

DevOps Activity 1 – Part 1

Created by Steve Robinson 13-April-2018

This version of the document is to demonstrate the part 1 of a complete process that will demonstrate building out a simple continuous integration pipeline that involves everyone in the class as a team.

DevOps Activity 1 – Part 1.....	1
Overview	1
Activity 1 -Part 1.....	1
What is Git.....	1
Prepare for Git	1
Install Notepad ++.....	1
Install Git on Windows.....	6
Verify Git installation	13
Next Steps	15
Cloning the Activity – Part 1 Repo from GitHub	15
Cloning a Repo	17

Overview

Activity 1 -Part 1

In Part 1, you are going to achieve the following:

- Install Notepad++ & Git locally on your Windows Desktop.
- Use Git to Clone a remote repo which contains further instructions/guides and scripts

Please progress through this guide, following steps as required.

What is Git

Git is a source control management system.

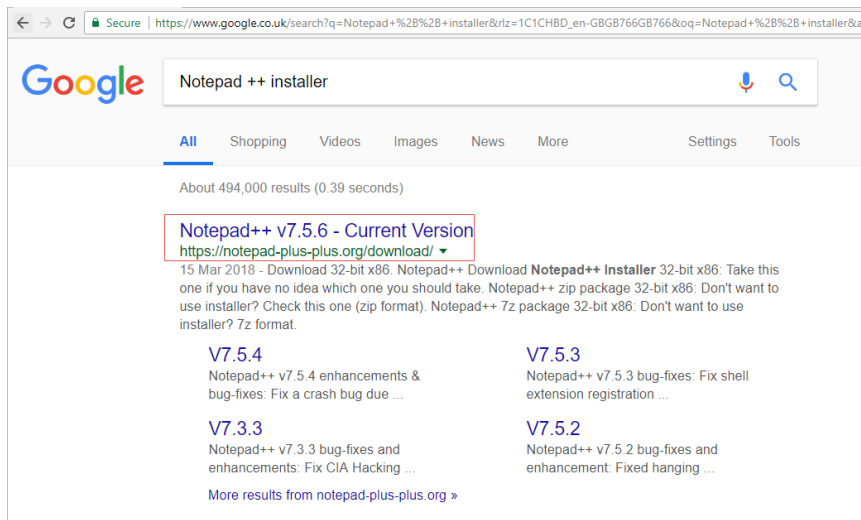
GitHub is an online managed Git Repo Service (SaaS)

Prepare for Git

To make it easier to edit files, we will install a better version of Windows Notepad, the program is called **Notepad ++**, we can then as part of installing **Git** enable Notepad++ as our preferred editor.

Install Notepad ++

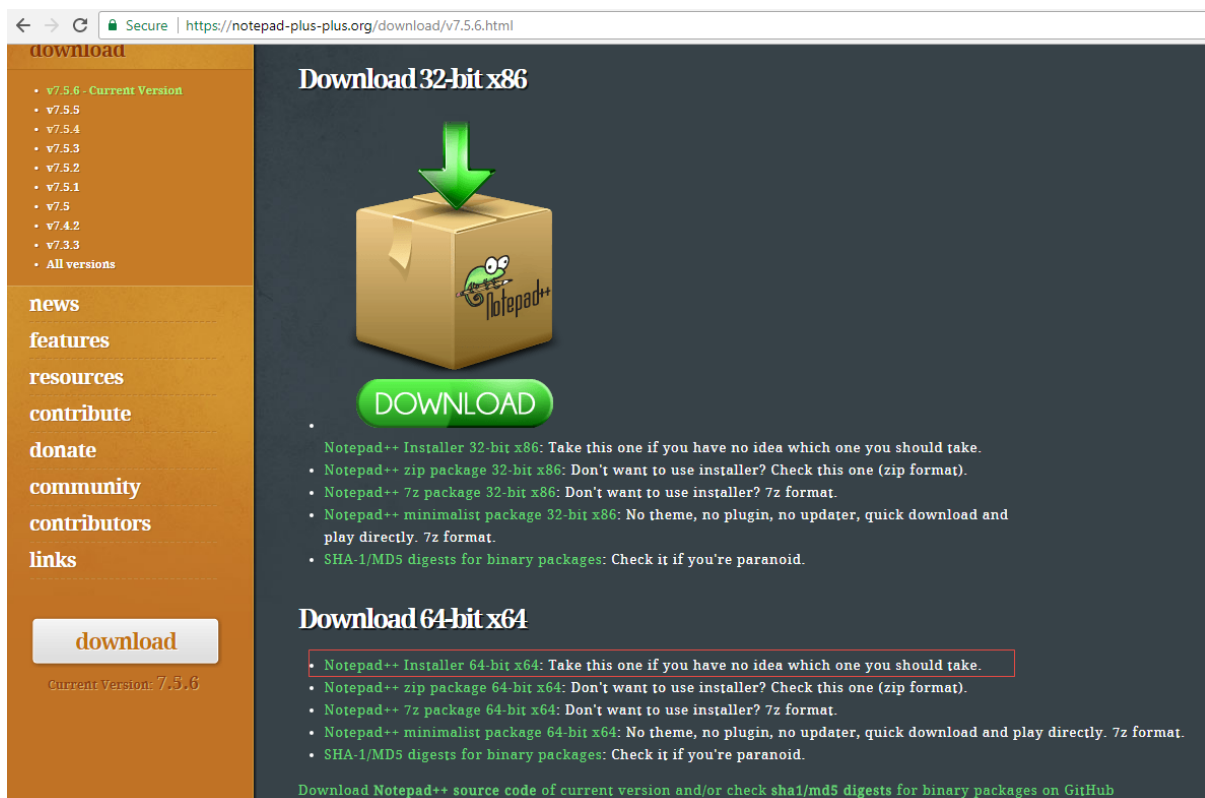
Do a Google search for **Notepad ++ installer**, and install the program



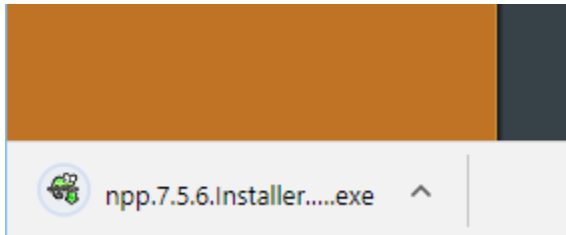
Locate and click the URL to get to the download page, you should end up on a page something like this:

<https://notepad-plus-plus.org/download/v7.5.6.html>

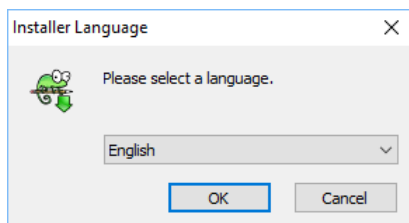
Locate the **64bit version**, and download, and install.



