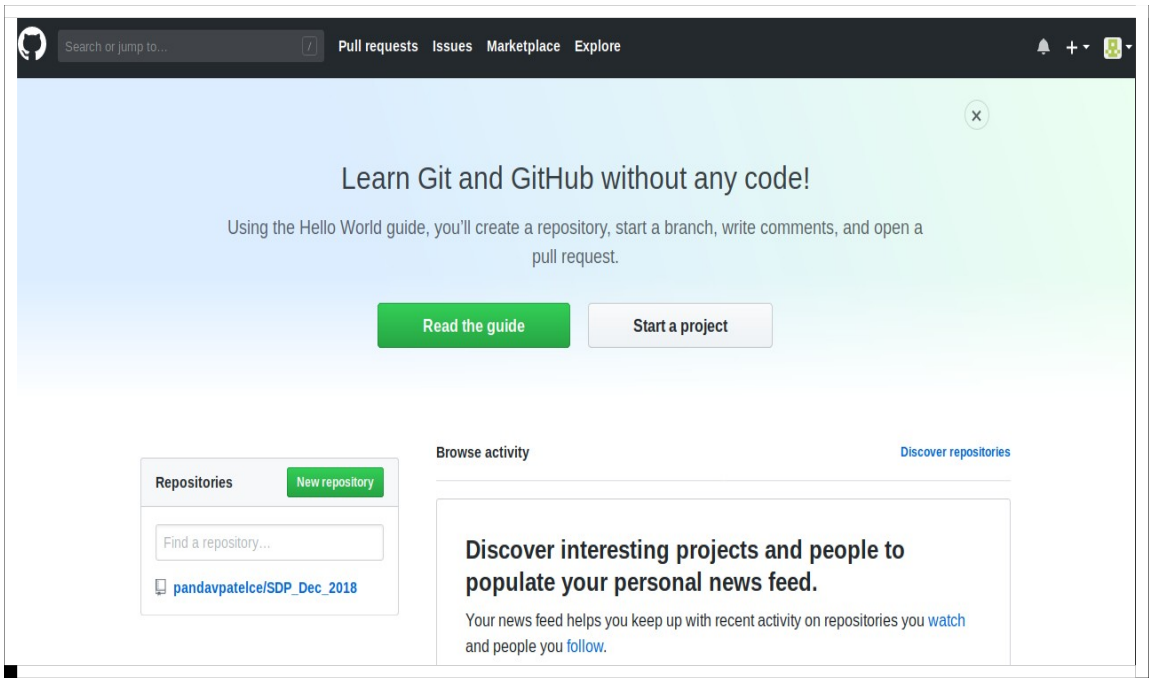
This document contains answer to frequently asked questions regarding github workflow.
This document assumes that you have some basic understaing of git – mainly you should know what is repository, cloning, branching, commit, remote repository; and what are the concepts of merge, pull and push. If you are not already familiar with those concepts, Its higly recommended to read first three chapters (Getting Started, Git Basics and Git Branching) of pro git (open source) book.

Here are the questions which would be answered by this doc. If you have a question that is not covered in this doc, please send it my way at pandavpatel.ce@ddu.ac.in and I will add it to this doc.

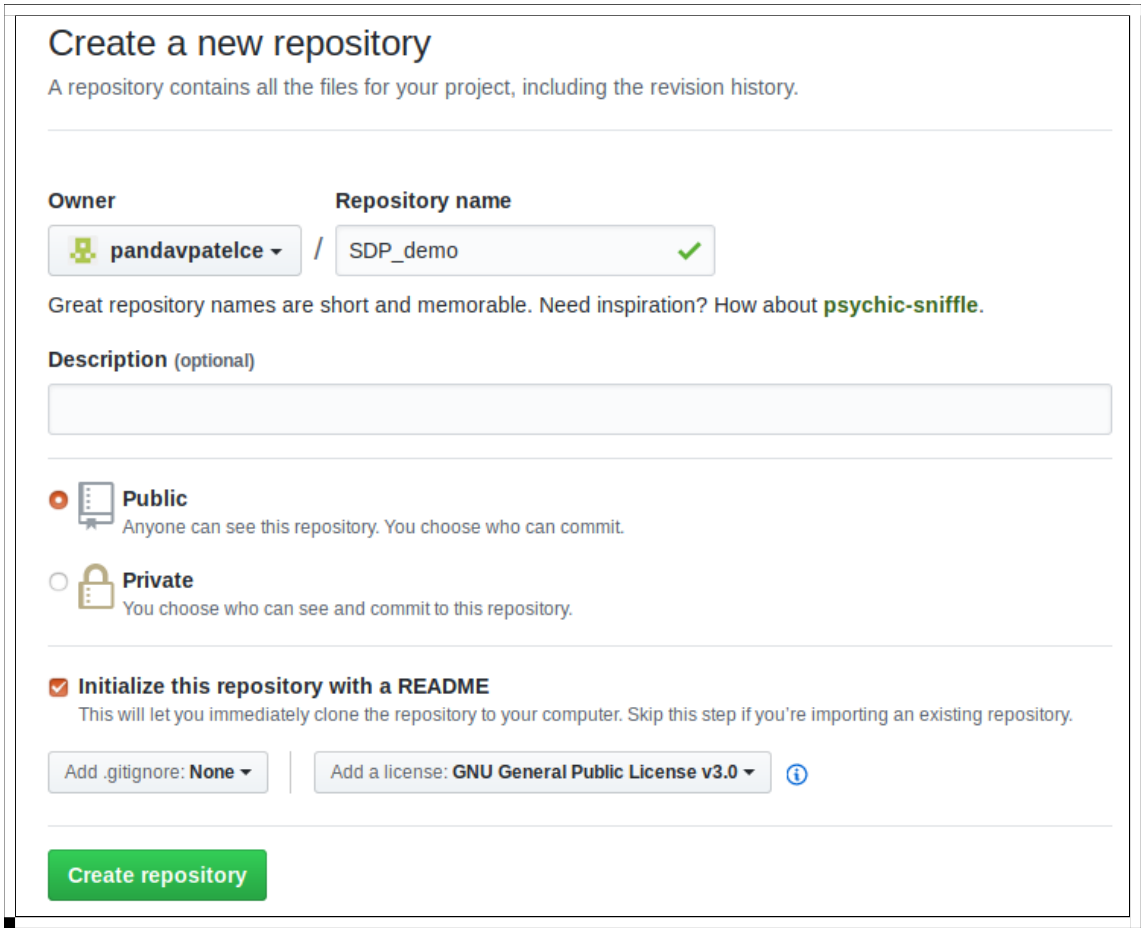
1. How to create github repository and how to add team members?
2. How to clone the repository of your project?
3. How to create new branch and checkout newly created branch?
4. How to make changes and commit those changes?
5. How to push new branch from local repository to the remote (origin) github repository?
6. How to push new commits from branch in local repository to already existing branch in remote repository?
7. How raise pull request to merge your work with default (master) branch on the remote (origin) repository?

