

Version Control Systems (Mercurial)

by Nikola Ilkov Bozhinov
11A grade in TUES
Sofia,Bulgaria

1.What is a Version Control System?

Version Control Systems (or as they are more commonly known **VCS systems**) are mainly used in team projects when it's best if more than one person works on the assignment. For this purpose the source code is uploaded to an internet storage known as „repository“ and each member of the team can „pull“ it from there and make changes. When the developer is done he can „push“ the code back to the repository and the team manager can combine all the new additions with the old source code in order to get the new product. The whole system which is responsible for the operations we make on the project in the repository is called VCS system.

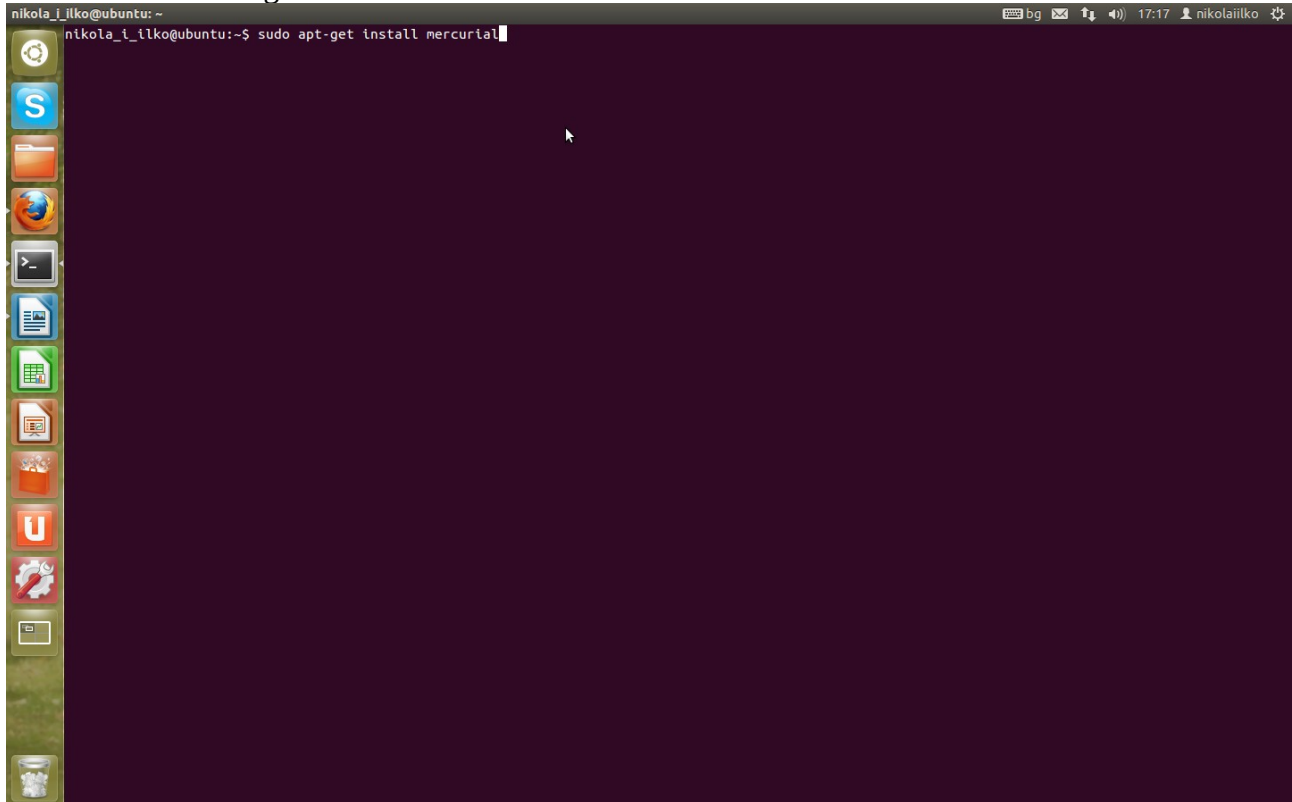
In this tutorial I will be using the VCS Mercurial.

2.How to install Mercurial on Ubuntu?

Step 1. Open your terminal (Bash)

Step 2. Copy this line: `sudo apt-get install mercurial`
and right click beside the empty line in your terminal (or you can just rewrite it).

You should be seeing this.



Step 3 Press Enter and when you are asked for your password type it (this will work only if you have the rights to install programs).

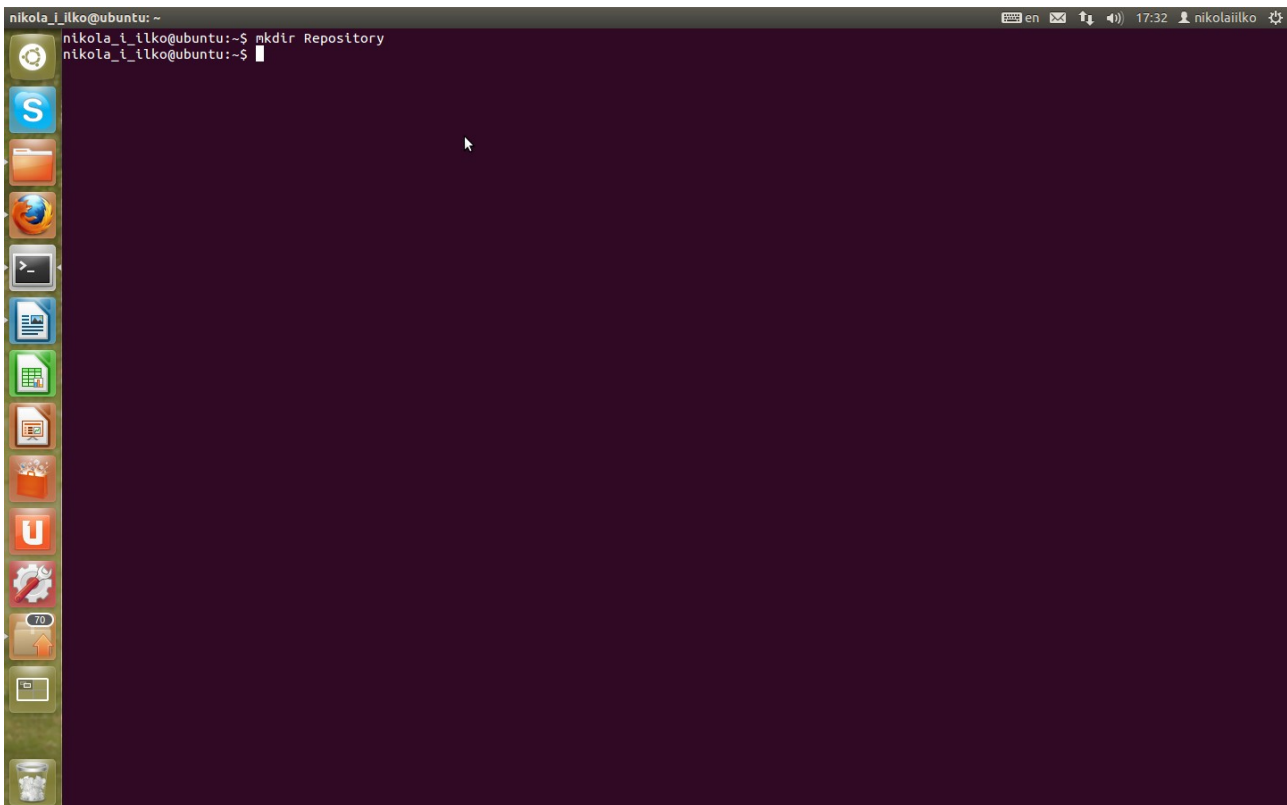
Now if everything is ok you should have Mercurial installed on your Ubuntu.