Look for the download, and Run/Open it



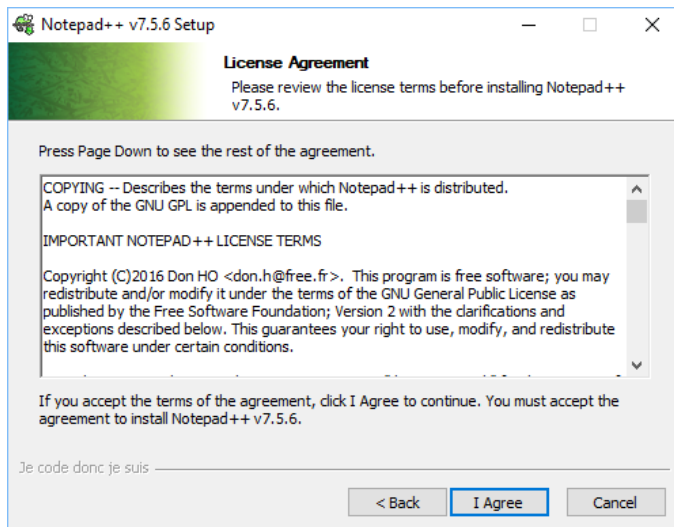
Accept any Windows elevated permission prompts (Windows requires Administrator Privileges) and continue.