How to create github repository and how to add team members?

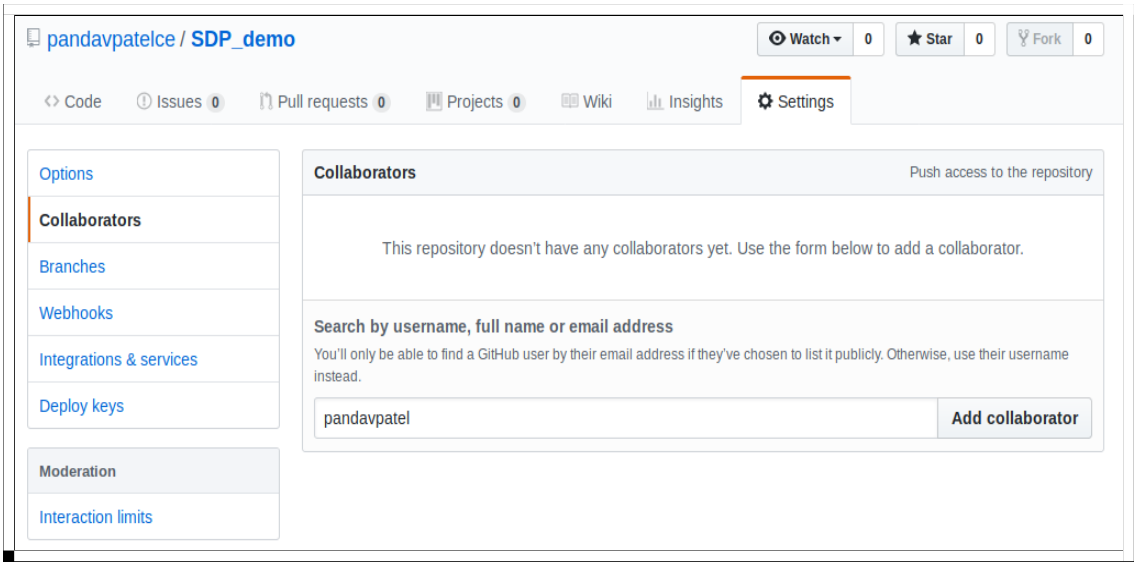
1. All members of the team need to create github account
2. One of the team member creates a repository for the project under his/her account
   a) Log in to github account
   b) click on ***New repository*** button



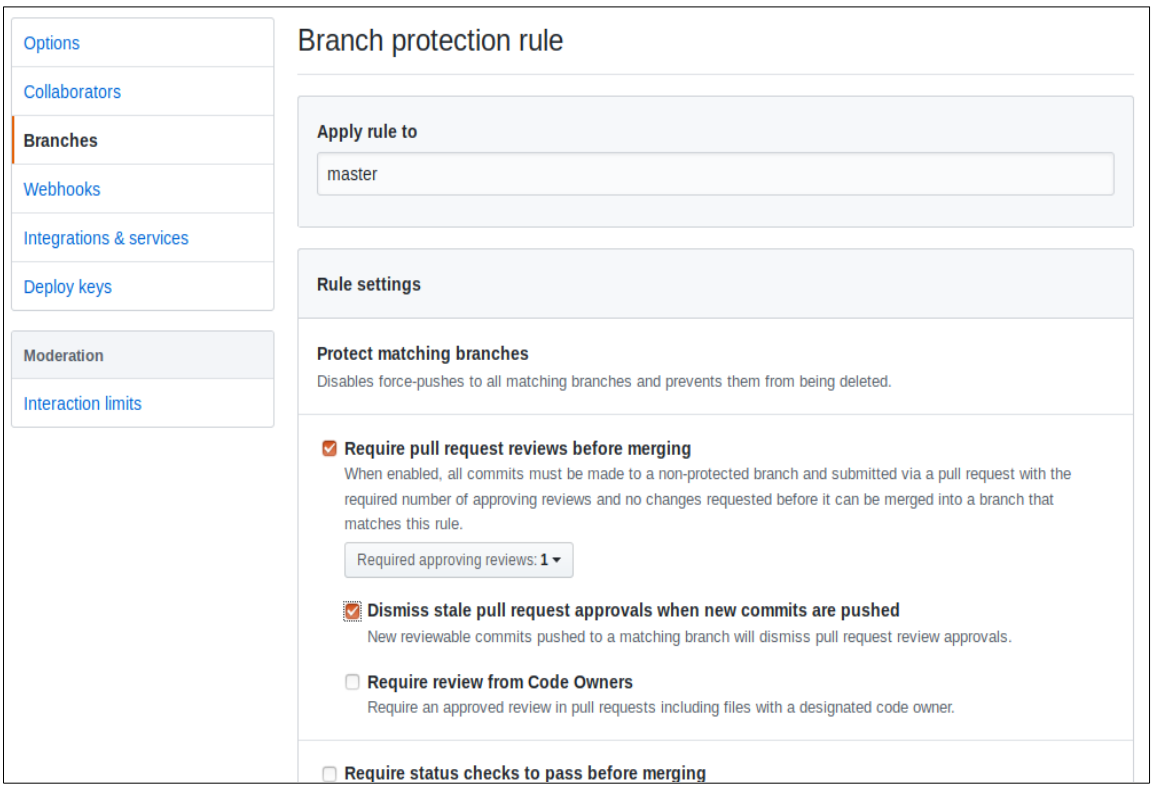
   c) Give repository name, initialize it with README and select license. Then click ***Create repository*** button

