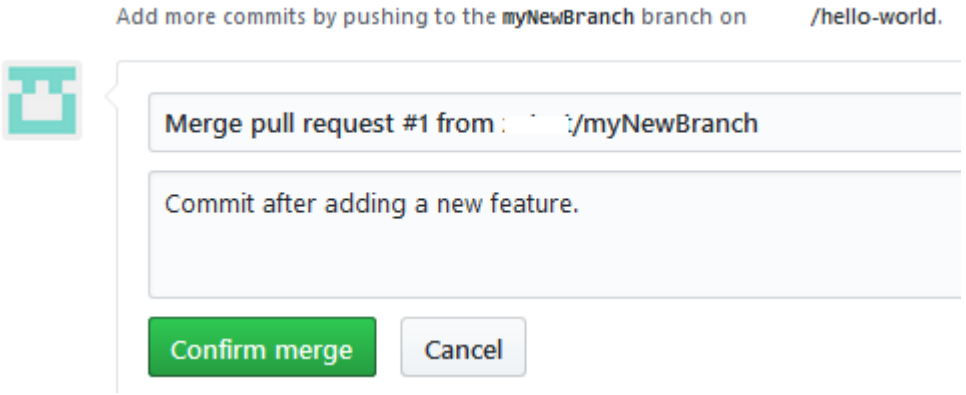
__ 57. Add a comment and click ‘Create Pull Request’.



__ 58. As you are the owner of this entire project, the request is visible to you. A green check indicates that there are no conflicts. If there are conflicts, the indicator would be grey and you will have to manually resolve conflicts.



__ 59. Click ‘Merge pull request’ to merge the branch with the master. Click ‘Confirm merge’ when prompted.



__ 60. You should get a “Pull request successfully merged and closed” message.

__ 61. Click the ‘hello-world’ link and go to the home page of your master code. View the contents of the HelloWorld.txt file and ensure that the change is available.

__ 62. The master on GitHub now, is not the same as the master on your local repository. From the command prompt window, enter the following command to pull the new master.

```
git checkout master
```

```
git pull origin master
```

__ 63. Let’s view a log of everything done so far. This shows all the commits, pull requests etc we have done so far.

```
git log
```

Section 7: (Optional) Clone from an existing repository and push to fork

Information: This section explains how to clone a new created by someone else on GitHub and create a fork.

__ 64. Navigate to the gitsample-tx directory.

```
cd ..
```

__ 65. From the gitsample-tx directory, enter the command to clone a remote repository from another developer.

```
git clone https://GitHub.com/upenr/twitter-ibm-watson-personality-insights.git
```

__ 66. Navigate to the newly cloned directory and view its contents.

```
cd twitter-ibm-watson-personality-insights
```

```
dir
```

```
C:\Users\y580\Desktop\gitsample-tx\twitter-ibm-watson-personality-insights>dir
 Volume in drive C has no label.
 Volume Serial Number is 540C-1B05

 Directory of C:\Users\y580\Desktop\gitsample-tx\twitter-ibm-watson-personality-insights

02/18/2020  06:09 PM    <DIR>          .
02/18/2020  06:09 PM    <DIR>          ..
02/18/2020  06:09 PM               127 .gitignore
02/18/2020  06:09 PM    <DIR>          .idea
02/18/2020  06:09 PM    <DIR>          app
02/18/2020  06:09 PM               561 build.gradle
02/18/2020  06:09 PM    <DIR>          gradle
02/18/2020  06:09 PM               747 gradle.properties
02/18/2020  06:09 PM             5,131 gradlew
02/18/2020  06:09 PM             2,404 gradlew.bat
02/18/2020  06:09 PM               689 README.md
02/18/2020  06:09 PM                16 settings.gradle
               7 File(s)          9,675 bytes
               5 Dir(s)  26,864,738,304 bytes free
```

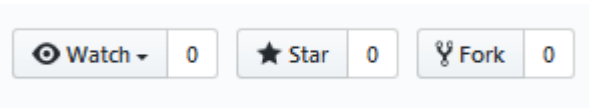
__ 67. Create a new branch from this master.

```
git checkout -b newBranch
```

__ 68. Go to the URL to create a new fork of the original code to your GitHub page.

https://github.com/upenr/twitter-ibm-watson-personality-insights

__ 69. On the top-right, click the Fork button.



__ 70. A fork should now be available to you.

Information: When you clone a repository, its URL is added as your 'origin'. If you want to make changes to your own fork, we will simply add your new fork as the 'origin' from now and push changes there.

__ 71. Let's assume you made changes and want to push this new branch to your fork on GitHub. From the command prompt, enter the username and password combination when prompted.

```
git remote set-url origin <URL_of_your_fork>
git push origin newBranch
```

__ 72. Refresh your fork and view the new branch that was just pushed.

End of exercise