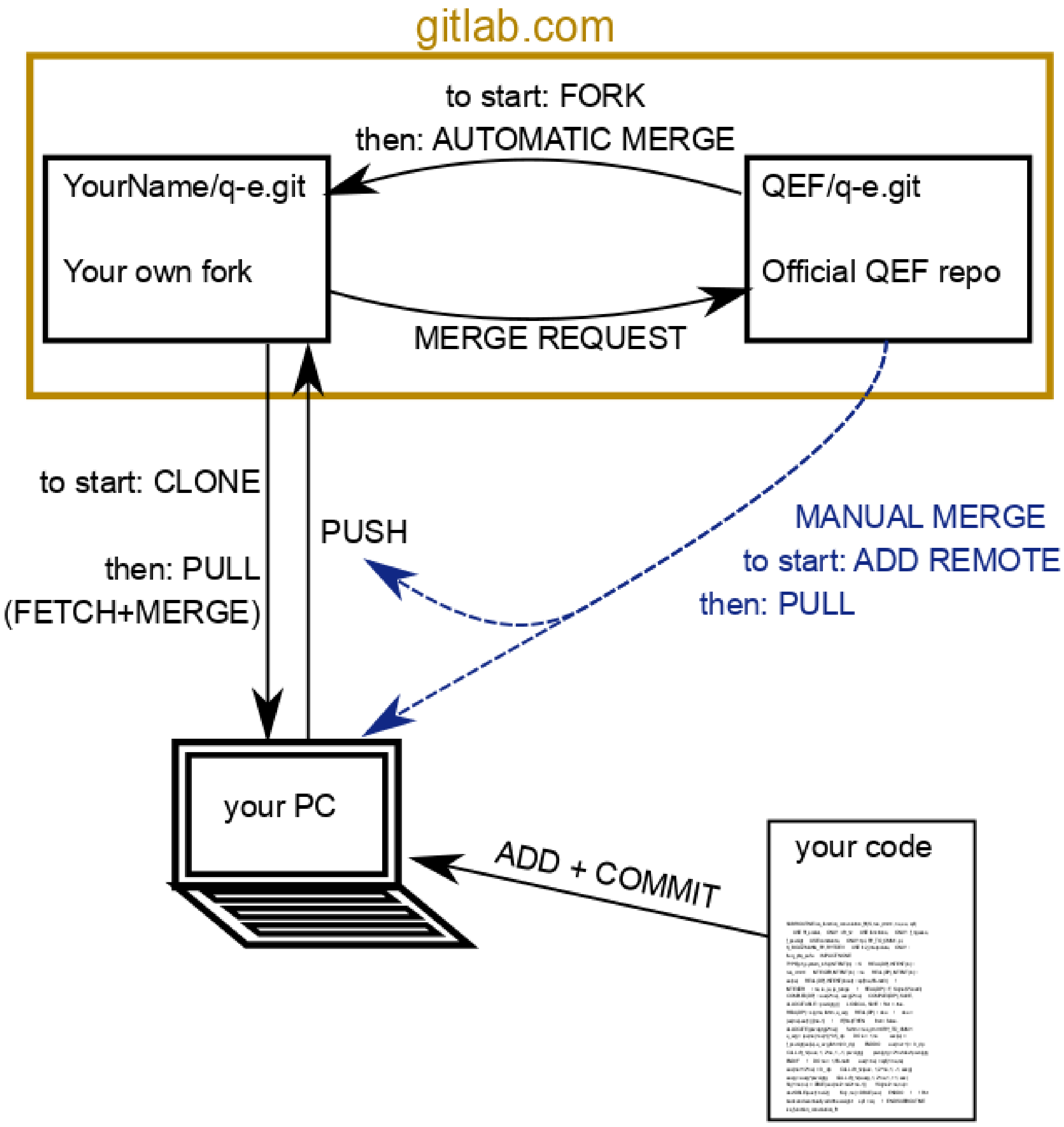
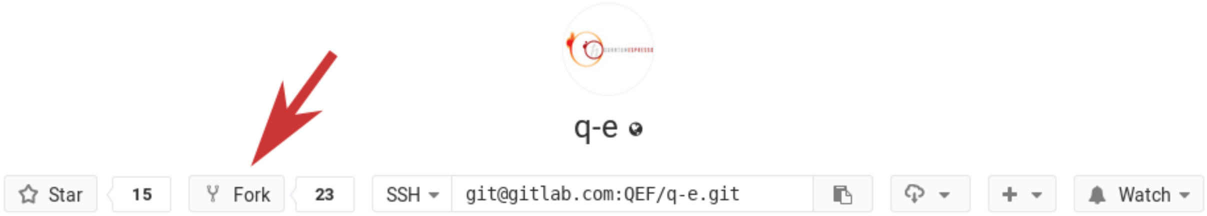


Getting started:

1. Create an account on gitlab.com; you can also use a github account or a google account
2. Create your own FORK of the Quantum-Espresso Foundation (QEF) repository, to do this go to http://gitlab.com/QEF/q-e and click the “Fork” button, then confirm. This will take a few minutes



3. Set up your fork: Go to http://gitlab.com/YourName/q-e and click on Settings. Some useful settings are:
   a) Set the default branch to “develop” (in the General section)

b) **Optional**: Set up the repository to AUTOMATICALLY MERGE, i.e. to stay up-to-date, with the official QEF repository. This can be done in the Repository settings, select "Pull" and as the URL of the repository put http://gitlab.com/QEF/q-e.

