holoclean is a algorithm for data cleaning and in this “How to” we will review the packages and methods that implement it in Python.

0- Installation

0.1 Download and using Anaconda

For Ubuntu:

- For the 32 bits version:
  Open a terminal and write the following commands

1. **wget https://3230d63b5fc54e62148e-c95ac804525aac4b6dba79b00b39d1d3.ssl.cf1.rackcdn.com/Anaconda-2.3.0-Linux-x86.sh**
2. **bash Anaconda-2.3.0-Linux-x86.sh**

- For the 64 bits:
  Open a terminal and write the following commands

1. **wget https://3230d63b5fc54e62148e-c95ac804525aac4b6dba79b00b39d1d3.ssl.cf1.rackcdn.com/Anaconda-2.3.0-Linux-x86_64.sh**
2. **bash Anaconda-2.3.0-Linux-x86_64.sh**

For MacOS:

- Follow the instructions on https://conda.io/docs/user-guide/install/macos.html to install Anaconda for MacOS

Installing on macOS

1. Download the installer:
   - Miniconda installer for macOS.
   - Anaconda installer for macOS.
2. Install:
   - Miniconda—In your Terminal window, run:

     ```
     bash Miniconda3-latest-MacOSX-x86_64.sh
     ```

   - Anaconda—Double-click the `.pkg` file.
3. Follow the prompts on the installer screens.

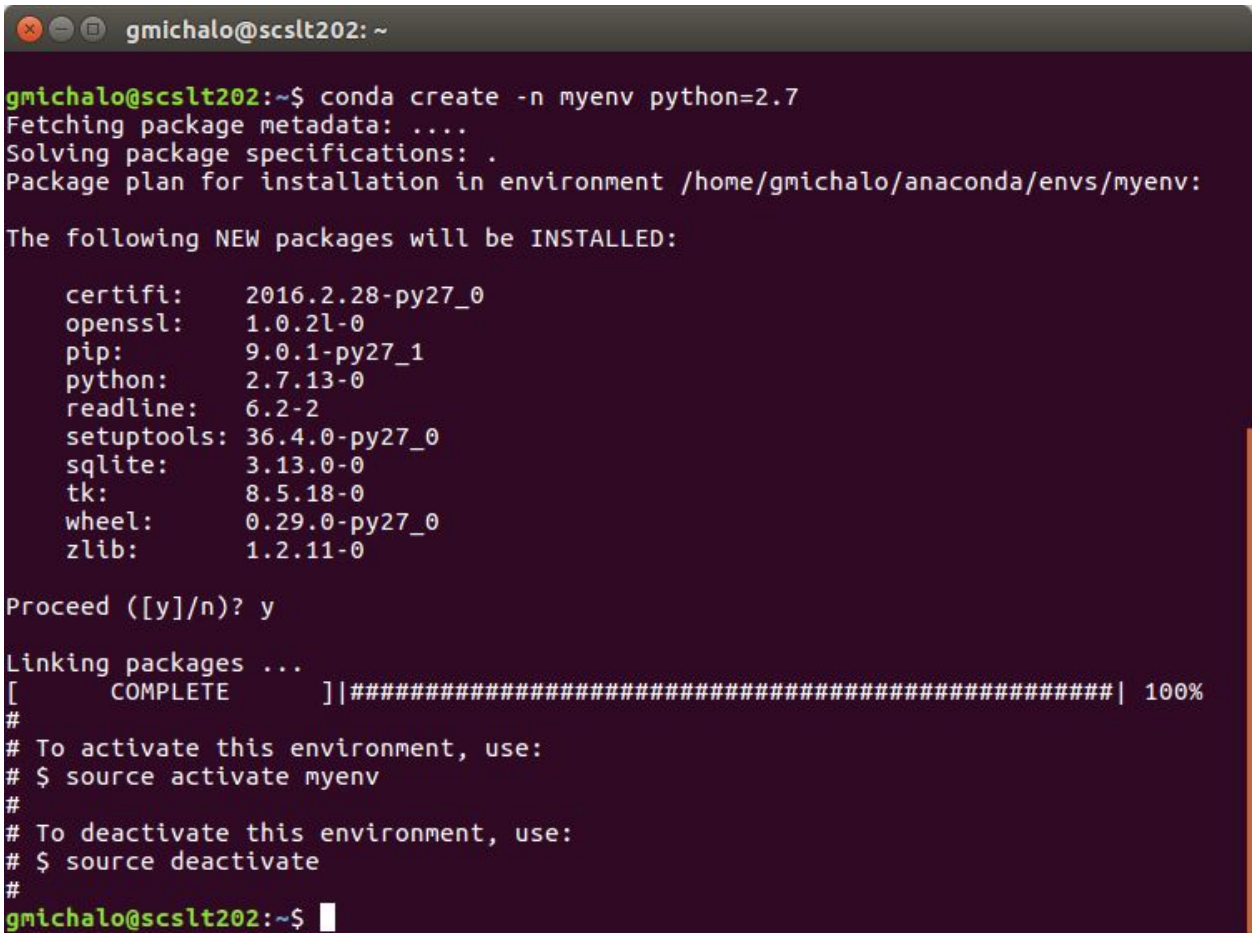
   If you are unsure about any setting, accept the defaults. You can change them later.
4. To make the changes take effect, close and then re-open your Terminal window.
5. Test your installation.

Now you have Anaconda you can create a new environment using Python 2.7 where you run the all the other programs

We can create a new environment by writing in a terminal

conda create -n myenv python=2.7

Where myenv is the name of your environment. If everything is fine you should see this message on your screen.



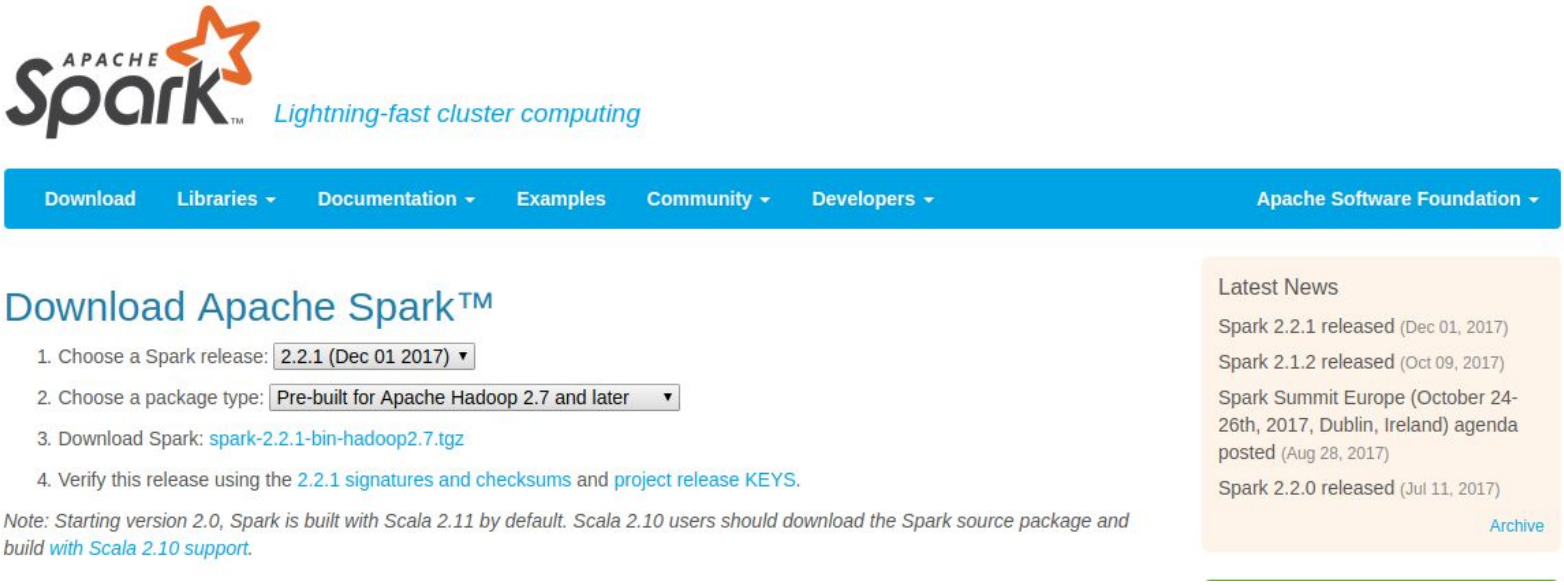
In order to activate this environment you should use the command:
Source activate myenv

And if you want to deactivate this environment you should use:
Source deactivate

0.2 Download and spark and pyspark and jupyter notebook

Visiting the web page

In order to download spark you should first go to the web page:
https://spark.apache.org/downloads.html
And choose the spark release you want to download. For our project, you should choose the options that you can see in the picture:



And then you press to download spark-2.2.0-bin-hadoop2.7.tgz

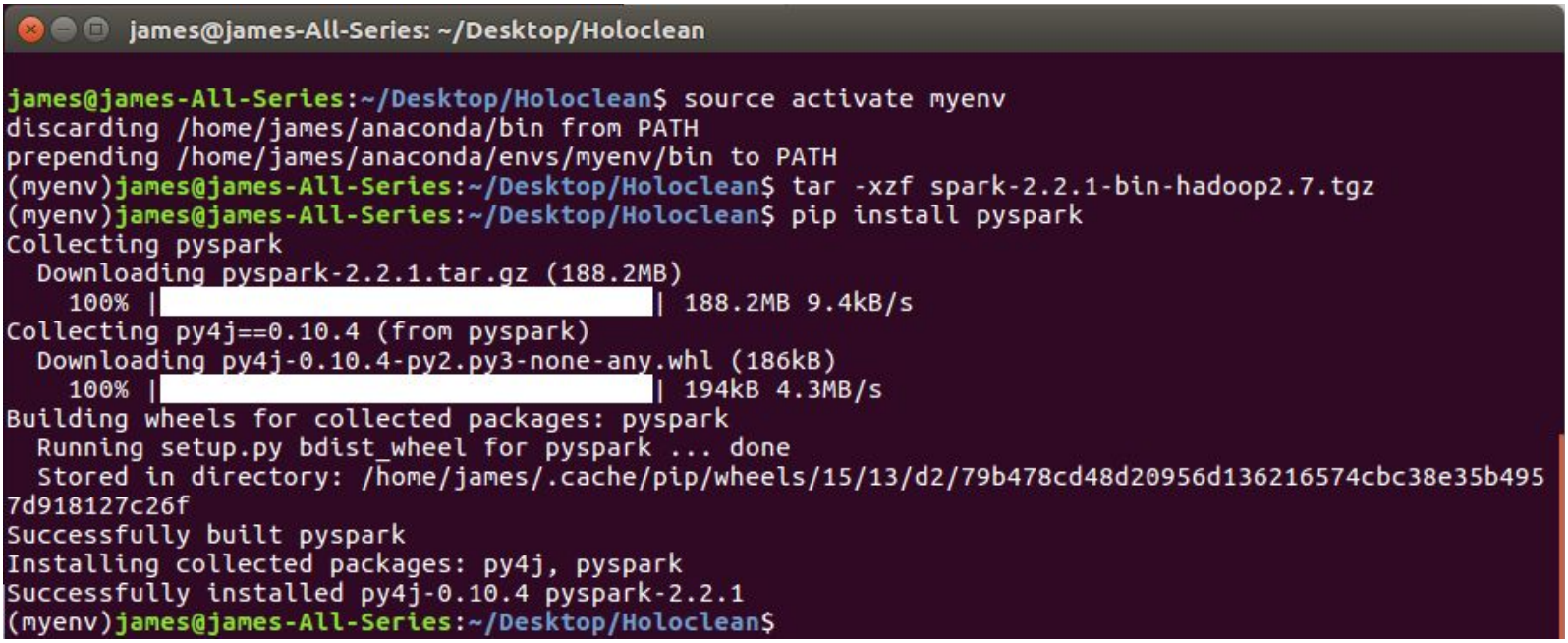
Installing spark and pyspark

After you download the tgz file you open the environment you create before (with **Source activate myenv** where myenv is the name of your environment)

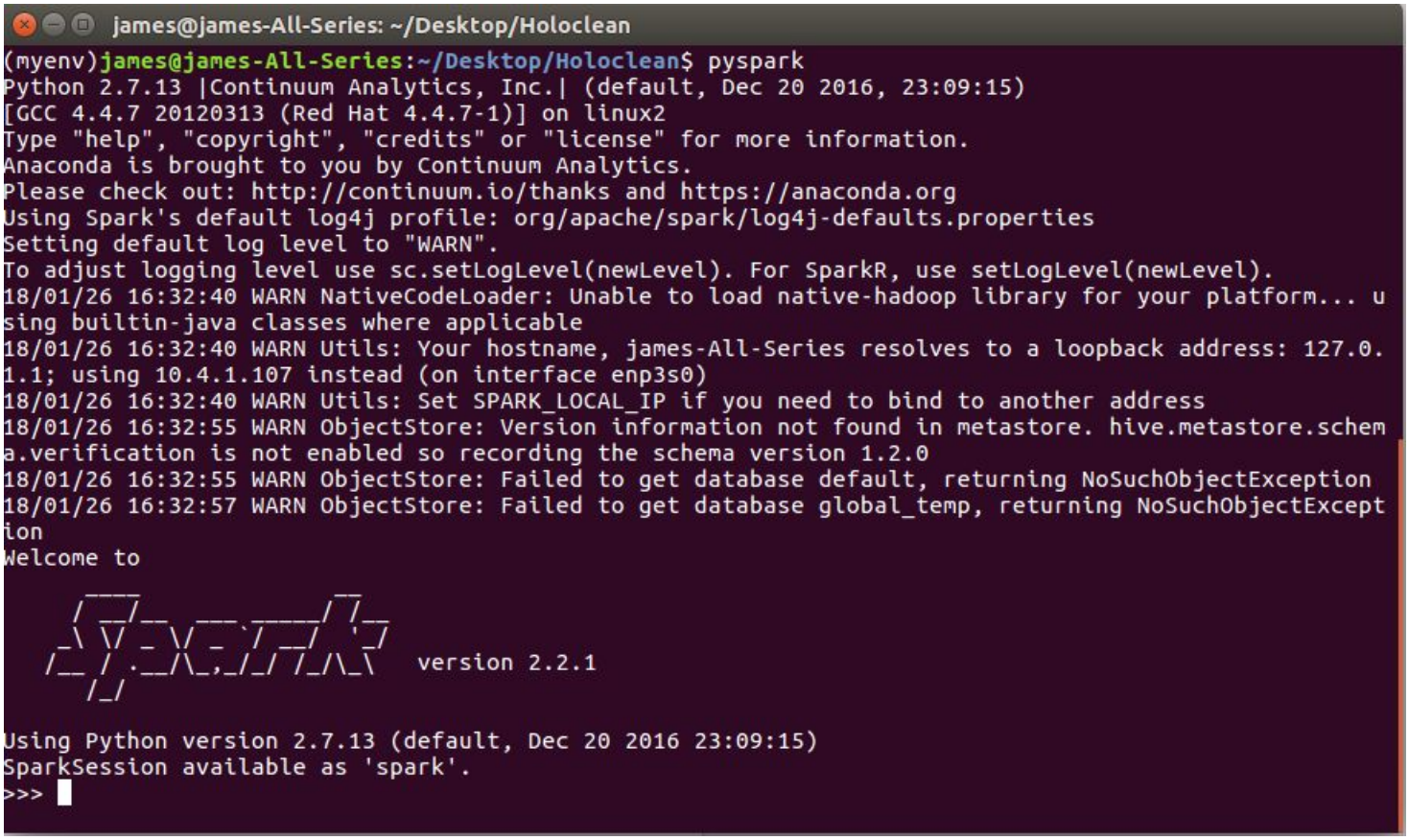
you go the folder where you downloaded the file:

And write the following command to extract the files
tar -xzf spark-2.2.0-bin-hadoop2.7.tgz

Then you download pyspark by using the following command on the terminal:
pip install pyspark



So if everything is working fine if you type:
pyspark
in your terminal you should see this image in your terminal



Congratulations !! we have just successfully install Spark and pyspark
If you want to leave the environment of spark you just type ***quit()***

Installing jupyter notebook and numpy

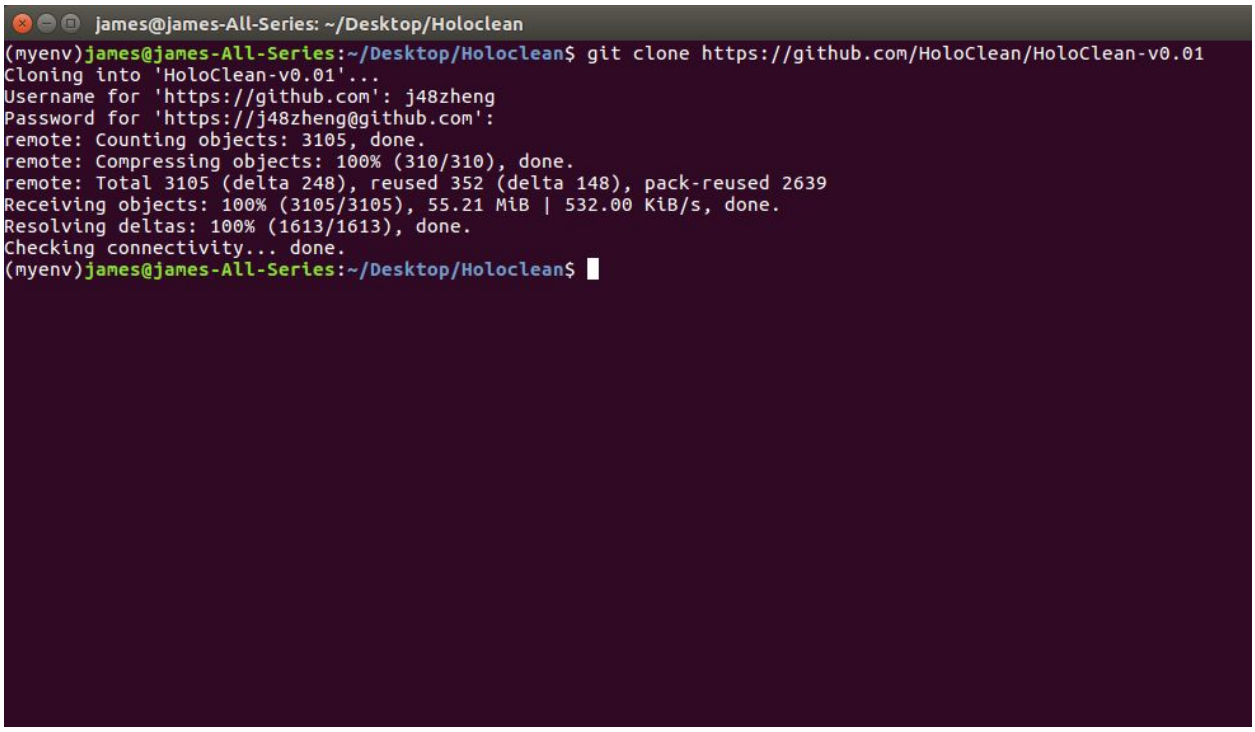
Lastly, you can now download jupyter notebook by typing :
pip install jupyter