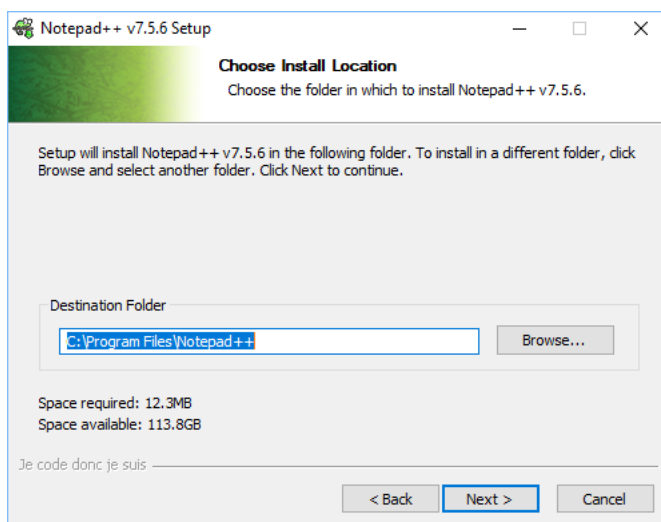
Select **English**, and click **OK**



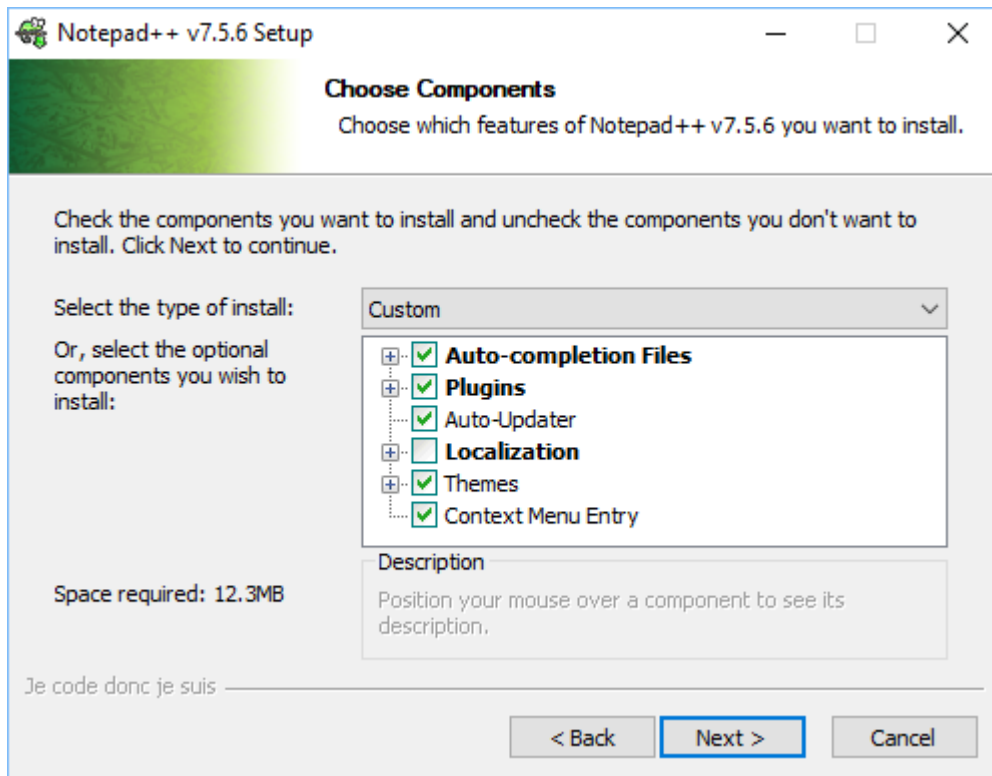
Click **Next**



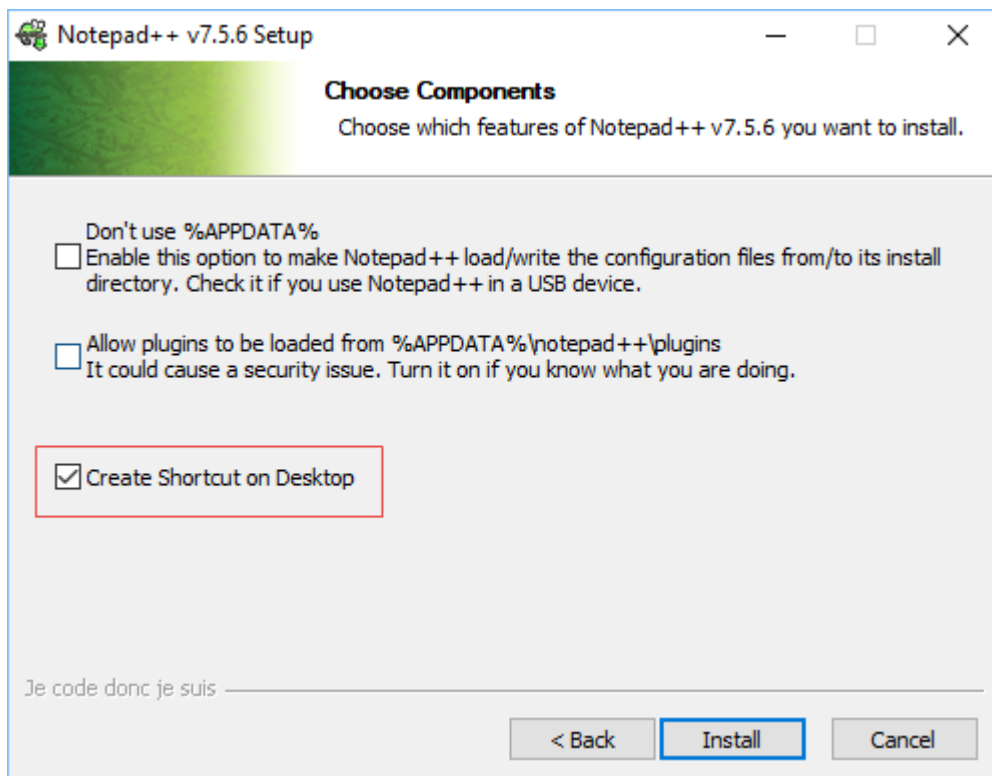
Click **I Agree** to continue



Leave default installation path, and click **Next**

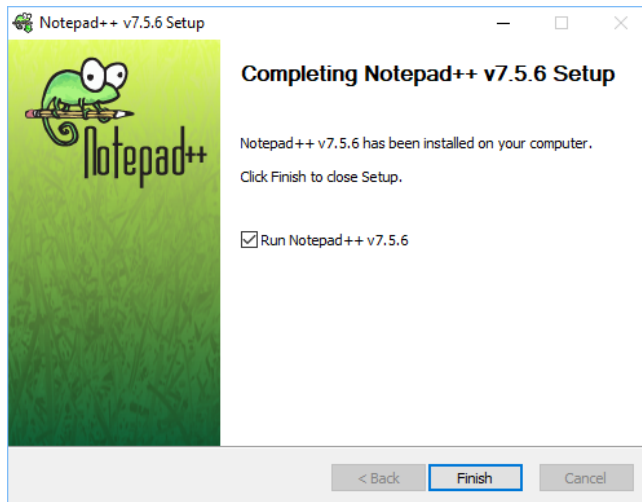


On the **Choose Components** screen, leave defaults and click **Next**

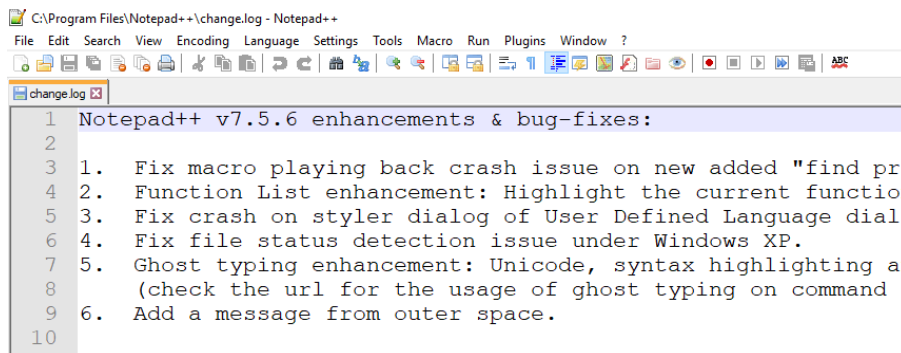


On the following screen, ensure that you check the **Create Shortcut on Desktop** to make life easier in the future i.e. locating Notepad++ in a hurry.

Click **Install**



Click **Finish**, and Notepad++ will launch because the **Run Notepad ++ v7.5.6** option is checked, this will allow you to ensure it has been installed.



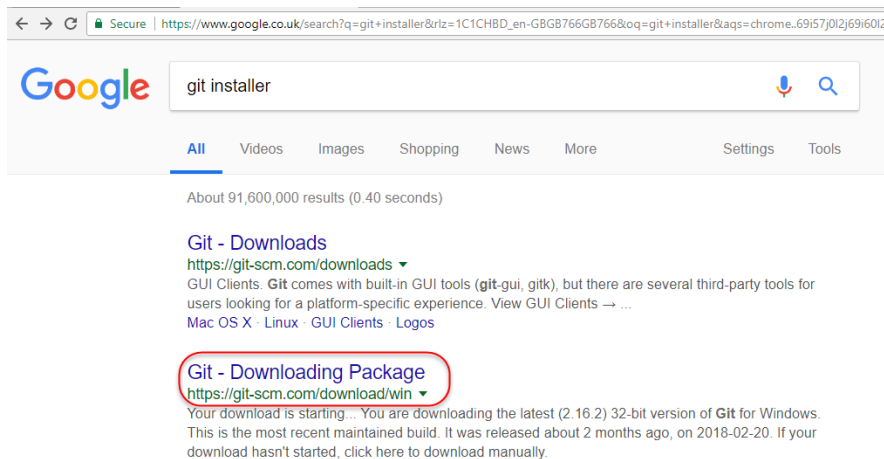
Then close notepad ++, using **File > Exit**

We have now installed an additional text editor, which be very useful as DevOps Engineers.

We will now move onto installing Git

Install Git on Windows

Use Google to locate and search for the latest Git for Windows installer.



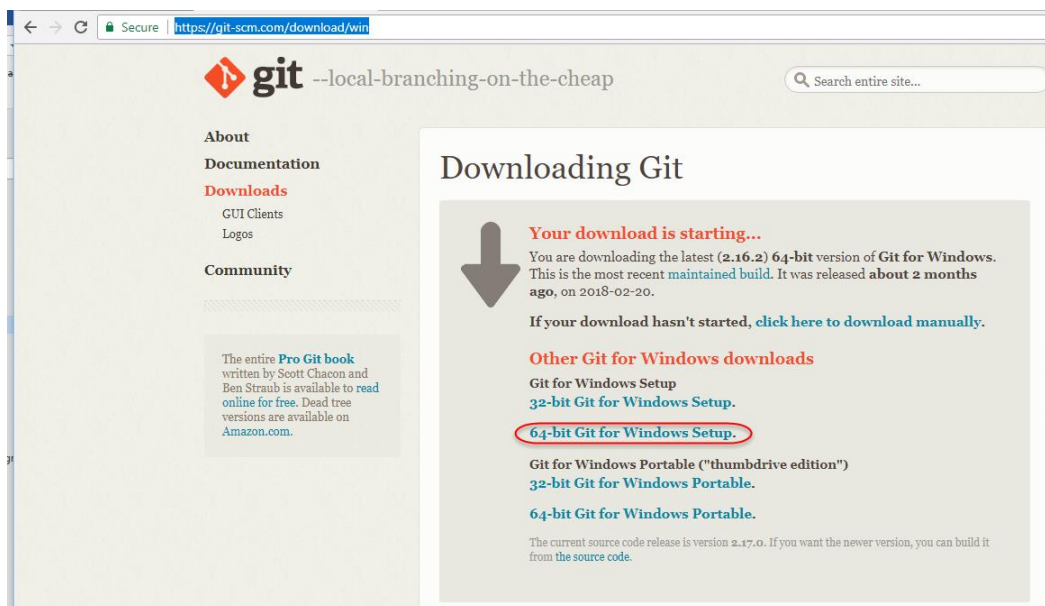
Download the latest Git for Windows installer.