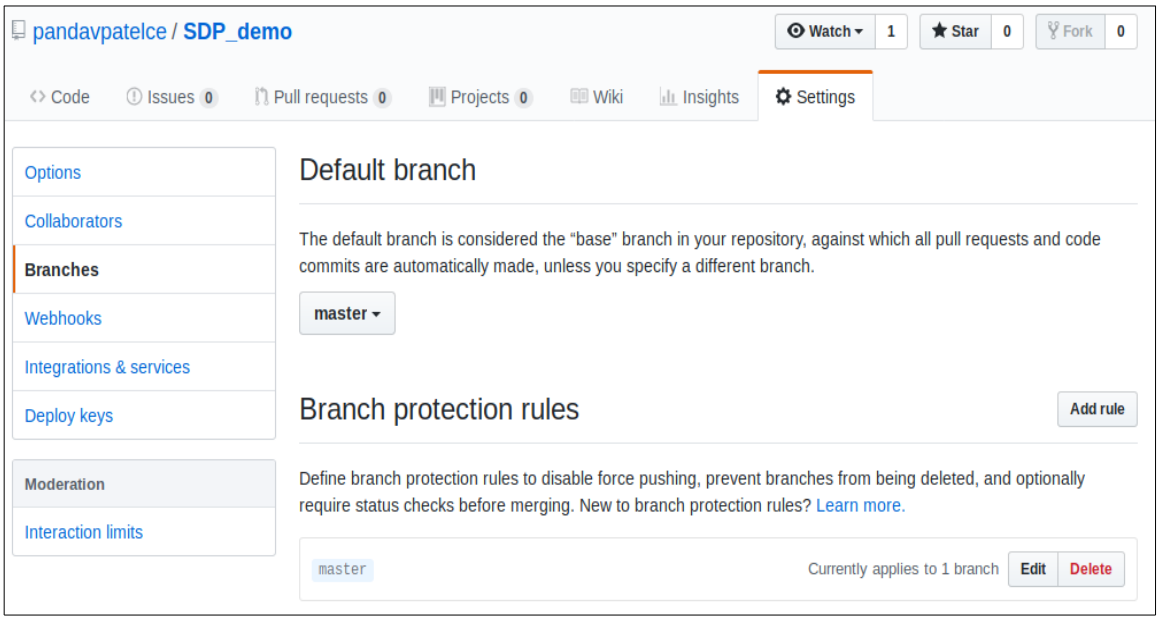


3. Add rest of the team members to repository
   a) Once you create repository, go to settings of the repository. On left side there is collaborators, click on that. Add rest of the team members using github username

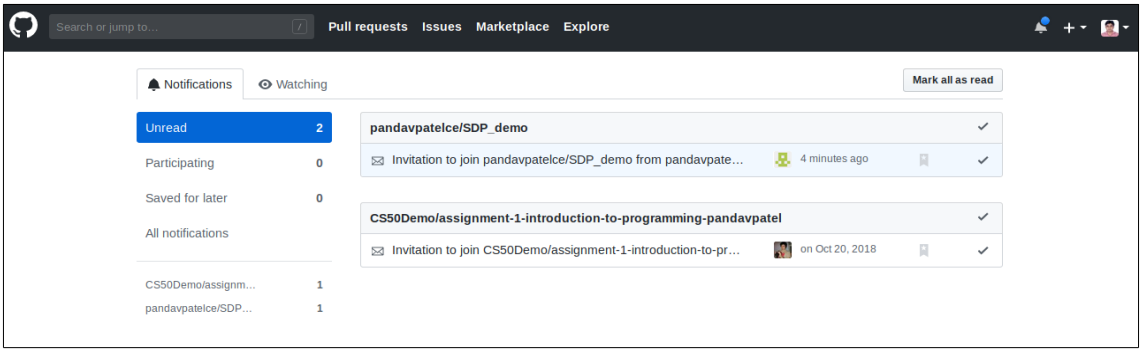


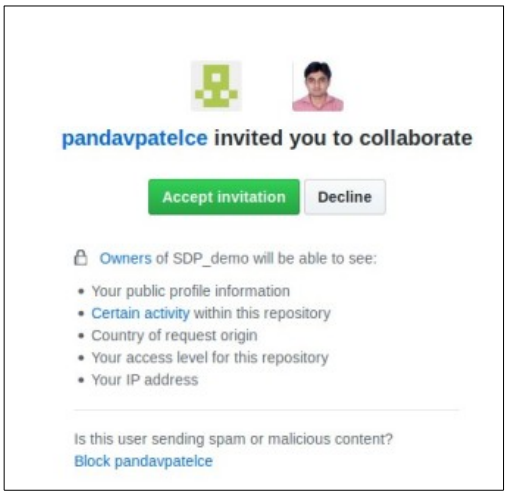
4. Add branch branch protection rule to master branch
   a) Tick "Require pull request reviews before merging". Tick "Dismiss stale pull request approvals when new commits are pushed". Then add rule and save changes.





5. Rest of the team members need to login to their github account and accept the invitation
   a) After login to the account, click on the bell icon on the top-right corner. Click on invitation notification for the repository
   b) Accept the invitation by clicking on ***Accept invitation*** button. Once you do accept the invitation, you will have access to the repository

