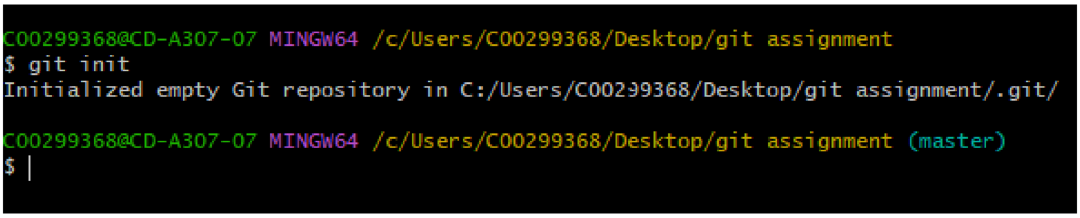
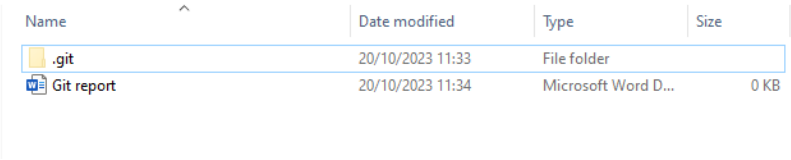
git init

Creates a new repository which is stored only in 1 location which is the local PC



```
C00299368@CD-A307-07 MINGW64 /c/Users/C00299368/Desktop/git assignment
$ git init
Initialized empty Git repository in C:/Users/C00299368/Desktop/git assignment/.git/

C00299368@CD-A307-07 MINGW64 /c/Users/C00299368/Desktop/git assignment (master)
$ |
```

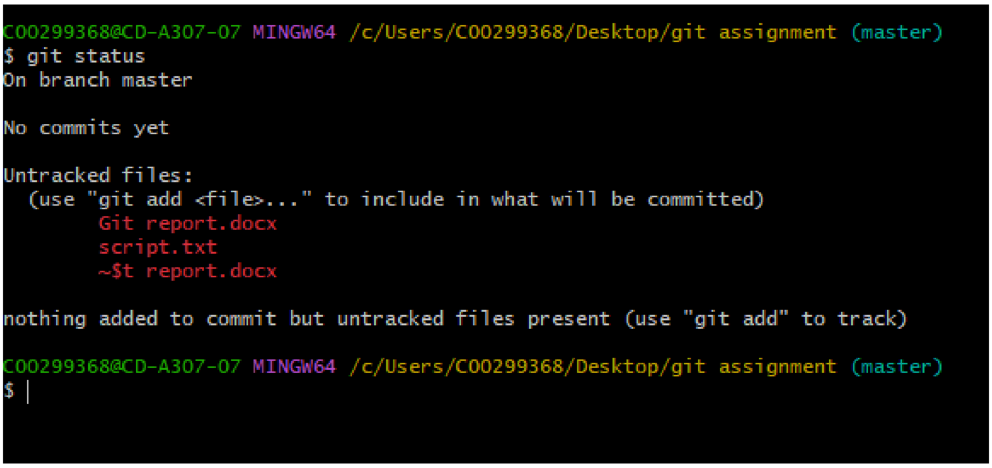
This creates a hidden .git file in the repository which can be viewed by going into the view tab and checking the "hidden files" box.



Name	Date modified	Type	Size
.git	20/10/2023 11:33	File folder	
Git report	20/10/2023 11:34	Microsoft Word D...	0 KB

git status

Shows the current status of your git repository and any files that are not staged or the ones that are staged

