Git operations for bitbucket

In order to access and submit the homework, you need to transfer the files from your bitbucket repository to your AWS machine. For this, you will need the link to your repository:

The transferring is done with the **git clone** command, which will create a working copy of your local repository.

git clone linktorepo

```
ubuntu@ip-10-0-0-44:~$ mkdir HW0
ubuntu@ip-10-0-0-44:~$ cd HW0/
ubuntu@ip-10-0-0-44:~/HW0$ git clone https://username@bitbucket.org/...link.git
```

After giving your password, the repository is copied to your machine.

```
ubuntu@ip-10-0-0-44:~/HW0$ ls
e6040_hw0_zo2131
ubuntu@ip-10-0-0-44:~/HW0$ cd e6040_hw0_zo2131/
ubuntu@ip-10-0-0-44:~/HW0/e6040_hw0_zo2131$ ls
E6040_HW0.ipynb
```

The **git status** command displays the state of the working directory and the staging area. It lets you see which changes have been staged, which haven't, and which files aren't being tracked by Git.

```
ubuntu@ip-10-0-0-44:~/HW0/e6040_hw0_zo2131$ git status
On branch master
Your branch is up-to-date with 'origin/master'.

nothing to commit, working directory clean
```

After changing the python notebook, the **git status** command will report that the file has been modified, and warn you that it is not staged.

```
ubuntu@ip-10-0-0-44:~/HW0/e6040_hw0_zo2131$ git status
On branch master
Your branch is up-to-date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   E6040_HW0.ipynb

no changes added to commit (use "git add" and/or "git commit -a")
```

The **git add** command adds a change in the working directory to the staging area. It tells Git that you want to include updates to a particular file in the next commit. However, **git add** doesn't really affect the repository in any significant way—changes are not actually recorded until you run **git commit**.

```
ubuntu@ip-10-0-0-44:~/HW0/e6040_hw0_zo2131$ git add E6040_HW0.ipynb
ubuntu@ip-10-0-0-44:~/HW0/e6040_hw0_zo2131$ git status
On branch master
Your branch is up-to-date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   E6040_HW0.ipynb

ubuntu@ip-10-0-0-44:~/HW0/e6040_hw0_zo2131$
```

The **git commit** command commits the staged snapshot to the project history. Committed snapshots can be thought of as “safe” versions of a project—Git will never change them unless you explicitly ask it to. Along with **git add**, this is one of the most important Git commands.

git commit -m "<message>"

```
ubuntu@ip-10-0-0-44:~/HW0/e6040_hw0_zo2131$ git commit -m 'First commit'
[master 6033415] First commit
 Committer: Ubuntu <ubuntu@ip-10-0-0-44.ec2.internal>
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:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

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

    git commit --amend --reset-author

 1 file changed, 1 insertion(+), 1 deletion(-)
ubuntu@ip-10-0-0-44:~/HW0/e6040_hw0_zo2131$
```

The **git push** command is how you transfer commits from your local repository to a remote repo. This has the potential to overwrite changes, so you need to be careful how you use it. To push back the file you previously changed to your bitbucket repo, use the **git push origin master** command.

The next time you will clone your repository, it will have the updated file. For further help please check:
https://www.atlassian.com/git/tutorials
https://confluence.atlassian.com/bitbucketserver/basic-git-commands-776639767.html

For convenience, here is the cheat sheet and workflow based on the operations above:

---

git clone https://.....git # create local repository for your homework

[... make some changes ...]

git status # check and track modifications in your local repository
git add file [file1 file2 ...] # stage the file who is ready for submit
git commit -m "MESSAGE" # commit all the staged files

[... make some changes and repeated above steps ...]

git push #upload your local repository onto BitBucket