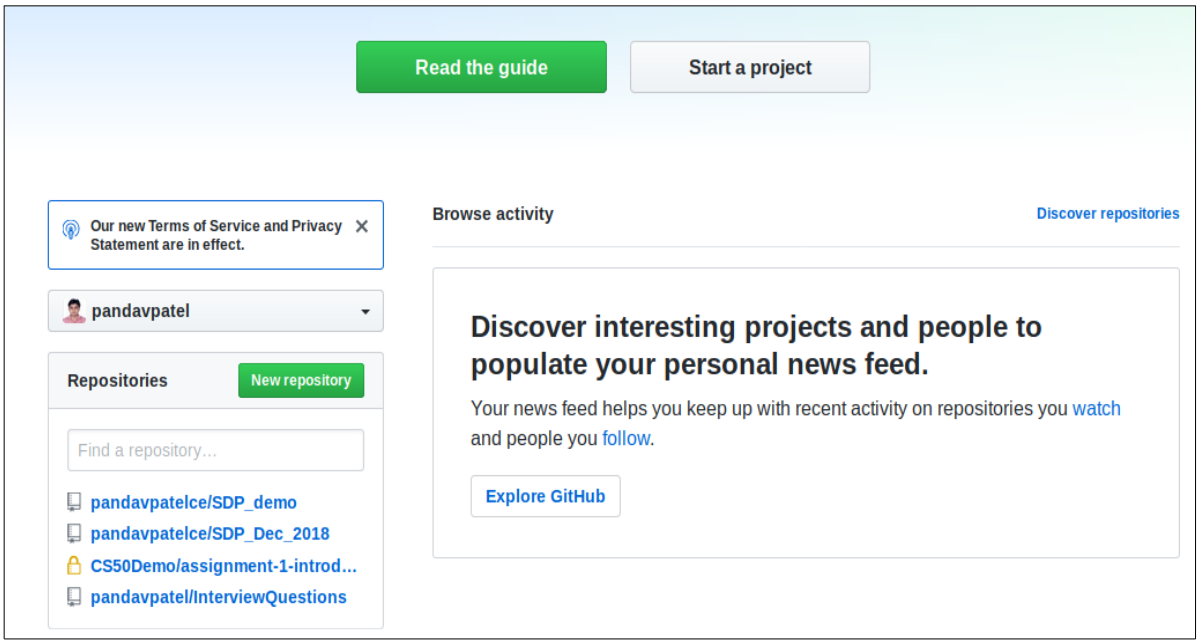




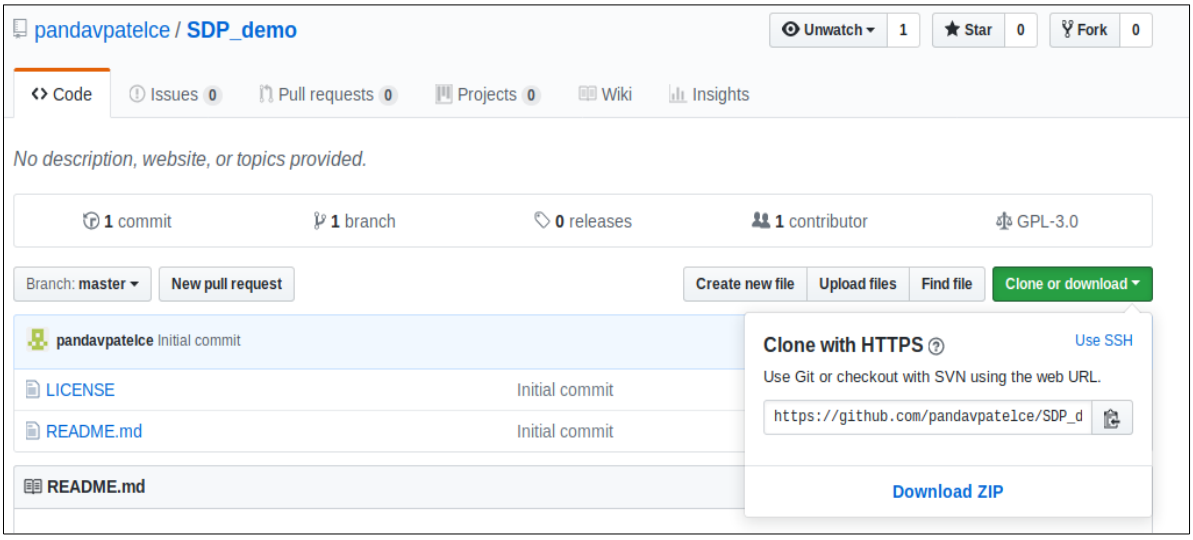
How to clone the repository of your project?

When you clone a repository, you are basically creating copy of the repository. Once you clone the repository, git will set which local branch is tracking which remote branch. So when you pull or push changes it goes to the remote branch that is being tracked by local branch.

1. Login to your github account
2. Click on the repository you want to clone



3. Once you are on the repository page, click on ***Clone or download*** button. It will open small pop-up, copy the URL from that pop-up



4. Go to github client on your machine (in my case git command from terminal). Change working directory to location where you want to clone the repository. And run command ***git clone <URL copied in previous step>***

```
pandav@pandav-ASAG3730-ASAG1730:~$ clear

pandav@pandav-ASAG3730-ASAG1730:~$ cd git
pandav@pandav-ASAG3730-ASAG1730:~/git$ mkdir demo
pandav@pandav-ASAG3730-ASAG1730:~/git/demo$ git clone https://github.com/pandavpatelce/SDP_demo.git
Cloning into 'SDP_demo'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), done.
pandav@pandav-ASAG3730-ASAG1730:~/git/demo$ ls
SDP_demo
pandav@pandav-ASAG3730-ASAG1730:~/git/demo$ cd SDP_demo/
pandav@pandav-ASAG3730-ASAG1730:~/git/demo/SDP_demo$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
pandav@pandav-ASAG3730-ASAG1730:~/git/demo/SDP_demo$
```

After command has run successfully, you will observe that new directory with repository name has been created in the working directory. Now that is your local repository. Which is like copy of remote repository on github with all history.

In the above screenshot it is clear that user is on the master branch and there are no changes that needs to be committed. Also there are no commits which needs to be pushed to remote repo. Also no commits need to be pulled from master branch on remote repo to master branch on local repo.

How to create new branch and checkout newly created branch?

There are command lines which can combine following steps into one step. But let us look at this step by step. You can read more about branches, here.

1. Create a new branch using command ***git branch <new_branch_name>***
   - ***git branch*** command without new branch name, will list all the branches

- ***** indicates current branch you are on

2. Checkout newly created branch
   - After running ***git branch <new_branch_name>*** command, new branch will be created, but it will not checkout new branch
   - That means you would still be in branch which you were using before running this command.
   - Any changes that you make, would be against branch which is currently checked out
   - To check out newly created branch use ***checkout <branch_to_checkout>*** command
   - After checking out new branch, all the changes that you make will be against that newly checked out branch.

How to make changes and commit those changes?

