It's standard practice to create separate branches for feature changes and submit them as "merge requests" to the main branch. Other people may fork your repo, make changes, and submit changes to the main codebase.

We can show this by creating and switching to a new branch.

```
PS C:\Tools\mycalculator> git branch dev
PS C:\Tools\mycalculator> git checkout dev
Switched to branch 'dev'
```

Next, modify the `Add` method in `Calculator.cs` to make it return an incorrect value.

```
PS C:\Tools\mycalculator> code .\Calculator\Calculator.cs
```

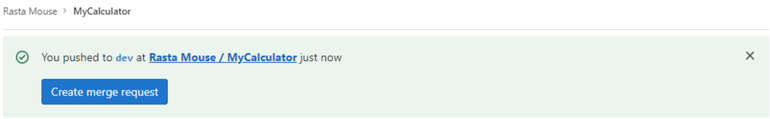
For example:

```
=> num1 * num2;
```

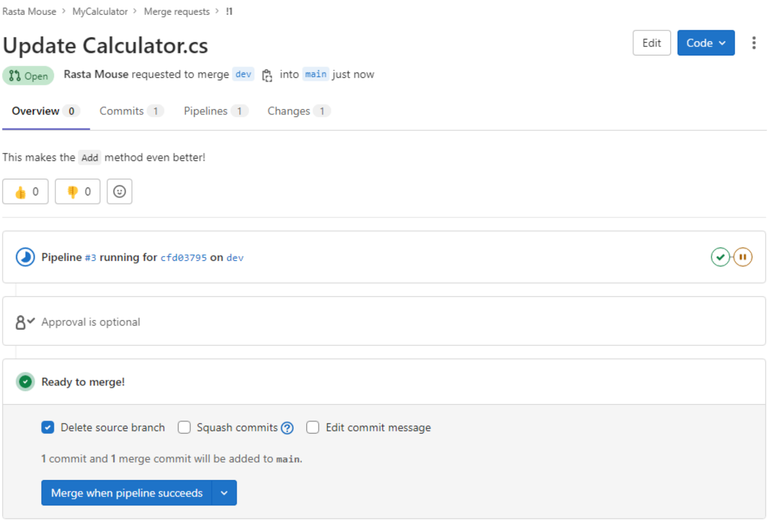
We'll then assume that's a job well done, commit it to the dev branch and push it to the repo.

```
PS C:\Tools\mycalculator> git add . && git commit -m "update Calculator.cs"
PS C:\Tools\mycalculator> git push -u origin dev
```

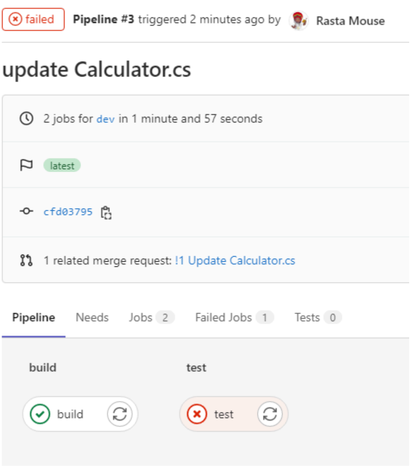
Within GitLab, you'll see a notification that the dev branch is ahead of main, and that we can create a request to merge the changes from dev into the main branch.



This will create a new merge request. You can go into the request and see what changes were made etc. As the owner of the repo, we would want some assurance that the changes will not break anything in our application. We would have to pull the changes to our dev machine, build and test the changes etc.



However, you can see that the CI/CD pipeline was automatically triggered when the merge request was created, which will run the build and test stages automatically. The build will pass but the test will fail.



We can drill down into the test stage and see the exact failures. For example:

```
Failed Calculator.Tests.CalculatorTests.AddTests(num1: -5, num2: 5, expected: 0) [< 1ms]

  Error Message:
   Assert.Equal() Failure

Expected: 0
Actual:   -25
```

As the repo owner, this lets us know that we shouldn't merge this change. As the developer, it shows us what we did wrong. We can then fix it, add a new commit to the merge request, and assuming the pipeline then passes, the changes can be merged.