And numpy by using the command:
pip install numpy

0.3 Download the git repo

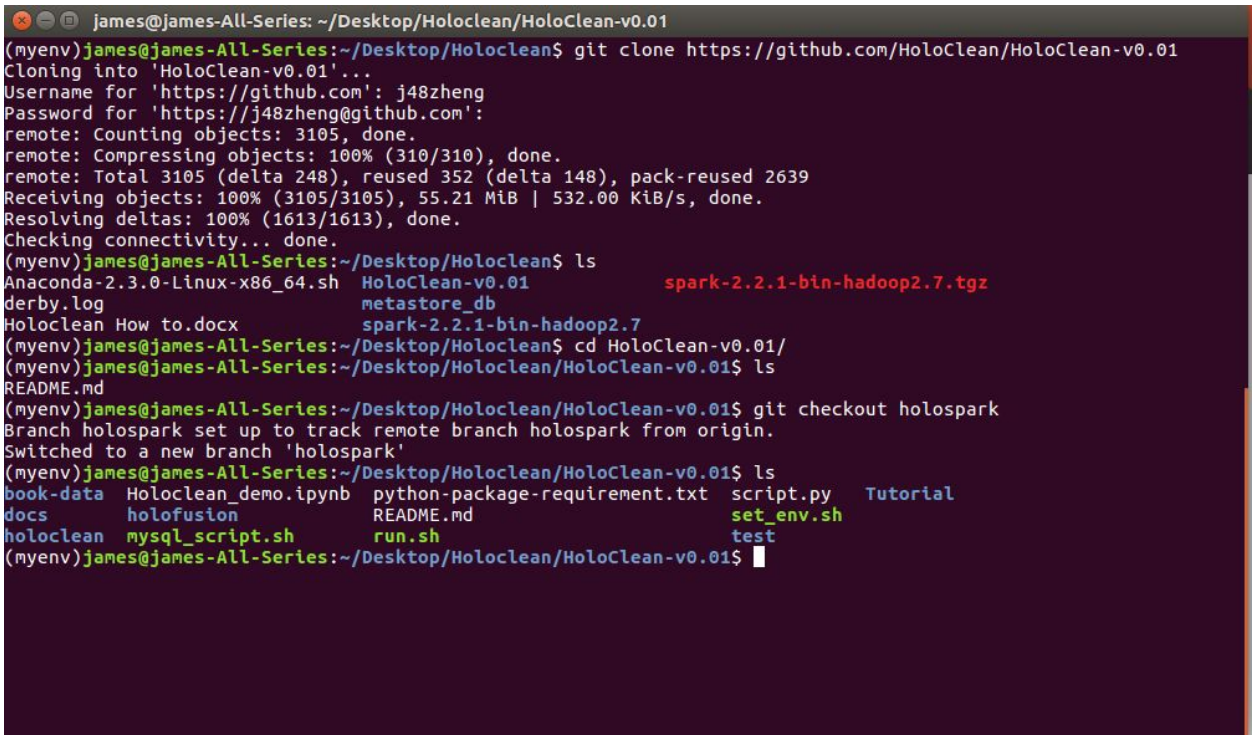
Clone the git repo

```
git clone https://github.com/HoloClean/HoloClean-v0.01
```



Check out the pytorch branch

```
git checkout pytorch
```



0.4 Setup MySql Server

For Ubuntu:

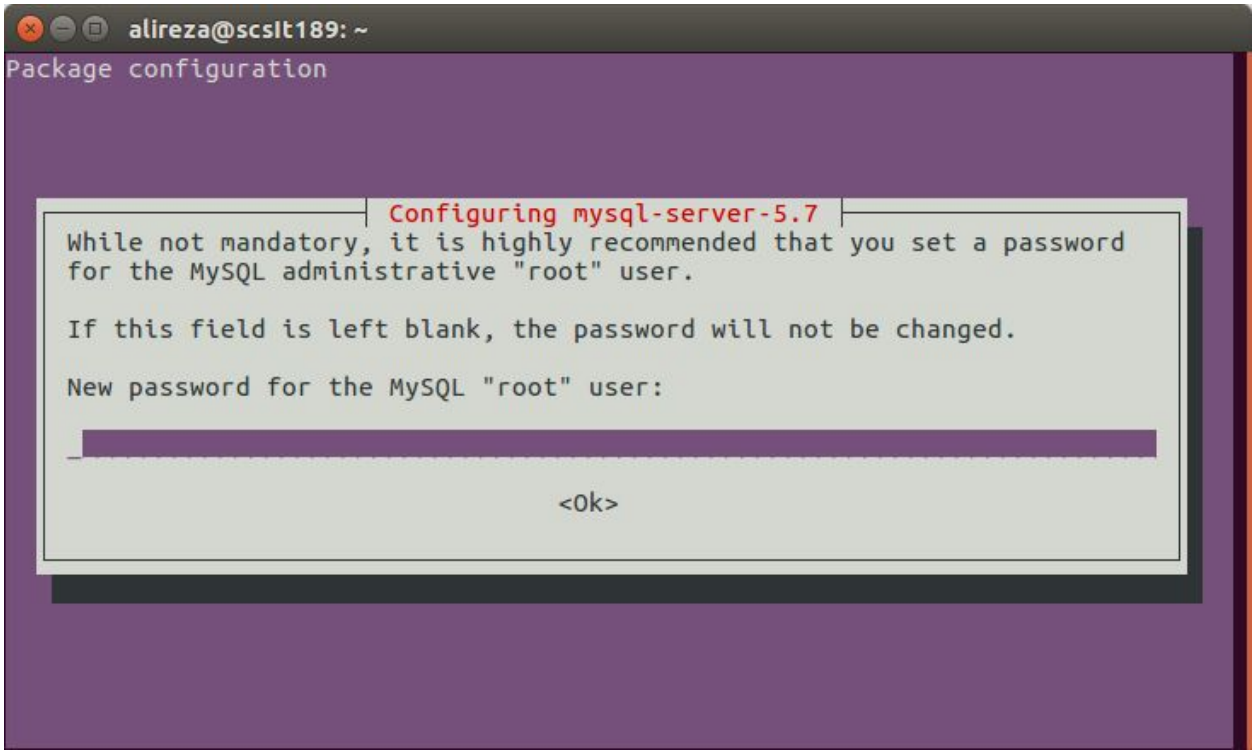
First update and upgrade your apt-get

```
sudo apt-get update
sudo apt-get upgrade
```

Then you can Install MySQL

```
sudo apt-get install mysql-server
```

The you should set password for your server



After that you repeat the password again and the installation will continue

```
alireza@scsIt189: ~
Preparing to unpack .../mysql-server-5.7_5.7.19-0ubuntu0.16.04.1_amd64.deb ...
Unpacking mysql-server-5.7 (5.7.19-0ubuntu0.16.04.1) ...
Selecting previously unselected package libhtml-template-perl.
Preparing to unpack .../libhtml-template-perl_2.95-2_all.deb ...
Unpacking libhtml-template-perl (2.95-2) ...
Selecting previously unselected package mysql-server.
Preparing to unpack .../mysql-server_5.7.19-0ubuntu0.16.04.1_all.deb ...
Unpacking mysql-server (5.7.19-0ubuntu0.16.04.1) ...
Processing triggers for man-db (2.7.5-1) ...
Processing triggers for libc-bin (2.23-0ubuntu9) ...
Processing triggers for ureadahead (0.100.0-19) ...
Processing triggers for systemd (229-4ubuntu19) ...
Setting up mysql-server-core-5.7 (5.7.19-0ubuntu0.16.04.1) ...
Setting up libevent-core-2.0-5:amd64 (2.0.21-stable-2ubuntu0.16.04.1) ...
Setting up mysql-server-5.7 (5.7.19-0ubuntu0.16.04.1) ...
update-alternatives: using /etc/mysql/mysql.cnf to provide /etc/mysql/my.cnf (my
.cnf) in auto mode
Renaming removed key_buffer and myisam-recover options (if present)
Setting up libhtml-template-perl (2.95-2) ...
Setting up mysql-server (5.7.19-0ubuntu0.16.04.1) ...
Processing triggers for libc-bin (2.23-0ubuntu9) ...
Processing triggers for systemd (229-4ubuntu19) ...
Processing triggers for ureadahead (0.100.0-19) ...
alireza@scsIt189:~$
```

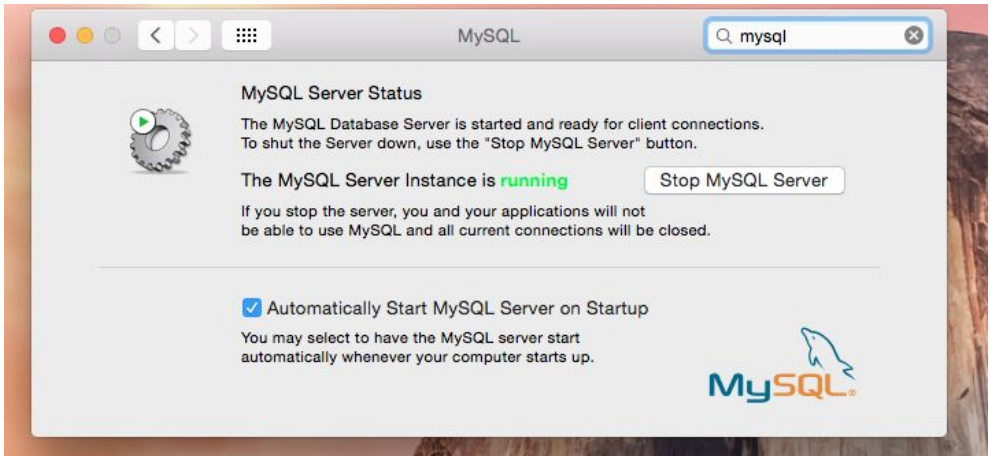
After this part we should install the service for client (python as client)

```
sudo apt-get install python-dev libmysqlclient-dev
```

For MacOS:

Install and run the .dmg file from https://dev.mysql.com/downloads/mysql/

After the installation is finished, open system preferences and click on the MySQL icon and make sure the MySQL Server Instance is running

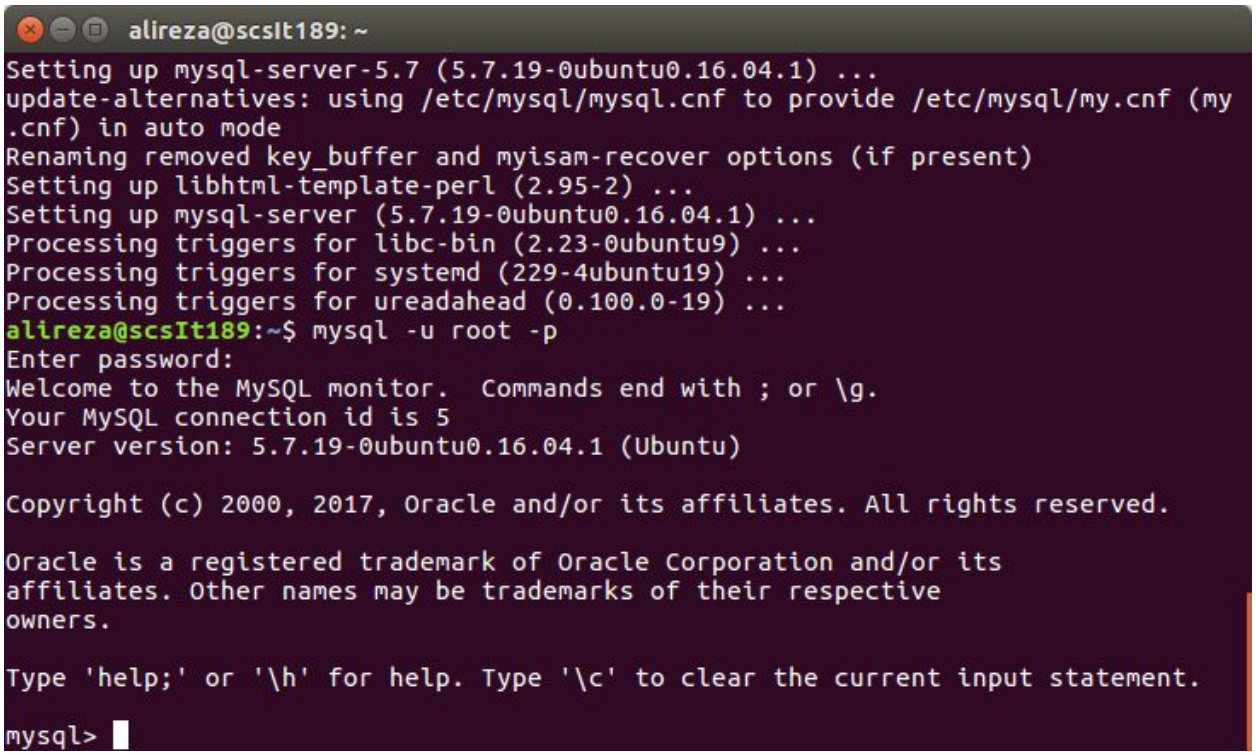


Then open terminal and run:

```
sudo usr/local/mysql/bin/mysql_secure_installation
```

The standard tool for interacting with MySQL is the mysql client, which installs with the mysql-server package.

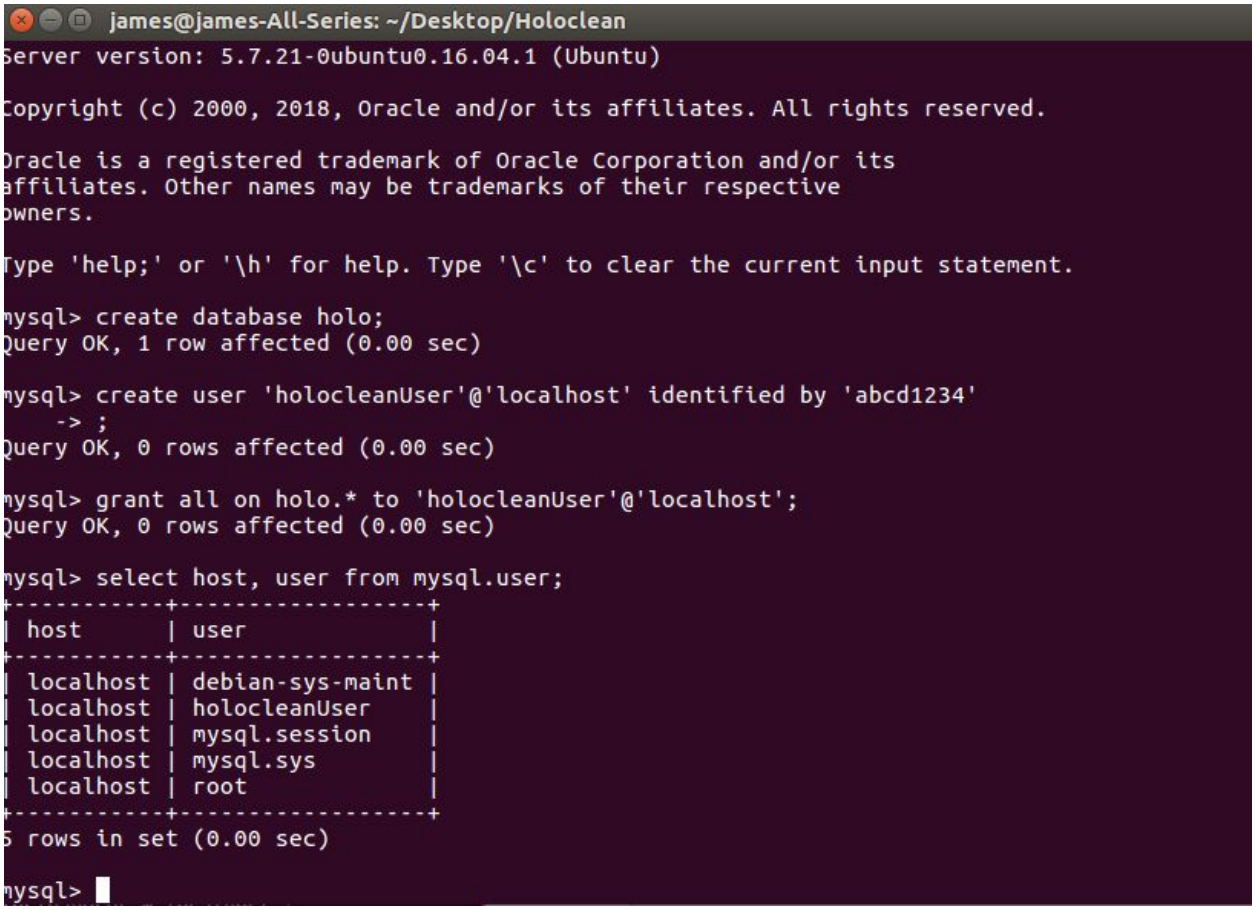
```
mysql -u root -p
```



Create a New MySQL User and Database

run script create the MySQL server

```
./mysql_script.sh
```



Now you can access to the database by your user setting

0.5- Installing required packages

Install required packages by running:

```
pip install python-package-requirement.txt
```