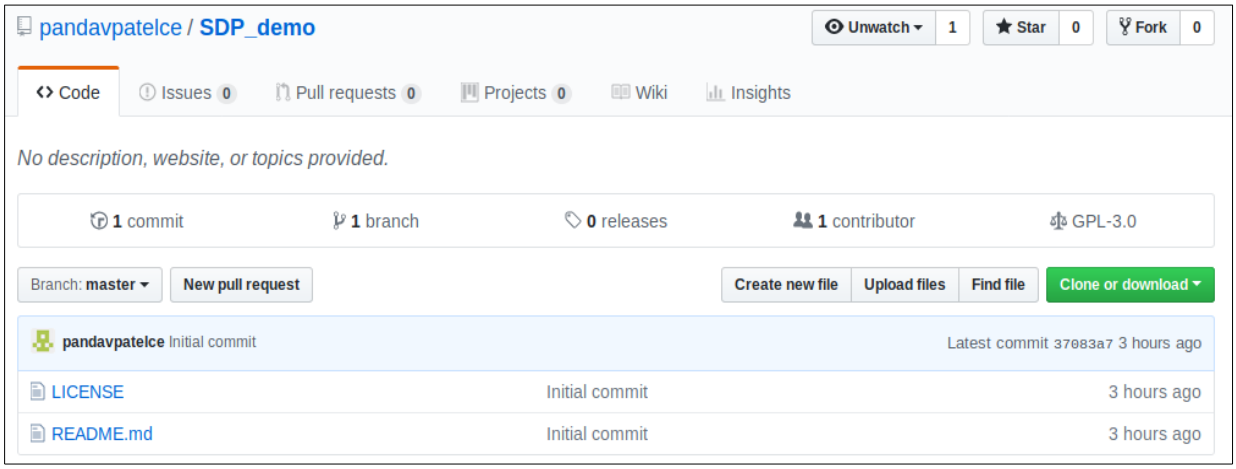
1. Make changes to files in the project
   - You can add and remove files too from the current branch
   - You can use editor of your choice for these changes
2. Stage changes that you want to commit
   - Git has something called working directory and staging area, you can read more about that, here.
   - Once you have modified, added or removed files, you need to stage them before you commit.
   - Its possible to skip staging all together (using certain option with git command), but lets look at it step by step
   - At this point you can use ***git status*** command to list all the changes which are not yet staged
   - You can use ***git add <file name>*** command to stage a particular file. You can use ***git add **** command to stage all the changes in your working area
3. Commit changes that you have staged
   - At this point you can use ***git status*** command to list all the changes which are staged for commit
   - Once you have added files to staging area, you can use ***git commit -m "commit message"*** command to commit your staged changes
   - At this point you can use ***git status*** command to verify that there are no changes reaming in staging area or working directory
   - You can use ***git log*** command to see you commits

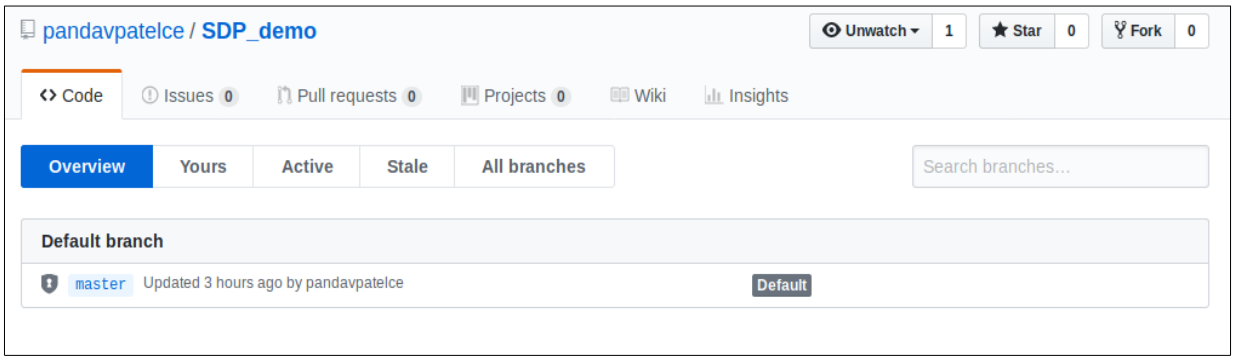
```
pandav@pandav-ASAG3730-ASAG1730:~/git/demo/SDP_demo$ git status
On branch feature1
nothing to commit, working tree clean
pandav@pandav-ASAG3730-ASAG1730:~/git/demo/SDP_demo$ vi test.txt
pandav@pandav-ASAG3730-ASAG1730:~/git/demo/SDP_demo$ git status
On branch feature1
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        test.txt

nothing added to commit but untracked files present (use "git add" to track)
pandav@pandav-ASAG3730-ASAG1730:~/git/demo/SDP_demo$ git add test.txt
pandav@pandav-ASAG3730-ASAG1730:~/git/demo/SDP_demo$ git status
On branch feature1
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   test.txt

pandav@pandav-ASAG3730-ASAG1730:~/git/demo/SDP_demo$ git commit -m "adding test.txt"
[feature1 e486999] adding test.txt
 1 file changed, 2 insertions(+)
 create mode 100644 test.txt
pandav@pandav-ASAG3730-ASAG1730:~/git/demo/SDP_demo$ git status
On branch feature1
nothing to commit, working tree clean
pandav@pandav-ASAG3730-ASAG1730:~/git/demo/SDP_demo$
```

```
pandav@pandav-ASAG3730-ASAG1730:~/git/demo/SDP_demo$ git log
commit e4869992d8704230588ba34e4a2e4d76184f2943 (HEAD -> feature1)
Author: pandavpatelce <pandavpatel.ce@ddu.ac.in>
Date:   Wed Jan 2 14:15:44 2019 +0530

    adding test.txt

commit 37083a7c1f36d66143e2813d087a74363741fb4e (origin/master, origin/HEAD, master)
Author: pandavpatelce <46277095+pandavpatelce@users.noreply.github.com>
Date:   Wed Jan 2 12:01:42 2019 +0530

    Initial commit
pandav@pandav-ASAG3730-ASAG1730:~/git/demo/SDP_demo$
```

How to push new branch from local repository to the remote (origin) github repository?

- You can work with multiple remote repositories using git. You can read more about that, here.
- What we are interested here is ***origin*** remote repository, for which git has created origin/master at the moment of cloning.
- If you observe closely, then there is only one branch on remote repository (origin), which is ***master*** branch. While local repository have ***master*** branch and ***feature1*** branch.