3. Initialise a repository and add files to it.

But of what good will Ubuntu be to us if we don't make a new project? Let us make one together and upload it to the internet (You must have an online repository, which can be easily made via registration in www.bitbucket.org)

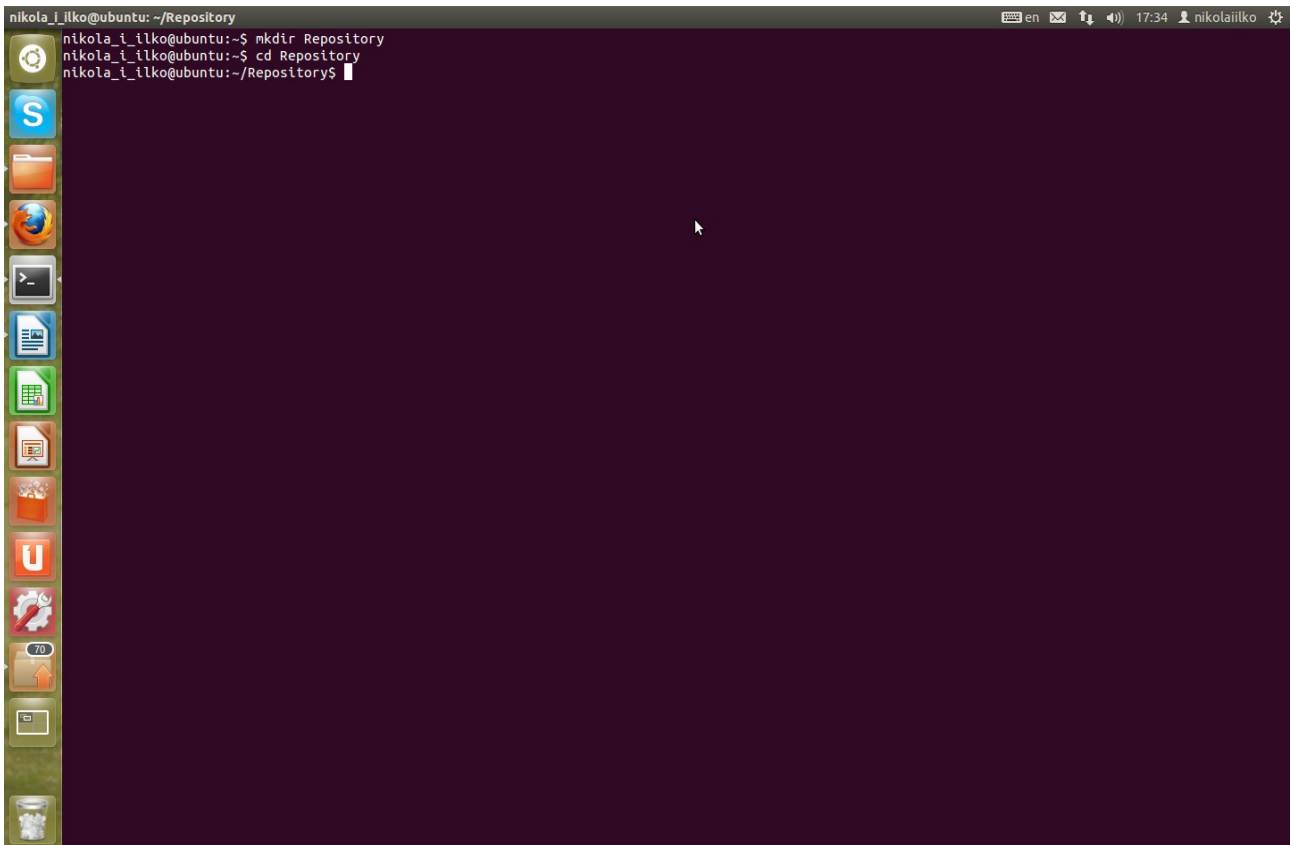
Step 1. Make a new folder with the command `mkdir <write_here_the_name_of_the_folder>` and this will create it.

Example:

A screenshot of a Linux terminal window. The window title is 'nikola_i_ilko@ubuntu: ~'. The terminal shows the command 'mkdir Repository' being entered and executed. The prompt changes from '~\$' to 'Repository\$' after the command is run. The terminal background is dark purple. On the left side of the terminal window, there is a vertical dock with various application icons including a gear, a blue 'S' icon, a folder, a globe, a terminal icon, a document, a spreadsheet, a presentation, a shopping cart, a 'U' icon, a wrench and screwdriver, a '70' icon, a mail icon, and a trash can. The top right corner of the terminal window shows system icons for language (en), network, volume, and the time 17:32, along with the user name 'nikolaiilko' and a settings gear icon.

