Git-Fork concept

Following is the brief overview of the standard Github workflow:

1. **fork** a project on github.
2. **clone** your github fork to your computer
3. create a topic **branch** for your own work in your local clone
4. **commit** changes to your local repository
5. **push** the changes to your github fork
6. send a **pull request** back to the original project

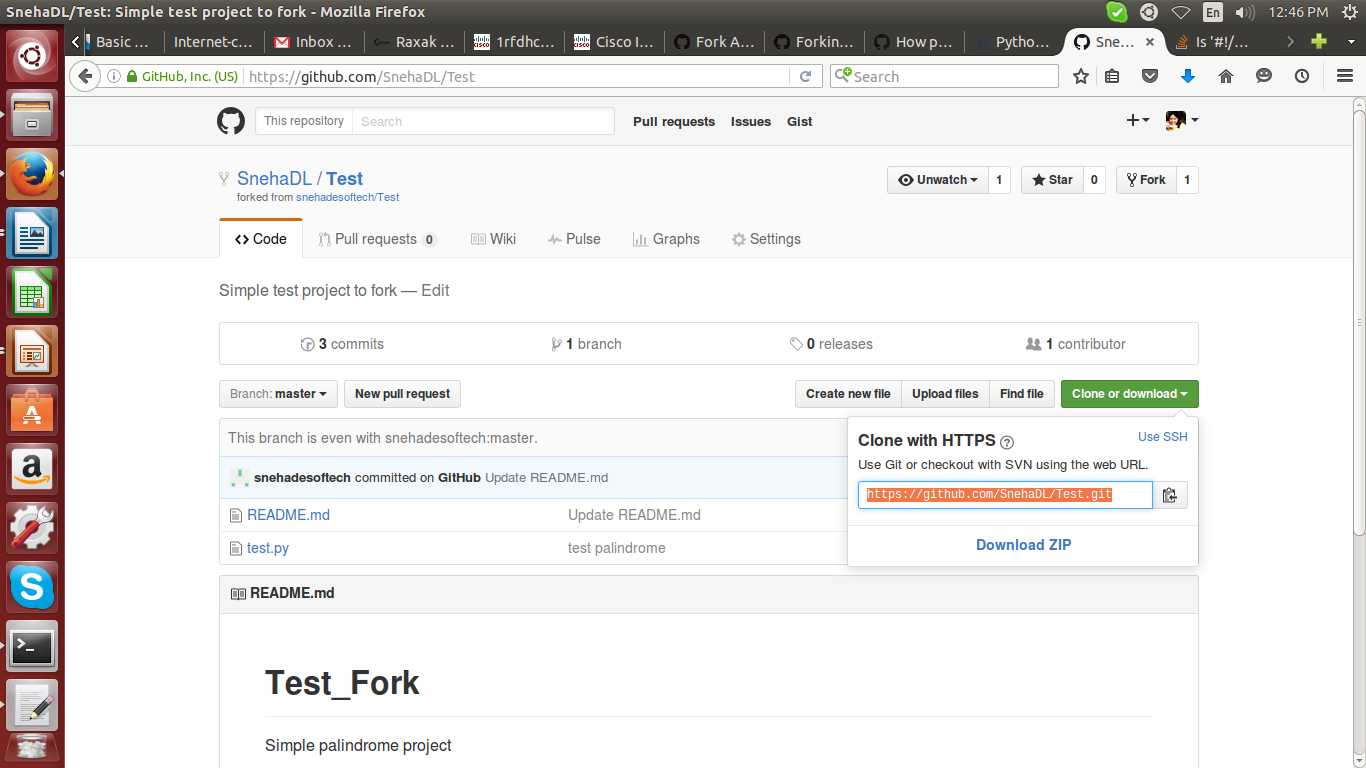
### The concept of fork

* A fork is a local copy of a repository.
* Forking a repository allows you to freely experiment with changes without affecting the original project.
* Most commonly, forks are used to either propose changes to someone else's project or to use someone else's project as a starting point for your own idea.

# Understanding fork with an example

1. On GitHub, navigate to the <https://github.com/snehadesoftech/Test> repository.
2. In the top-right corner of the page, click Fork.
3. After that, goto terminal and follow steps to:

* clone the forked repo
* create a new branch
* make changes
* add the changes
* commit and push the code.



Execute the commands mentioned below to push the code changes:

*# git clone https://github.com/SnehaDL/Test.git*

*# cd Test/*

*/Test# ls -ltr*

*total 8*

*-rw-r--r-- 1 root root 387 Jul 14 12:46 test.py*

*-rw-r--r-- 1 root root 520 Jul 14 12:46 README.md*

*/Test# vi test.py*

*/Test# git diff test.py*

*diff --git a/test.py b/test.py*

*index 62a63c8..a5a8470 100644*

*--- a/test.py*

*+++ b/test.py*

*@@ -1,6 +1,7 @@*

*# Program to check if a string*

*# is palindrome or not*

*+#!/usr/bin/python*

*# take input from the user*

*my\_str = input("Enter a string: ")*

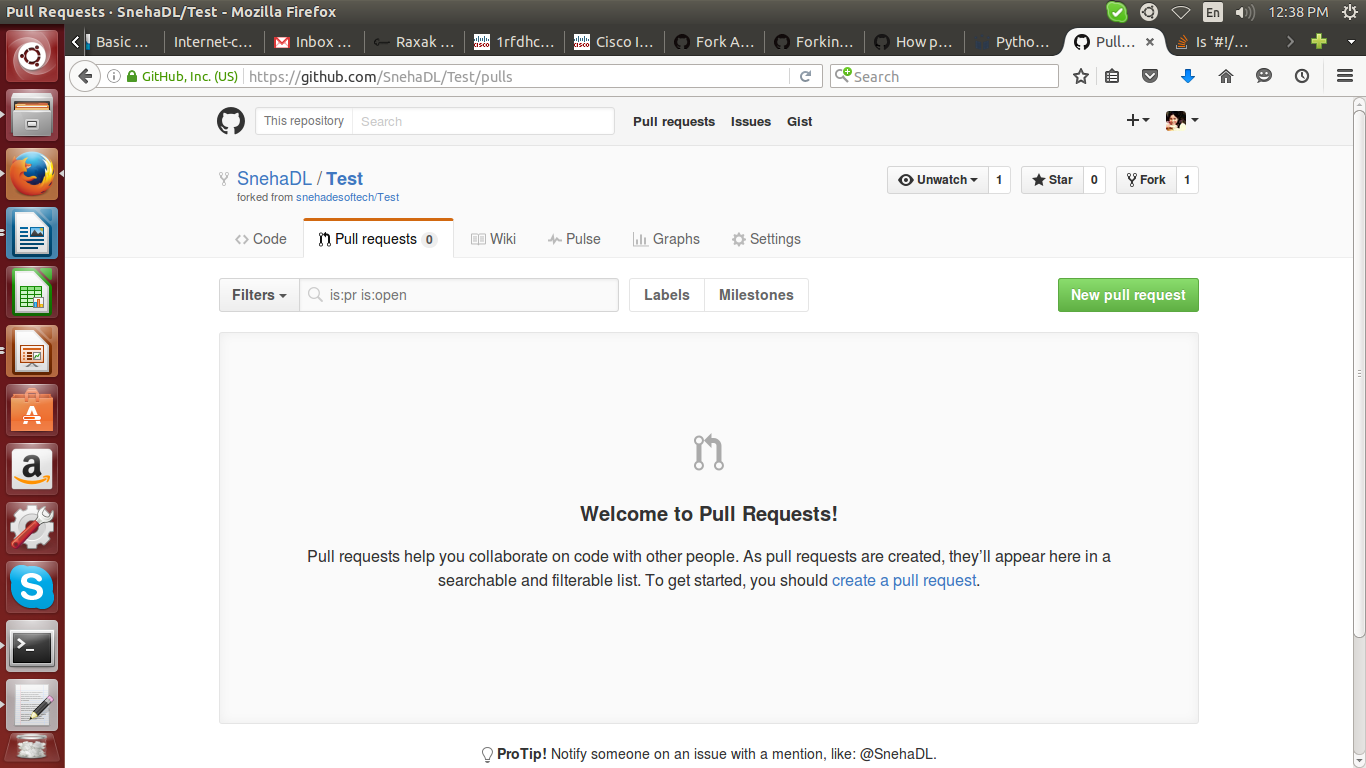
*/Test# git add –all*

*/Test# git commit -m "adding a line"*

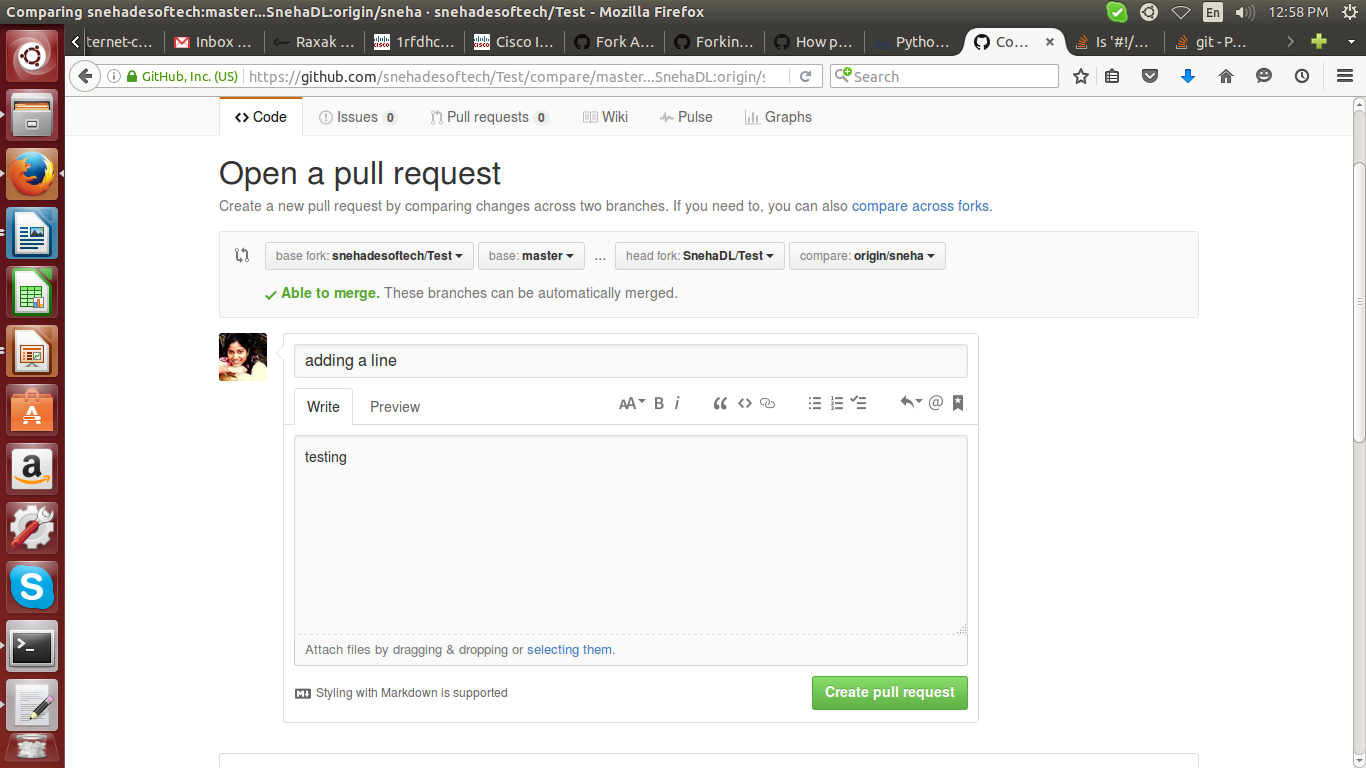
*/Test# git push origin origin/sneha*

4. Now the changes will be pushed to your Fork repository.

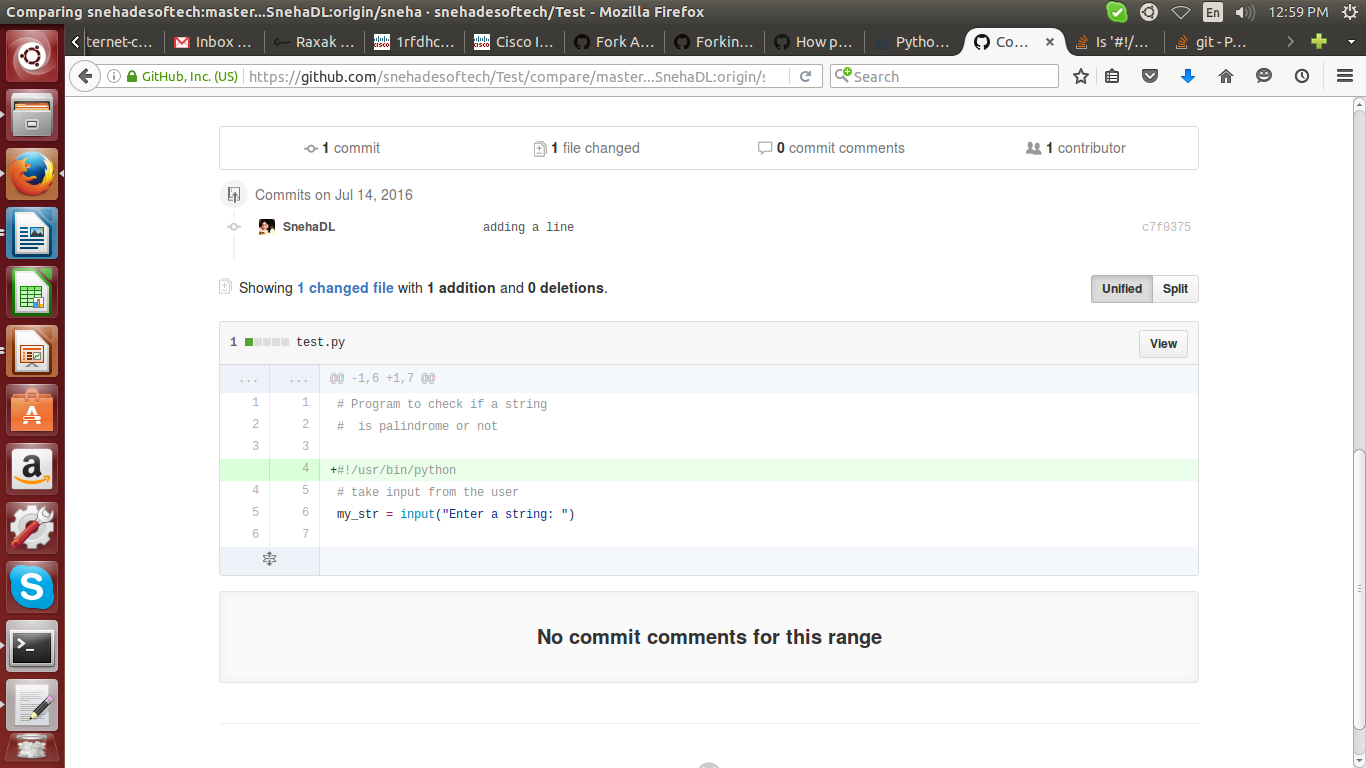
5. We have to create a pull request from the Fork to the main repo by clicking on the '**New pull request button'** shown below