0.6- Install pytorch

Follow instructions at:
http://pytorch.org/
To install pytorch for your OS
Make sure to install version 0.3.0 or later

1- Session

Holoclean use Spark session to be executed over large data

1.1 Download drivers for MySQL

In order to use spark and Mysql, we need to download the drivers from:

https://dev.mysql.com/downloads/connector/j/5.1.html

Afterwards we untar the file and when we start a spark session in holoclean we put as an argument the whole path to the jar file. For example:
holoclean_se._start_spark_session("/home/user/Downloads/mysql-connector-java-5.1.44/mysql-connector-java-5.1.44-bin.jar")

2- Preliminaries

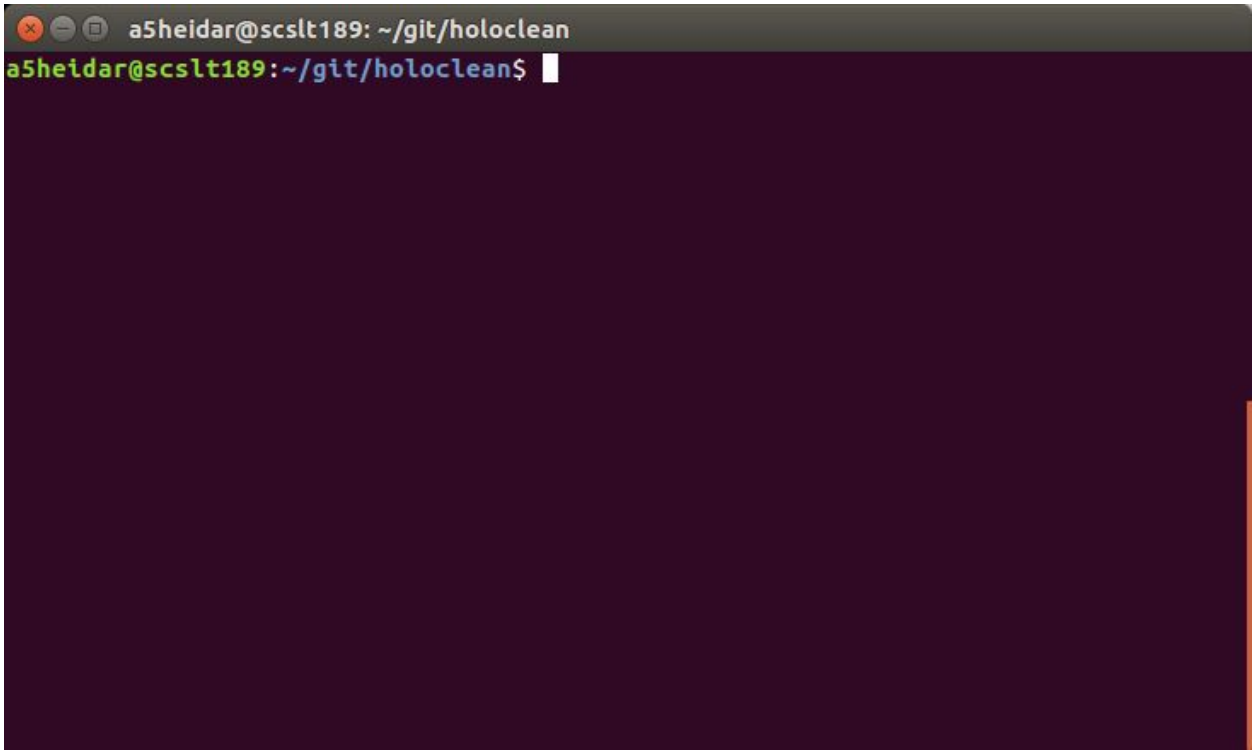
2.1 Hello Git!

To be part of team you should know how to communicate with them as you might know programmer language is code and their social network is Github so in this section we would learn how to connect to git.

Requirement :

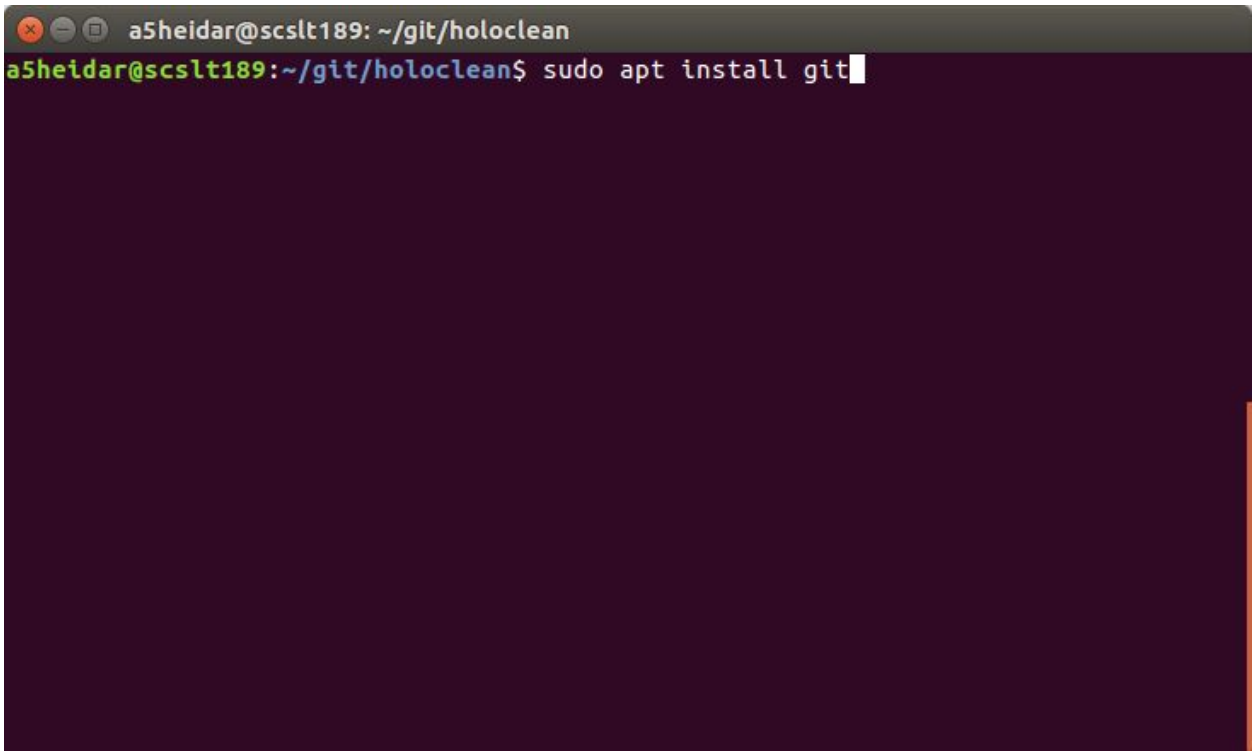
- Linux
- Github Account
- Internet access

Open your terminal

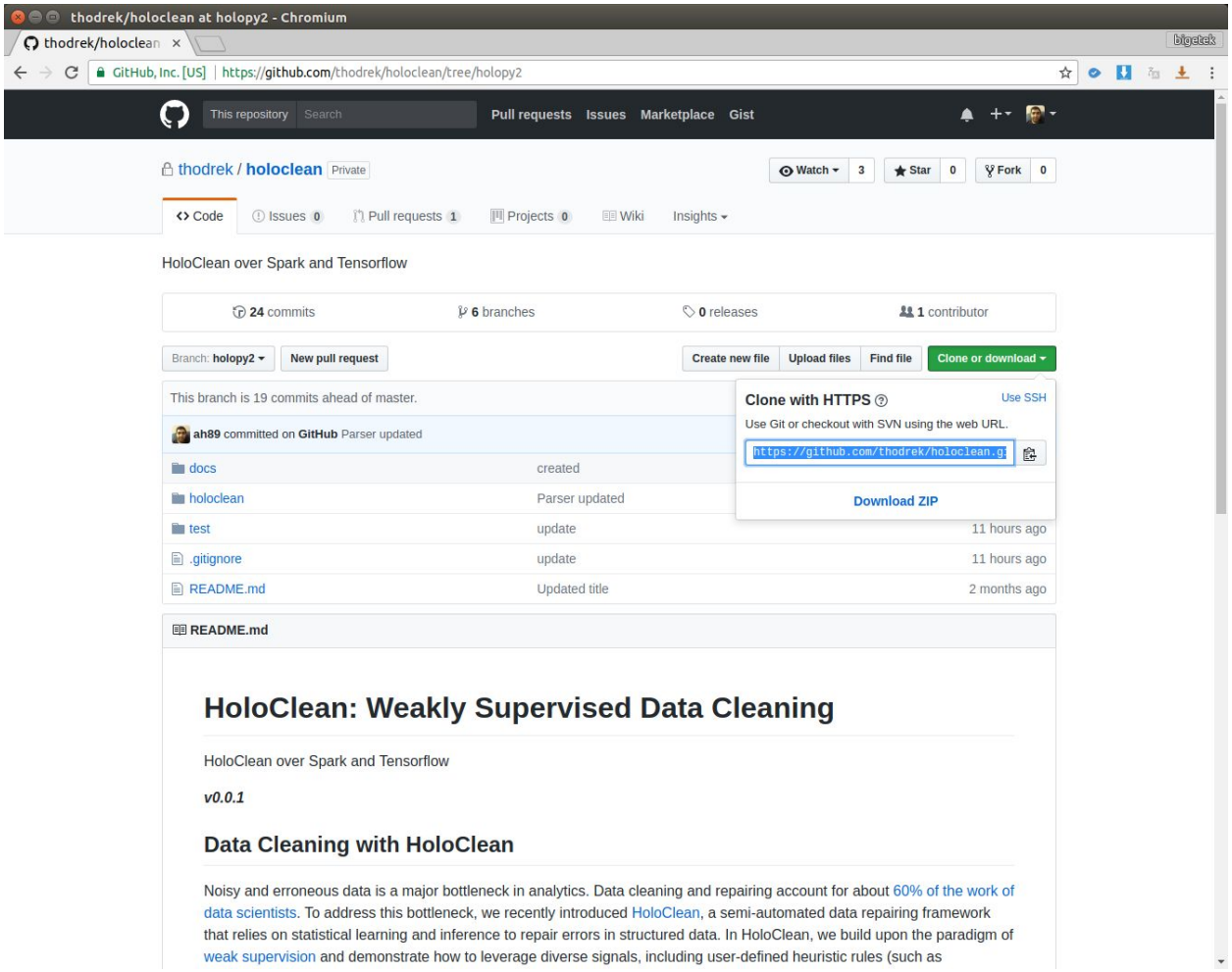


For step you need to **install git** for this you can use this command

sudo apt install git



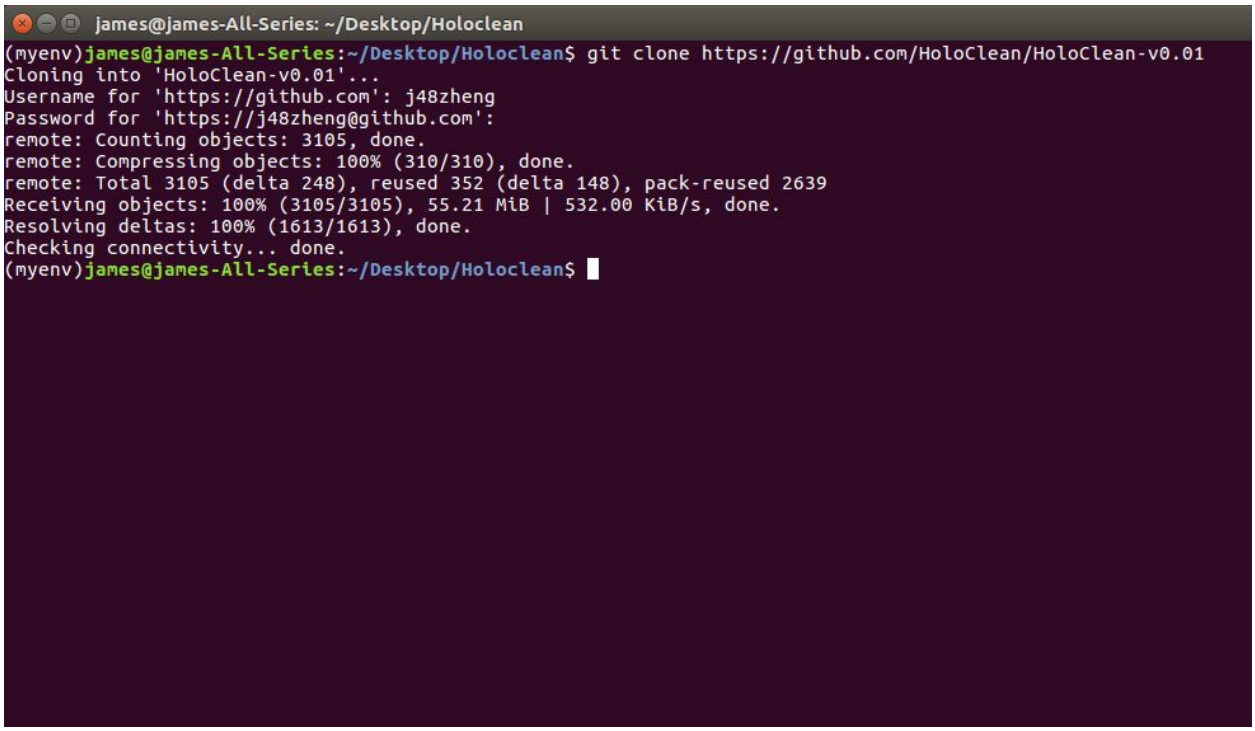
And after that you can access to the git directory that given to you or you can get it after invited to a github project. For geting clone address go to your dashboard in your github account and on the top right click on download and clone and get the address clone

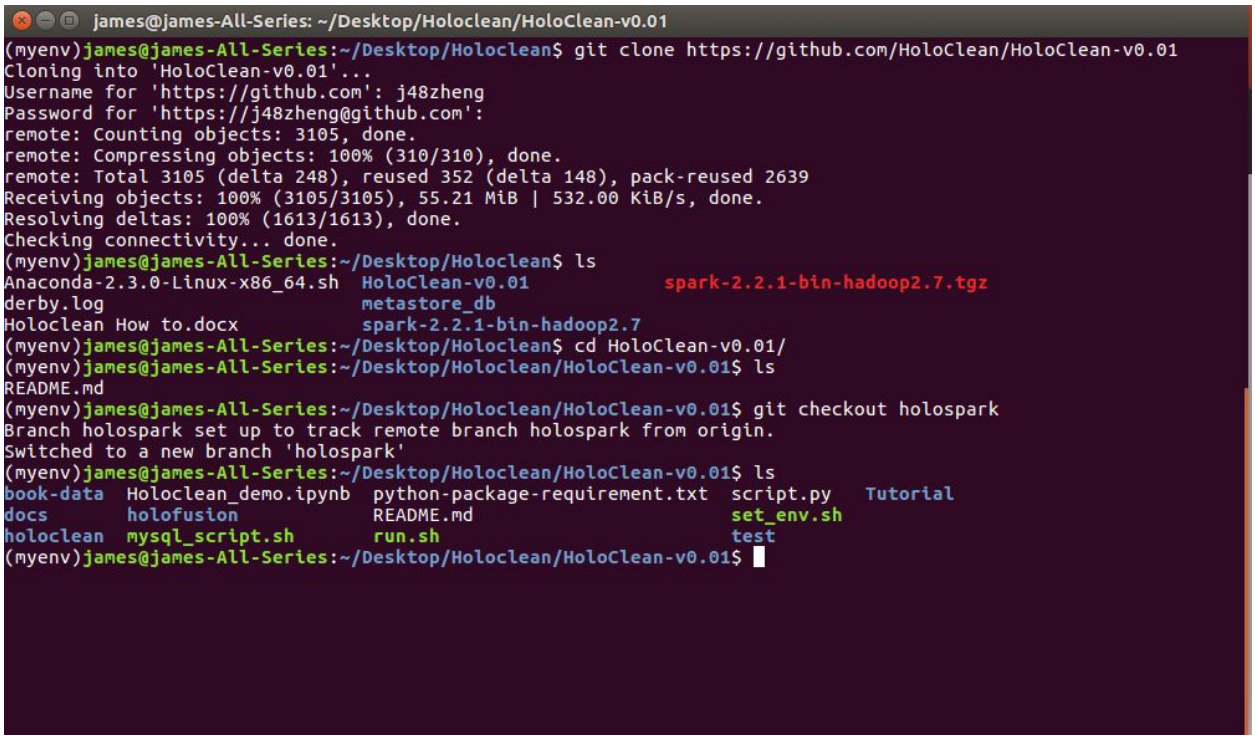


Copy that address and keep it!
For a project first you need to make directory in your local machine so we made folder with name ‘git’ on my machine and then go in that direction we want to clone the data from that git that we get it's address from our dashboard as you can see in following:

git clone https://github.com/HoloClean/HoloClean-v0.01

After that it will ask your Github username and password and then you have git files on your local machine.