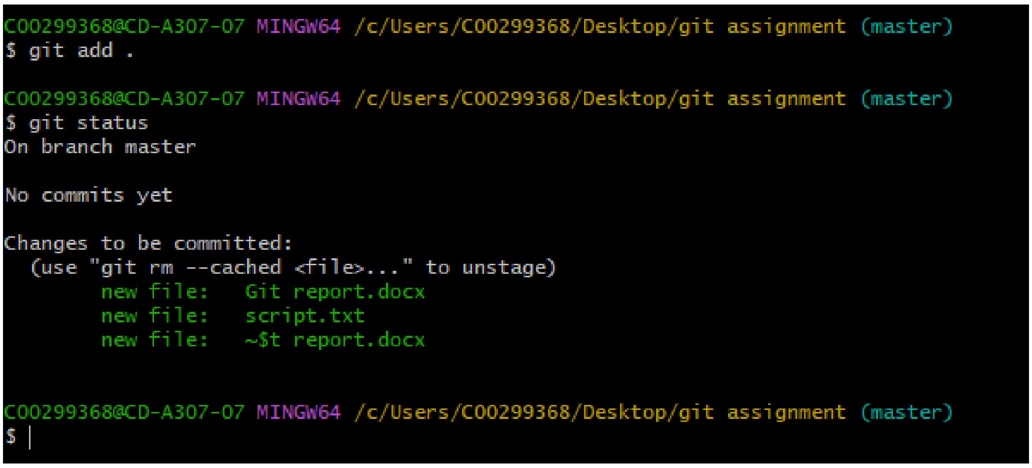


```
C00299368@CD-A307-07 MINGW64 /c/Users/C00299368/Desktop/git assignment (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Git report.docx
        script.txt
        ~$t report.docx

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

C00299368@CD-A307-07 MINGW64 /c/Users/C00299368/Desktop/git assignment (master)
$ |
```

Red files are new files which are in the repository which have not yet been staged and will not be saved when you execute the "git commit" command.

git add

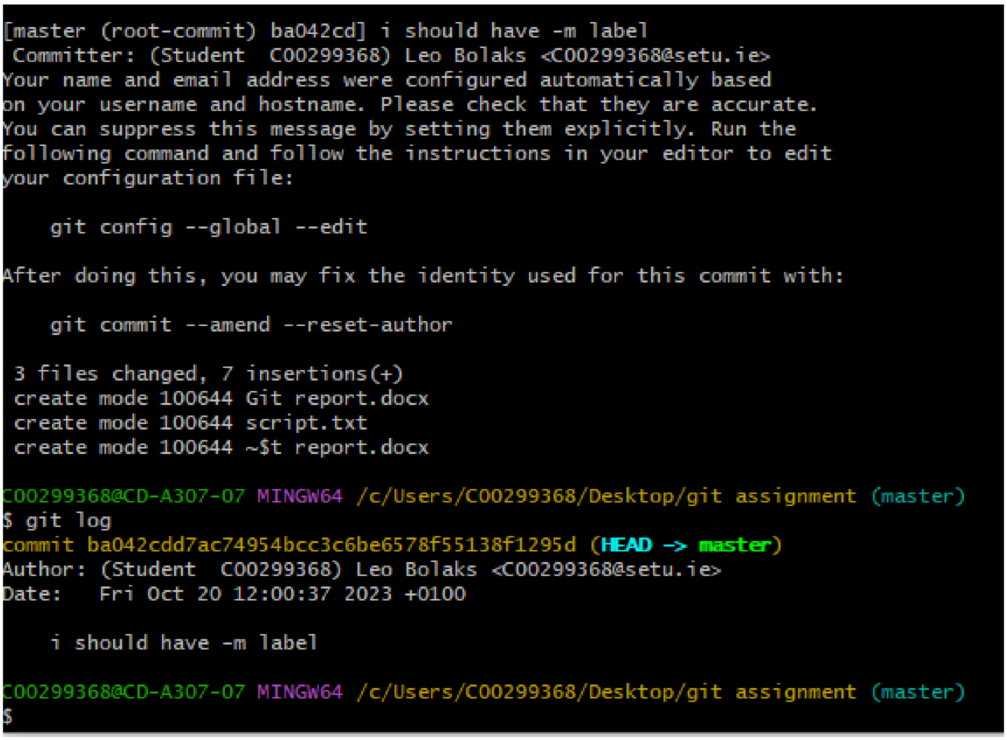
Git add command stages the files in the repository and they will be displayed in green, this can be checked by re-using the git status command, now when you do a commit, these files will be saved and you will be able to see them in the github repository.



```
C00299368@CD-A307-07 MINGW64 /c/Users/C00299368/Desktop/git assignment (master)
$ git add .

C00299368@CD-A307-07 MINGW64 /c/Users/C00299368/Desktop/git assignment (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Git report.docx
        new file:   script.txt
        new file:   ~$t report.docx


C00299368@CD-A307-07 MINGW64 /c/Users/C00299368/Desktop/git assignment (master)
$ |
```