You should have located this page/URL

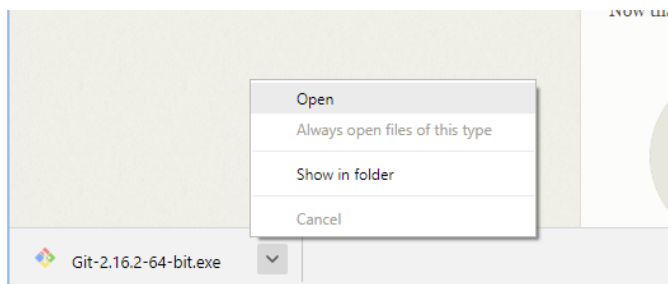
<https://git-scm.com/download/win>

Download the appropriate installer.

Note we use 64bit unless otherwise stated.



Once downloaded, Run/Open the installer



Acknowledge any required elevated (Administrator) privileges and continue

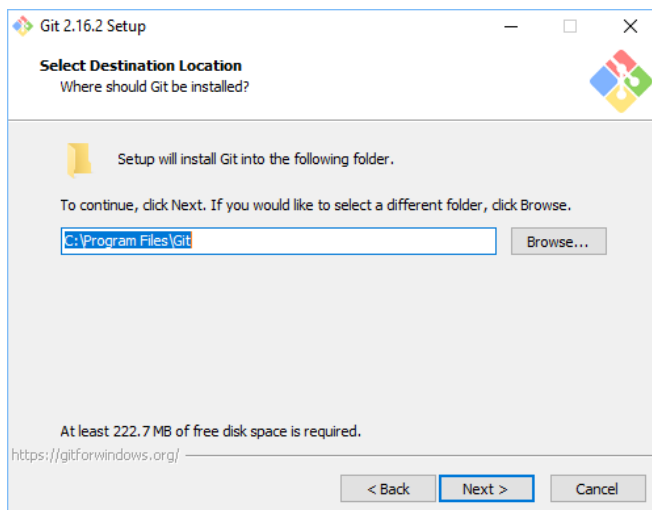
When you've successfully started the installer, you should see the Git Setup wizard screen. Follow the **Next** and **Finish** prompts to complete the installation. The default options are pretty sensible for most users. But we have added some screen captured to ensure everyone has the same setup.

Please follow these steps below.



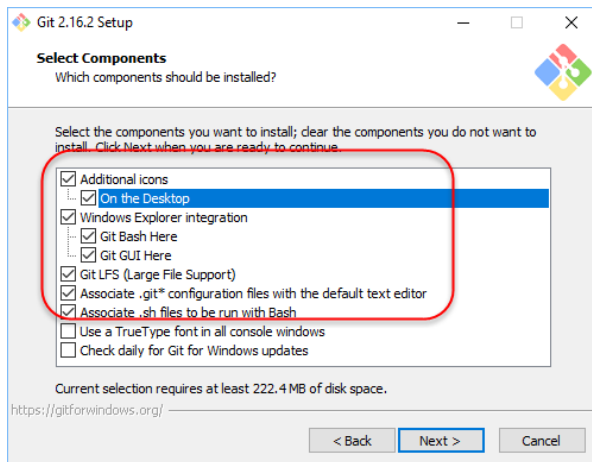
Click **Next**

We will use the default Git installation folder



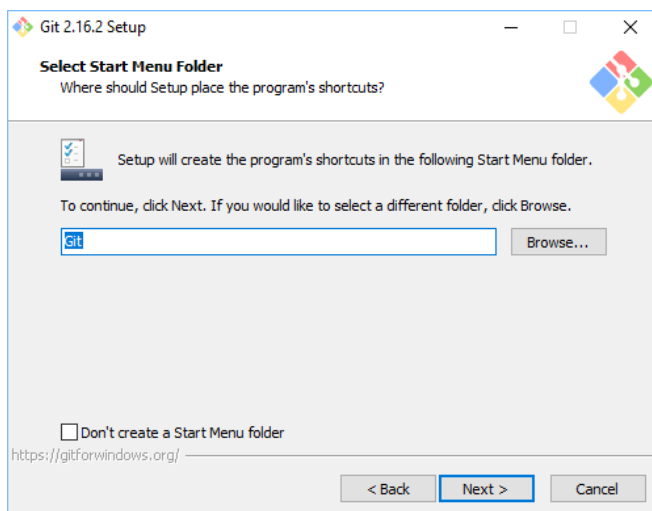
Click **Next**

Associate the appropriate files



Ensure you have **Additional Icons on Desktop** set to make it easy to launch Git when required.

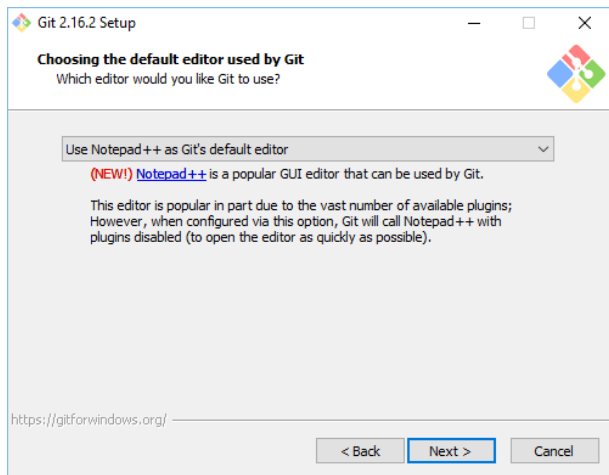
Click **Next**



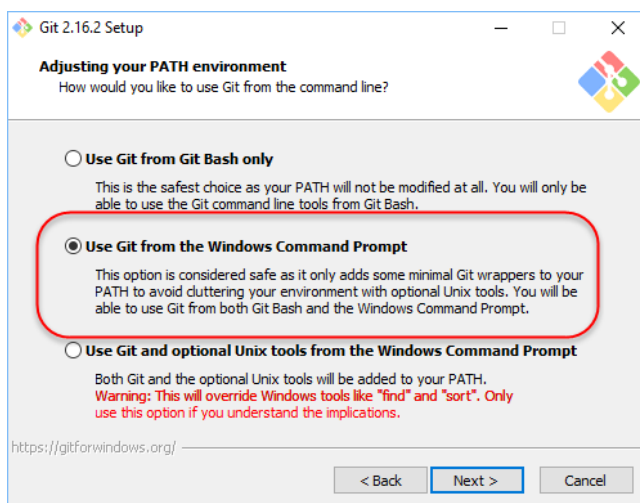
Click **Next**

In the following screen we will set **Notepad ++** as our default Git editor.