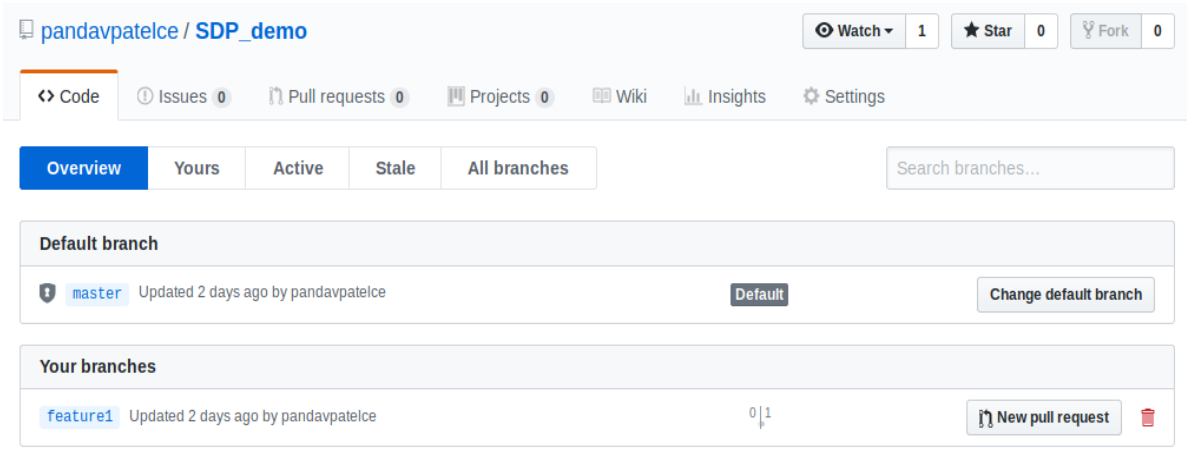




- How to push newly created ***feature1*** branch from local repository to remote repository?
- You can do that using command ***`git push -u origin <new_branch_name>`***

```
pandav@pandav-ASAG3730-ASAG1730:~/git/demo/SDP_demo$ git push -u origin feature1
Username for 'https://github.com': pandavpatelce
Password for 'https://pandavpatelce@github.com':
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 388 bytes | 388.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'feature1' on GitHub by visiting:
remote:      https://github.com/pandavpatelce/SDP_demo/pull/new/feature1
remote:
To https://github.com/pandavpatelce/SDP_demo.git
 * [new branch]      feature1 -> feature1
Branch 'feature1' set up to track remote branch 'feature1' from 'origin'.
pandav@pandav-ASAG3730-ASAG1730:~/git/demo/SDP_demo$ git status
On branch feature1
Your branch is up to date with 'origin/feature1'.

nothing to commit, working tree clean
pandav@pandav-ASAG3730-ASAG1730:~/git/demo/SDP_demo$
```

- If you observe in the above screenshot, ***git status*** message not says “Your branch is up to date with 'origin/feature1'”. Means now there is a new ***feature1*** branch on remote (origin) repository. We can confirm it by looking at it from github page.



How to push new commits from branch in local repository to already existing branch in remote repository?

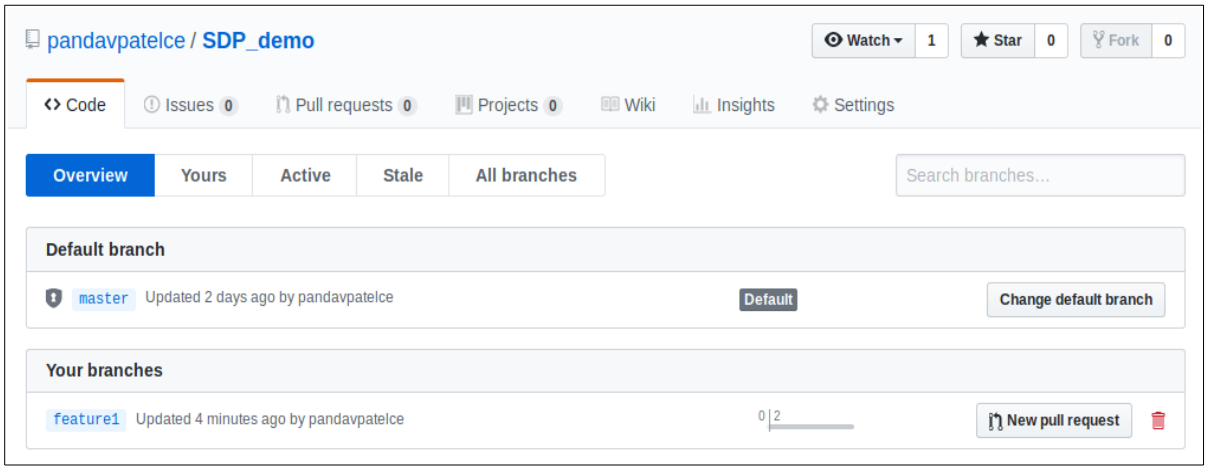
- If you want to do some further work into feature1 branch, then you can continue working on feature1 branch on your local machine. Commit those changes on local machine and then can push your changes to ***origin/feature1*** branch using ***git push*** command.
- Make sure that your have commited your changes and current checked out branch is the branch you want to push (in this case feature1)

```
pandav@pandav-ASAG3730-ASAG1730:~/git/demo/SDP_demo$ git status
On branch feature1
Your branch is up to date with 'origin/feature1'.

nothing to commit, working tree clean
pandav@pandav-ASAG3730-ASAG1730:~/git/demo/SDP_demo$ vi test.txt
pandav@pandav-ASAG3730-ASAG1730:~/git/demo/SDP_demo$ git status
On branch feature1
Your branch is up to date with 'origin/feature1'.

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:   test.txt

no changes added to commit (use "git add" and/or "git commit -a")
pandav@pandav-ASAG3730-ASAG1730:~/git/demo/SDP_demo$ git add test.txt
pandav@pandav-ASAG3730-ASAG1730:~/git/demo/SDP_demo$ git status
On branch feature1
Your branch is up to date with 'origin/feature1'.

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

        modified:   test.txt

pandav@pandav-ASAG3730-ASAG1730:~/git/demo/SDP_demo$ git commit -m "making changes to test.txt"
[feature1 7e6c688] making changes to test.txt
 1 file changed, 1 insertion(+)
pandav@pandav-ASAG3730-ASAG1730:~/git/demo/SDP_demo$
```

```
pandav@pandav-ASAG3730-ASAG1730:~/git/demo/SDP_demo$ git status
On branch feature1
Your branch is ahead of 'origin/feature1' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
pandav@pandav-ASAG3730-ASAG1730:~/git/demo/SDP_demo$ git push
Username for 'https://github.com': pandavpatelce
Password for 'https://pandavpatelce@github.com':
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 302 bytes | 302.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/pandavpatelce/SDP_demo.git
   e486999..7e6c688  feature1 -> feature1
pandav@pandav-ASAG3730-ASAG1730:~/git/demo/SDP_demo$ git status
On branch feature1
Your branch is up to date with 'origin/feature1'.

nothing to commit, working tree clean
pandav@pandav-ASAG3730-ASAG1730:~/git/demo/SDP_demo$
```