git commit

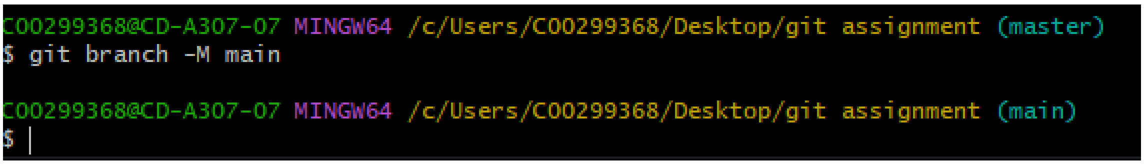
The commit command is basically a save which creates a new node which you can go back to if there has been a mistake further up the nodes, the commit command saves all the staged files (the ones in green) to the repository



```
[master (root-commit) ba042cd] i should have -m label
 Committer: (Student  C00299368) Leo Bolaks <C00299368@setu.ie>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 3 files changed, 7 insertions(+)
 create mode 100644 Git report.docx
 create mode 100644 script.txt
 create mode 100644 ~$t report.docx

C00299368@CD-A307-07 MINGW64 /c/Users/C00299368/Desktop/git assignment (master)
$ git log
commit ba042cdd7ac74954bcc3c6be6578f55138f1295d (HEAD -> master)
Author: (Student  C00299368) Leo Bolaks <C00299368@setu.ie>
Date:   Fri Oct 20 12:00:37 2023 +0100

    i should have -m label

C00299368@CD-A307-07 MINGW64 /c/Users/C00299368/Desktop/git assignment (master)
$
```

git branch

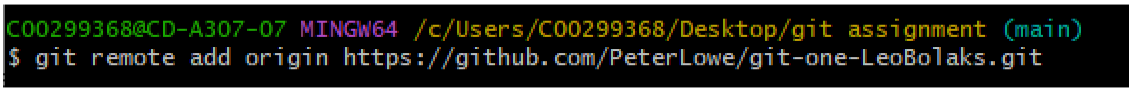
The branch command I used was “git branch -M main” this command renames the repository from “master” to “main” but different variations of this command can create new branches, edit them, or delete them.



```
C00299368@CD-A307-07 MINGW64 /c/Users/C00299368/Desktop/git assignment (master)
$ git branch -M main

C00299368@CD-A307-07 MINGW64 /c/Users/C00299368/Desktop/git assignment (main)
$ |
```