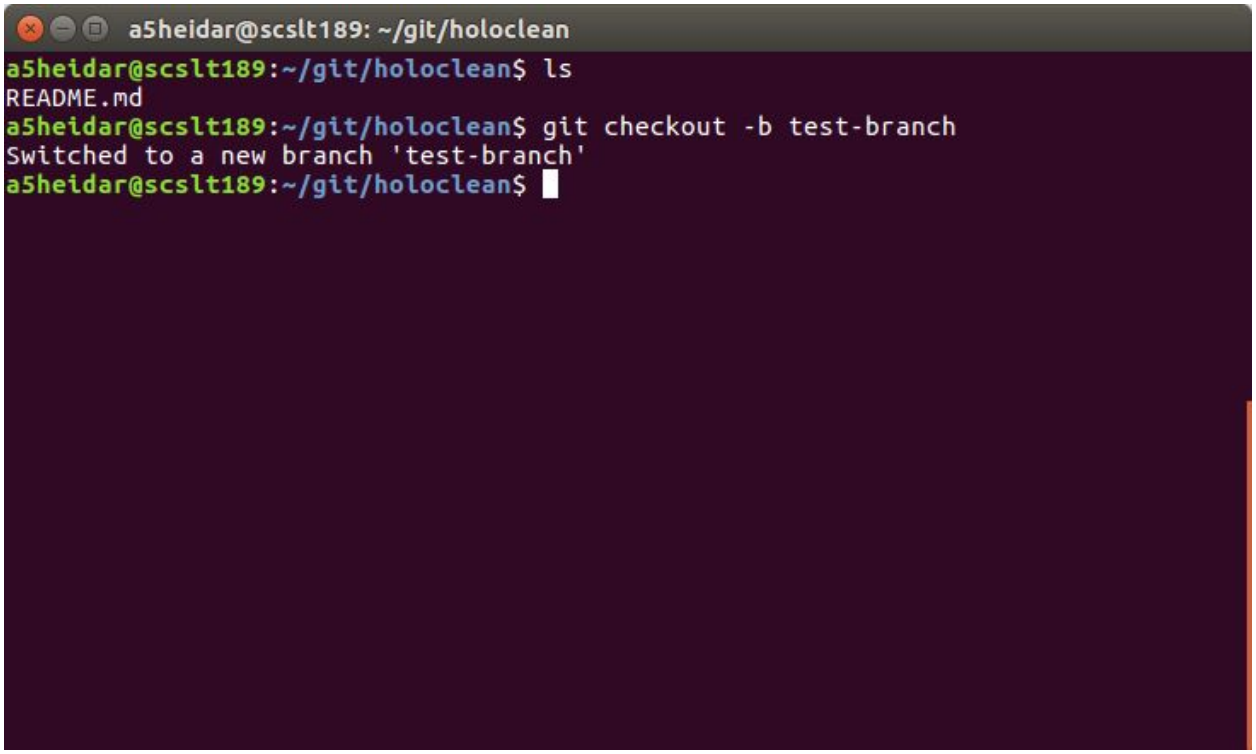

james@james-All-Series: ~/Desktop/Holoclean
(myenv)james@james-All-Series:~/Desktop/Holoclean$ git clone https://github.com/HoloClean/HoloClean-v0.01
Cloning into 'HoloClean-v0.01'...
Username for 'https://github.com': j48zheng
Password for 'https://j48zheng@github.com':
remote: Counting objects: 3105, done.
remote: Compressing objects: 100% (310/310), done.
remote: Total 3105 (delta 248), reused 352 (delta 148), pack-reused 2639
Receiving objects: 100% (3105/3105), 55.21 MiB | 532.00 KiB/s, done.
Resolving deltas: 100% (1613/1613), done.
Checking connectivity... done.
(myenv)james@james-All-Series:~/Desktop/Holoclean$


james@james-All-Series: ~/Desktop/Holoclean/HoloClean-v0.01
(myenv)james@james-All-Series:~/Desktop/Holoclean$ git clone https://github.com/HoloClean/HoloClean-v0.01
Cloning into 'HoloClean-v0.01'...
Username for 'https://github.com': j48zheng
Password for 'https://j48zheng@github.com':
remote: Counting objects: 3105, done.
remote: Compressing objects: 100% (310/310), done.
remote: Total 3105 (delta 248), reused 352 (delta 148), pack-reused 2639
Receiving objects: 100% (3105/3105), 55.21 MiB | 532.00 KiB/s, done.
Resolving deltas: 100% (1613/1613), done.
Checking connectivity... done.
(myenv)james@james-All-Series:~/Desktop/Holoclean$ ls
Anaconda-2.3.0-Linux-x86_64.sh HoloClean-v0.01 spark-2.2.1-bin-hadoop2.7.tgz
derby.log metastore_db
Holoclean How to.docx spark-2.2.1-bin-hadoop2.7
(myenv)james@james-All-Series:~/Desktop/Holoclean$ cd HoloClean-v0.01/
(myenv)james@james-All-Series:~/Desktop/Holoclean/HoloClean-v0.01$ ls
README.md
(myenv)james@james-All-Series:~/Desktop/Holoclean/HoloClean-v0.01$ git checkout holospark
Branch holospark set up to track remote branch holospark from origin.
Switched to a new branch 'holospark'
(myenv)james@james-All-Series:~/Desktop/Holoclean/HoloClean-v0.01$ ls
book-data Holoclean_demo.ipynb python-package-requirement.txt script.py Tutorial
docs holofusion README.md set_env.sh
holoclean mysql_script.sh run.sh test
(myenv)james@james-All-Series:~/Desktop/Holoclean/HoloClean-v0.01$

2.2 Branching

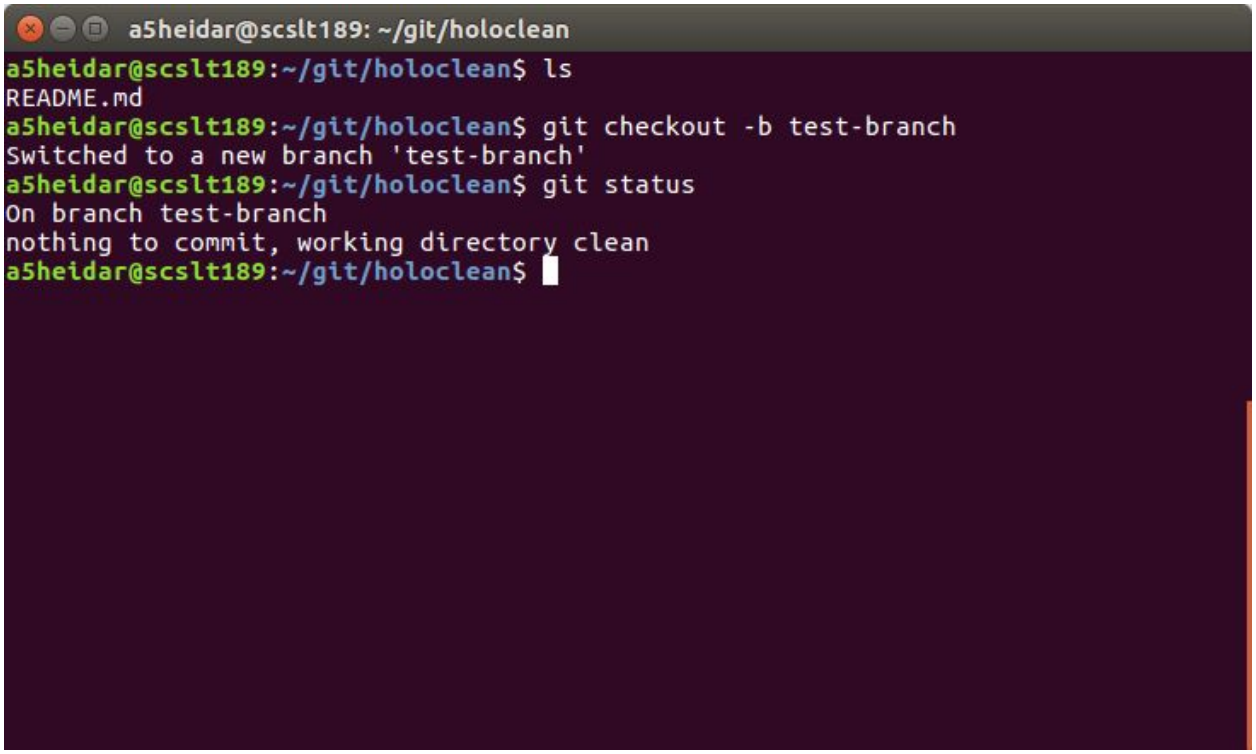
Git branches are gloriously excellent for safely making and testing changes. We can make branch by following command:

git checkout -b test-branch



If you want to see your branch you can use the following commands:

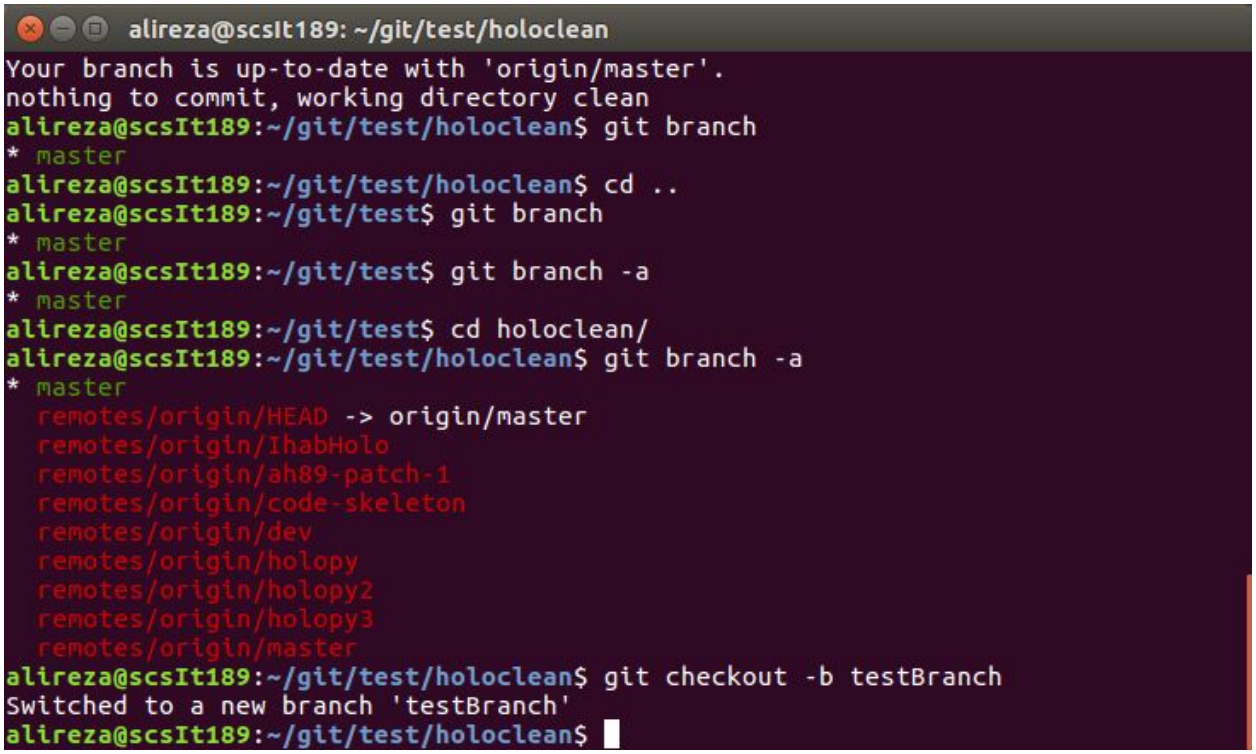
git status



2.3 Merge codes from a direction

If you want to start from somewhere that another branch is (not from scratch beginning) you can merge the code from there on your own branch

In here we made a new branch ‘testBranch’



i

Now as you can see in the image we want fill our branch with another branch code and start from where that branch is for this we need to merge our branch code with other.The command that can be used is :

git merge <another branch code>

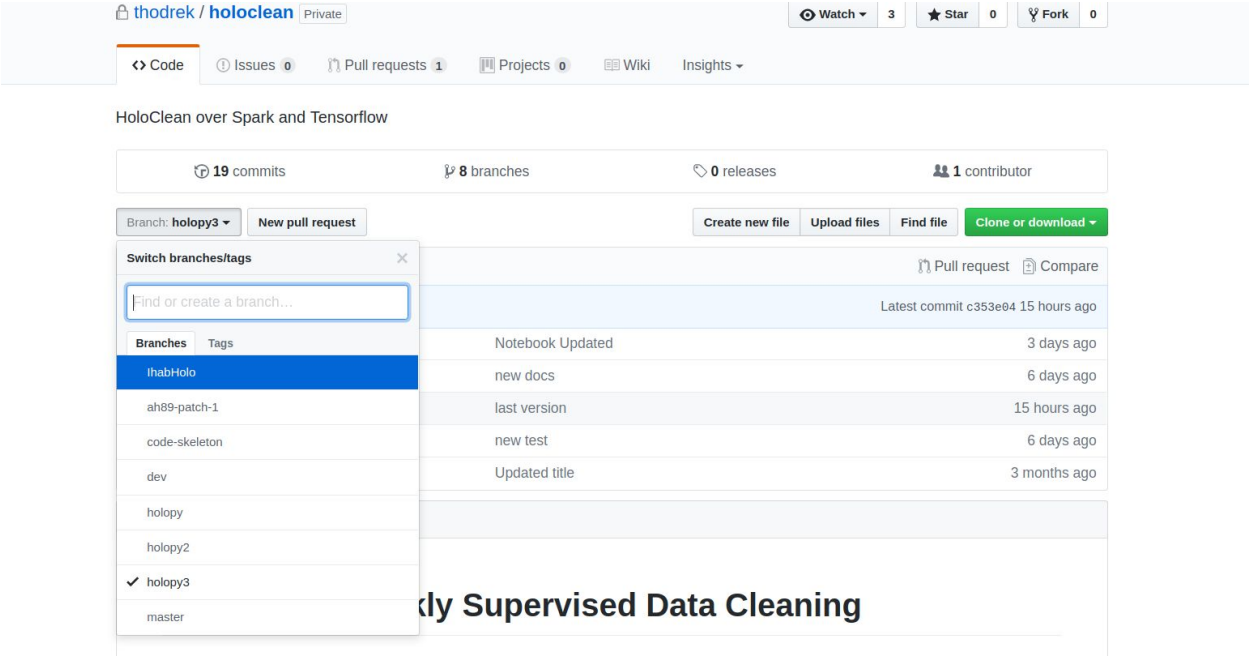
```
alireza@scsIt189: ~/git/test/holoclean
  remotes/origin/ah89-patch-1
  remotes/origin/code-skeleton
  remotes/origin/dev
  remotes/origin/holopy
  remotes/origin/holopy2
  remotes/origin/holopy3
  remotes/origin/master
alireza@scsIt189:~/git/test/holoclean$ git checkout -b testBranch
Switched to a new branch 'testBranch'
alireza@scsIt189:~/git/test/holoclean$ git status
On branch testBranch
nothing to commit, working directory clean
alireza@scsIt189:~/git/test/holoclean$ git status
On branch testBranch
nothing to commit, working directory clean
alireza@scsIt189:~/git/test/holoclean$ git merge holopy3
merge: holopy3 - not something we can merge

Did you mean this?
        origin/holopy3
alireza@scsIt189:~/git/test/holoclean$ git merge origin/holopy3
Updating 0c8fcd6..c353e04
Fast-forward
 .../.ipynb_checkpoints/Grounding-checkpoint.ipynb  | 543 ++++++++++++++++++
```

Now if you list your file you can see that you have same files from your target branch on your own branch.

```
alireza@scsIt189: ~/git/test/holoclean
 create mode 100644 holoclean/metastore_db/seg0/ca11.dat
 create mode 100644 holoclean/metastore_db/seg0/ca21.dat
 create mode 100644 holoclean/metastore_db/seg0/cb1.dat
 create mode 100644 holoclean/metastore_db/seg0/cc0.dat
 create mode 100644 holoclean/metastore_db/seg0/cd1.dat
 create mode 100644 holoclean/metastore_db/seg0/ce1.dat
 create mode 100644 holoclean/metastore_db/seg0/cf0.dat
 create mode 100644 holoclean/metastore_db/service.properties
 create mode 100644 holoclean/models/__init__.py
 create mode 100755 holoclean/models/learning_framework.py
 create mode 100644 holoclean/models/learning_framework.pyc
 create mode 100755 holoclean/repairingEngine.py
 create mode 100644 holoclean/test.csv
 create mode 100755 holoclean/utilities/data_parser.py
 create mode 100644 holoclean/utilities/data_parser.pyc
 create mode 100644 test/learning_test.py
 create mode 100644 test/prediction_test.py
 create mode 100644 test/tests.txt
alireza@scsIt189:~/git/test/holoclean$ ls
docs  holoclean  README.md  test  Tutorial
alireza@scsIt189:~/git/test/holoclean$ git status
On branch testBranch
nothing to commit, working directory clean
alireza@scsIt189:~/git/test/holoclean$
```

As you can see you are in your own branch but this branch is on you local so cannot see it on the github website