We see this option, because we installed Notepad++ earlier to give us a better text-editor & user experience for use with Git tools.

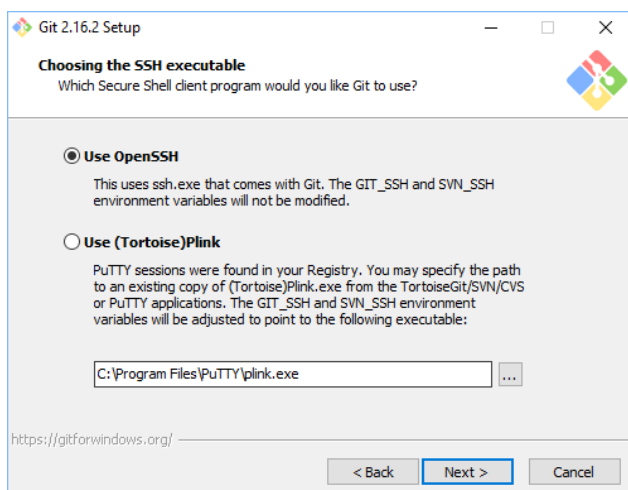


Click **Next**



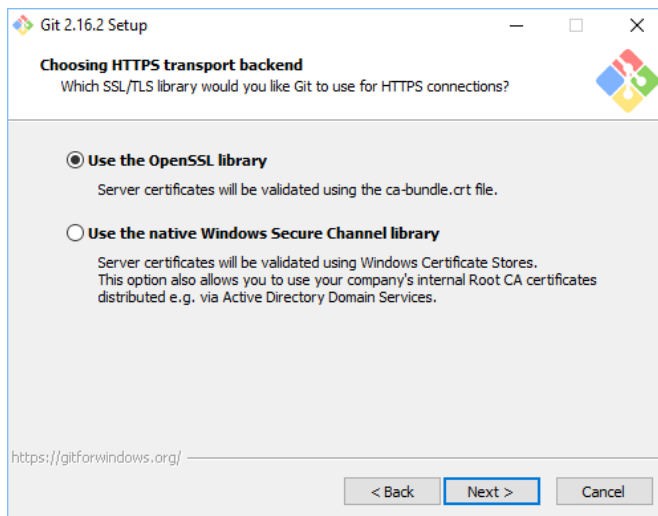
Select **Use Git from the Windows Command Prompt** option

Click **Next**



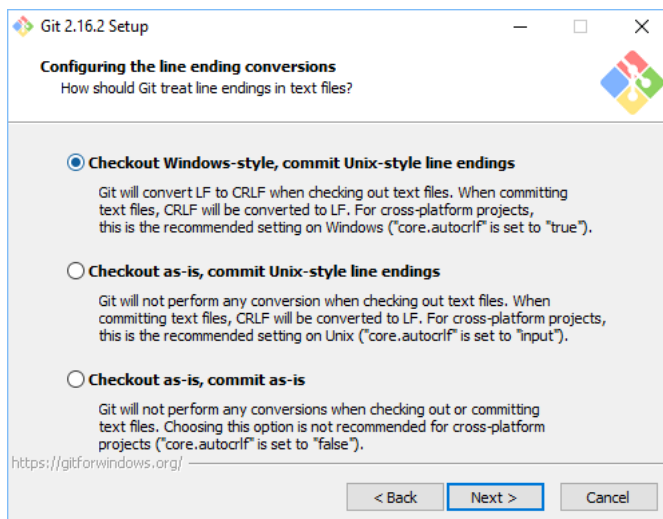
We will let Git **Use OpenSSH** for SSH Client (Default Option)

Click **Next**



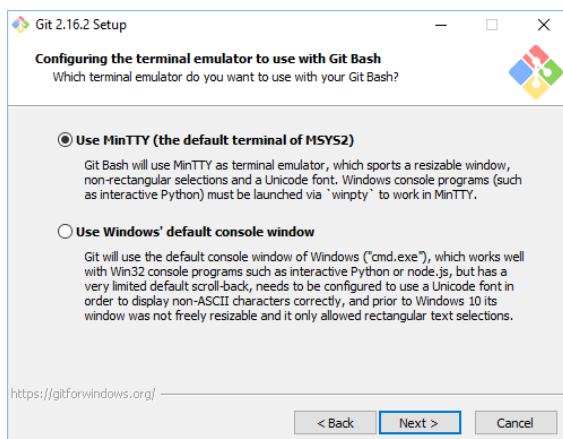
Choose **Use the OpenSSL library** (Default)

Click **Next**



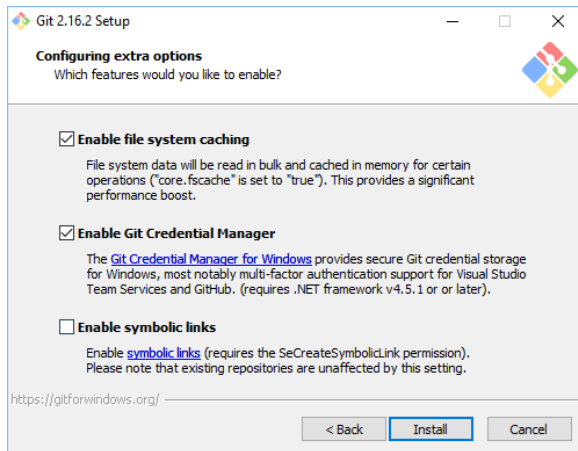
Choose **Checkout Windows-style, commit Unix-style line endings**