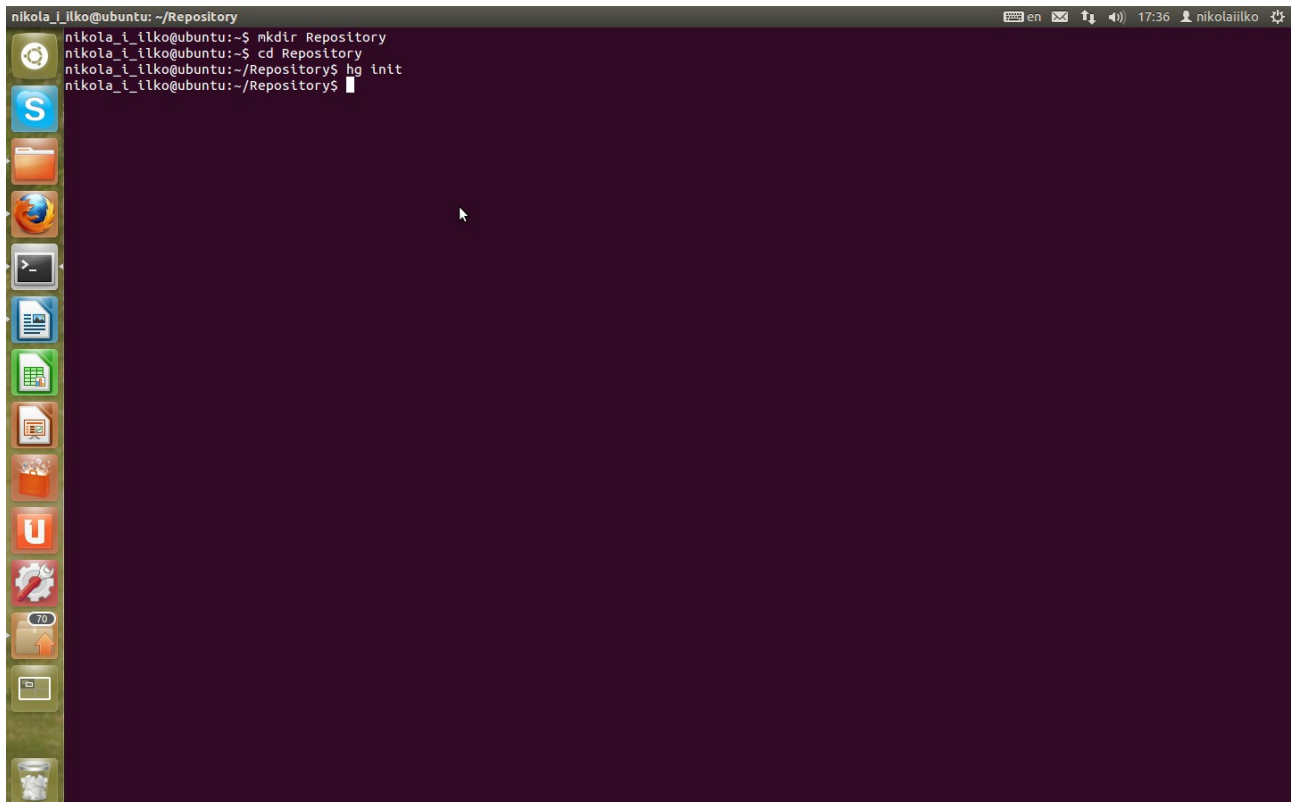
Step 2. Change your directory by using the command:
`cd <name_of_directory_you_wish_to_go_in>`

```
nikola_ilko@ubuntu: ~/Repository
nikola_ilko@ubuntu:~$ mkdir Repository
nikola_ilko@ubuntu:~$ cd Repository
nikola_ilko@ubuntu:~/Repository$
```

A terminal window on an Ubuntu system. The prompt is 'nikola_ilko@ubuntu: ~/Repository'. The user has entered 'mkdir Repository', then 'cd Repository', and the prompt is now 'nikola_ilko@ubuntu:~/Repository\$'. The terminal background is dark purple. On the left side, there is a vertical dock with various application icons including a search icon, a file manager, a web browser, a terminal, a document, a spreadsheet, a presentation, a mail client, a calendar, and a trash bin. The top of the window shows system status icons for language (en), network, volume, and battery, along with the time '17:34' and the user's name 'nikolaiilko'.

Step 3. To initialise(create) a new project type the command:
hg init

```
nikola_ilko@ubuntu: ~/Repository
nikola_ilko@ubuntu:~$ mkdir Repository
nikola_ilko@ubuntu:~$ cd Repository
nikola_ilko@ubuntu:~/Repository$ hg init
nikola_ilko@ubuntu:~/Repository$
```

A terminal window on an Ubuntu system, similar to the previous one. The prompt is 'nikola_ilko@ubuntu: ~/Repository'. The user has entered 'mkdir Repository', then 'cd Repository', and now 'hg init'. The prompt is now 'nikola_ilko@ubuntu:~/Repository\$'. The terminal background is dark purple. On the left side, there is a vertical dock with various application icons including a search icon, a file manager, a web browser, a terminal, a document, a spreadsheet, a presentation, a mail client, a calendar, and a trash bin. The top of the window shows system status icons for language (en), network, volume, and battery, along with the time '17:36' and the user's name 'nikolaiilko'.

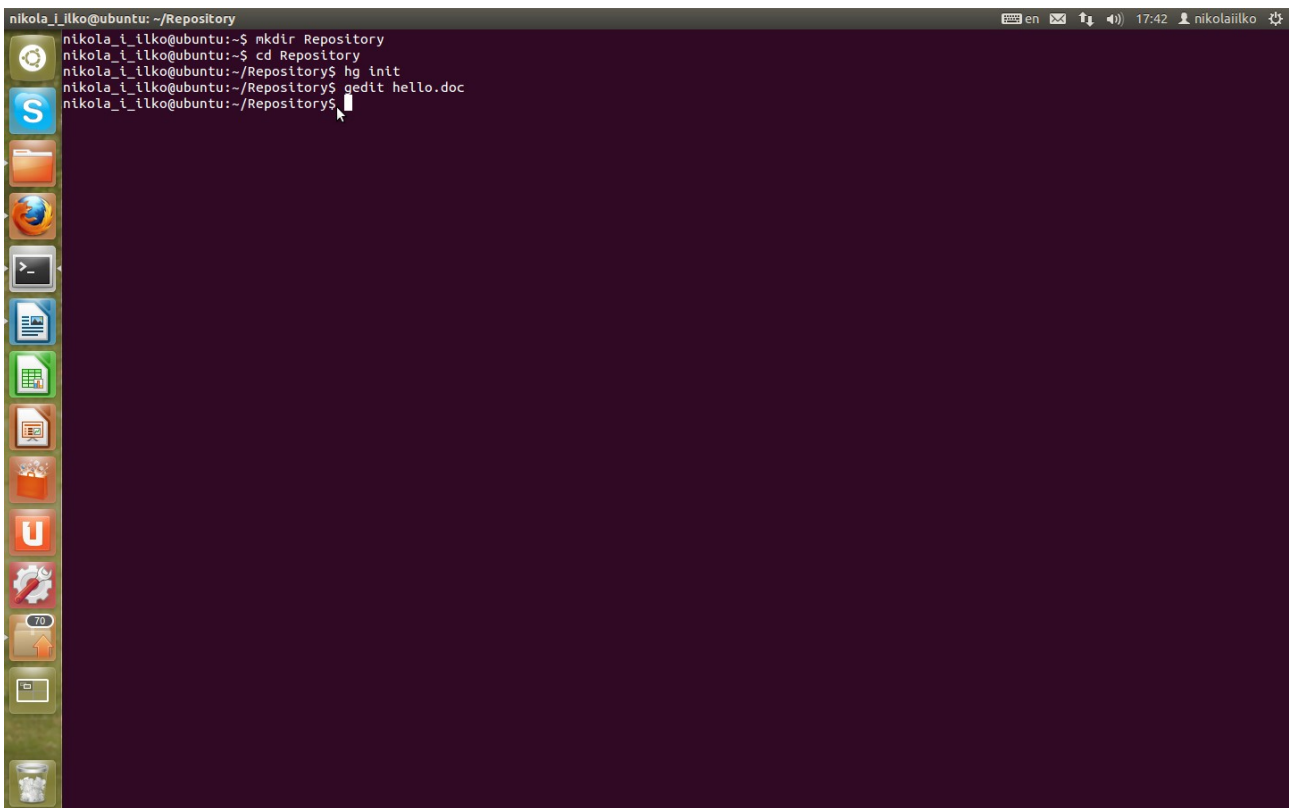
Congradulations!!!You made your first repository.

