# GitHub\_GettingStarted

This document contains notes pertaining to the Pluralsight course “GitHub: Getting Started”. The purpose of this document is to document the steps needed to start using Git and GitHub for a computer that has never been connected to GitHub, or for a user on a computer that needs a private connection.

Only a few of the clips of the Pluralsight course are documented. The names of those clips are copied to the section headers in this document.

## Demo: Setting up Your Environment

This clip contains instructions for **git** – user name, email address, editor used, . . .

## Foundations of Git

This clip describes a few of the commands needed to get started with using git in a new repository.

## Demo: Getting Started with Git

This clip illustrates setting up a new **git** repository. When I shadowed this course, I worked with the folder – on the Windows 10 computer

C:\Users\sncole\source\GitRepositories\VSAndOtherPlatformProjects\

JavaScript\_GettingStarted\web-dev-starter

The experience is slightly different from what is shown in Pluralsight, because the folder – at the outset – is not empty. After navigating – in GitBash – to this folder, the following are the commands that I entered.

$ git init creates the repository

$ ls -la shows the files in the folder; the subfolder .git is included

$ git status no commits yet

$ git add README.md added README.md (NAME IS CASE-SENSITIVE) - preparation for commit

$ git status response shows that README.md is staged

$ git commit –m “some comment” adds README.md to the local respository

$ git status nothing available for commit

$ git add . adds remaining files – status “staged”

$ git log newly staged files not mentioned; “log” shows only commits

$ git commit –am “new files added” adds several files to local repository

$ git log shows 2 commits

***The next 2 sections deal with creating SSH (Secure Shell) password and applying that SSH password for initializing a new project on a GitHub account. If you do not yet have a GitHub account refer to the Pluralsight clip “Demo Setting up Your GitHub Account”. Use Google to find the website where you can supply an email address and password for a new account.***

## Demo: Working with SSH

To find out if we already have an SSH key, from the root directory (users\sncole on my computers), type

$ ls / migrate to the root directory

$ ls –al ~/.ssh search for a folder named “.ssh”

On my computer the response includes

/c/Users/sncole/.ssh

id-rsa id\_rsa.pub known\_hosts

This tells me that the folder c:\Users\sncole contains a folder titled .ssh, and that .ssh contains three files id\_rsa, id\_rsa.pub, and known\_hosts. id\_rsa and id\_rsa.pub pertain to the SSH key; known\_hosts pertain to GitHub. This is not surprising, because - sometime in the past - I created an SSH key, and I used it to connect a repository to a GitHub account. But if the computer is new, you should see a response that indicates that .ssh is missing.

If .ssh exists, and if you know the shortcut password, then you can skip the remainder of this section. Otherwise, . . .

* $ ssh-keygen -t rsa -b 4096 -C "*programmer’s email address*" The response is instructions to supply a filename and its path to hold the key. SSH generates a public/private key pair, and it asks for the file where this information should be stored.
* Type the **enter** key to accept the default (/c/Users/sncole0/.ssh/id\_rsa). Next SSH asks you to type the passphrase.
* Type the passphrase carefully; do not expect any visual response from the individual keystrokes.
* After you have typed the passphrase for the 2nd time, SSH generates the key, and it stores the result in id-rsa and id\_rsa.pub.
* Verify that the agent is running by typing “$ eval $(ssh-agent –s)”. On my computer the response was “–s: No such file or directory”.
* I tried removing “-s” from the command, and the response changed to “Agent pid 1207”, which sounds like a reasonable positive response.

***I did not shadow the remainder of this clip. But perhaps I got around the missing steps when I created the repository.***

## Demo: Working with Repositories

Navigate to the GitHub web site, and log into the account that has been set up – where you want your **git** repository to reside. The landing page shows existing repositories in the left-hand panel. Our intent at this point is to create a new repository.

* Click the **New** command button near the top of this left-hand panel. GitHub navigates to page titled **Create a new repository**.
* Type a descriptive name in the **Repository name** text box. (I believe it should suggest the project or repository that you are setting up.)
* Type a description of the repository into the **Description (optional)** text bos.
* I recommend clicking the **Private** radio button. For business applications, one can (supposedly) specify users that are permitted to work with this repository.
* For simplicity I ignored adding README, .gitignore, and license. (I usually add .gitignore in the **git** repository on the local computer. This option suggests that it is probably a good idea to maintain .gitignore as one of the files in the **git** repository.)
* Click the **Create repository** command button. GitHub navigates to a page titled **Quick Setup – if you’ve done this kind of thing before**.

***The Quick Setup page and the instructions below are not displayed in the Pluralsight course. But perhaps the reason was that I had not copied the SSH key (as instructed by Pluralsight) immediately after key generation.***

* Make certain that SSH is selected (has a grey background).
* Follow the instructions under ..**or create a new repository on the command line**.
  + $ git branch –M main
  + $ git remote add origin *text\_from\_top\_of\_page*
  + $ git push –u origin main

*text\_from\_top\_of\_page* is the text to the right of the SSH button. You can copy this text into the clipboard by clicking the icon at the right edge of this text box. (If you are using GitBash, you can insert this clipboard text by typing the **Insert** key on the keyboard. GitHub will ask for the SSH shortcut code.

* Type the shortcut code. GitHub will display a confirmation that the committed files have been uploaded from your computer.
* Back at GitHub – click the tab **< > Code**. GitHub will display the files in the newly created repository.