6. Open a new pull request by choosing the base branch and the target branch.

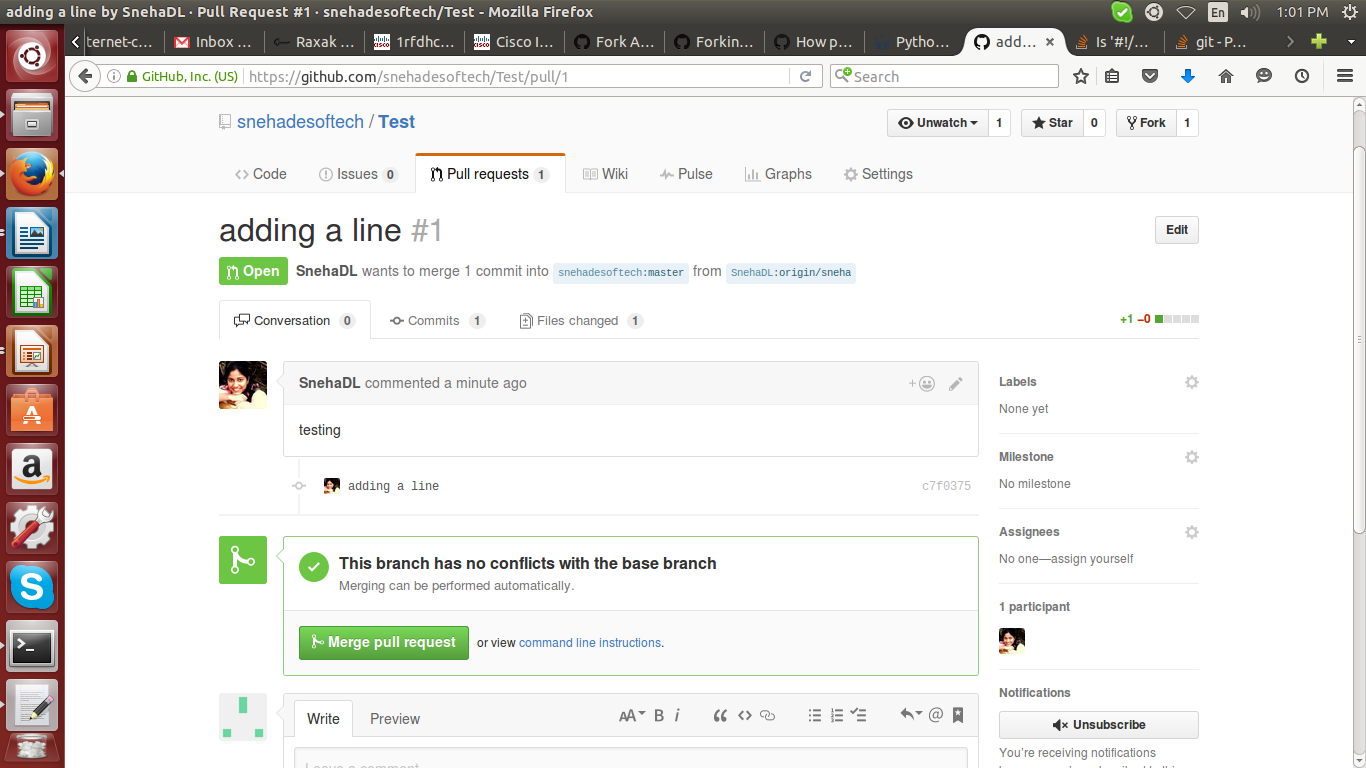


7. You can see the code difference at the bottom of the page and add description and comments to your pull request.



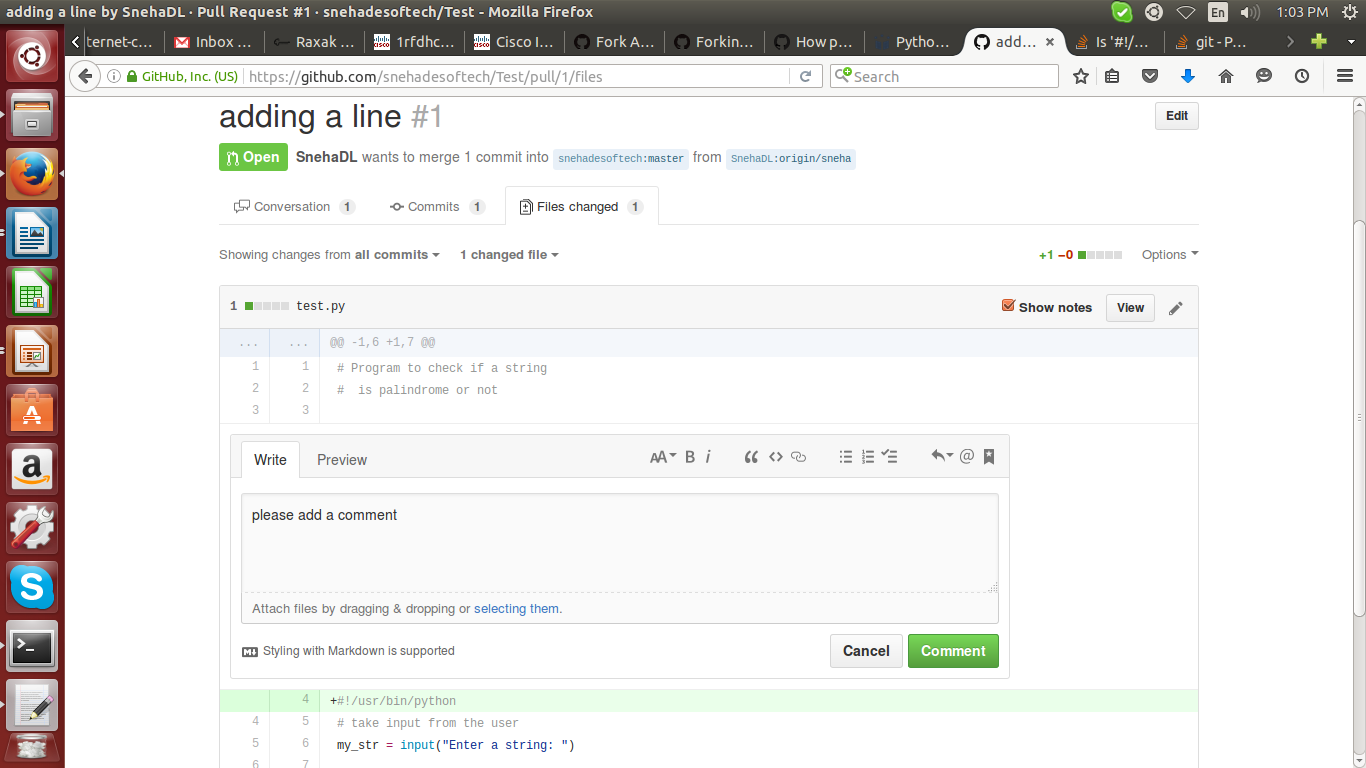
8. Once the pull request is created, the moderator can review it and put some comments.