Step 4. Create a file inside your current directory. This can be done in several ways.

-type: `gedit <filename>` in the terminal and make a new file via `gedit`

-type: `nano <filename>` in the terminal and make a new file via `nano`

-go to your folder manually and create a new file

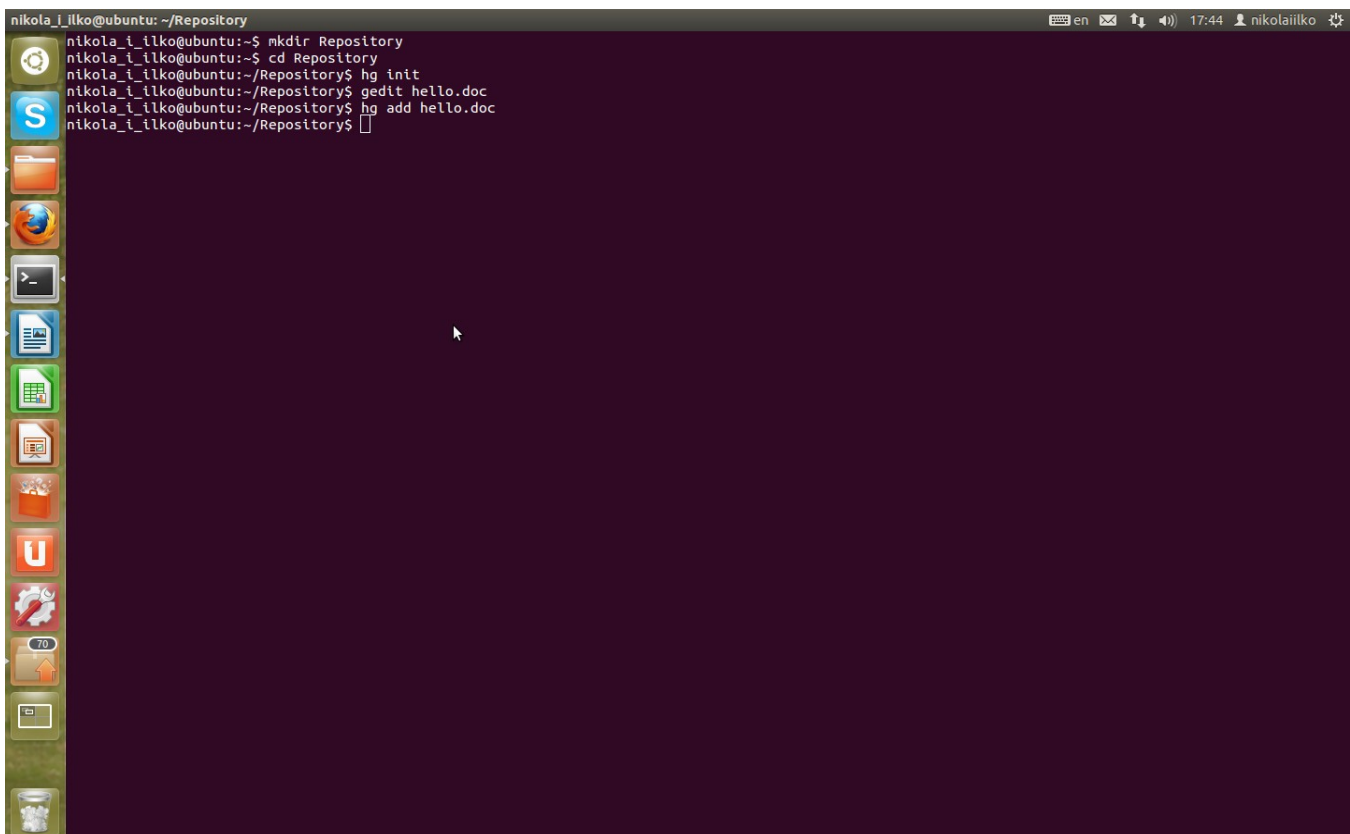


```
nikola_i_ilko@ubuntu: ~/Repository
nikola_i_ilko@ubuntu:~$ mkdir Repository
nikola_i_ilko@ubuntu:~$ cd Repository
nikola_i_ilko@ubuntu:~/Repository$ hg init
nikola_i_ilko@ubuntu:~/Repository$ gedit hello.doc
nikola_i_ilko@ubuntu:~/Repository$
```

Write whatever you want inside it. For mine I will use `gedit` and name it `hello.doc`. I wrote „Hello World!!!“ inside and closed it after saving.

Step 5. Let's add this file to the repository so it can be tracked. Use the command:

`hg add <filename>`



```
nikola_i_ilko@ubuntu: ~/Repository
nikola_i_ilko@ubuntu:~$ mkdir Repository
nikola_i_ilko@ubuntu:~$ cd Repository
nikola_i_ilko@ubuntu:~/Repository$ hg init
nikola_i_ilko@ubuntu:~/Repository$ gedit hello.doc
nikola_i_ilko@ubuntu:~/Repository$ hg add hello.doc
nikola_i_ilko@ubuntu:~/Repository$
```

It can be used for folders too and will add all the files inside of it.