4. CLONE the fork to your PC, with the command:
   `git clone git@gitlab.com:YourName/q-e.git`
   This will create the directory "q-e" with a copy of the git repository in it.

ALL COMMANDS FROM HERE ON MUST BE ISSUED INSIDE THE q-e DIRECTORY!

5. ADD the official QEF repository as an additional REMOTE with the command
   `git remote add qef git@gitlab.com:QEF/q-e.git`
   You can check with "git remote" that now two remote sources will appear. The one called "origin" which is your fork, the other one called "qef"

Everyday work

Whenever you want to save your work status, you can commit it to your local repository, this can be done as often as needed: we recommend you do it at least once per day, or anytime you have done some significant modification, or just before starting one. In order to save your work you will have to tell git which files you want to save with the "ADD" command, then finalize the changes with the "COMMIT" command.

```
git add file1.f90 file2.f90

git commit
```

Alternatively you can use the command "git commit -a" which will commit ALL the changes you have done in the entire source tree (i.e. even in directories which are up from your current location!). Git will ask you to write a short message explaining the commit, then save and quit the editor (per default vim: you quit+save with "[ESC]:x[ENTER]") to finalize the commit.

As soon as you have done a commit, or whenever you have internet connection it is good practice to mirror the commit in your remote fork on gitlab. You can do this with the PUSH command:

```
git push
```

Staying up to date

You should regularly, i.e. before starting each work session, keep your repository up to date. You have set up your fork to automatically stay in sync with the QEF repository, then it may be enough to ask git to FETCH the changes from the remote fork and MERGE them in your current source three. The operations fetch+merge can also be combined as PULL, which is exactly the same thing:

```
git fetch

git merge
```

– OR –

```
git pull
```

If any actual merging has been done in the process, git will automatically create a "merge commit", it is better to finalize it as-is, without specifying any message, in this way the gitlab system can automatically identify it at threat is in a special way. Do not forget to push it.

If there are any conflicts, i.e. if some change cannot be merged automatically, git will ask you to do it manually: You will have to edit the problematic files and looks for the conflicts, they are marked in a very obvious way. Then commit the changes to finalize the merge, you won't be able to commit anything else until the merge is over.

If you have not set up the automatic merge, or if for some reason it failed, you will have to do a MANUAL MERGE of your fork with the QEF repository. Instruct git to go get the code from the QEF repository and merge it with your source tree, i.e. to pull from the qef remote:

```
      git pull qef develop
```

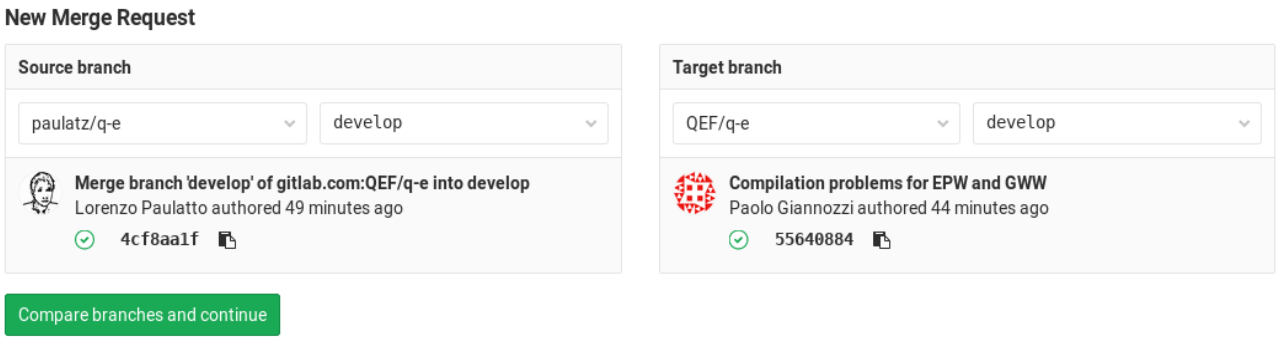
Again, leave the merge message as-is, resolve any conflict and commit the changes, then push to save everything in your online fork.

As long as you do not change the commit messages of merge operations, gitlab should be able to avoid conflict between automatic and manual merges, preventing the same modifications to be merged twice.

Publishing your work

Whenever you want to contribute your code back to the QEF repository, you have to open a MERGE REQUEST. This is not a git command, but a feature of gitlab.

Got to your repository page, and from the menu select merge request, then click to create a new request.



New Merge Request

Source branch

paulatz/q-e | develop

Merge branch 'develop' of gitlab.com:QEF/q-e into develop
Lorenzo Paulatto authored 49 minutes ago
4cf8aa1f

Target branch

QEF/q-e | develop

Compilation problems for EPW and GWW
Paolo Giannozzi authored 44 minutes ago
55640884

Select the branch of your project that you want to merge and the branch in the QEF repository. Usually they will both be "develop".

Go on, insert a message explaining the reason of your request, you may get comments and requests to improve or change your code. **Note that if you keep updating your repository the merge request will change!** This is useful to comply with the comments you may get in your request, but if you want to keep working on the code, while the merge request is open, it may be a good idea to create a specific branch, in your fork, to manage the merge request (check online to find more about managing branches).