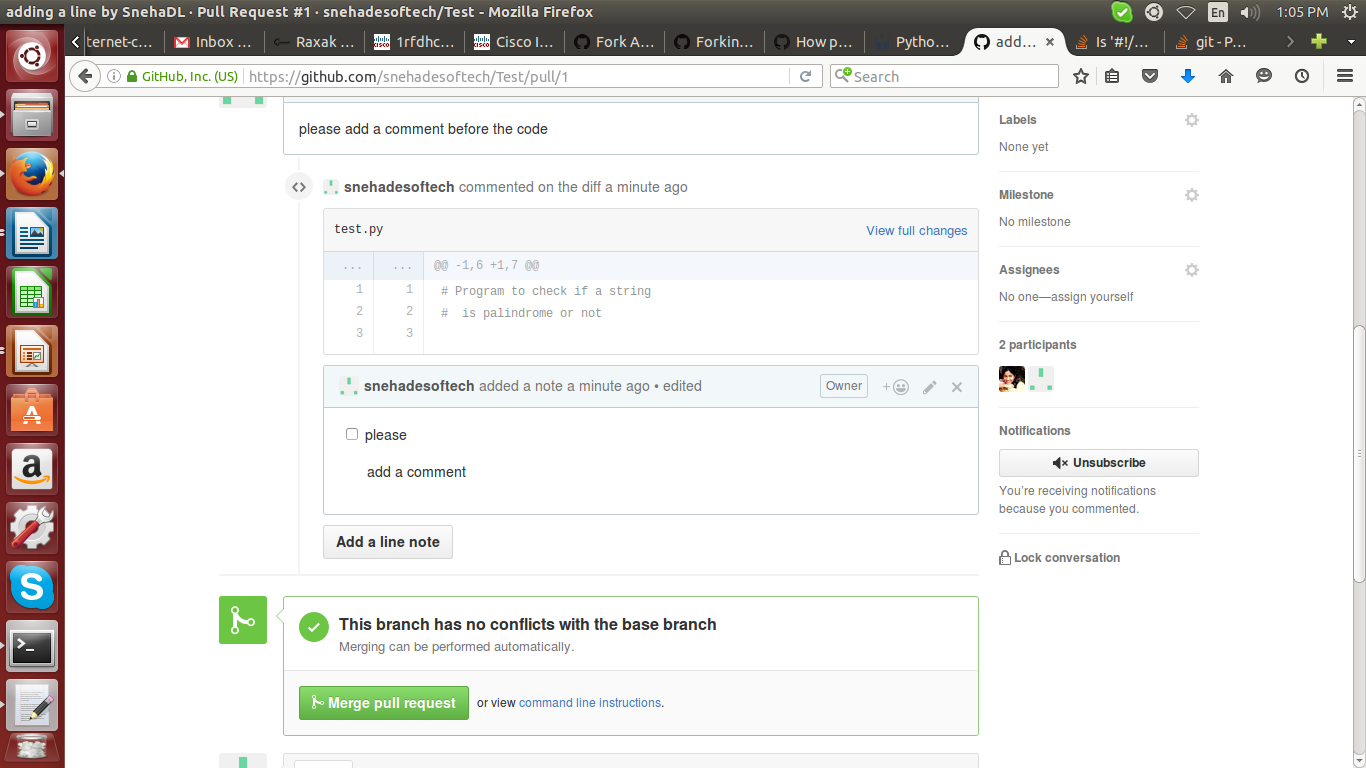
Following image shows the pull request waiting for the review.



9. The reviewer can add inline comments for the expected code changes as shown below

To add comments, and write a comment.