git remote

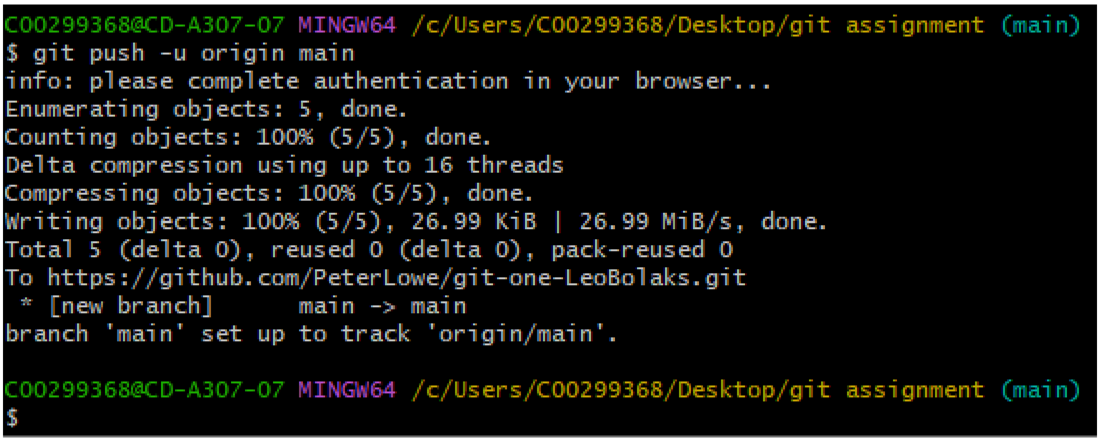
The remote command allows you to connect to a repository somewhere in the cloud, in this case the name of the cloud repository is remote, you can now push all the commited files into the repository



```
C00299368@CD-A307-07 MINGW64 /c/Users/C00299368/Desktop/git assignment (main)
$ git remote add origin https://github.com/PeterLowe/git-one-LeoBolaks.git
```

git push

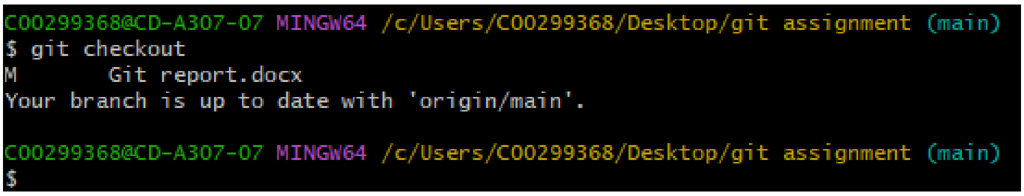
The push command pushes all of the files you have commited to the repository which is not on your local machine, in this case I pushed all files I have commited to the repository named “origin”



```
C00299368@CD-A307-07 MINGW64 /c/Users/C00299368/Desktop/git assignment (main)
$ git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 26.99 KiB | 26.99 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/PeterLowe/git-one-LeoBolaks.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

C00299368@CD-A307-07 MINGW64 /c/Users/C00299368/Desktop/git assignment (main)
$
```

git checkout

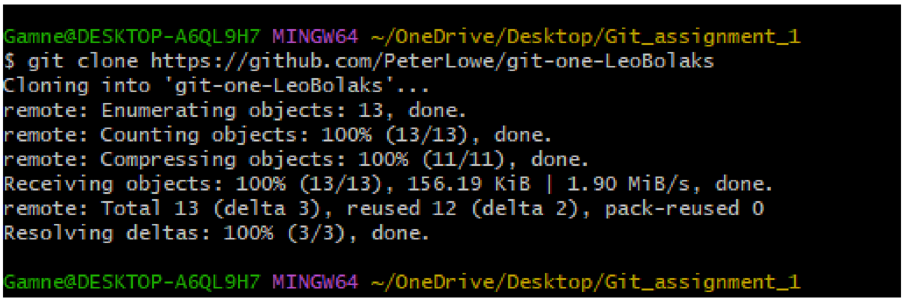
the checkout command allows me to compare whether the node I'm working on is the same or different from the main directory in this case my branch is up to date with the origin/main repository.



```
C00299368@CD-A307-07 MINGW64 /c/Users/C00299368/Desktop/git assignment (main)
$ git checkout
M       Git report.docx
Your branch is up to date with 'origin/main'.

C00299368@CD-A307-07 MINGW64 /c/Users/C00299368/Desktop/git assignment (main)
$
```

