Basic Git Commands

Objective: In this lab, you will configure and initialize Git.

Update the Ubuntu OS

1. Update all your Ubuntu packages. We do this because this VM image needs to be updated. The -y automates the installs (rather than ask for your approval to install them)
   $ sudo apt-get update -y && sudo apt-get upgrade -y

Verify Git is installed on your remote workstation

2. **$ git --version**
   Running this will confirm the version. Any version 2.X or higher is OK.

Configure Git on your workstation

1. **$ git config --global user.name "<your-firstname> <your-lastname>"**

2. **$ git config --global user.email "<your-email@address>"**

   EXAMPLES:
   $ git config --global user.name "James Bond"
   $ git config --global user.email "jbond@mi6.gov.uk"

3. Verify setting with
   $ git config --global --list

Set the editor for your workstation. Choose either nano, vi, vim or emacs.

4. To set the editor environment variable, use **ONE** of these four options:
   $ export EDITOR=nano
   $ export EDITOR=vi
   $ export EDITOR=vim
   $ export EDITOR=emacs

5. To open editor:
   $ git config --global --edit

6. Modify your name, save and quit

Create and initialize a local Git repo

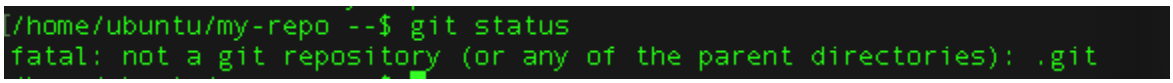
7. Change to your home directory
   $ cd ~

8. Create a new directory, which we will initialize as git-enabled
   $ mkdir my-repo

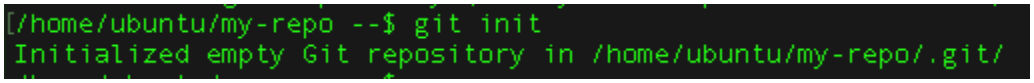
9. Change into the new directory
   $ cd my-repo

10. Check the current git status of the new directory
    $ git status

    This should fail because you have not yet initialized the directory.



11. Initialize the directory as a git-enabled
    $ git init



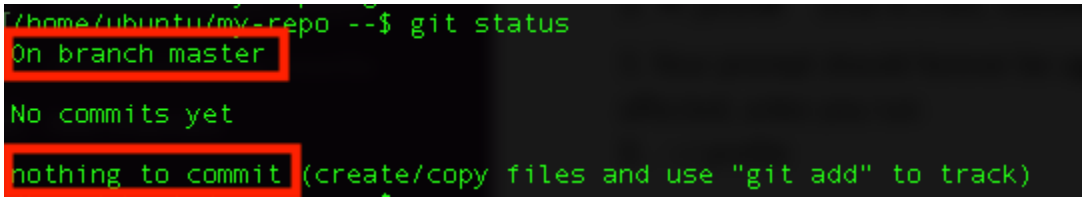
12. Verify that the directory has been initialized
    $ ls -al

13. Verify the existence of the .git directory
    a. cd into and look at the .git data files
       $ cd .git
       $ ls -la

```
ubuntu@ip-172-31-22-45:~/my-repo/.git$ ls -la
total 40
drwxrwxr-x 7 ubuntu ubuntu 4096 Apr  1 07:30 .
drwxrwxr-x 3 ubuntu ubuntu 4096 Apr  1 06:25 ..
-rw-rw-r-- 1 ubuntu ubuntu   23 Apr  1 06:25 HEAD
drwxrwxr-x 2 ubuntu ubuntu 4096 Apr  1 06:25 branches
-rw-rw-r-- 1 ubuntu ubuntu   92 Apr  1 06:34 config
-rw-rw-r-- 1 ubuntu ubuntu   73 Apr  1 06:25 description
drwxrwxr-x 2 ubuntu ubuntu 4096 Apr  1 06:25 hooks
drwxrwxr-x 2 ubuntu ubuntu 4096 Apr  1 06:25 info
drwxrwxr-x 4 ubuntu ubuntu 4096 Apr  1 06:25 objects
drwxrwxr-x 4 ubuntu ubuntu 4096 Apr  1 06:25 refs
```

b. Return to the my-repo directory
   $ cd ..