10. Tasks can be created and assigned to the developer as shown below

11. To complete the code review changes, developer must follow the below steps:

* checkout the branch again in the local repo
* make changes
* add changes
* commit changes
* push the code

Following are the commands to be used:

*root@MWPYT106:/home/sneha/Desktop/Raxak/tesfork/Test# git checkout origin/origin/sneha*

*root@MWPYT106:/home/sneha/Desktop/Raxak/tesfork/Test# git checkout origin/sneha*

*Switched to branch 'origin/sneha'*

*root@MWPYT106:/home/sneha/Desktop/Raxak/tesfork/Test# ls -ltr*

*total 8*

*-rw-r--r-- 1 root root 520 Jul 14 12:46 README.md*

*-rw-r--r-- 1 root root 405 Jul 14 12:51 test.py*

*root@MWPYT106:/home/sneha/Desktop/Raxak/tesfork/Test# vi test.py*

*root@MWPYT106:/home/sneha/Desktop/Raxak/tesfork/Test# git diff*

*diff --git a/test.py b/test.py*

*index a5a8470..75ae3ca 100644*

*--- a/test.py*

*+++ b/test.py*

*@@ -1,6 +1,6 @@*

*# Program to check if a string*

*# is palindrome or not*

*-*

*+#adding the path for python*

*#!/usr/bin/python*

*# take input from the user*

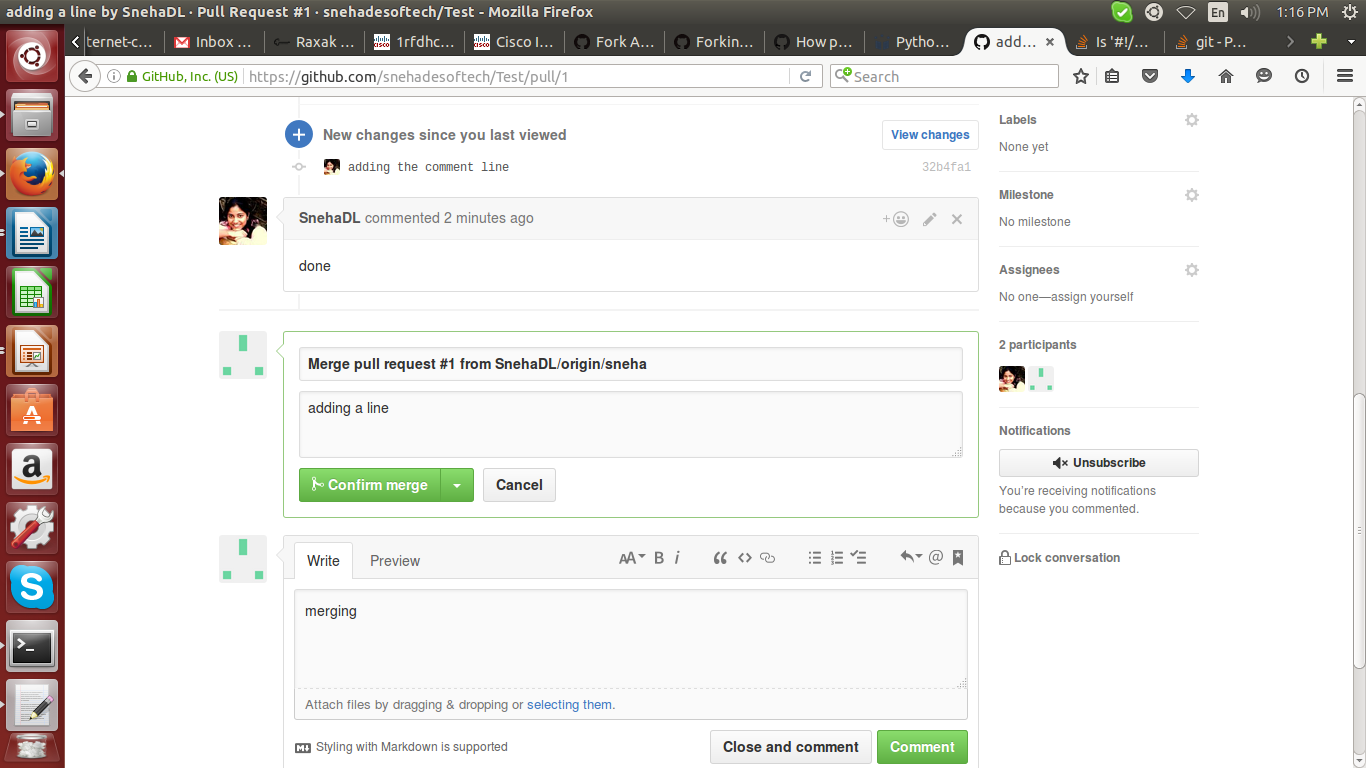
*my\_str = input("Enter a string: ")*

*root@MWPYT106:/home/sneha/Desktop/Raxak/tesfork/Test# git add -all*

*root@MWPYT106:/home/sneha/Desktop/Raxak/tesfork/Test# git commit -m "adding the comment line"*

*root@MWPYT106:/home/sneha/Desktop/Raxak/tesfork/Test# git push origin origin/sneha*

12. Once the reviewed code and comments are resolved, the reviewer can merge the code.



Lets take a scenario where the master branch has changes newly added which are to be pulled and merged in the local fork to create a new pull request.