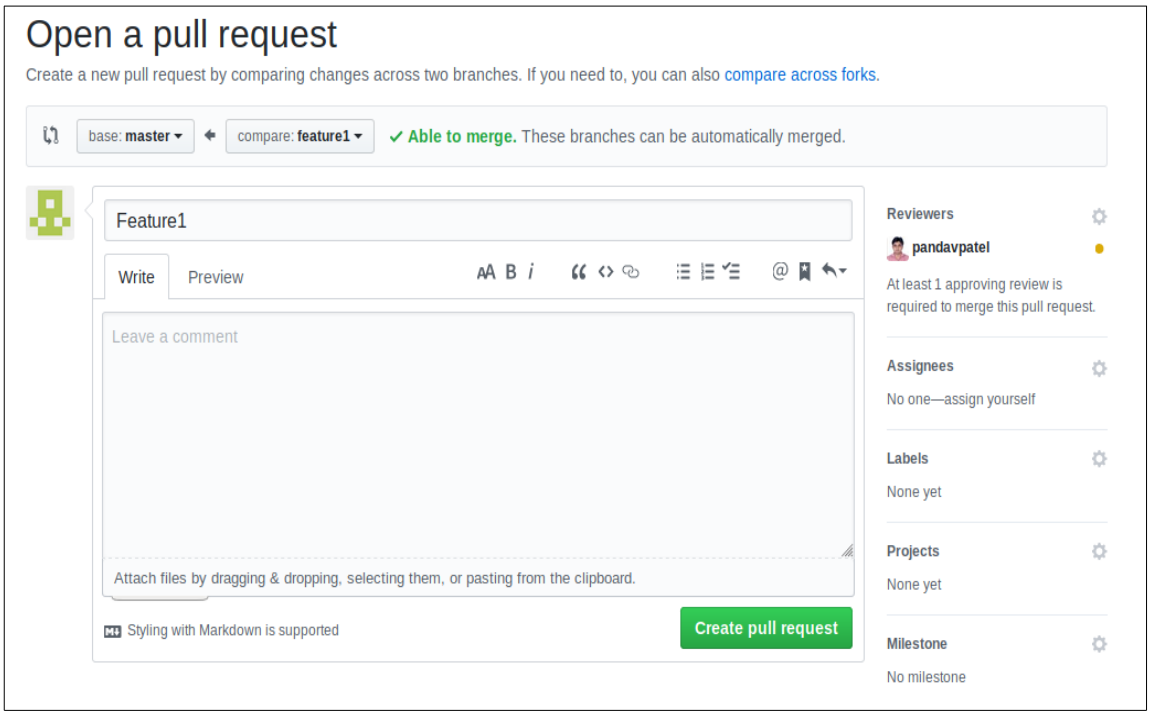
How raise pull request to merge your work with default (master) branch on the remote (origin) repository?

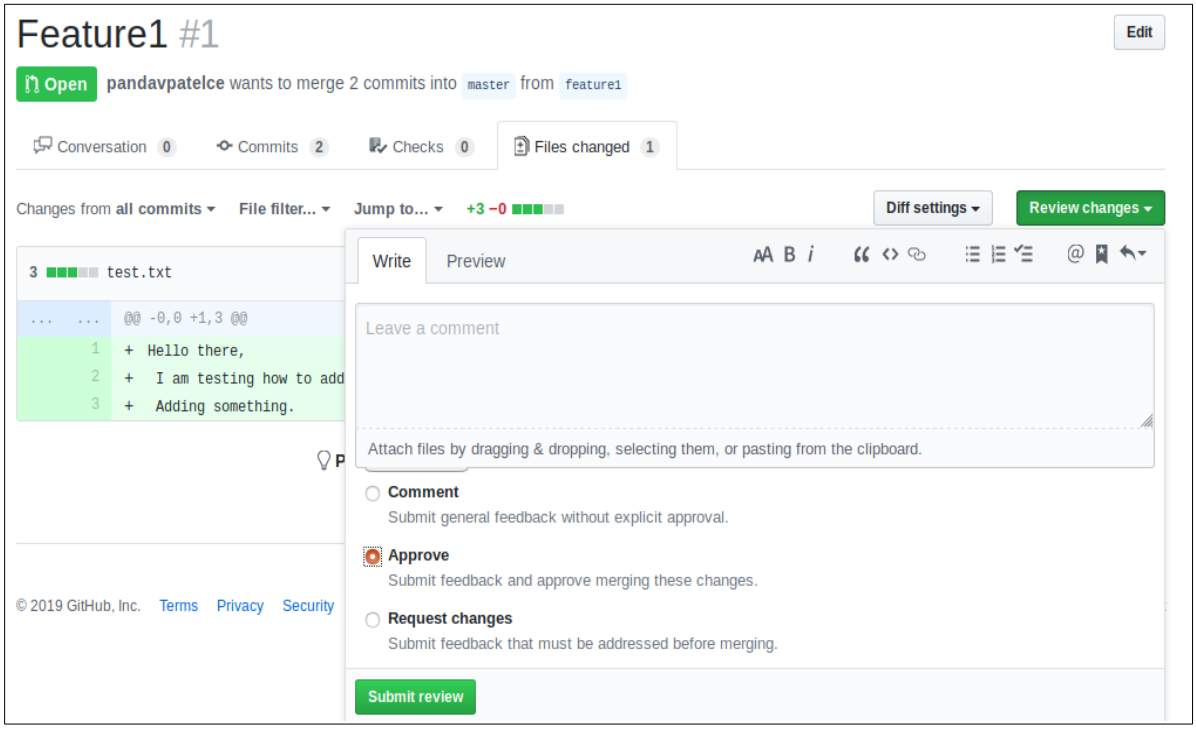
- In below screenshot you can see that feature1 branch is 2 (***0|2***) commits ahead of master branch
- Also there is ***New pull request*** button against feature1 branch. Click that to create pull request