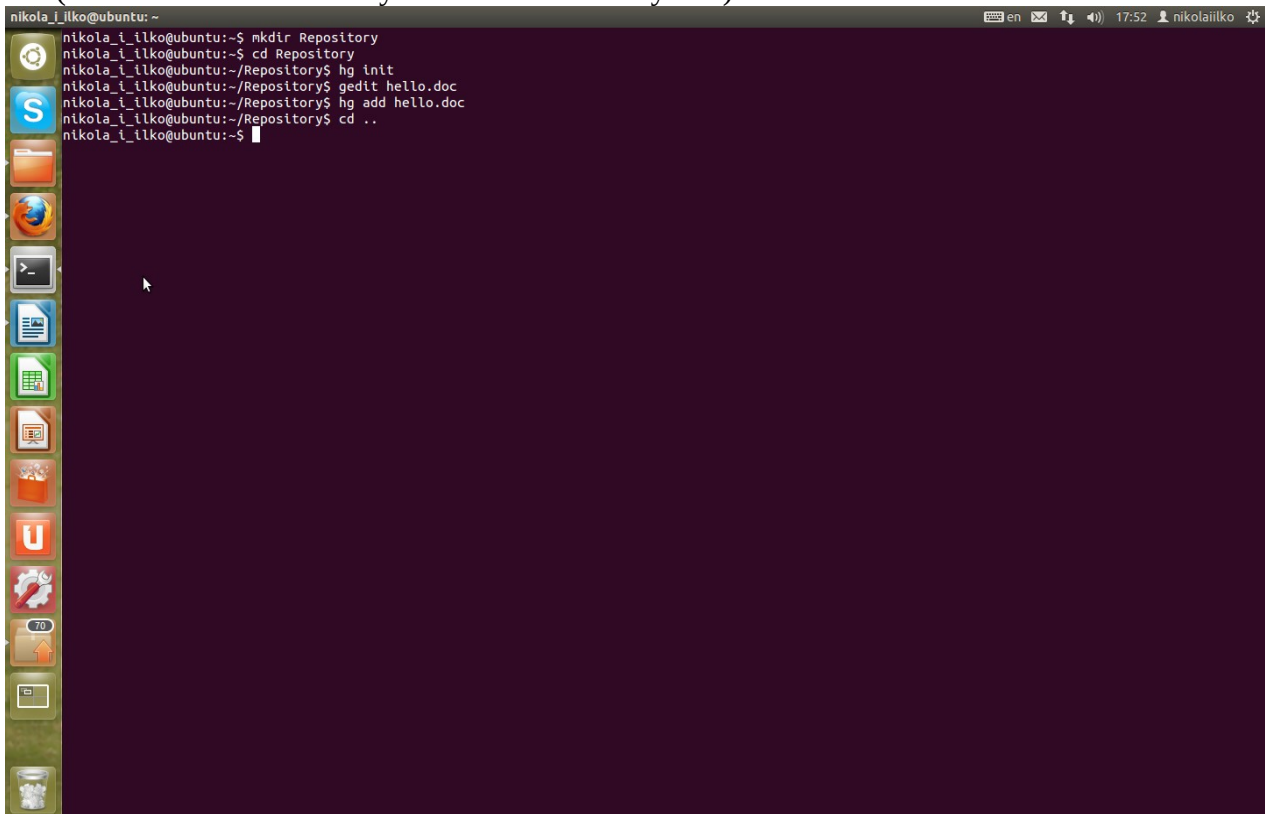
Step 6. Everything is good we have added a file for tracking but in fact mercurial still doesn't really know we have made changes to our repository. In order to yell this to the VCS we need to use the command :

`hg commit -m „This is the message we can later read about the current changes to the repository“`

But first we need to do something else-identify ourselves.

this can either done by returning back to the main Directory with the command:

`cd ..` (it returns us one directory back in the directory tree)



```
nikola_i_ilko@ubuntu: ~
└─$ mkdir Repository
nikola_i_ilko@ubuntu: ~
└─$ cd Repository
nikola_i_ilko@ubuntu: ~/Repository
└─$ hg init
nikola_i_ilko@ubuntu: ~/Repository
└─$ gedit hello.doc
nikola_i_ilko@ubuntu: ~/Repository
└─$ hg add hello.doc
nikola_i_ilko@ubuntu: ~/Repository
└─$ cd ..
nikola_i_ilko@ubuntu: ~
└─$
```

Now write:
`gedit .hgrc`

```
nikola_i_ilko@ubuntu: ~  
nikola_i_ilko@ubuntu:~$ mkdir Repository  
nikola_i_ilko@ubuntu:~$ cd Repository  
nikola_i_ilko@ubuntu:~/Repository$ hg init  
nikola_i_ilko@ubuntu:~/Repository$ gedit hello.doc  
nikola_i_ilko@ubuntu:~/Repository$ hg add hello.doc  
nikola_i_ilko@ubuntu:~/Repository$ cd ..  
nikola_i_ilko@ubuntu:~$ gedit .hgrc
```

and write this:

[ui]

username= your_name <your_email>

Example:

[ui]

username = Nikola Bojinov <nikola_bojinovTUES@abv.bg> (use the same e-mail as the one you used to create the bitbucket account)