Click **Next**

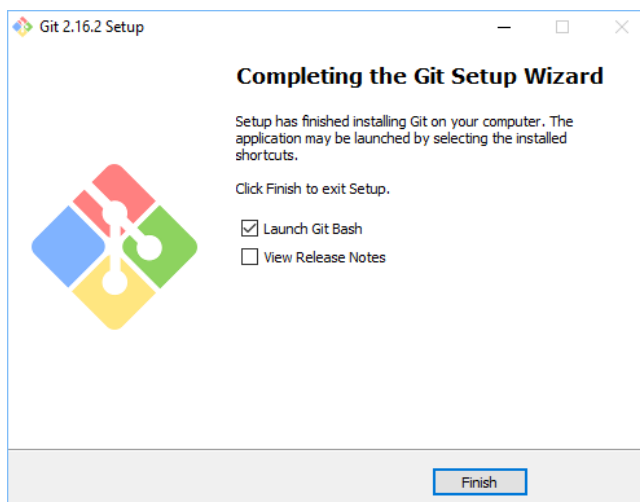
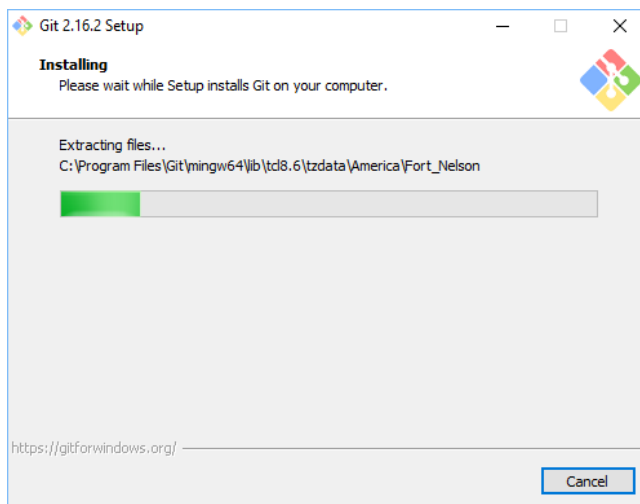


Choose **Use MinTTY (the default terminal of MSYS2)** which is the default option

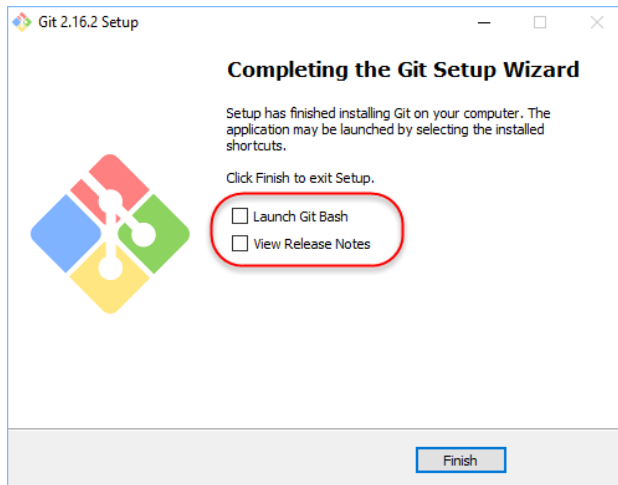
Click **Next**



On the **Configuring extra options** screen, Leave the defaults, and click **Install**



On the Completing the Git Setup Wizard screen, uncheck Launch Git Bash, and uncheck View Release Notes.



Click Finish

Verify Git installation

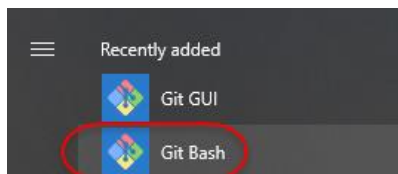
We will now check to see if Git is working as intended and set a couple of configuration options.

Open a Git Bash session using the Desktop icon or Windows Menu

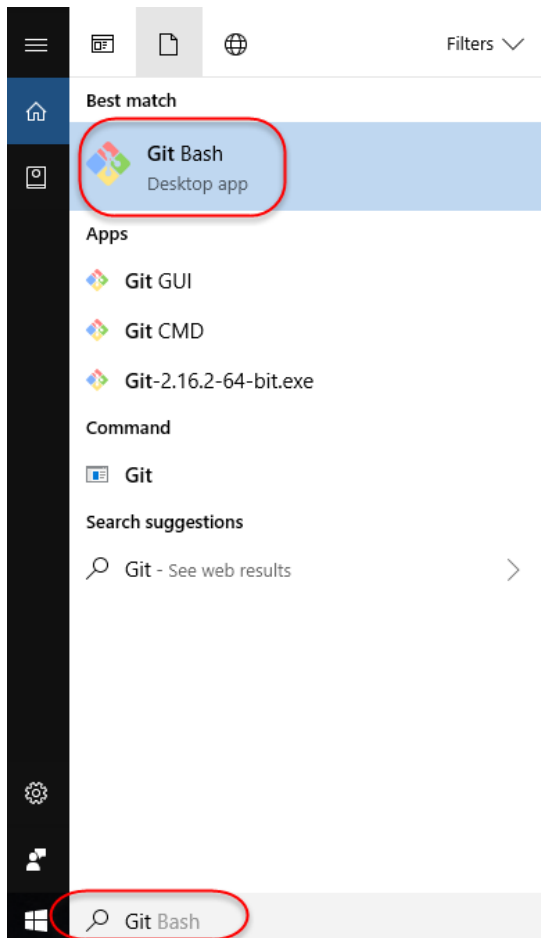
Example of the Git Bash desktop shortcut



Example of the Git Bash start menu option



Example using Search, click the Windows Start button, type Git in the search bar



With an open Git Bash session, type the following command

```
git --version
```

Result:

A screenshot of a Git Bash terminal window. The title bar shows 'MINGW64; c:/Users/Steve Robinson'. The terminal content shows the prompt 'Steve Robinson@SurfaceLaptop MINGW64 ~' followed by the command '\$ git --version' and the output 'git version 2.16.2.windows.1'. The prompt is then shown again with a cursor: '\$ |'.

