Hands-on Lab: Using Git from Your Own Desktop

Estimated time needed: : 30 mins

Objectives

After completing this lab, you will be able to:

1. Clone your GitHub repository locally.
2. Make changes to the cloned files.
3. Add a new file.
4. Check the status.
5. Commit changes.
6. Generate Personal Access Token.
7. Push the changes back to GitHub.

Pre-requisites

GitHub account, with a project in it, as illustrated in this lab.

GitBash or Git installed on your local desktop, as in this lab.

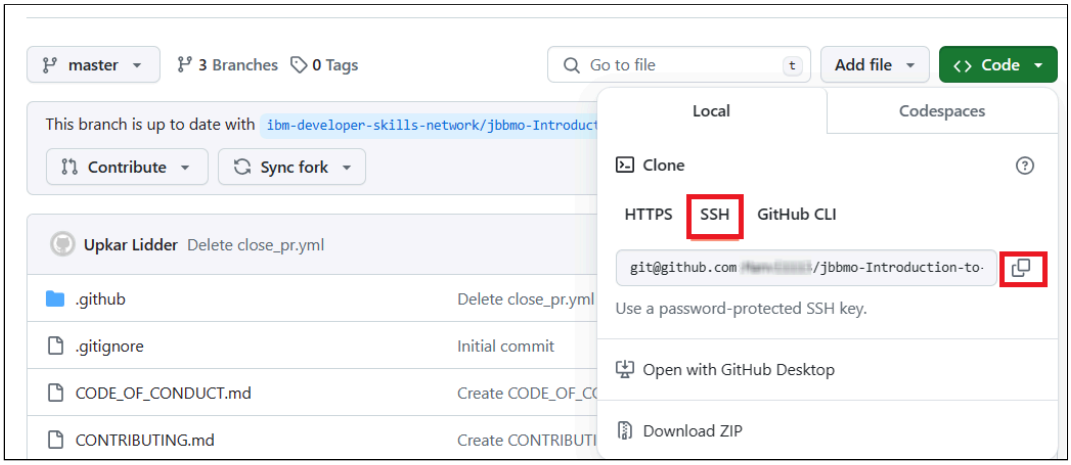
Create SSH keys, as in this lab

Add SSH Key to GitHub, as in this lab

Exercise 1: Clone a repo

To clone a repo, you need the SSH URL of the repo.

1. To get the SSH URL, login into GitHub.
2. Navigate to the repo you wish to clone.
3. Click the 'Code' button.
4. Click the 'clipboard icon' to copy the SSH URL. Paste this URL where you can access it later.



5. On your desktop, open a terminal or GitBash, if you are using Windows OS.
6. Navigate to a directory where you wish to clone the repo.
7. Run the command `git clone <your repo ssh url>`
8. This will clone the repo on GitHub into your current directory.
9. You can see all the downloaded files under a directory named as your repo name.

```
~
$ git clone git@github.com:[redacted]/vftvk-Simple-Interest-Calculator.git
Cloning into 'vftvk-Simple-Interest-Calculator'...
remote: Enumerating objects: 28, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 28 (delta 4), reused 6 (delta 2), pack-reused 16
Receiving objects: 100% (28/28), 9.82 KiB | 1.40 MiB/s, done.
Resolving deltas: 100% (5/5), done.
```

10. To ensure that every file was downloaded, navigate to the cloned directory and list the files.

```
$ cd vftvk-Simple-Interest-Calculator/
~/vftvk-Simple-Interest-Calculator (master)
$ ls
LICENSE  README.md  index.html  script.js  style.css
```

Exercise 2: Make changes to cloned files

1. Using your favourite editor, make the changes to the HTML file.

```
<!doctype html>
<html>
    <head>
        <title>Web App to compute Simple Interest</title>
    <script src="script.js"></script>
    <link rel="stylesheet" href="style.css">
    </head>

    <body>
        <h1>Simple Interest Calculator</h1>

        <input type="number"  id="principal"> Amount <br/>
        <input type="number"  id="rate"> Rate <br/>
        <input type="number"  id="years"> No. of Years <br/>
        Interest : <span id="result"></span><br>

"simple_interest_calc.html" 21L, 542C written
```

2. `git status` will show all the modified files.

```
(base) sr@rameshs-air simple_interest_calculator % ls
README.md                       simple_interest_calc.html
script.js                       style.css
(base) sr@rameshs-air simple_interest_calculator % git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   simple_interest_calc.html

no changes added to commit (use "git add" and/or "git commit -a")
(base) sr@rameshs-air simple_interest_calculator %
```

Exercise 3: Add a new file to the local repo

1. Let us add a new file to the local repo.
2. Using a text editor, create a new file **browser-support.txt**.

3. Add "Chrome, Firefox, Edge" into the file.

4. Save the file.

Exercise 4: Check the status

1. Run `git status` to see info on the modified files.

```
(base) sr@rameshs-air simple_interest_calculator % ls
README.md                       simple_interest_calc.html
script.js                       style.css
(base) sr@rameshs-air simple_interest_calculator % git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   simple_interest_calc.html

no changes added to commit (use "git add" and/or "git commit -a")
(base) sr@rameshs-air simple_interest_calculator %
```

2. Add the file to the repository for committing using `git add browser-support.txt`.

```
                                    ~/vftvk-Simple-Interest-Calculator (master)
$ git add browser-support
```

Exercise 5: Commit and push the changes

1. Git commit will record all the changes into the local stating area. To commit the changes you have made, run `git commit -m 'added a new file browser-support.txt'`.

```
$ git commit -m "added a new file browser-support"
[master 14bf790] added a new file browser-support
 1 file changed, 1 insertion(+)
 create mode 100644 browser-support
```

Now all the changes you have made this far, get committed locally.

2. The git `push` command will enable you to sync all the changes made locally to the GitHub web repository. Run the `git push` command in GitBash terminal.

```
(base) sr@rameshs-air simple_interest_calculator % git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 304 bytes | 304.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:rsannareddy/simple_interest_calculator.git
   2858800..bcf175b  master -> master
(base) sr@rameshs-air simple_interest_calculator %
```

You can now visit the GitHub repository page and check to ensure that the revised and newly added files are in place.

Summary

In this lab, you have learned how to clone a GitHub repository, make changes to it, commit the changes locally, and push it back to GitHub.

Author(s)

Ramesh Sannareddy

Other Contributor(s)

Rav Ahuja