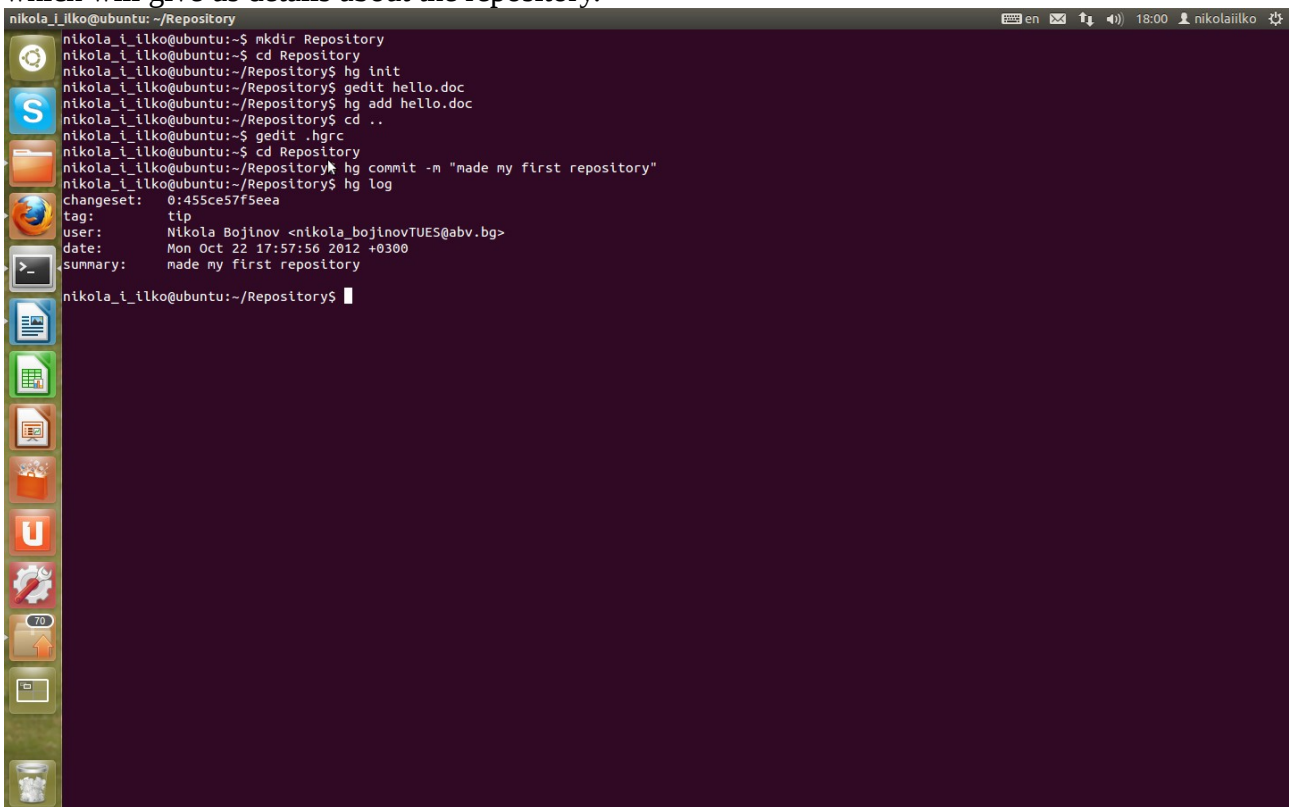
and save it.

```
nikola_i_ilko@ubuntu: ~/Repository  
nikola_i_ilko@ubuntu:~$ mkdir Repository  
nikola_i_ilko@ubuntu:~$ cd Repository  
nikola_i_ilko@ubuntu:~/Repository$ hg init  
nikola_i_ilko@ubuntu:~/Repository$ gedit hello.doc  
nikola_i_ilko@ubuntu:~/Repository$ hg add hello.doc  
nikola_i_ilko@ubuntu:~/Repository$ cd ..  
nikola_i_ilko@ubuntu:~$ gedit .hgrc  
nikola_i_ilko@ubuntu:~$ cd Repository  
nikola_i_ilko@ubuntu:~/Repository$ hg commit -m "made my first repository"  
nikola_i_ilko@ubuntu:~/Repository$
```

Now that we have identified ourselves go back to the repository directory with the cd command and use the commit command described earlier.

If everything is ok you shouldn't have received an error... If not please go back and check if you have done everything properly.

To check out new repository we will use the command:
`hg log`
which will give us details about the repository.

A terminal window screenshot showing the execution of hg log. The terminal title is 'nikola_ilko@ubuntu: ~/Repository'. The output of the command is as follows:

```
nikola_ilko@ubuntu:~/Repository$ hg log
changeset:  0:455ce57f5eea
tag:        tip
user:       Nikola Bojinov <nikola_bojinovTUES@abv.bg>
date:       Mon Oct 22 17:57:56 2012 +0300
summary:    made my first repository
```

Briefly:

changeset contains 2 numbers the first shows the order of our update compared to the ones before it. Because we haven't had any changes it is now 0 but if we had another it would have grown with one.

For example this is one of my other repositories:


```
nikola_t_ilko@ubuntu:~/newrepo
nikola_t_ilko@ubuntu:~/Repository$ cd ..
nikola_t_ilko@ubuntu:~$ cd newrepo
nikola_t_ilko@ubuntu:~/newrepo$ hg log
changeset: 3:461aca5218df
tag:      tip
user:     Nikola Bojinov <nikola_bojinovTUES@abv.bg>
date:     Wed Oct 10 16:33:27 2012 +0300
summary:  new java homework

changeset: 2:b8dc90654cd2
user:     nbojinov
date:     Tue Oct 09 10:44:47 2012 +0300
summary:  made a .java file with an array of integers

changeset: 1:a6da21a26bd0
user:     nikola
date:     Tue Oct 09 09:52:11 2012 +0300
summary:  adding screenshot

changeset: 0:ee524aa47df8
user:     Nikola Bojinov <nikola_bojinovTUES@abv.bg>
date:     Mon Oct 08 16:35:34 2012 +0300
summary:  made new directories

nikola_t_ilko@ubuntu:~/newrepo$
```

As you can see the newest one has the number 3 and all others with one less...

The second part of the number is a unique sequence which is used for accessing each changeset but we won't talk about this now.

The line

```
tag      tag
```

shows us the changeset we are using now.

I think all other are pretty obvious.