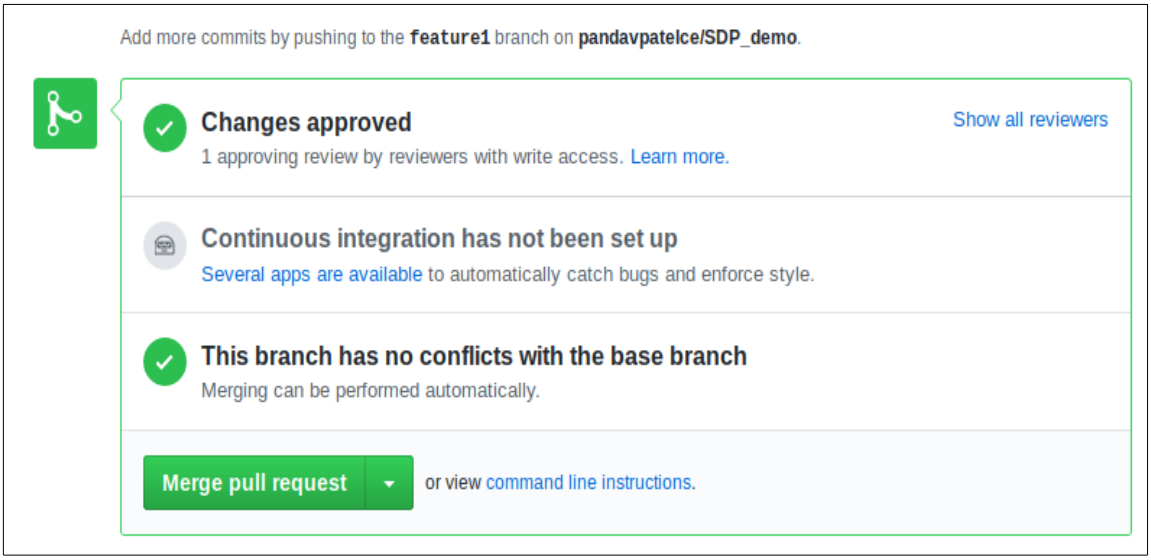
- Add reviewers (who will give comments on your code), and other details. Then click C***reate pull request*** button
- Make sure your base branch is ***master*** in case you want to merge your changes to ***master*** branch. Which would be the case most of the time.
- Once pull request is created reviewers can give comments and you need to make chages to address those comments. Then push new changes to ***feature1*** branch and pull request would

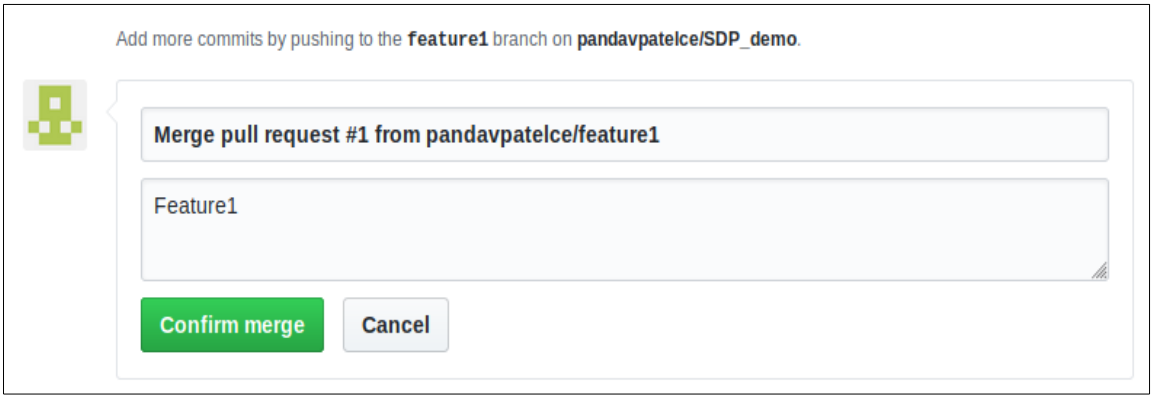
be updated with new changes. Once reviewer is satisfied with the quality of code. He/she can approve the pull request.

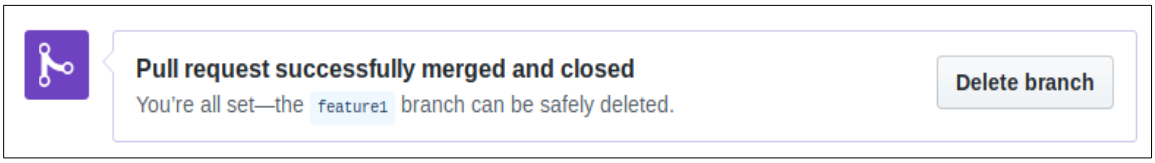




- Once pull request is approved you need to merge pull request and once merge is done, your changes become part of base branch (***master*** branch in this case)
- Once changes are merged into master and if you are not planning any changes to feature1 branch. Then you can delete that branch


Add more commits by pushing to the feature1 branch on pandavpatelce/SDP_demo.
Changes approved
1 approving review by reviewers with write access. Learn more.
Show all reviewers
Continuous integration has not been set up
Several apps are available to automatically catch bugs and enforce style.
This branch has no conflicts with the base branch
Merging can be performed automatically.
Merge pull request
or view command line instructions.


Add more commits by pushing to the feature1 branch on pandavpatelce/SDP_demo.
Merge pull request #1 from pandavpatelce/feature1
Feature1
Confirm merge
Cancel


Pull request successfully merged and closed
You're all set—the feature1 branch can be safely deleted.
Delete branch