git clone

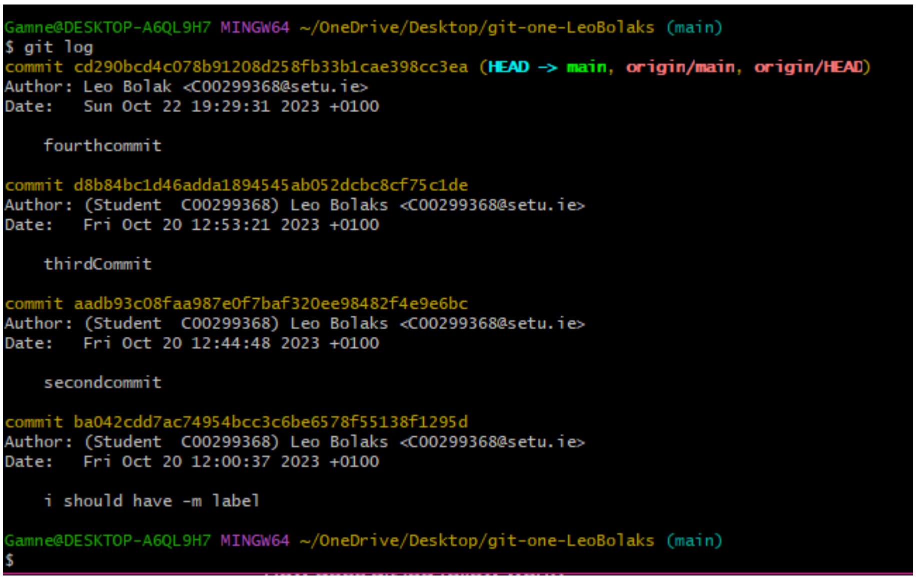
The git clone command does exactly what it's name is, it clones the repository from the cloud onto your machine that you are using, using this allows you to keep working on your repository from any machine you want



```
Gamne@DESKTOP-A6QL9H7 MINGW64 ~/OneDrive/Desktop/Git_assignment_1
$ git clone https://github.com/PeterLowe/git-one-LeoBolaks
Cloning into 'git-one-LeoBolaks'...
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (11/11), done.
Receiving objects: 100% (13/13), 156.19 KiB | 1.90 MiB/s, done.
remote: Total 13 (delta 3), reused 12 (delta 2), pack-reused 0
Resolving deltas: 100% (3/3), done.

Gamne@DESKTOP-A6QL9H7 MINGW64 ~/OneDrive/Desktop/Git_assignment_1
```

git log

The git log command allows you to view all previous commits that have been done by a user in the repository, you can use this to search for and view specific changes.



```
Gamne@DESKTOP-A6QL9H7 MINGW64 ~/OneDrive/Desktop/git-one-LeoBolaks (main)
$ git log
commit cd290bcd4c078b91208d258fb33b1cae398cc3ea (HEAD -> main, origin/main, origin/HEAD)
Author: Leo Bolak <C00299368@setu.ie>
Date:   Sun Oct 22 19:29:31 2023 +0100

    fourthcommit

commit d8b84bc1d46adda1894545ab052dcbc8cf75c1de
Author: (Student  C00299368) Leo Bolaks <C00299368@setu.ie>
Date:   Fri Oct 20 12:53:21 2023 +0100

    thirdCommit

commit aadb93c08faa987e0f7baf320ee98482f4e9e6bc
Author: (Student  C00299368) Leo Bolaks <C00299368@setu.ie>
Date:   Fri Oct 20 12:44:48 2023 +0100

    secondcommit

commit ba042cdd7ac74954bcc3c6be6578f55138f1295d
Author: (Student  C00299368) Leo Bolaks <C00299368@setu.ie>
Date:   Fri Oct 20 12:00:37 2023 +0100

    i should have -m label

Gamne@DESKTOP-A6QL9H7 MINGW64 ~/OneDrive/Desktop/git-one-LeoBolaks (main)
$
```