Task : Configure your VCS. Initialize an empty repository

First of all, I downloaded and installed git for linux using `sudo apt-get install git` and set up the configuration details of my GitHub user. Basically, I introduced my user name and email from my GitHub account. Then, I created a local empty repository `git init repozitoriu`.

```
cornelia@cornelia-300E4C-300E5C-300E7C:~$ git config --global user.name "RedFox96"
cornelia@cornelia-300E4C-300E5C-300E7C:~$ git config --global user.email "cornelusasirbu@gmail.com"
cornelia@cornelia-300E4C-300E5C-300E7C:~$ git init repozitoriu
Initialized empty Git repository in /home/cornelia/repozitoriu/.git/
```

Than I connected to GitHub with SSH so I can connect without supplying my username and password at each visit. I generated a new SSH key with the command `ssh- keygen –t rsa 4096 –C email@example.com` and chose a passphrase. Then, I added the new key to the ssh-agent by typing in `ssh-add ~/.ssh/id_rsa`. After that, I just copied the generated key and added it on the GitHub site.

```
cornelia@cornelia-300E4C-300E5C-300E7C:~$ ssh-keygen -t rsa -b 4096 -C "cornelusasirbu@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/cornelia/.ssh/id_rsa): /home/cornelia/.ssh/id_rsa
Created directory '/home/cornelia/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/cornelia/.ssh/id_rsa.
Your public key has been saved in /home/cornelia/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:7huEE0aWbcyQV0/8TNPvactZlJp87cFysdK0TeRVdy8 cornelusasirbu@gmail.com
The key's randomart image is:
+---[RSA 4096]----+
|      +B ....  .=|
|     oo *  o. o B|
|      oo    .+E+=|
|     . o      o==|
|      o S   . *+B|
|       +     * @=|
|        o     B.=|
```

```
cornelia@cornelia-300E4C-300E5C-300E7C:~$ eval "$(ssh-agent -s)"
Agent pid 12543
cornelia@cornelia-300E4C-300E5C-300E7C:~$ ssh-add ~/.ssh/id_rsa
Enter passphrase for /home/cornelia/.ssh/id_rsa:
Identity added: /home/cornelia/.ssh/id_rsa (/home/cornelia/.ssh/id_rsa)
cornelia@cornelia-300E4C-300E5C-300E7C:~$ sudo apt-get install xclip
Reading package lists... Done
```

I created a new repository My-Repo on GitHub. Initialized the local directory as a Git repository. Added the files in my new local repository. This stages for the first commit. Commited the files that I’ve staged in my local repository than pushed the changes.

```
cornelia@cornelia-300E4C-300E5C-300E7C:~$ echo "#My-Repo" >> README.md
cornelia@cornelia-300E4C-300E5C-300E7C:~$ git init
Initialized empty Git repository in /home/cornelia/.git/
cornelia@cornelia-300E4C-300E5C-300E7C:~$ git add README.md
cornelia@cornelia-300E4C-300E5C-300E7C:~$ git commit -m "My not first commit"
[master (root-commit) e3c2d4b] My not first commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
cornelia@cornelia-300E4C-300E5C-300E7C:~$ git remote add origin https://github.com/RedFox96/My-Repo.git
cornelia@cornelia-300E4C-300E5C-300E7C:~$ git push -u origin master
Username for 'https://github.com': RedFox96
Password for 'https://RedFox96@github.com':
Counting objects: 3, done.
Writing objects: 100% (3/3), 231 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/RedFox96/My-Repo.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
```

Task: Create branches. Commit to different branches

I crated a branch named MadWorld by using the command git branch MadWorld.
Then, I called the command git checkout MadWorld which switched me from the master branch to the newly created MadWorld branch. Being on this branch, I committed several files and pushed them on my remote repository.

```
cornelia@cornelia-300E4C-300E5C-300E7C:~$ git branch MadWorld
cornelia@cornelia-300E4C-300E5C-300E7C:~$ git checkout MadWorld
Switched to branch 'MadWorld'
cornelia@cornelia-300E4C-300E5C-300E7C:~$ git add hello.py
cornelia@cornelia-300E4C-300E5C-300E7C:~$ git commit -m "commit MadWorld"
[MadWorld 2f70c22] commit MadWorld
 1 file changed, 1 insertion(+)
 create mode 100644 hello.py
cornelia@cornelia-300E4C-300E5C-300E7C:~$ git push origin MadWorld
Username for 'https://github.com': RedFox96
Password for 'https://RedFox96@github.com':
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 304 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/RedFox96/My-Repo.git
 * [new branch]      MadWorld -> MadWorld
```

For the second branch, I used a slightly different command: `git checkout –b beta`. This creates a new branch and automatically switches to it. The –b option is a convenience flag that tells Git to run git branch <new-branch> before running gitcheckout <new-branch>. Again, I made two commits to this branch and pushed them to my remote repository.

```
cornelia@cornelia-300E4C-300E5C-300E7C:~$ git checkout -b Rainbow
Switched to a new branch 'Rainbow'
cornelia@cornelia-300E4C-300E5C-300E7C:~$ git add hello.rb
cornelia@cornelia-300E4C-300E5C-300E7C:~$ git commit -m "commit Rainbow"
[Rainbow dd9e8a5] commit Rainbow
 1 file changed, 1 insertion(+)
 create mode 100644 hello.rb
cornelia@cornelia-300E4C-300E5C-300E7C:~$ git add hello.c
cornelia@cornelia-300E4C-300E5C-300E7C:~$ git commit -m "commit Rainbow"
[Rainbow c391fd1] commit Rainbow
 1 file changed, 8 insertions(+)
 create mode 100644 hello.c
cornelia@cornelia-300E4C-300E5C-300E7C:~$ git push origin Rainbow
Username for 'https://github.com': RedFox96
Password for 'https://RedFox96@github.com':
Counting objects: 6, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 610 bytes | 0 bytes/s, done.
Total 6 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/RedFox96/My-Repo.git
 * [new branch]      Rainbow -> Rainbow
```