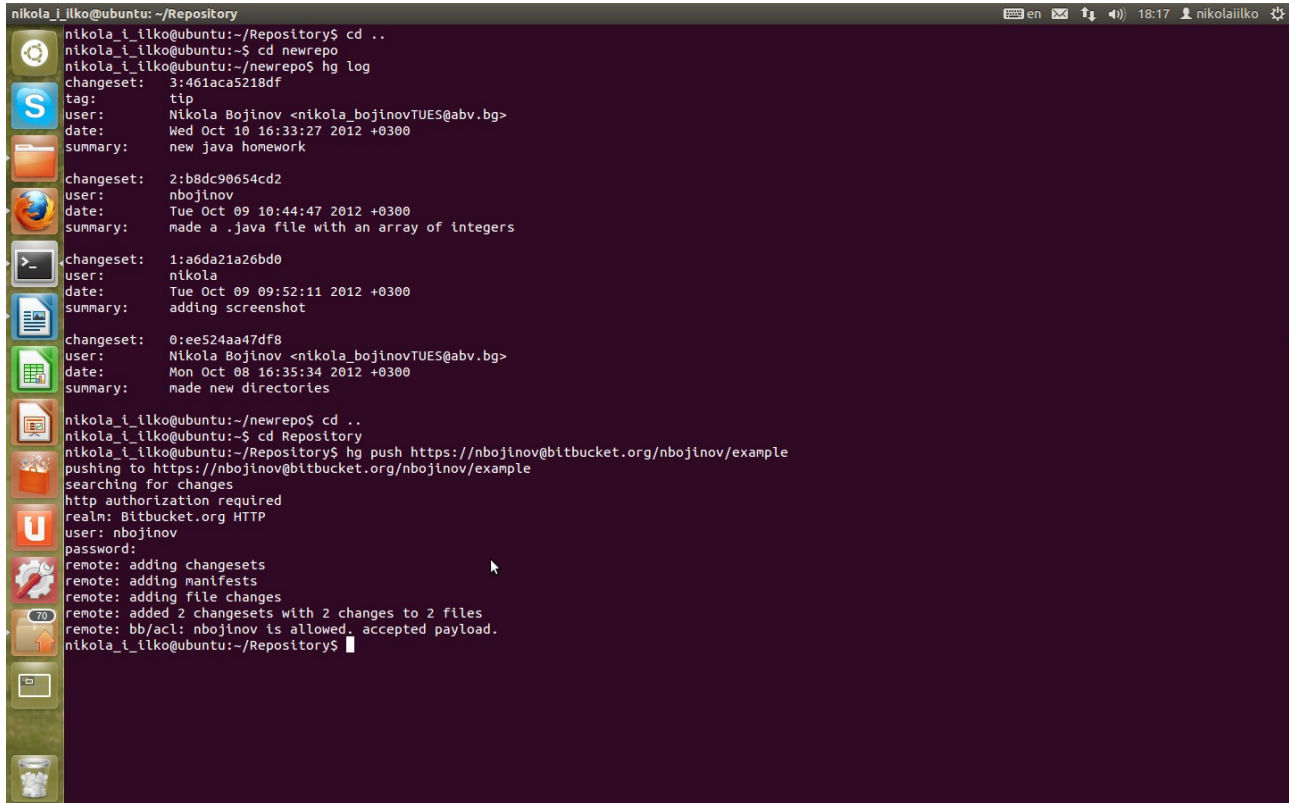
4.Upload the content to an online repository.

We can now upload our new content to the repository you have in bitbucket. Mine is at <https://nbojinov@bitbucket.org/nbojinov/example> and the command is:

hg push (while you are in the folder of the repository)

so the whole line should look like this

hg push https://nbojinov@bitbucket.org/nbojinov/example



```
nikola_ilko@ubuntu:~/Repository
nikola_t_ilko@ubuntu:~/Repository$ cd ..
nikola_t_ilko@ubuntu:~$ cd newrepo
nikola_t_ilko@ubuntu:~/newrepo$ hg log
changeset: 3:461aca5218df
tag: tip
user: Nikola Bojinov <nikola_bojinovTUES@abv.bg>
date: Wed Oct 10 16:33:27 2012 +0300
summary: new java homework

changeset: 2:b8dc90654cd2
user: nbojinov
date: Tue Oct 09 10:44:47 2012 +0300
summary: made a .java file with an array of integers

changeset: 1:a6da21a26bd0
user: nikola
date: Tue Oct 09 09:52:11 2012 +0300
summary: adding screenshot

changeset: 0:ee524aa47df8
user: Nikola Bojinov <nikola_bojinovTUES@abv.bg>
date: Mon Oct 08 16:35:34 2012 +0300
summary: made new directories

nikola_t_ilko@ubuntu:~/newrepo$ cd ..
nikola_t_ilko@ubuntu:~$ cd Repository
nikola_t_ilko@ubuntu:~/Repository$ hg push https://nbojinov@bitbucket.org/nbojinov/example
pushing to https://nbojinov@bitbucket.org/nbojinov/example
searching for changes
http authorization required
realm: Bitbucket.org HTTP
user: nbojinov
password:
remote: adding changesets
remote: adding manifests
remote: adding file changes
remote: added 2 changesets with 2 changes to 2 files
remote: bb/acl: nbojinov is allowed. accepted payload.
nikola_t_ilko@ubuntu:~/Repository$
```

When you are asked for password write the one you used for your bitbucker account (don't worry if you can't see the keys you have pressed it's supposed to be like that :))

5.Pull and Update.

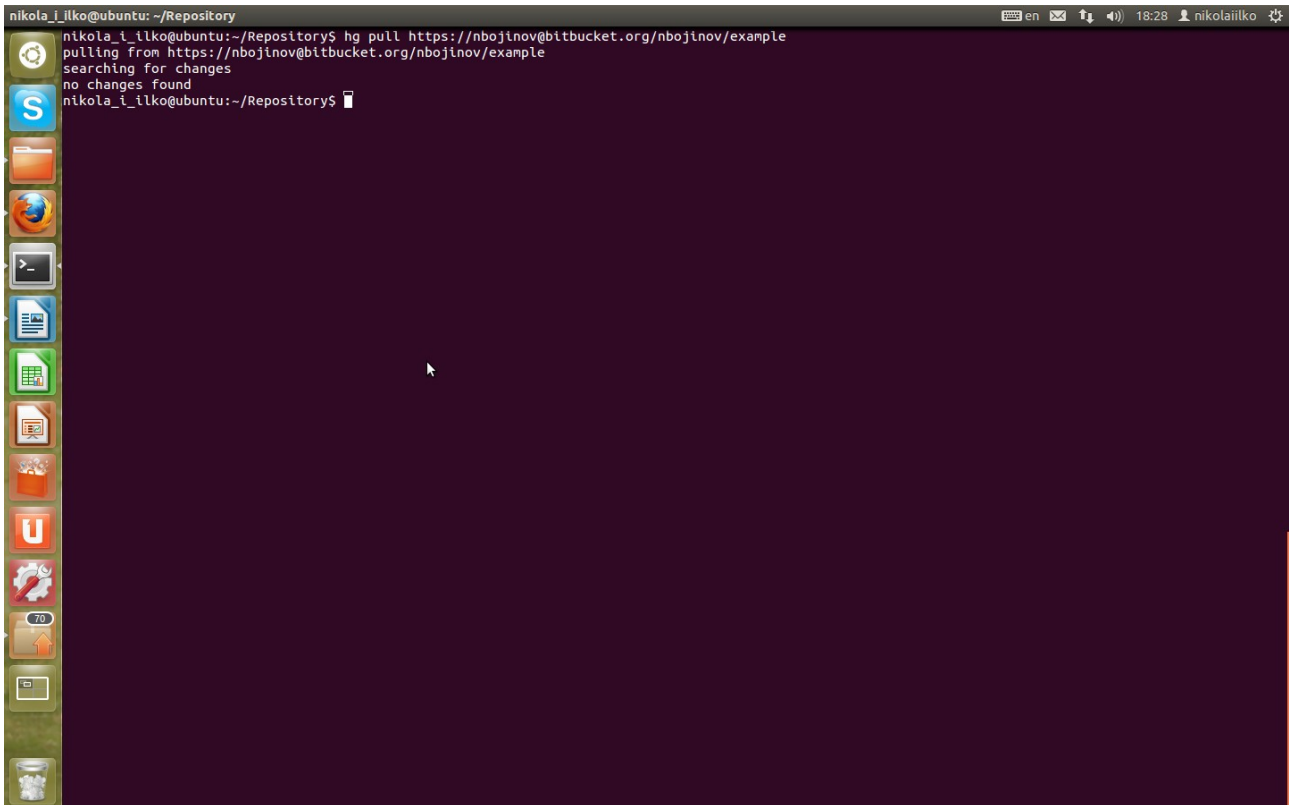
When working with many people on the same project there will definitely be changes to the project while you aren't working on it.