14. Run **$ git status** again. This time you should see:
    a. That you are on the master branch
    b. That there are no commits yet
    c. The names of any untracked files



First Git commands

15. Change directories back to the **my-repo** directory if needed

16. Create a README.md file, using your text editor

17. Add some text and save the file

18. Run **$ git status** to see how Git handles this new file
    Under 'Untracked files' you should see "README.md", showing that this file has been changed but has not yet been added

19. Add the README file to the Git staging area
    $ git add README.md

20. Check the status and see the changes since adding the file to the staging area

```
/home/ubuntu/my-repo --$ git status
On branch master

No commits yet

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

/home/ubuntu/my-repo --$
```

21. Commit the file to the local git repo. Add the text below in yellow to the commit log:

```
/home/ubuntu/my-repo --$ git commit
```

```
Added a README.md file
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
#
# Initial commit
#
# Changes to be committed:
#       new file:   README.md
#
~
```

22. Check the current git status after committing

$git status

```
/home/ubuntu/my-repo --$ git status
On branch master
nothing to commit, working tree clean
```

23. Rerun the git commit command to see what happens if you try to commit again without first making changes and adding the files to the staging area. Note there is nothing to commit, because everything has already been committed

```
/home/ubuntu/my-repo --$ git commit
On branch master
nothing to commit, working tree clean
```

24. Run git log, to see the commit history:

```
/home/ubuntu/my-repo --$ git log
commit 2b56a40b99c0ef46171d7f80fe37103de2be532c (HEAD -> master)
Author: Elon D. Bar-Evan <elon@technotrainer.com>
Date:   Mon Nov 16 17:05:20 2020 +0000
```

25. Run git show to see the details of a particular commit.
    Replace [SHA] below with your actual SHA:
    $ git show [SHA]

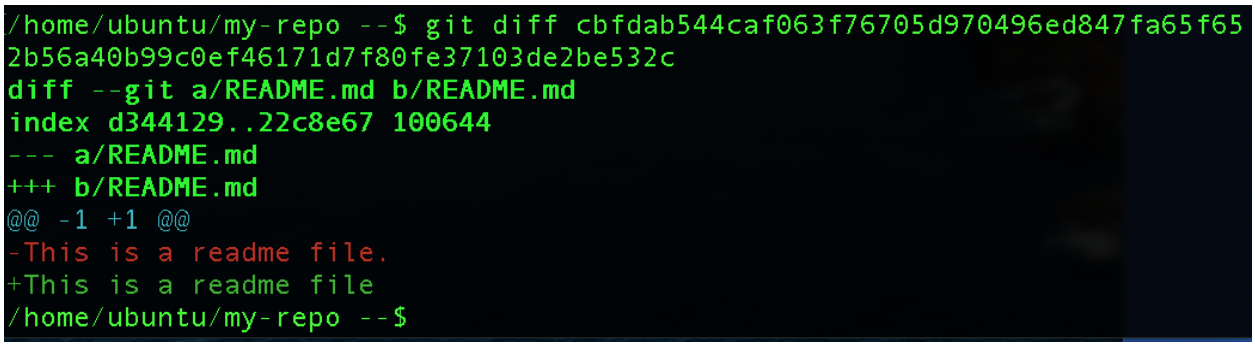
```
/home/ubuntu/my-repo --$ git show 2b56a40b99c0ef46171d7f80fe37103de2be532c
commit 2b56a40b99c0ef46171d7f80fe37103de2be532c (HEAD -> master)
Author: Elon D. Bar-Evan <elon@technotrainer.com>
Date:   Mon Nov 16 17:05:20 2020 +0000

    Added a README.md file

diff --git a/README.md b/README.md
new file mode 100644
index 0000000..22c8e67
--- /dev/null
+++ b/README.md
@@ -0,0 +1 @@
+This is a readme file
/home/ubuntu/my-repo --$
```

26. Update file to compare multiple commits
    a. Modify and save the README file again
    b. Add the README file to the git staging area
       $ git add README.md

c. Run git commit, then add a commit message
   $ git commit
d. Now look at the commit history and note that there are two different SHA's
   $ git log

27. Compare differences
    $ git diff <first SHA> <2nd SHA>

    Notice in the example below, we added a dot '.' to the file. You can see it in red, versus the line below the red one in green



28. Entire Commit Process

    1. Modify and save the README.md file again
    2. **$ git status** (to see the file that has been modified but is untracked)
    3. **$ git add README.md** (to stage the file)
    4. **$ git status** (to see there is something to commit)
    5. **$ git commit** (to commit it)
    6. **$ git log** (to see the commit summary)
    7. **$ git show [SHA]**
       This will show the commit details. Note: [SHA] should be replaced with the

40-digit unique identifier

8. **$ git diff [SHA-1] [SHA-2]** (to see the difference between multiple commits. Remember again to replace [SHA-1] and [SHA-2] with your actual two SHA's)

Notify your instructor that you are done with the lab

END OF LAB