Run the following commands to configure your Git username and email using the following commands, replacing Emma's name with your own. These details will be associated with any commits that you create:

You will be assigned a user-name and email by the instructor for an example

USERNAME = `awslogin1`

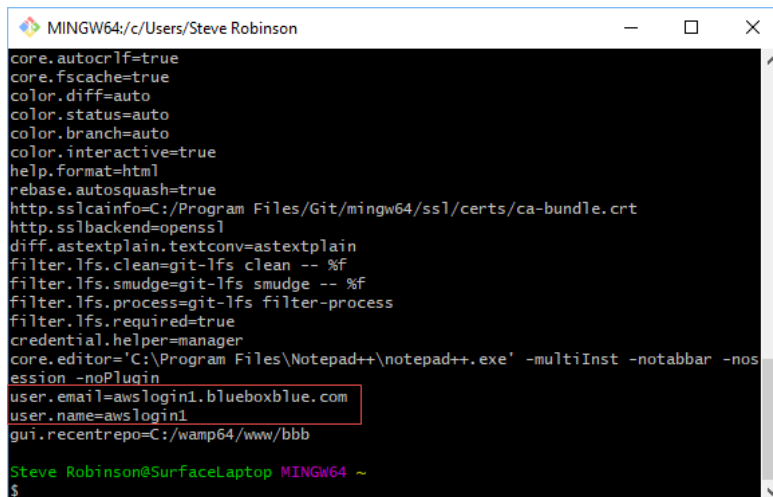
EMAIL = `awslogin1@blueboxblue.com`

Note: Before you run these commands, please ensure that you have the correct username and password assigned to you, as later we will all be checking in code changes that are made by each individual as part of a team effort.

```
git config --global user.name "awslogin1"
git config --global user.email "awslogin1.blueboxblue.com"
git config --list
```

Result

You will see your configuration settings after you have run the `git config --list` command



```
MINGW64; c/Users/Steve Robinson
core.autocrlf=true
core.fscache=true
color.diff=auto
color.status=auto
color.branch=auto
color.interactive=true
help.format=html
rebase.autosquash=true
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
http.sslbackend=openssl
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
credential.helper=manager
core.editor='C:\Program Files\Notepad++\notepad++.exe' -multiInst -notabbar -nos
ession -noPlugin
user.email=awslogin1.blueboxblue.com
user.name=awslogin1
gui.recentrepo=C:/wamp64/www/bbb

Steve Robinson@SurfaceLaptop MINGW64 ~
$
```

Close the Git Bash terminal, either by typing exit, or closing the window using a mouse.

Congratulations you have completed Part 1

Next Steps

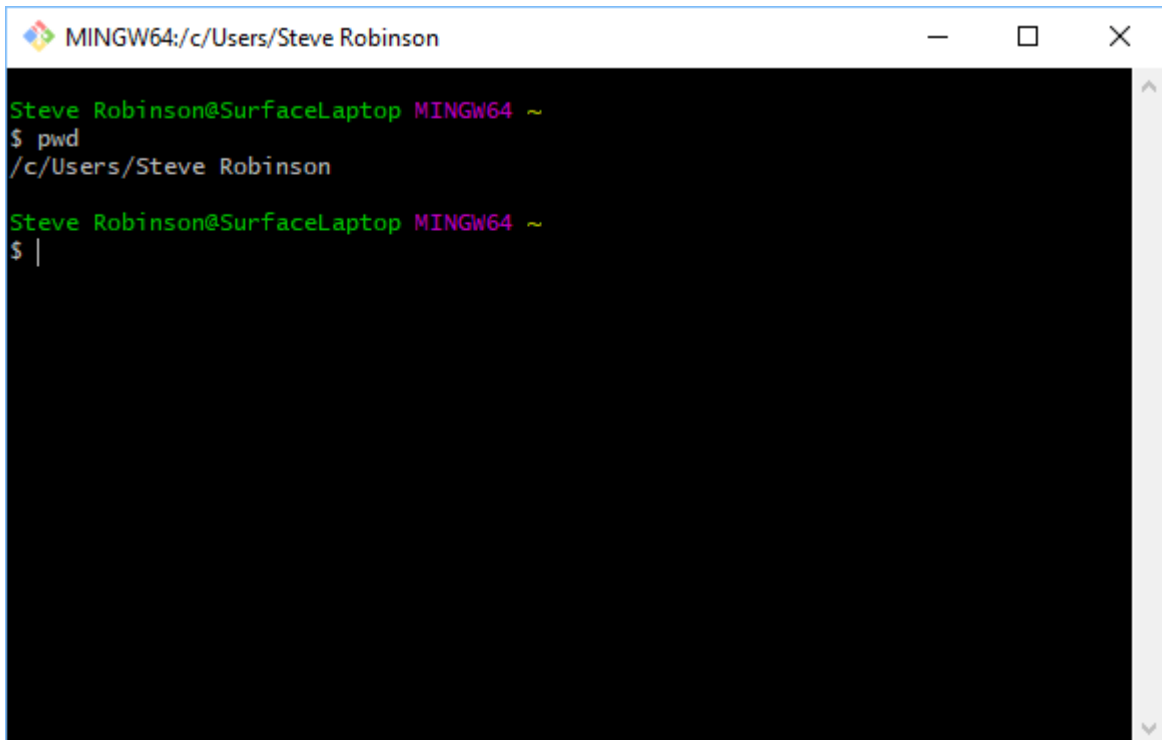
We will now move onto Section 2 where we will use Git to clone a remote repository from GitHub (A software as a service [SaaS] for online Git Repos).

This repo will contain instructions for part 2 where we will install some more tools and create a local Linux virtual machine using automation/scripts.

Cloning the Activity – Part 1 Repo from GitHub

Open a Git Bash terminal (shell)

Type the command `pwd`, which will display the current working folder/directory

A screenshot of a MINGW64 terminal window. The title bar shows 'MINGW64:/c/Users/Steve Robinson'. The terminal content shows the prompt 'Steve Robinson@SurfaceLaptop MINGW64 ~' followed by the command '\$ pwd' and its output '/c/Users/Steve Robinson'. Below this, the prompt is shown again with a cursor: 'Steve Robinson@SurfaceLaptop MINGW64 ~ \$ |'.

```
MINGW64:/c/Users/Steve Robinson

Steve Robinson@SurfaceLaptop MINGW64 ~
$ pwd
/c/Users/Steve Robinson

Steve Robinson@SurfaceLaptop MINGW64 ~
$ |
```

As we can see you will have a path similar but appropriate to your machines configuration and user-name.

```
/c/Users/Steve Robinson
```

We will create a folder called local-repos. Type the following command, one at a time and press enter to execute.

```
mkdir local-repos
cd local-repo
ls -ltra
```

