Git Fundamentals Lab

BME554L - Fall 2025 - Palmeri

Dr. Mark Palmeri, M.D., Ph.D.

2025-08-25

Table of contents

Git Practice	1
Initial main Branch Development	1
Working on a Branch (Local)	2
Merge Request Time!	2
Create an Annotated Tag	3
Fix a Merge Conflict (Locally)	3
How to Ask for Help	4
What to Submit	4

Git Practice

This week we will go through a very simple git workflow to get you familiar with the process and to confirm that your toolchain works for future assignments.

Initial main Branch Development

- 1. Make sure that you have git installed, ssh keys configured, and your Duke GitLab account is setup, as detailed here
- 2. Be sure to review the tutorials outlined on the git setup page, as needed, to complete the tasks below.
- 3. Create a **fork**-not a **clone**-of the git-fundamentals-lab in your userspace.
- 4. Add Dr. Palmeri (mlp6) as a Maintainer of your forked repository (Left Sidebar: Manage-> Members -> Invite members)
- 5. Clone your forked repository to your laptop using the URL available under the blue Code button: Clone with SSH.

⚠ Warning

You want to clone your forked repository, not the parent repository you forked from!

Working on a Branch (Local)

- 1. Switch to the about_me branch.
- 2. Edit AboutMe.md to replace all of the FILL ME IN placeholders with your own informa-
- 3. Add and commit these text changes with a meaningful commit message.
- 4. Add and commit a new image file named fun picture.png in a directory called images/ to complete that section of the AboutMe.md file.
- 5. Push this updated branch to the remote of your GitLab repository.
- 6. Confirm that you can see your latest commit on this branch on the GitLab website.

Merge Request Time!

- 1. Create a new Merge Request on the GitLab to merge the about me branch into the main branch.
 - Assign the Merge Request to yourself.
 - You do not need a Reviewer.
 - You do **not** want to *squash* or *rebase* your branch commits
 - You do not want to delete the source branch.
- 2. Check that the check_image and check_about_me pipline jobs have passed.
 - Build -> Pipelines
 - Click on the latest pipeline
 - Click on the check_image or check_about_me job to see the output of the test if it Failed.
- 3. If the pipeline jobs for this branch have failed, you will need to fix the issue(s) and push the changes to your about_me branch.
 - If you are not sure what the issue is, click on the failed job to see the output of the test.
 - If you are not sure how to fix it, ask for help!
- 4. Once your CI jobs pass, approve your Merge Request on the GitLab website and Merge your about_me branch into main.
- 5. Checkout the main branch on your laptop and pull the latest changes from the GitLab server.

6. Confirm that you can see the updated AboutMe.md file in your main branch, along with its associated commit in your git commit history.

Create an Annotated Tag

- 1. On main, create an annotated tag for this commit associated with your merged about_me branch called v1.0.0, with the message "include the about me info".
- 2. Push this annotated tag to the GitLab server (origin).
- 3. Confirm that you can see your latest commit and tag on the GitLab website.
- 4. You should see that the check_v1_0_0_tag pipeline job has passed if this was successful.

Fix a Merge Conflict (Locally)

- 1. Inspect the contents of AboutDrPalmeri.md in main.
- 2. Switch to the about_dr_p branch, and look at the content (slightly different) of AboutDrPalmeri.md.
- 3. Switch back to the main branch, and merge the about_dr_p branch into main.
 - You will be told that there is a Merge Conflict:

```
$ git merge about_dr_p
CONFLICT (content): Merge conflict in AboutDrPalmeri.md
Automatic merge failed; fix conflicts and then commit the result.
```

• git status will show you the files that are in conflict, and what your next likely steps will be to resolve this conflict:

4. Staying on the main branch, open AboutDrPalmeri.md in your text editor, and inspect the <<<<<, ======, and >>>>> markers that indicate the conflicting lines.

- 5. Keep the version of the conflicting line that is in main, and delete the line that was incoming from about_dr_p.
- 6. Be sure to delete the <<<<<, ======, and >>>>>> markers (lines).
- 7. Add and commit this merge conflict resolution.
- 8. Create an annotated tag called v1.0.1 with the message "fix Dr. P's age".
- 9. Push your latest commits and tags to main on the GitLab server (origin).
- 10. All of your CI jobs should pass if everything was successful.

How to Ask for Help

- 1. If you have a general / non-coding question, you should ask your TAs / Dr. Palmeri on Ed to allow any of them to respond in a timely manner.
- 2. Push you code to your GitLab repository, ideally with your active development on a non-main branch.
- 3. Create an Issue in your repository.
 - Add as much detail as possible as to your problem, and add links to specific lines / section of code when possible.
 - Assign the label "Bug" or "Question", as appropriate.
 - Be sure to specify what branch you are working on.
 - Assign the Issue to one of the TAs.
 - If your TA cannot solve your Issue, they can escalate the Issue to Dr. Palmeri.
- 4. You will get a response to your Issue, and maybe a new branch of code will be pushed to help you with some example syntax that you can use git diff to visualize.

What to Submit

Your fork of this GitLab repository will be running this CI script: .gitlab-ci.yml

- 1. Go to the Build -> Pipelines page of your GitLab repository.
- 2. Take a screenshot of the latest pipeline that shows all of the tests passing, along with your repository information, as shown below (except all of those Failed tests should report Passed).

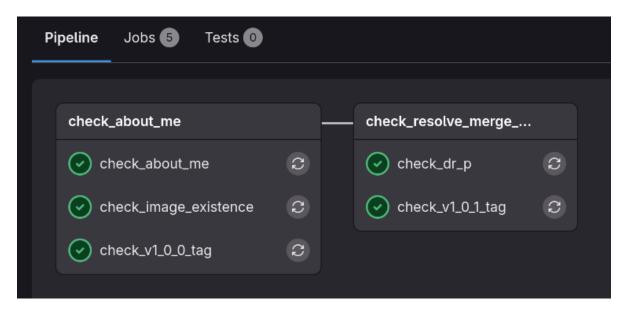


Figure 1: ci_pipeline

1. Submit this screenshot to Gradescope assignment, along with completing the prompted questions.