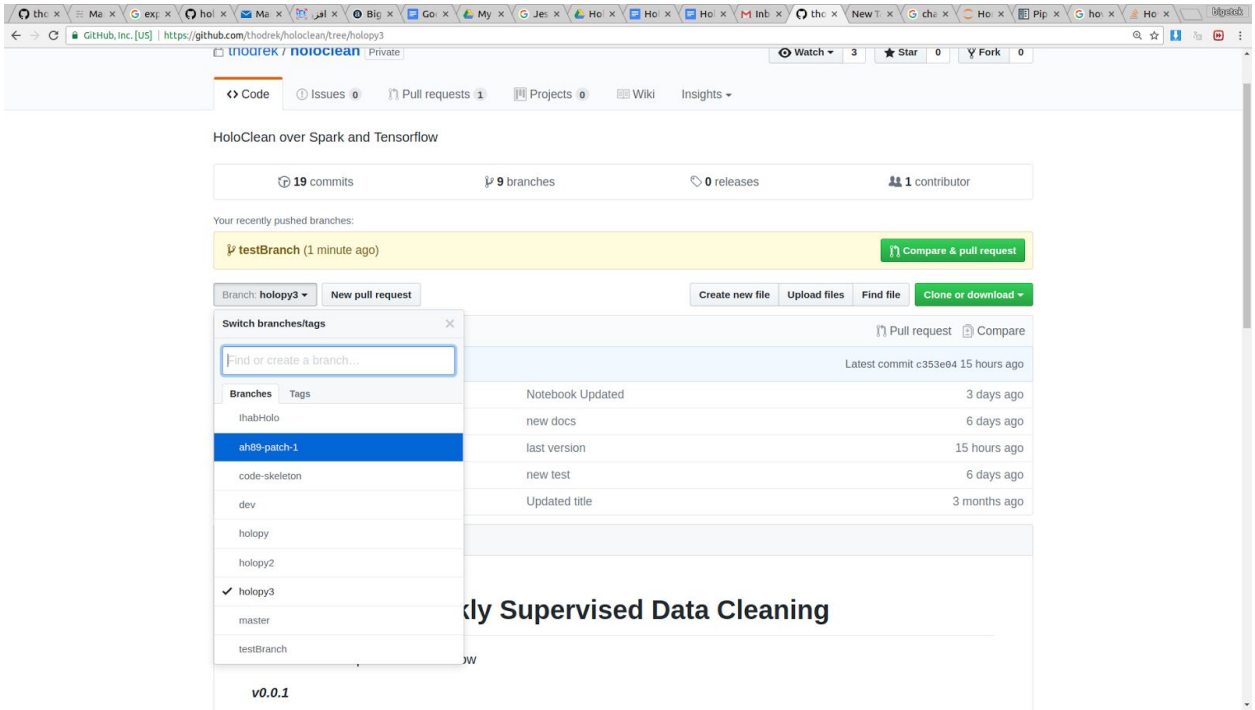


The last step is to push your code to the remote it can be done by

git push --set-upstream origin <branchName>

```
alireza@scsIt189: ~/git/test/holoclean
alireza@scsIt189:~/git/test/holoclean$ git push --set-upstream origin testBranch
Username for 'https://github.com': ah89
Password for 'https://ah89@github.com':
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/thodrek/holoclean.git
 * [new branch]      testBranch -> testBranch
Branch testBranch set up to track remote branch testBranch from origin.
alireza@scsIt189:~/git/test/holoclean$
```

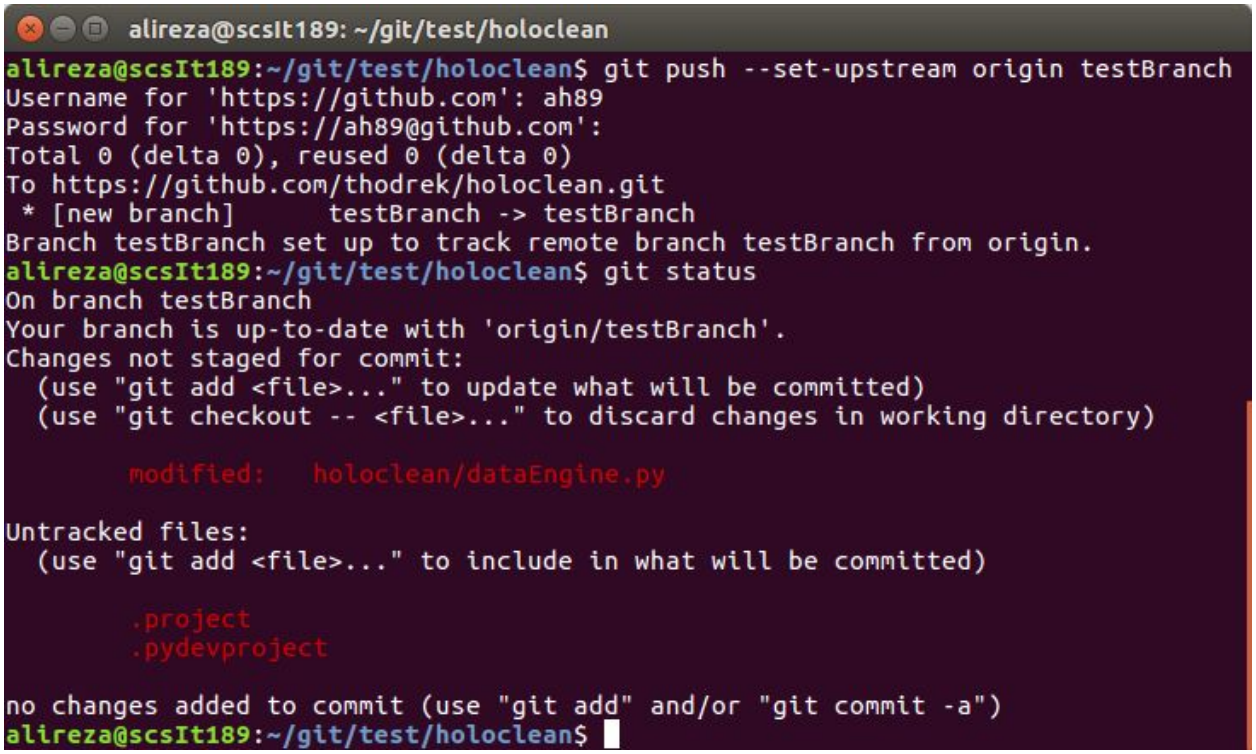
And after that you can see your branch name in the repo



2.4 Push your code to the repo

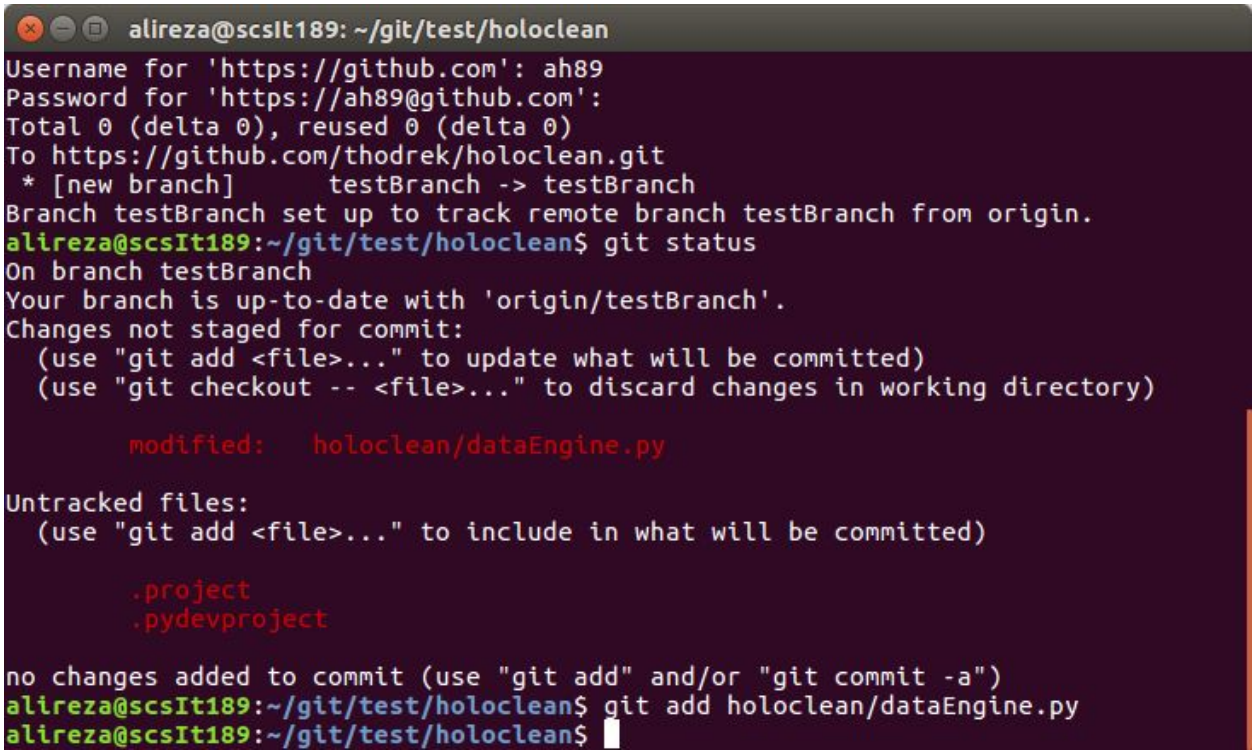
After you done with programming you can push it in your branch but a logical way is first to what changes have been made so check the status of your branch

git status



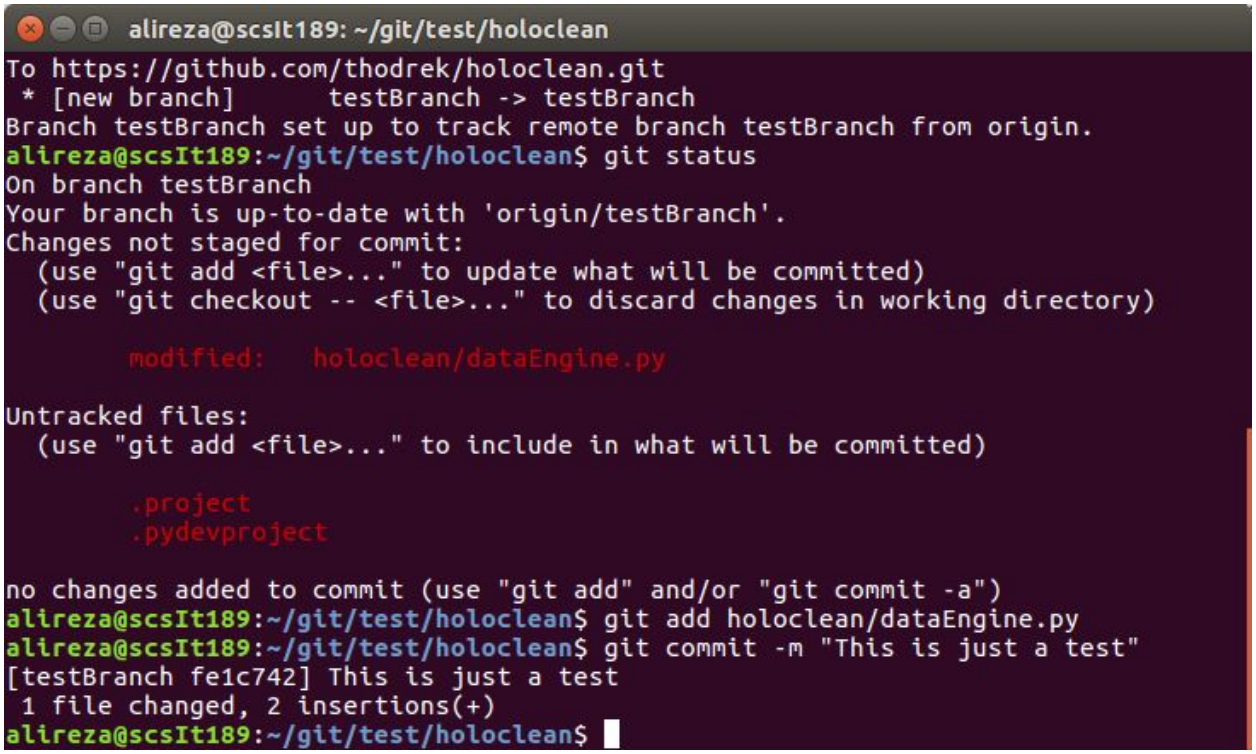
You can see the files that you have by add them to the git queue you can be sure they will update the remote repo in next push

git add <file name>



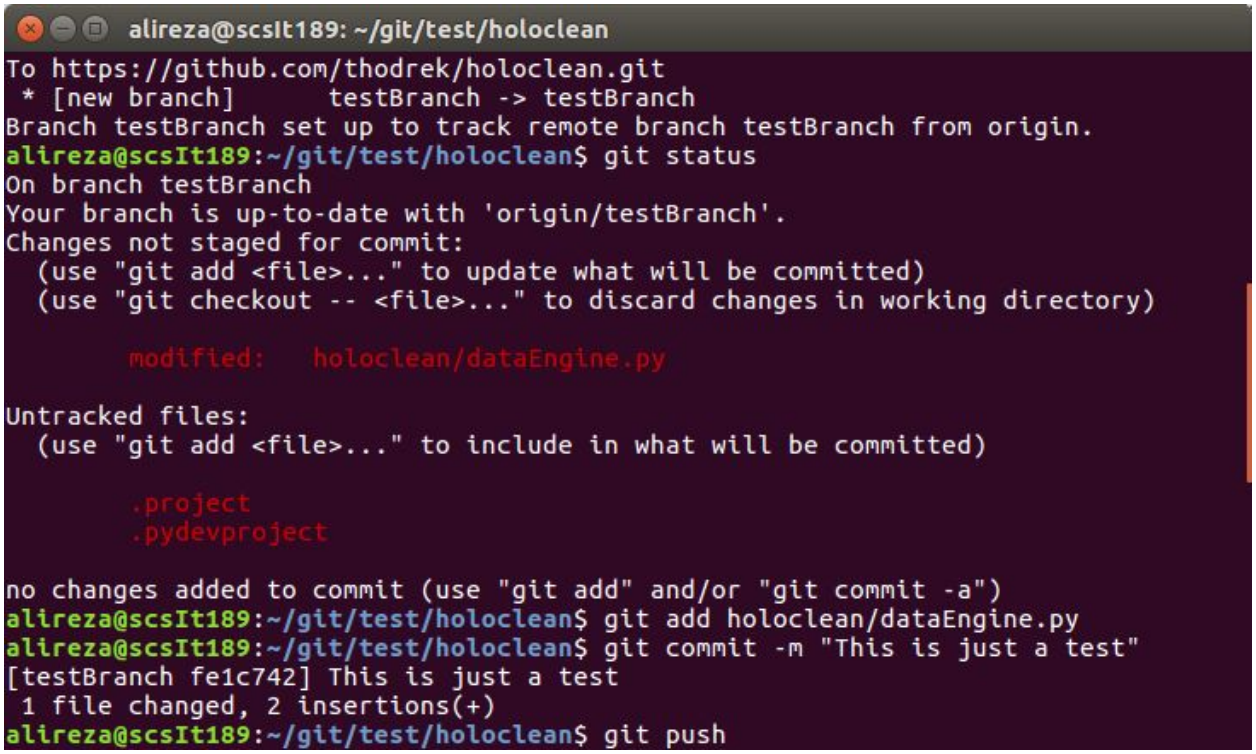
After you added this file you can make comment about your added file by using commit

git commit -m “Your comment come here!”

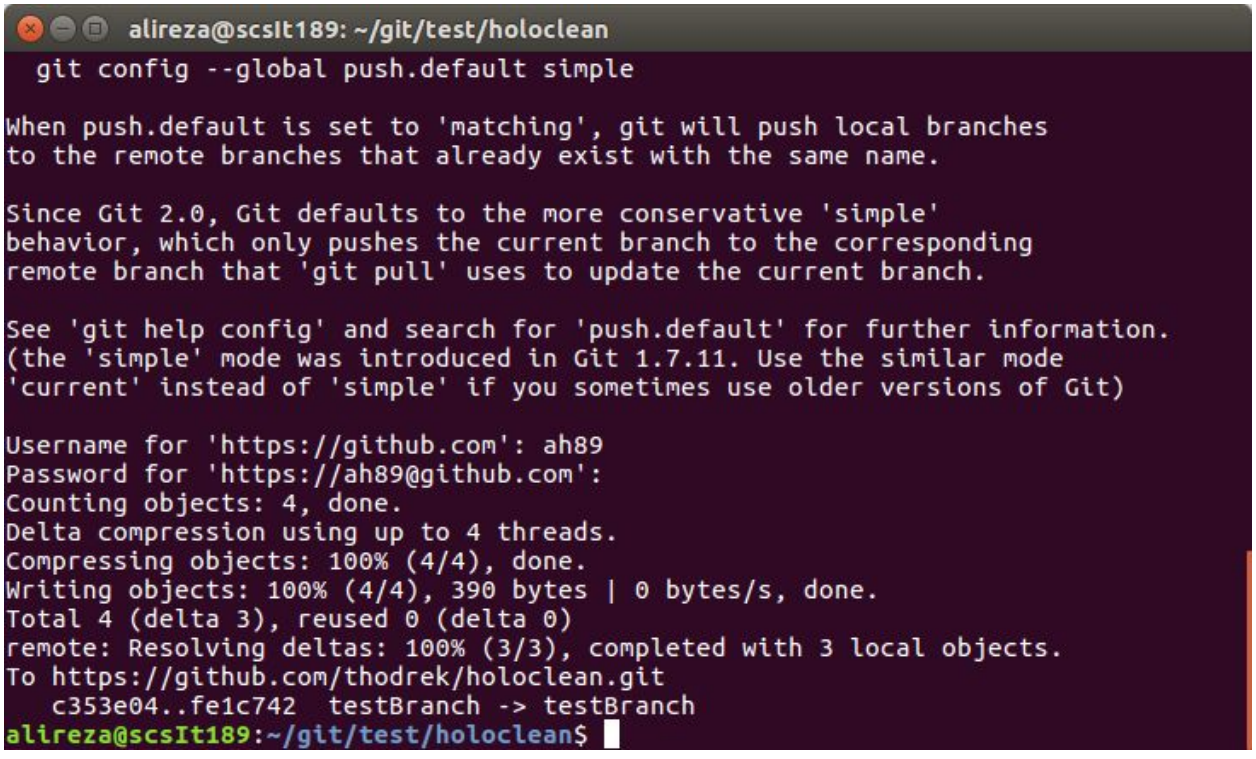


Now your code has been added to git queue. You might have some other changes and want to add them to with proper comment after that you need to push the code to your remote repo you can do it by using

git push



And you can see the push has been done successfully.