13. Lets create a new branch from the master and push the changes.

*# git checkout new*

*Switched to branch 'new'*

*root@MWPYT106:/home/sneha/main/Test# git push origin new*

*Counting objects: 5, done.*

*Delta compression using up to 4 threads.*

*Compressing objects: 100% (3/3), done.*

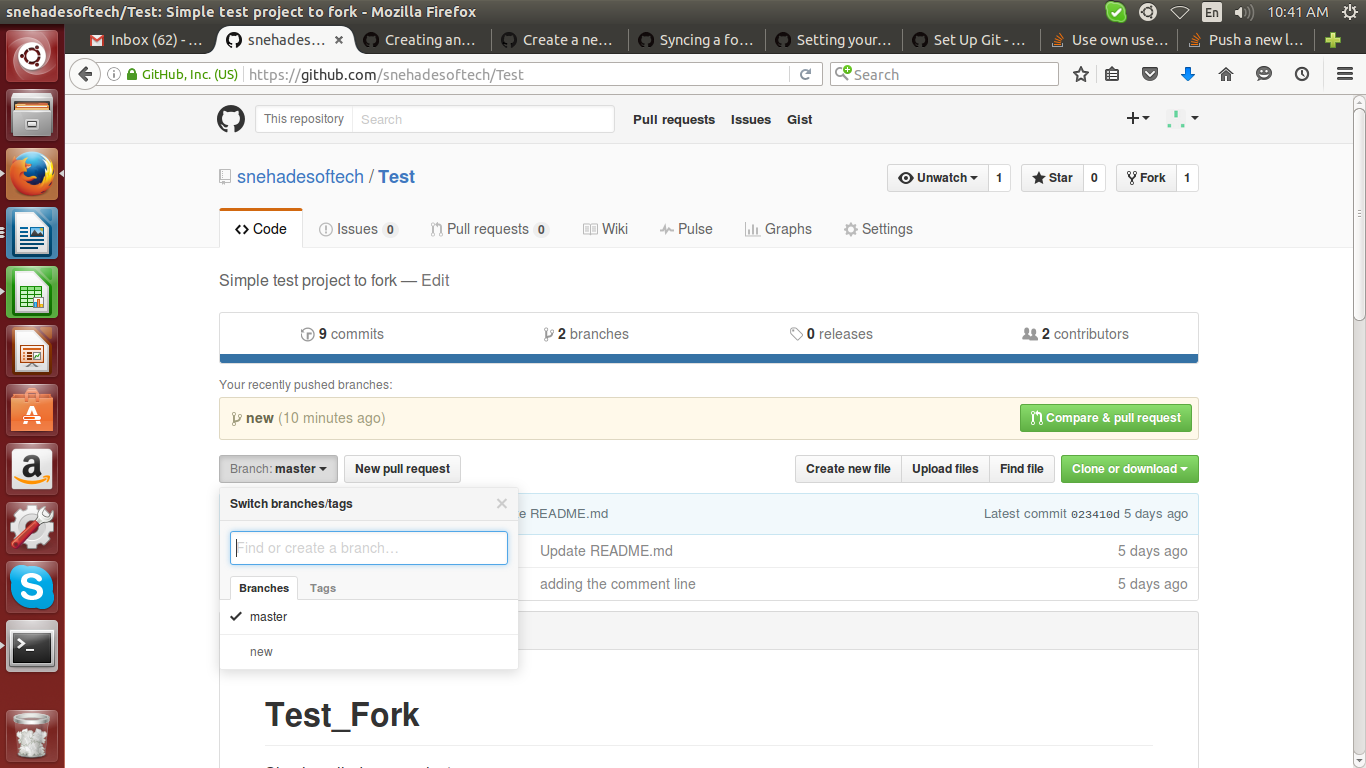
*Writing objects: 100% (3/3), 449 bytes | 0 bytes/s, done.*

*Total 3 (delta 1), reused 0 (delta 0)*

*To https://github.com/snehadesoftech/Test.git*

*\* [new branch] new -> new*

Branch 'new' appears in the GUI as shown below:



Following is the diff between this new branch and the fork.