The result will be a new folder, and move to the new directory i.e. `cd` (change directory) and a listing of the contents of this new folder, which is currently empty


```

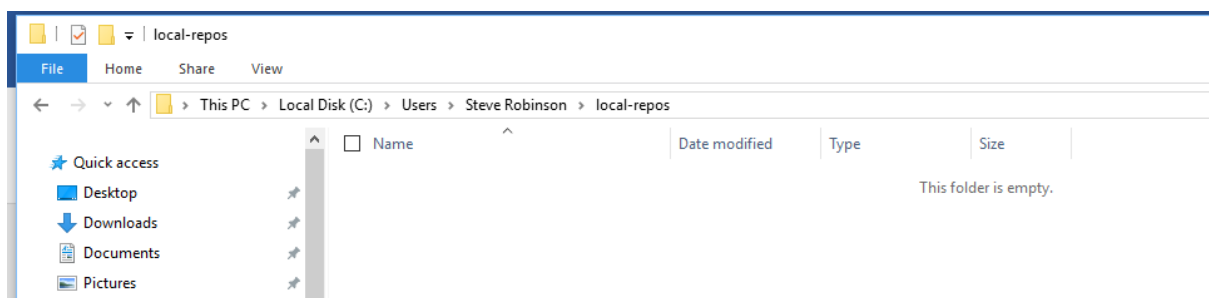
Steve Robinson@SurfaceLaptop MINGW64 ~
$ cd local-repos/

Steve Robinson@SurfaceLaptop MINGW64 ~/local-repos
$ ls -ltra
total 20
drwxr-xr-x 1 Steve Robinson 197121 0 Apr 13 12:51 ../
drwxr-xr-x 1 Steve Robinson 197121 0 Apr 13 12:51 ../

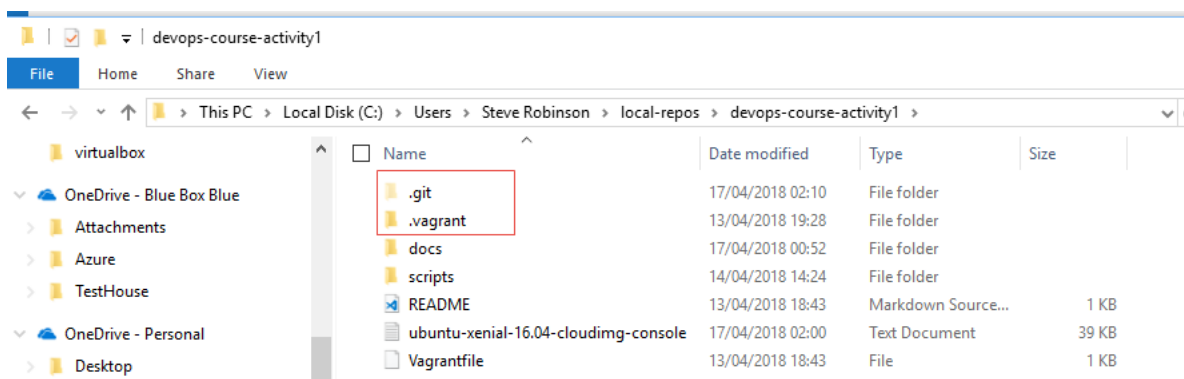
Steve Robinson@SurfaceLaptop MINGW64 ~/local-repos
$

```

If we browse to the folder using File-Explorer we see the following...



If we change the view options, to show hidden files as shown below we can see the hidden Git repo settings/config files.



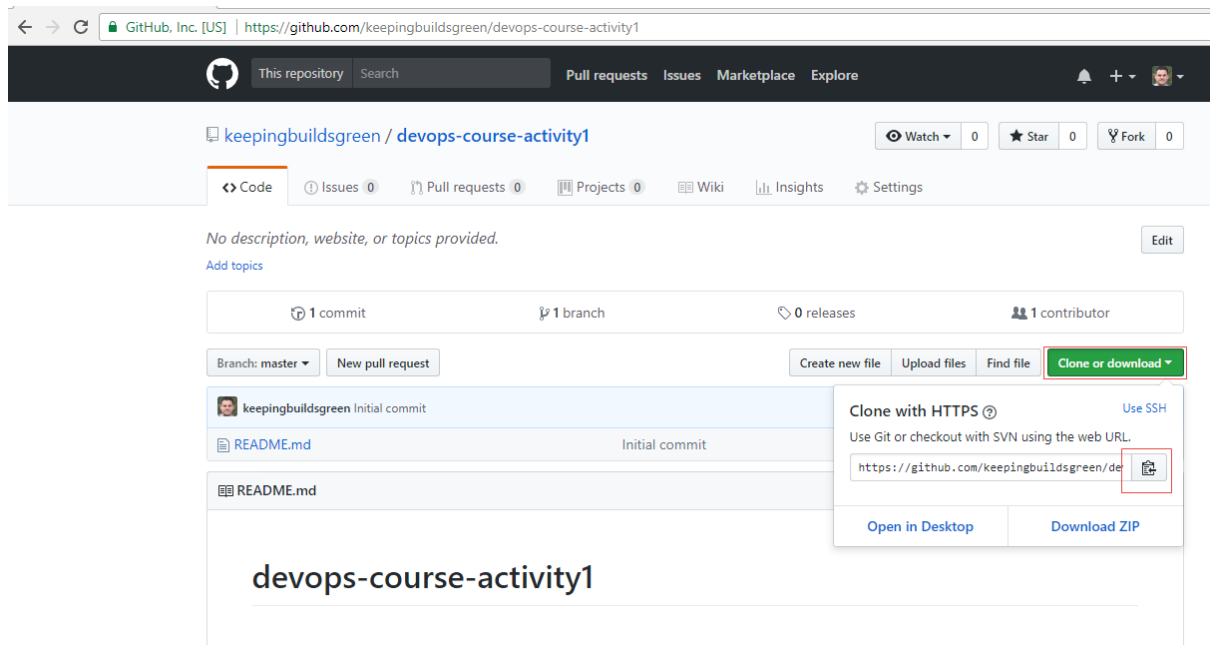
Cloning a Repo

A repo has been created online, and that repo has a few resources. Not all repos are dedicated to code, sometimes they have scripts, and or documents.

Please navigate to the following URL

<https://github.com/keepingbuildsgreen/devops-course-activity1>

We can see that there is a public repository that I have created that contains some resources.



We will click the clone button to copy the repo URL which is:

<https://github.com/keepingbuildsgreen/devops-course-activity1.git>

Using this URL, we can clone the repo using the following **Git** command sequence in an open **Git Bash** terminal into the

`git clone https://github.com/keepingbuildsgreen/devops-course-activity1.git`

Result:

```
$ git clone https://github.com/keepingbuildsgreen/devops-course-activity1.git
Cloning into 'devops-course-activity1'...
remote: Counting objects: 46, done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 46 (delta 10), reused 38 (delta 7), pack-reused 0
Unpacking objects: 100% (46/46), done.
```

```
$ git clone https://github.com/keepingbuildsgreen/devops-course-activity1.git
Cloning into 'devops-course-activity1'...
remote: Counting objects: 46, done.
remote: Compressing objects: 100% (33/33), done.
remote: Total 46 (delta 10), reused 38 (delta 7), pack-reused 0
Unpacking objects: 100% (46/46), done.
```

Open the repo location in a browser to view the resources, and open the File named

[DevOps-Acticity1-Part2.pdf](#)

Have a read through, to see what comes next?