2.5 Push your code to the repo

Sometimes you might need to ask merge your code to master branch which is the original branch but before that you need to ask for review in that condition you should make a pull request. Pull request need the branch name that you want to merge and the target branch name and directory it's command scheme comes in following:

git request-pull <start> <url> [<end>]

For example if you want to feel this scheme it would be like:

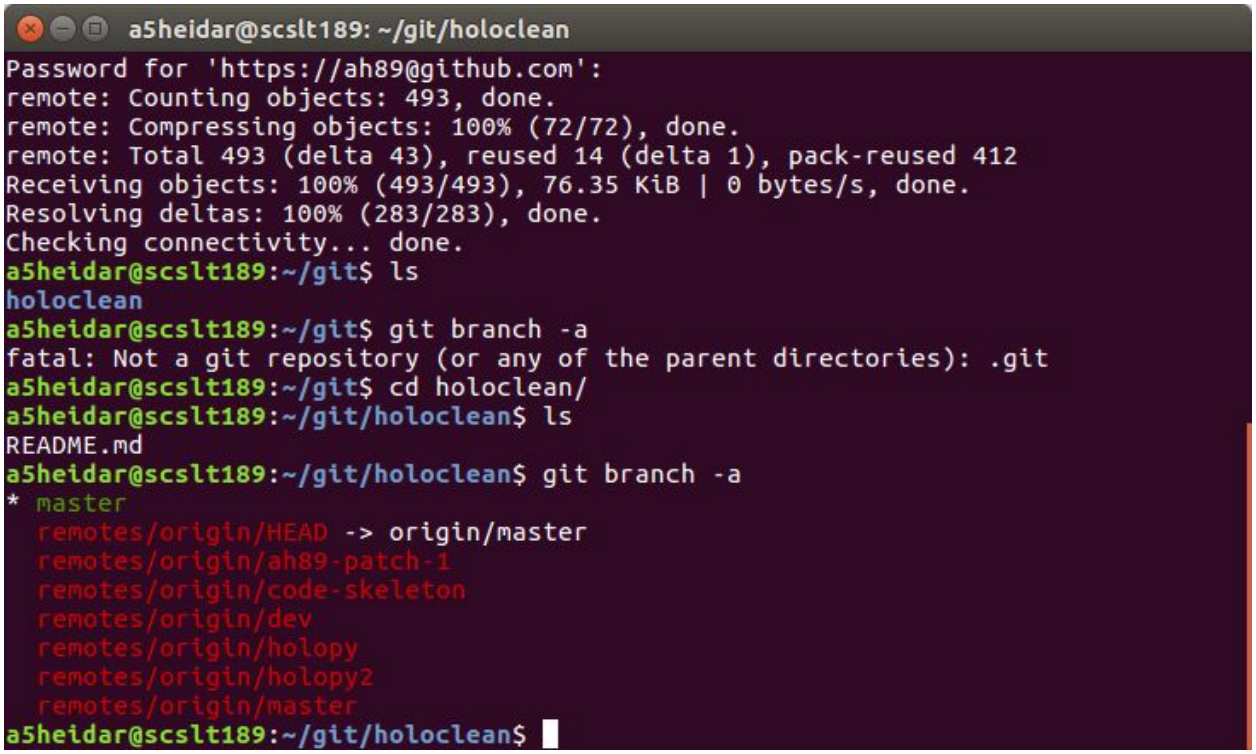
git request-pull test-branch https://github.com/thodrek/holoclean.git master

In this command make a pull request to merge our test branch to the master branch with given url.

2.6 clone a specific branch

For starting programming you need to start to continue a specific branch codes so at first we need list of branches , got to a git direction and use following command :

```
git branch -a
```



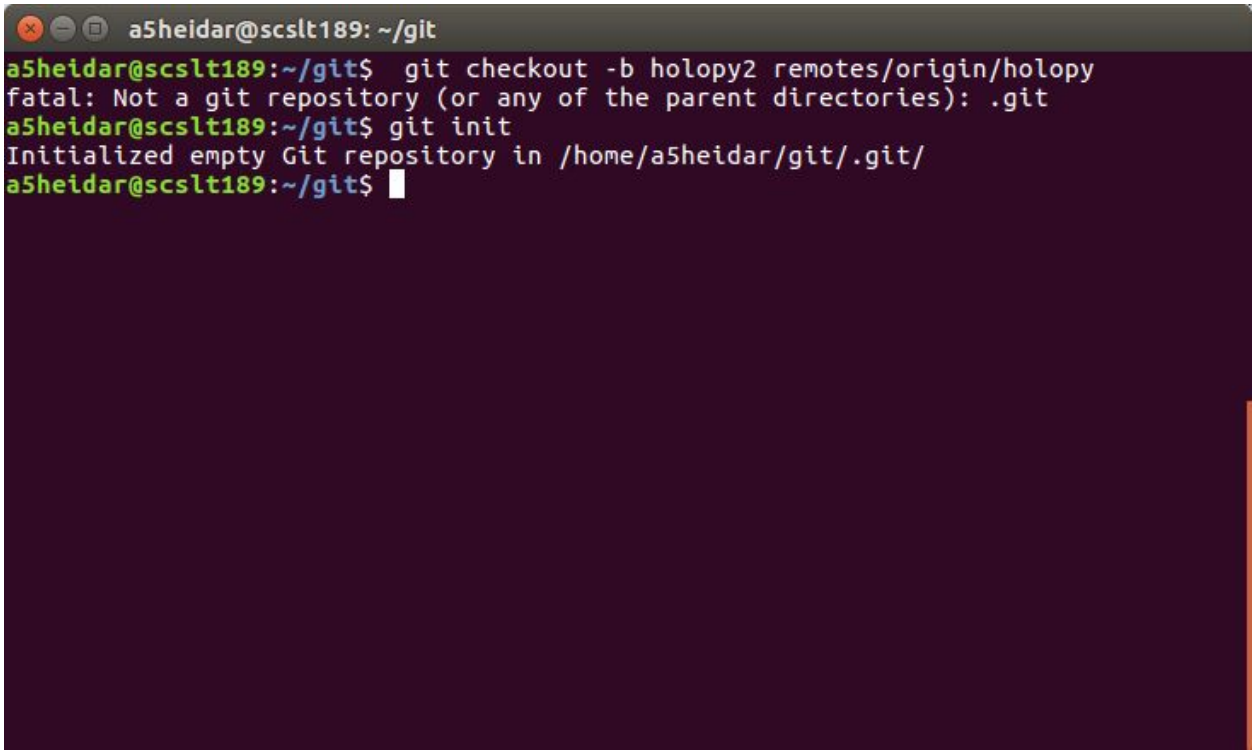
The next step is to clone the code from specific branch to our local repo. This act can be done with following command :

```
git checkout [name of branch]
```

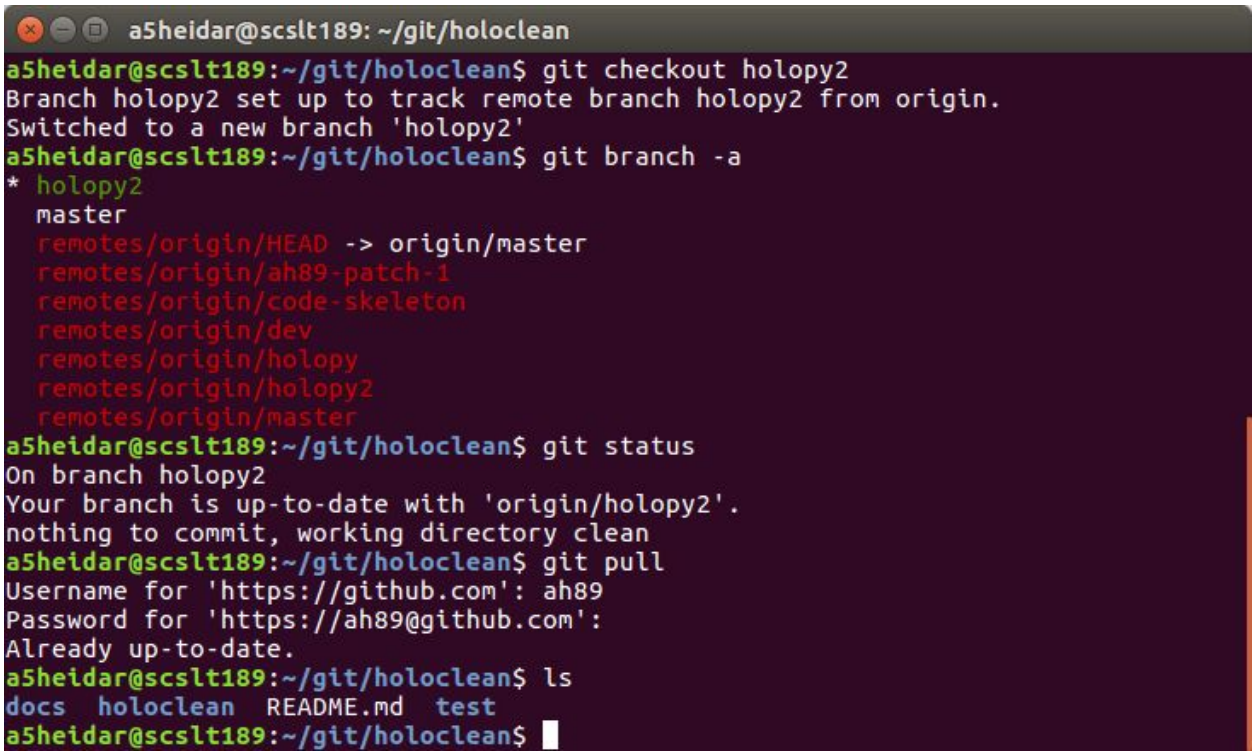
Before going further you need initial you git directory in your local machine with following command :

```
git init
```

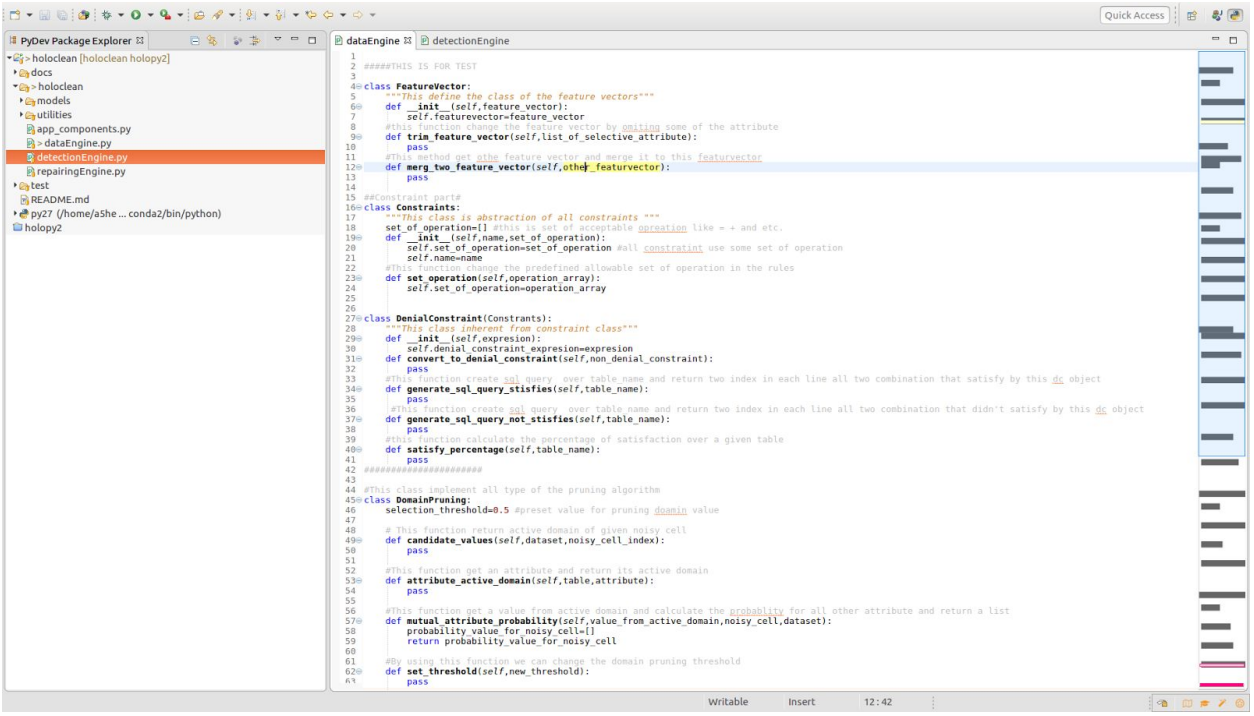
In our example we mean our /git directory as you can see in following screenshot



Now is the time to get codes of holopy2 branch from repo

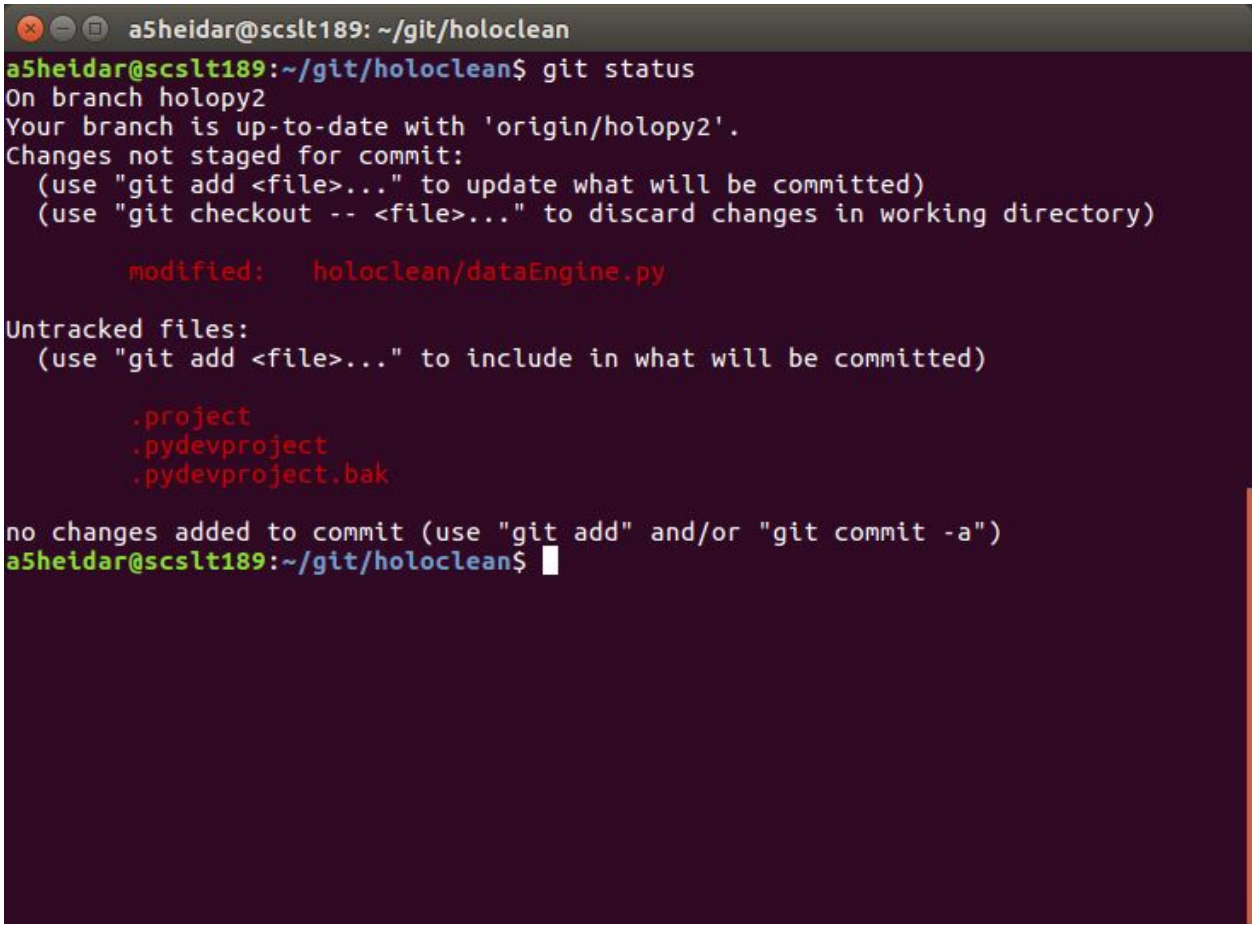


We changed something in the code and add comment on the first line of script:



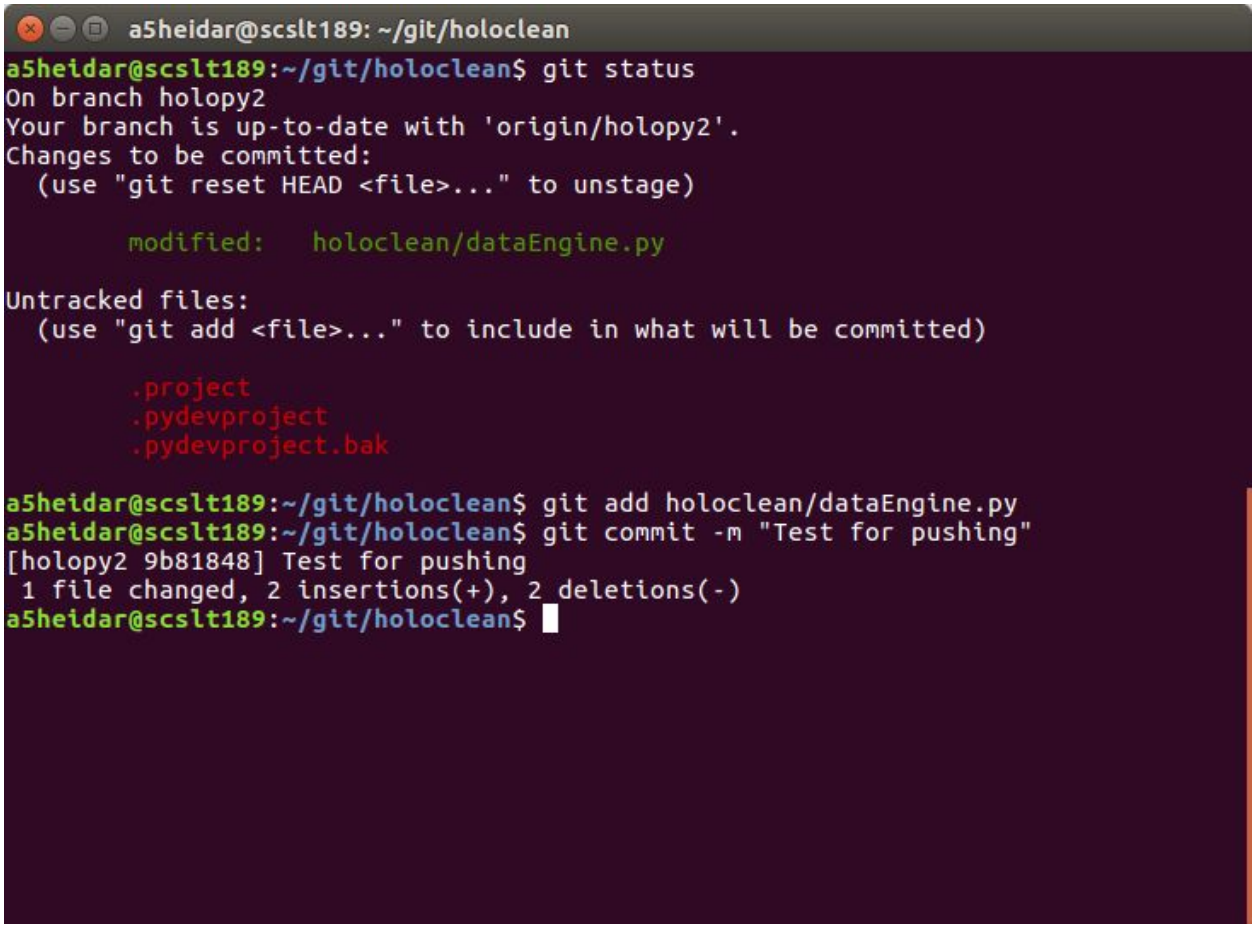
With status we see this changes in the command line :

git status



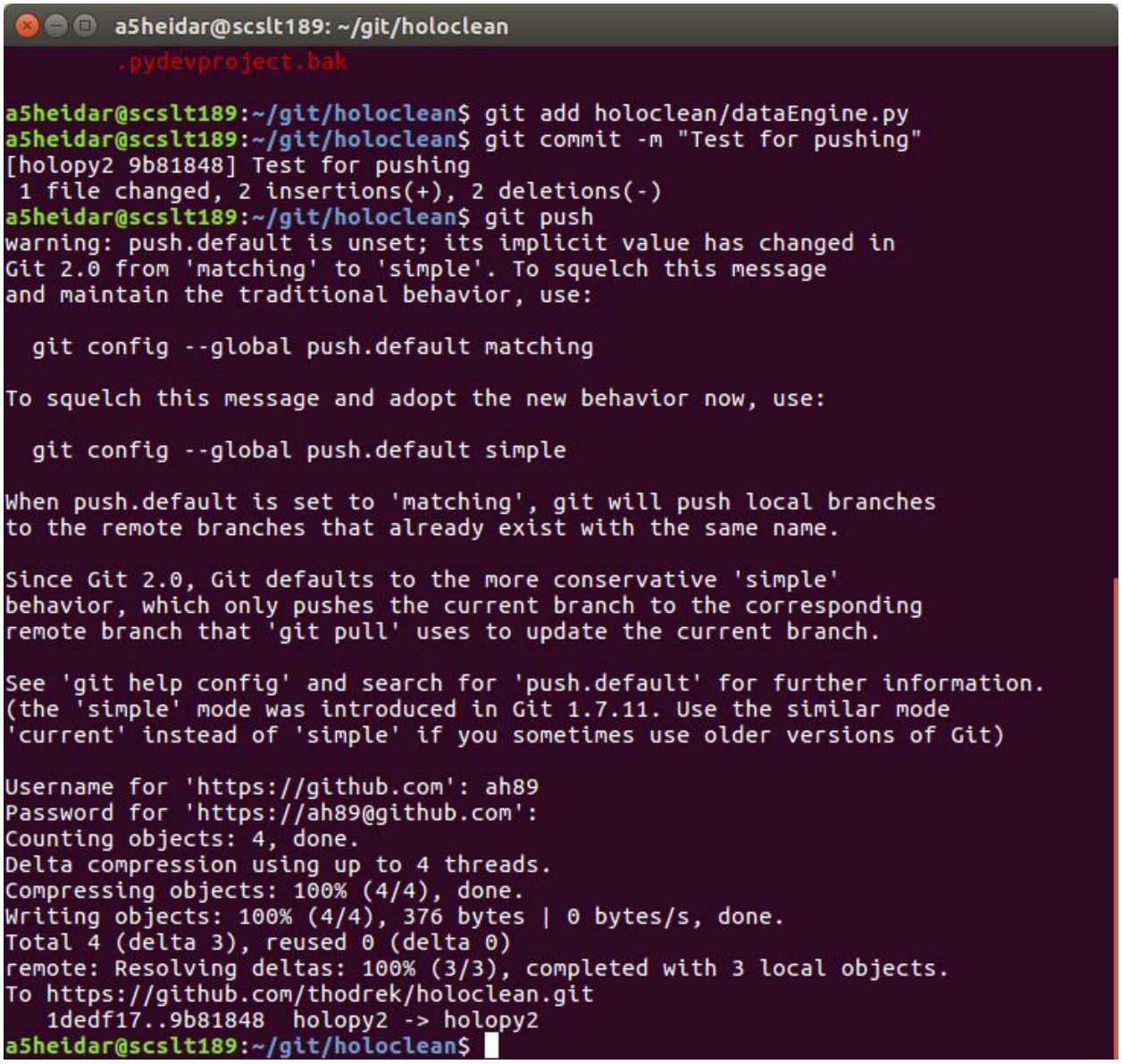
You get your status and you were in the right branch then you can push your changes to remote repo for this you first should make “commit” so by “add” you specify your file that you want to make commit for it and by commit the comment about that changes will attach.

git add [red file in the status for example holoclean/dataEngine.py]
git commit -m "some description like : Test for pushing"

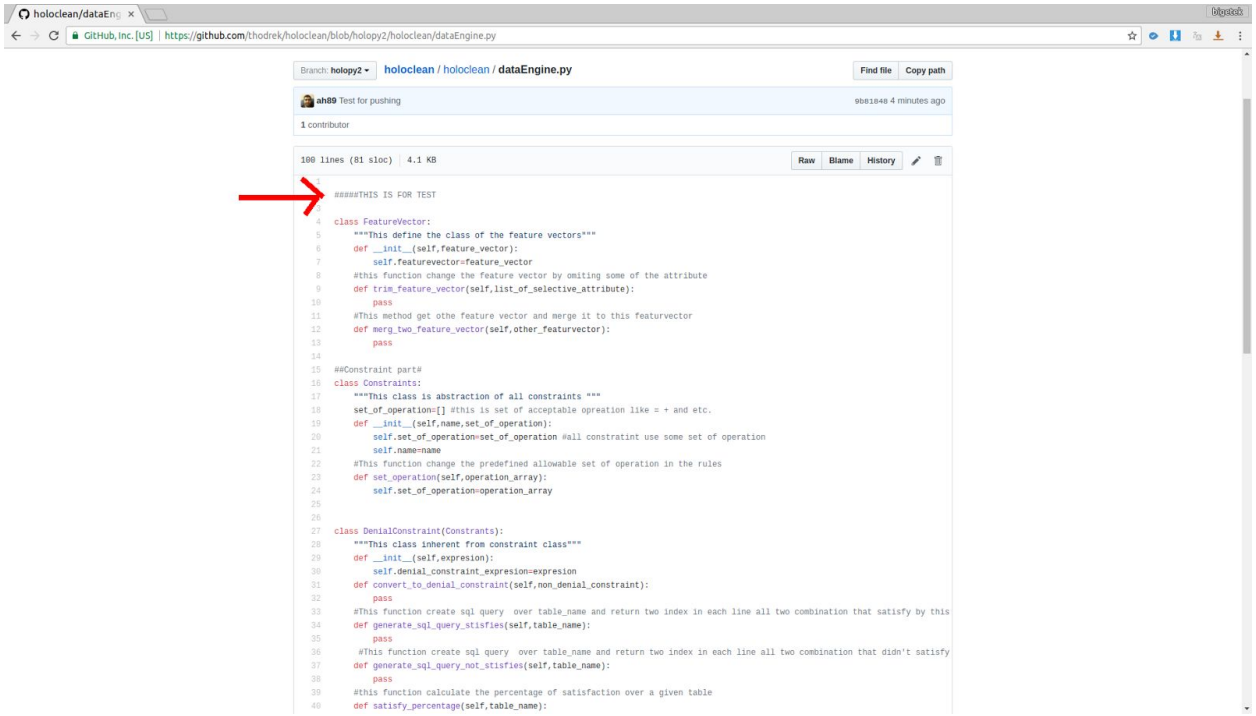


After that you can push all commits that you made in that branch you got in you status

git push



We can got to github.com and check if the codes been pushed correctly.

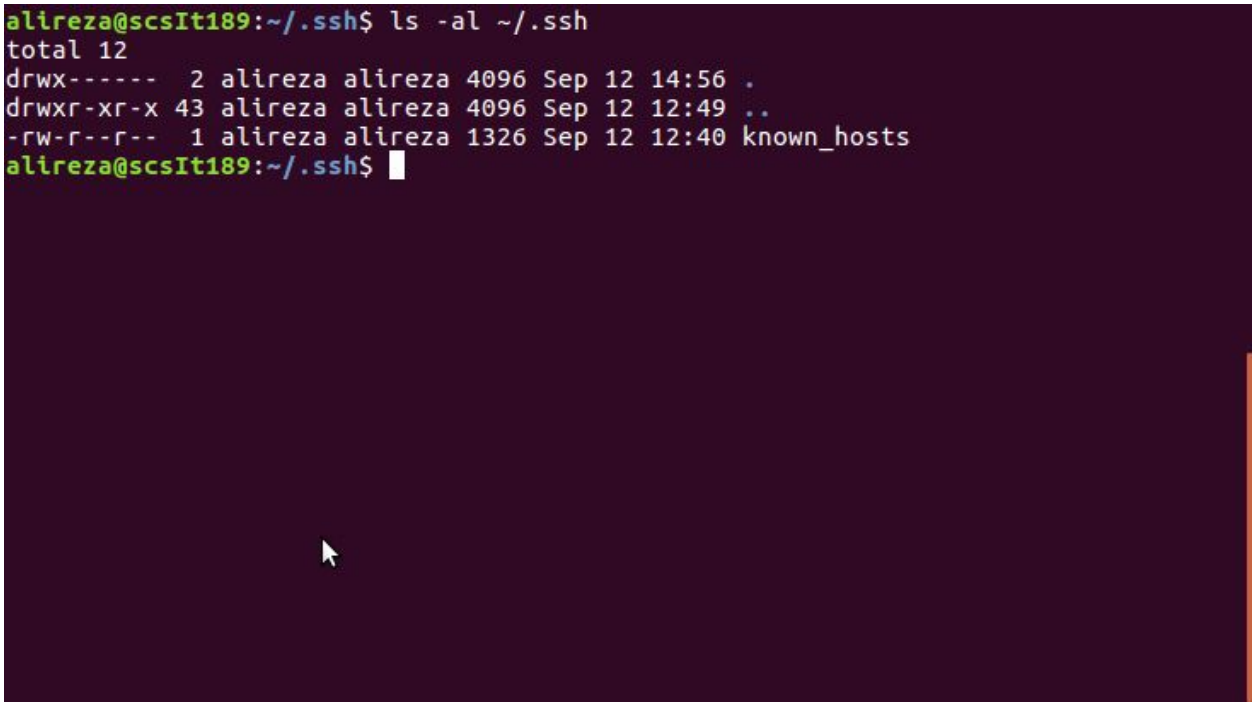


You can also change your branch to other branches by checkout command
git checkout [Some other branches]

2.7 Set SSH on Github

Sometimes you entering username and password are frustrating so you have enough confidence to your machine and so you can set SSH on your github account to avoid enter password every time you push. To this end you should make key on your machine first so at first you can check if you have any key before using following command :

ls -al ~/.ssh

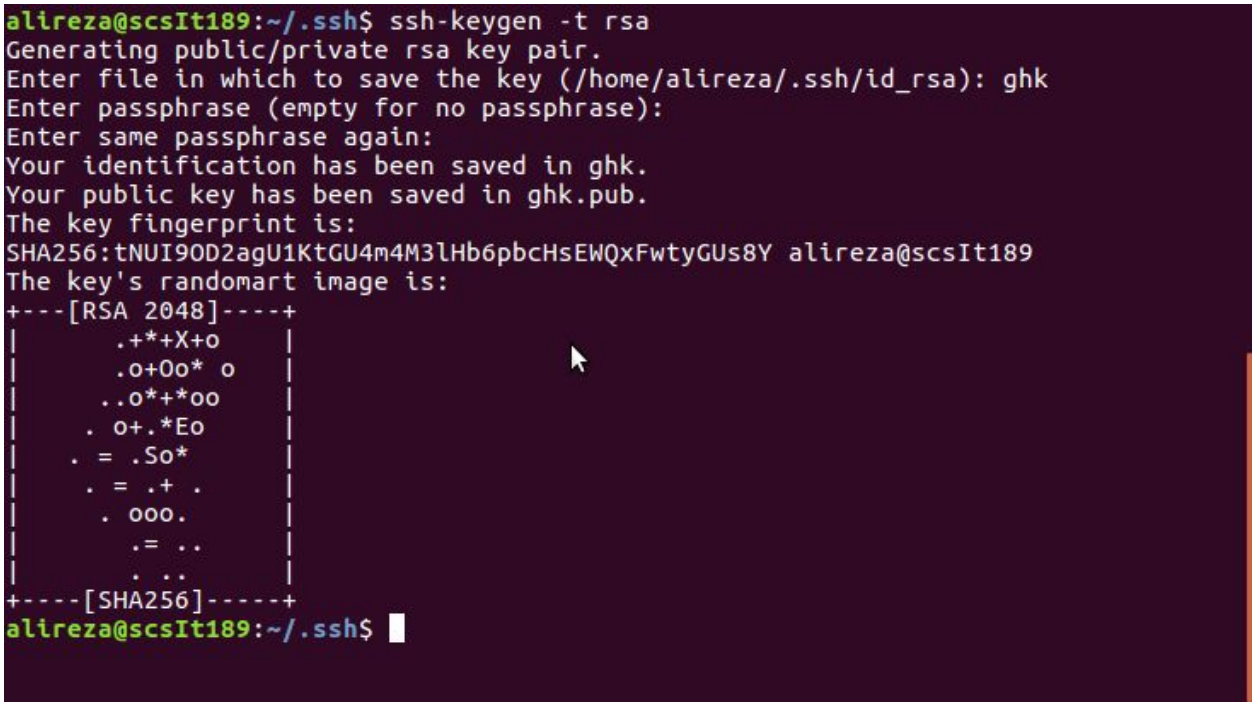


2.7.1 Generating a new SSH key and adding it to the ssh-agent

For creating a new ssh that can be used in your github you should create a new one with email that you create your github account .

ssh-keygen -t rsa

After this step you can choose passphrase to a security layer if you want to know more see this but we doesn't enter any passphrase.



2.7.2 Adding your SSH key to the ssh-agent

After you made a key you need to introduce it to you ssh agent which manage all you ssh connection first run an agent :

eval "$(ssh-agent -s)"

```
alireza@scsIt189:~/.ssh$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/alireza/.ssh/id_rsa): ghk
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ghk.
Your public key has been saved in ghk.pub.
The key fingerprint is:
SHA256:tNUI9OD2agU1KtGU4m4M3lHb6pbcHsEWQxFwtyGUs8Y alireza@scsIt189
The key's randomart image is:
+---[RSA 2048]----+
|       .+*+X+o    |
|       .o+Oo* o   |
|      ..o*+*oo    |
|     . o+.*Eo     |
|    . = .So*      |
|     . = .+ .     |
|      . ooo.      |
|        .= ..     |
|        . ..      |
+----[SHA256]-----+
alireza@scsIt189:~/.ssh$ eval "$(ssh-agent -s)"
Agent pid 16293
alireza@scsIt189:~/.ssh$
```

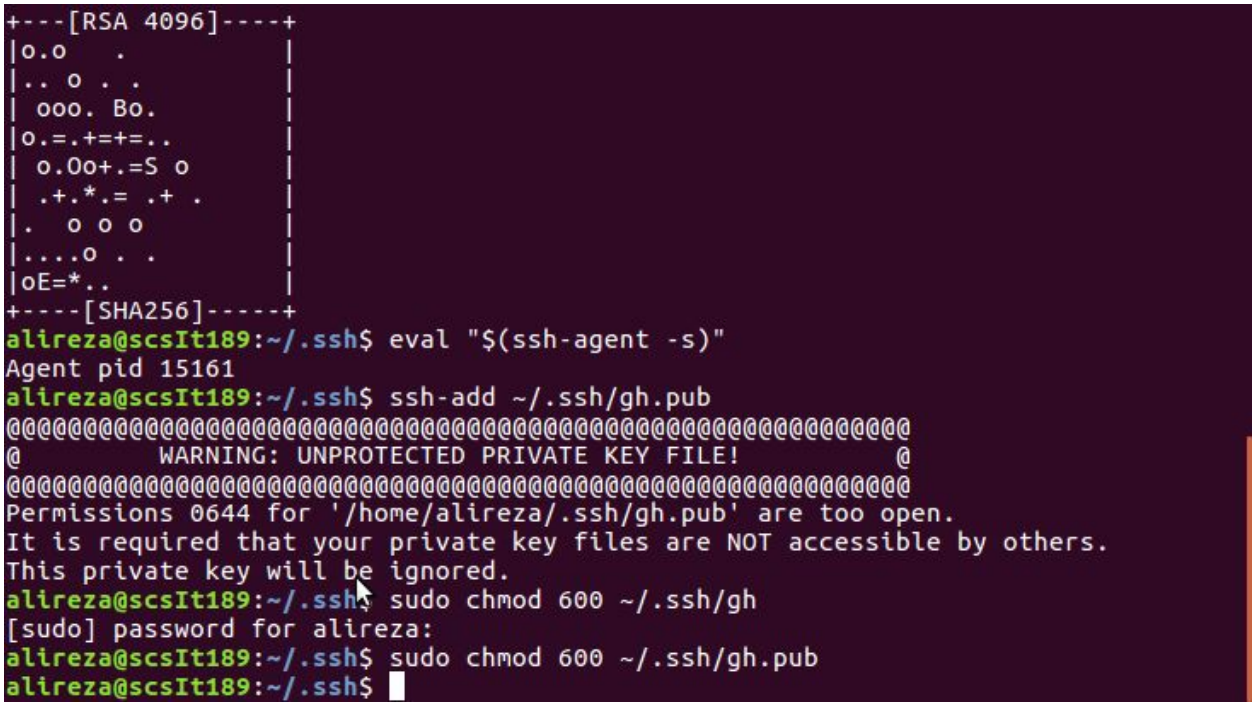
Then add your key our key name is ‘gh.pub’

ssh-add ~/.ssh/<Private key>

```
Enter file in which to save the key (/home/alireza/.ssh/id_rsa): ghk
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ghk.
Your public key has been saved in ghk.pub.
The key fingerprint is:
SHA256:xaFkuWNMwUHbB06FPxkc4by2xc7hGH2z8mD8fK3G710 alireza@scsIt189
The key's randomart image is:
+---[RSA 2048]----+
|       o*++++o   |
|       o+B.=o    |
|       oo.=.+o   |
|        =. .++   |
|       .S.  +.=..|
|           ..O oo|
|            o*+.E|
|             . O =|
|              ..B=|
+----[SHA256]-----+
alireza@scsIt189:~/.ssh$ eval "$(ssh-agent -s)"
Agent pid 17294
alireza@scsIt189:~/.ssh$ ssh-add ~/.ssh/ghk
Identity added: /home/alireza/.ssh/ghk (/home/alireza/.ssh/ghk)
alireza@scsIt189:~/.ssh$
```