In order to get these changes use the commands:

`hg pull <site>`(to pull the changes from the repository)

`hg update` (to get these new changes working)

if there are no new changes you will get this message:

A screenshot of a Linux terminal window. The window title is "nikola_i_ilko@ubuntu: ~/Repository". The terminal shows the following text:

```
nikola_i_ilko@ubuntu:~/Repository$ hg pull https://nbojinov@bitbucket.org/nbojinov/example
pulling from https://nbojinov@bitbucket.org/nbojinov/example
searching for changes
no changes found
nikola_i_ilko@ubuntu:~/Repository$
```

The terminal output indicates that the pull operation was successful but no new changes were found in the repository. The terminal window is part of a desktop environment with a sidebar of application icons on the left and a system tray on the top right.

6.Clone.

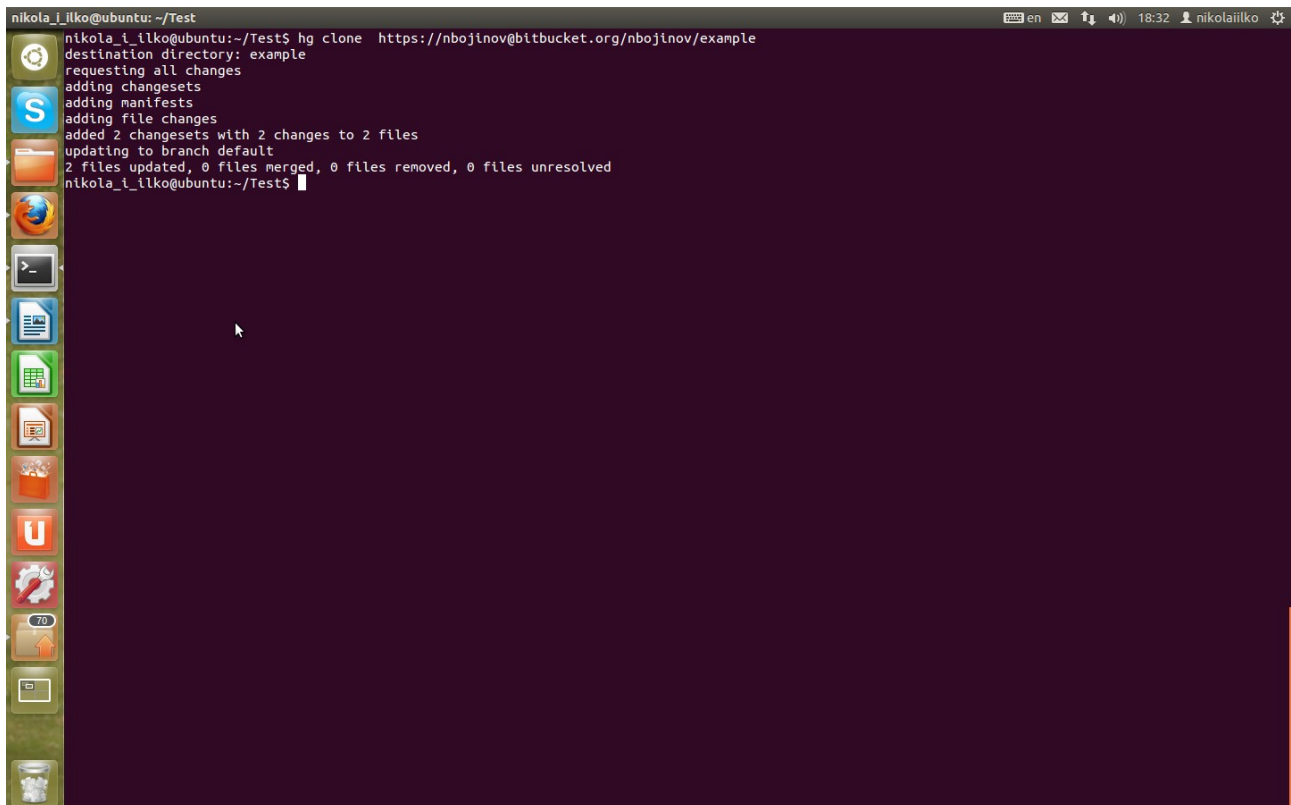
The clone command is used in a combination with HG INIT when you don't have a copy of an repository and want to download it.

Step 1.Initialise a new repository in a new folder as shown above.

Step 2. Use the command:

```
hg clone <site>
```

example:

A screenshot of a terminal window on an Ubuntu system. The terminal shows the execution of the command 'hg clone https://nbojinov@bitbucket.org/nbojinov/example'. The output indicates that the destination directory is 'example', changes are requested, and two changesets are added to the default branch. The terminal output is as follows:

```
nikola_i_ilko@ubuntu: ~/Test
nikola_i_ilko@ubuntu:~/Test$ hg clone https://nbojinov@bitbucket.org/nbojinov/example
destination directory: example
requesting all changes
adding changesets
adding manifests
adding file changes
added 2 changesets with 2 changes to 2 files
updating to branch default
2 files updated, 0 files merged, 0 files removed, 0 files unresolved
nikola_i_ilko@ubuntu:~/Test$
```

If the repository isn't private you should be able to download it easily nad have a working copy of it.

I hope that this tutorial will be useful to you. Feel free to copy information and use it as long as you mention the source.