# *diff main/Test/test.py fork/Test/test.py*

*3,4c3*

*< #adding the path for python*

*< #!/usr/bin/python*

*---*

*>*

*19,26d17*

*<*

*< def bubble\_sort(items):*

*< """ Implementation of bubble sort """*

*< for i in range(len(items)):*

*< for j in range(len(items)-1-i):*

*< if items[j] &gt; items[j+1]:*

*< items[j], items[j+1] = items[j+1], items[j] # Swap!*

*root@MWPYT106:/home/sneha/main/Test# git branch -r*

*origin/HEAD -> origin/master*

*origin/master*

*origin/new*

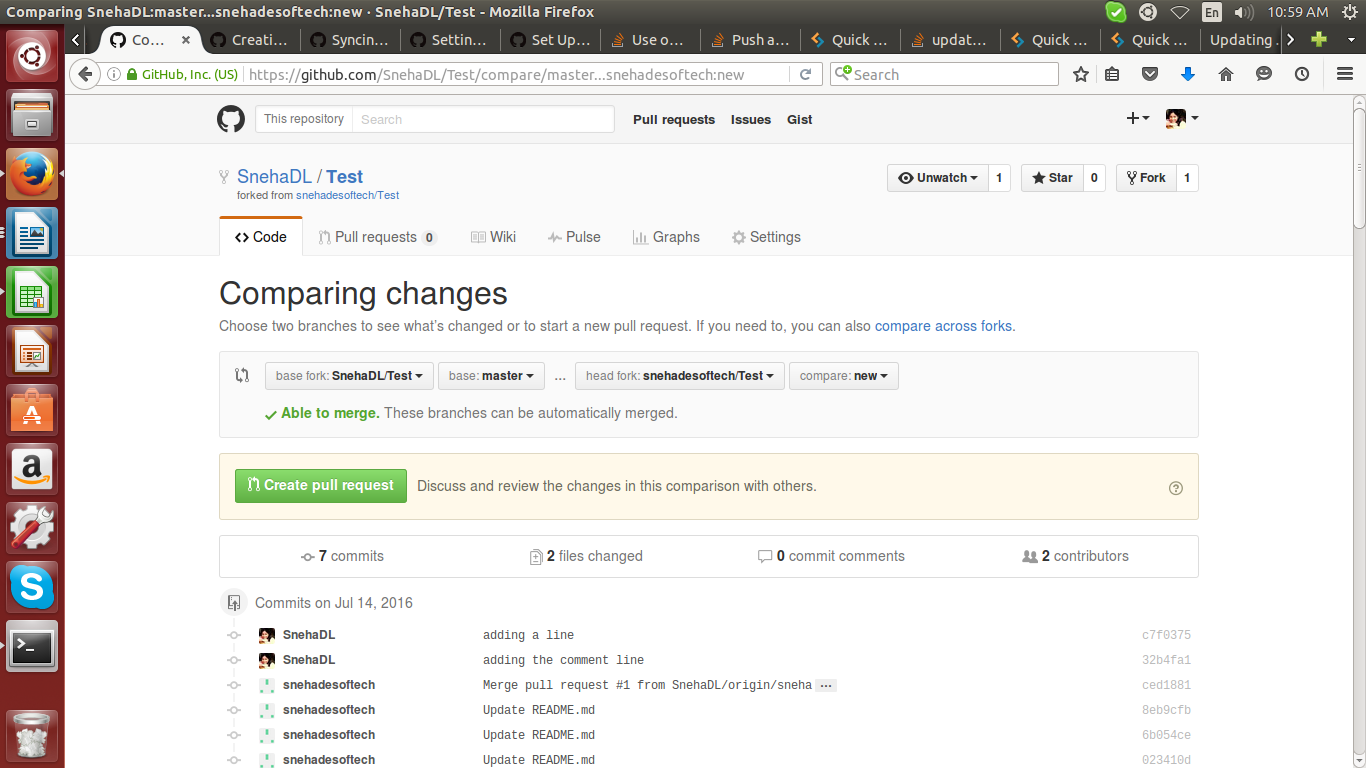
*root@MWPYT106:/home/sneha/fork/Test# git branch -r*

*origin/HEAD -> origin/master*

*origin/master*

*origin/origin/sneha*

The changes can be viewed from the GitHub UI as well, click on new pull request and click 'Swicth the base' or manually swicth the bases to find the diff as shown below.



In order to bring these changes to fork, follow the steps mentioned below:

14. Specify the remote upstream repository that needs to be synced with the fork.

root@MWPYT106:/home/sneha/fork/Test# git remote add upstream https://SnehaDL@github.com/snehadesoftech/Test.git

15. Fetch the branches and their respective commits from the upstream repository. Commits to master will be stored in a local branch, upstream/master.

r*oot@MWPYT106:/home/sneha/fork/Test# git fetch upstream*

*remote: Counting objects: 3, done.*

*remote: Compressing objects: 100% (2/2), done.*

*remote: Total 3 (delta 1), reused 3 (delta 1), pack-reused 0*

*Unpacking objects: 100% (3/3), done.*

*From https://github.com/snehadesoftech/Test*

*\* [new branch] master -> upstream/master*

*\* [new branch] new -> upstream/new*

*root@MWPYT106:/home/sneha/fork/Test# git branch -r*

*origin/HEAD -> origin/master*

*origin/master*

*origin/origin/sneha*

*upstream/master*

*upstream/new*

16. Check out your fork's branch that you are working on

*root@MWPYT106:/home/sneha/fork/Test# git checkout origin/sneha*

*Branch origin/sneha set up to track remote branch origin/sneha from origin.*

*Switched to a new branch 'origin/sneha'*

*root@MWPYT106:/home/sneha/fork/Test# git checkout sneha*

*error: pathspec 'sneha' did not match any file(s) known to git.*

*root@MWPYT106:/home/sneha/fork/Test# git checkout origin/sneha*

*Switched to branch 'origin/sneha'*

*Your branch is up-to-date with 'origin/origin/sneha'.*

17. Merge the changes from upstream/new into your local branch. This brings your fork's new branch into sync with the upstream repository, without losing your local changes.

*root@MWPYT106:/home/sneha/fork/Test# git merge upstream/new*

*Updating 32b4fa1..fa89ff3*

*Fast-forward*

*README.md | 64 +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++-*

*test.py | 8 ++++++++*

*2 files changed, 71 insertions(+), 1 deletion(-)*

*root@MWPYT106:/home/sneha/fork/Test# vi test.py*

*root@MWPYT106:/home/sneha/fork/Test# git add --all*

*root@MWPYT106:/home/sneha/fork/Test# git commit -m "adding the bubblesort comment"*

*[origin/sneha 33f8d95] adding the bubblesort comment*

*1 file changed, 1 insertion(+)*

*root@MWPYT106:/home/sneha/fork/Test# git push origin origin/sneha*

*Counting objects: 9, done.*

*Delta compression using up to 4 threads.*

*Compressing objects: 100% (6/6), done.*

*Writing objects: 100% (6/6), 738 bytes | 0 bytes/s, done.*

*Total 6 (delta 2), reused 0 (delta 0)*

*To https://github.com/SnehaDL/Test.git*

*32b4fa1..33f8d95 origin/sneha -> origin/sneha*

18. Now create the Pull request again with these changes.