Note : You might can see error because you haven't set your private key permission you change the permission of private key to 600 .

sudo chmod 600 ~/.ssh/gh
sudo chmod 600 ~/.ssh/gh.pub

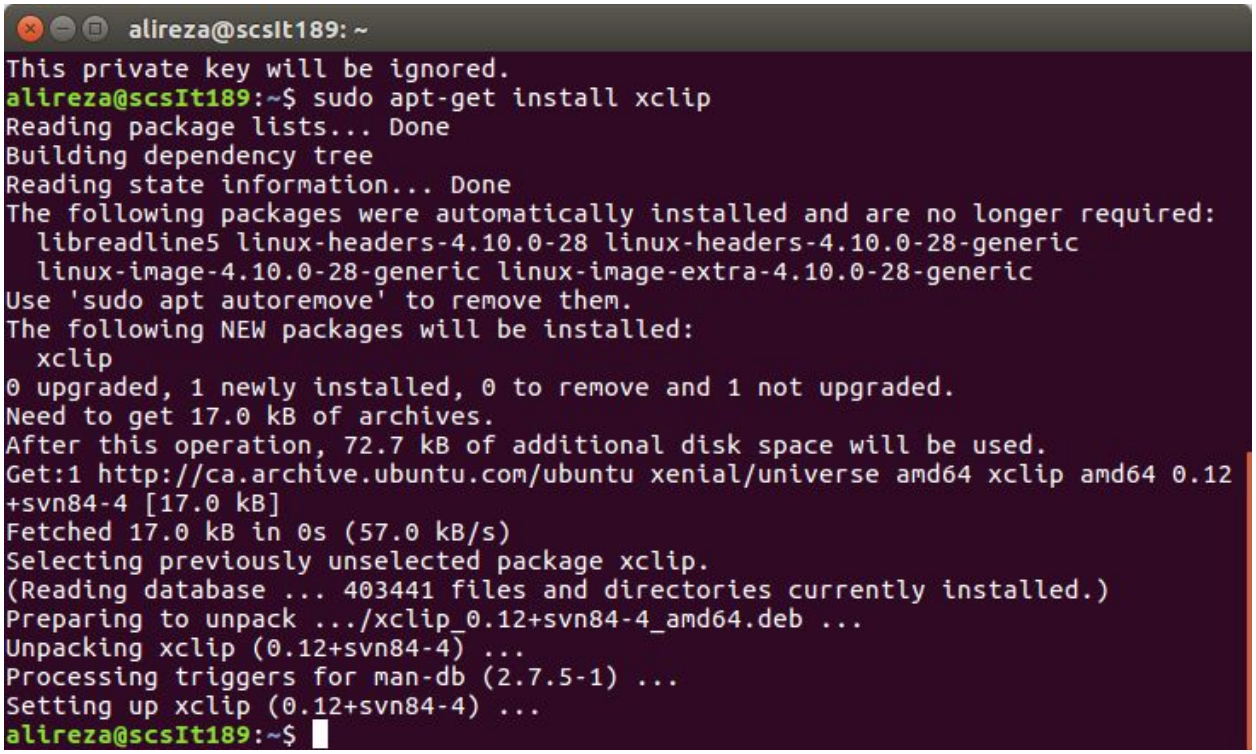


And then add your private key to you agent

1.3 Adding a new SSH key to your GitHub account

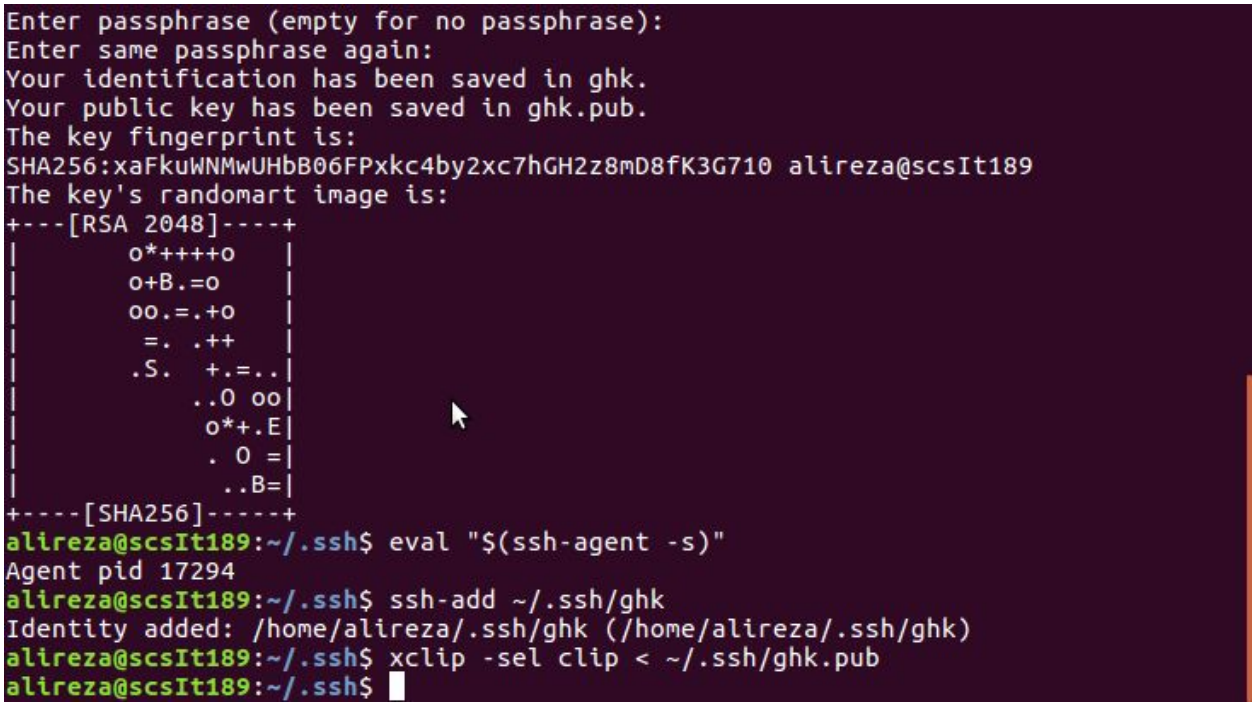
Copy your RSA key (this is very sensitive so we need to install an application for this)

sudo apt-get install xclip

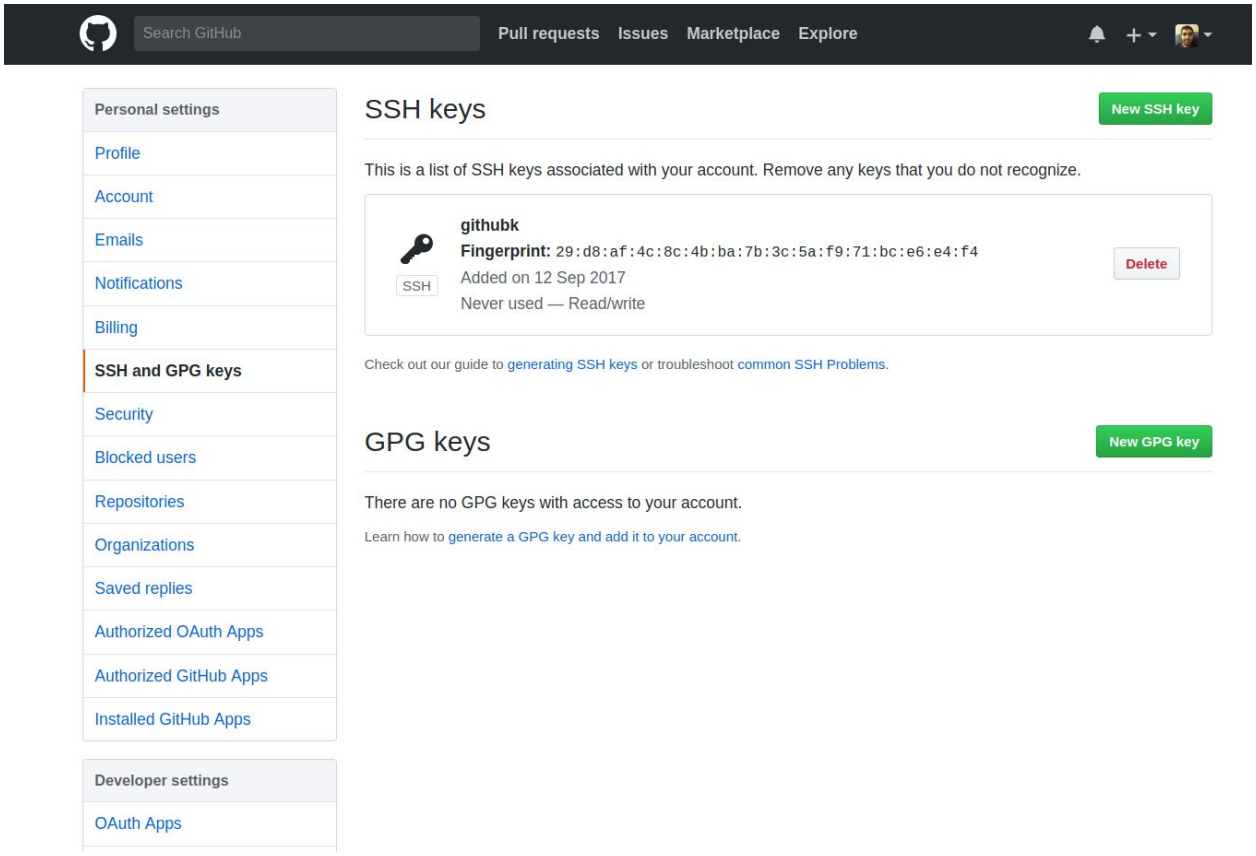


the n copy your public key using xclip as follow :

xclip -sel clip < [Directory of your .pub file]

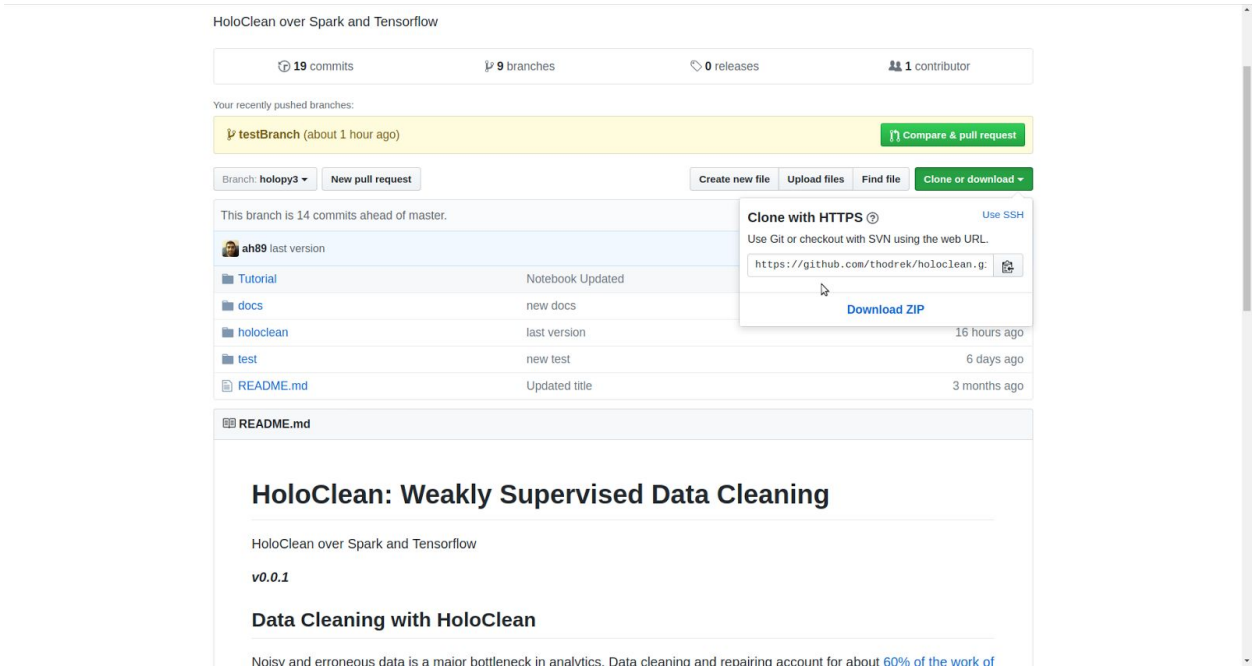


The go to your **github account> Settings > SSH and GPG keys > New SSH key or Add SSH key** and then paste your ssh key on ‘key’ field and click **Add SSH key**

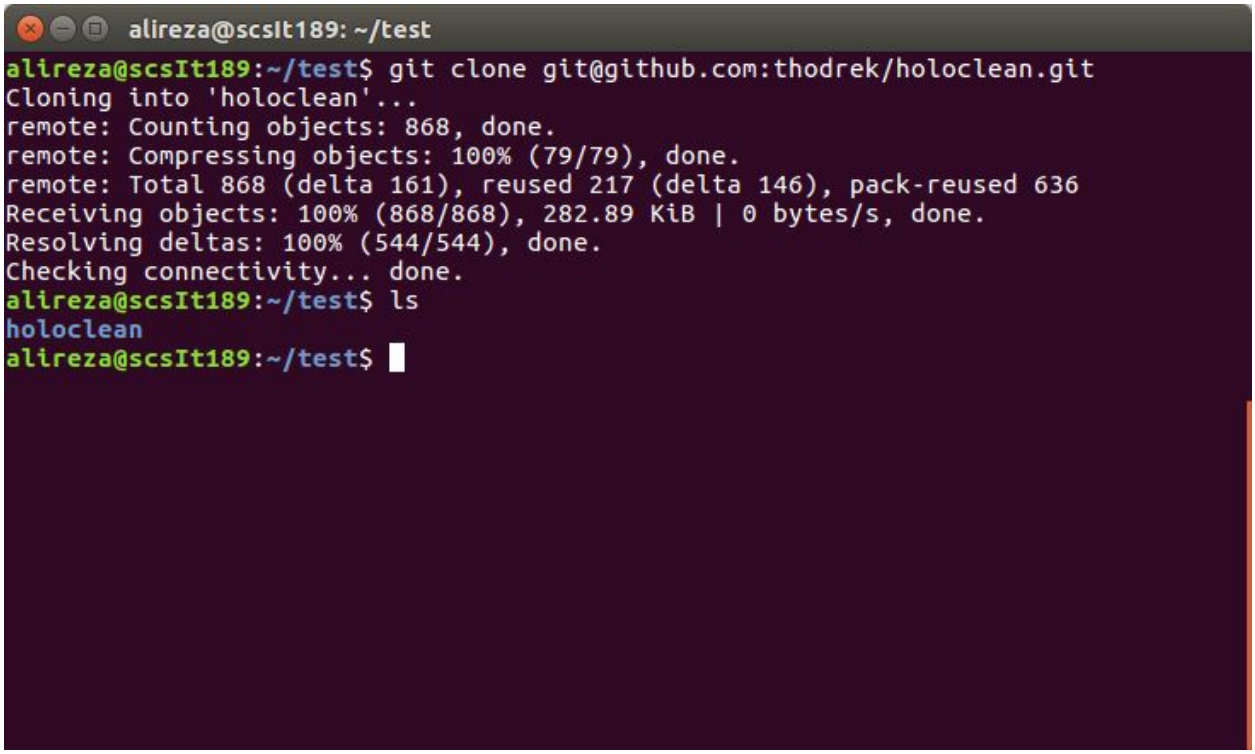


1.4 How use SSH with github

For using it you should clone with SSH download method when you go to github account you can see option for SSH clone address too.



After that you can clone with .git address that you copied to your system



3- Project

Each session might run multiple project in it so it has method in it that by calling it a project will be started

4- Reading data

This part a reading object by getting a directory for dataset and constraints and noisy cells (that created in the error detection process) try to read them and pass them to the parsing package

5- Parsing

6- Learning

This package has three module one make the data manipulation for training and testing data, second part by a model that we choose start to learn model with preset iteration and step , we can change this two parameters

7- Interaction and feedback

This package has to communicate with the users show result to them and grab their