**Git Branch + Merge + CI/CD**

**Purpose:** We are going to demonstrate the “Feature Branch” workflow by working an example. After practicing merging and branching, we will work through the CI/CD step.

Start with the following HTML [here](https://ceclnx01.cec.miamioh.edu/~dunnnm2/cse451-dunnnm2-web/BranchandMerge/451-a.html).

1) Form a group of **3-5 people**. The more people you have, the more fun this lab becomes.

2) Assign a group leader through desired format (rock-paper-scissors, trial by combat, etc..)

3) The group leader should designate each person in the group a player number, 1- 5, including themselves. The [default HTML page](https://ceclnx01.cec.miamioh.edu/~dunnnm2/cse451-dunnnm2-web/BranchandMerge/451-a.html) is built for 5 people, but the divs containing the different colors can be removed as needed.

4) The group leader should assign one person in the group to create a GitLab repo, and add each person in the group as a Contributor to that repo.

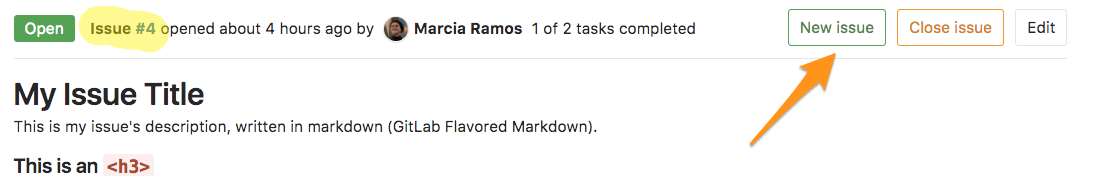
5) Have the same person, through the standard git-add, git-commit, git-push workflow upload the following files to the repo.

6) Recall, the feature/topic workflow states that the central repo, the master branch in this case that was just created, is representative of our full version history

7) With maintainers permissions intact, all members of the team should clone the repo that has been set up by one of your team members in step 4 and 5.

8) All team members should go on to the shared team repo on the GitLab site, and each create a GitLab issue that is titled the name of the desired feature they intend to create. Documentation can be found [here](https://docs.gitlab.com/ee/user/project/issues/managing_issues.html).

9) Everyone should locate the id number for the issue they have created



10) Once everyone has their own version of the repo, everyone should branch off master and name their branch after the small feature they intend to add combined with the issue number. If you are issue #4, and you want to name your branch printLetterA, you should name your branch 4-printLetterA .Your feature can be as simple as changing the text that is already existent, or writing a short php script that prints information.

How to branch:

$ git checkout -b x-branch-name

WHERE x is the issue number without the # symbol,

11) Everyone should implement their desired feature, git add and then git commit it.

<?php print “Hello World” ?>

12) Attempt to git push your merge to GitLab. The remote server will generate a link to fill out a merge request. Create a merge request on GitLab to merge your branch into master with your finished feature. (Sometimes it pays to be first).

remote:

remote: To create a merge request for 6-test, visit:

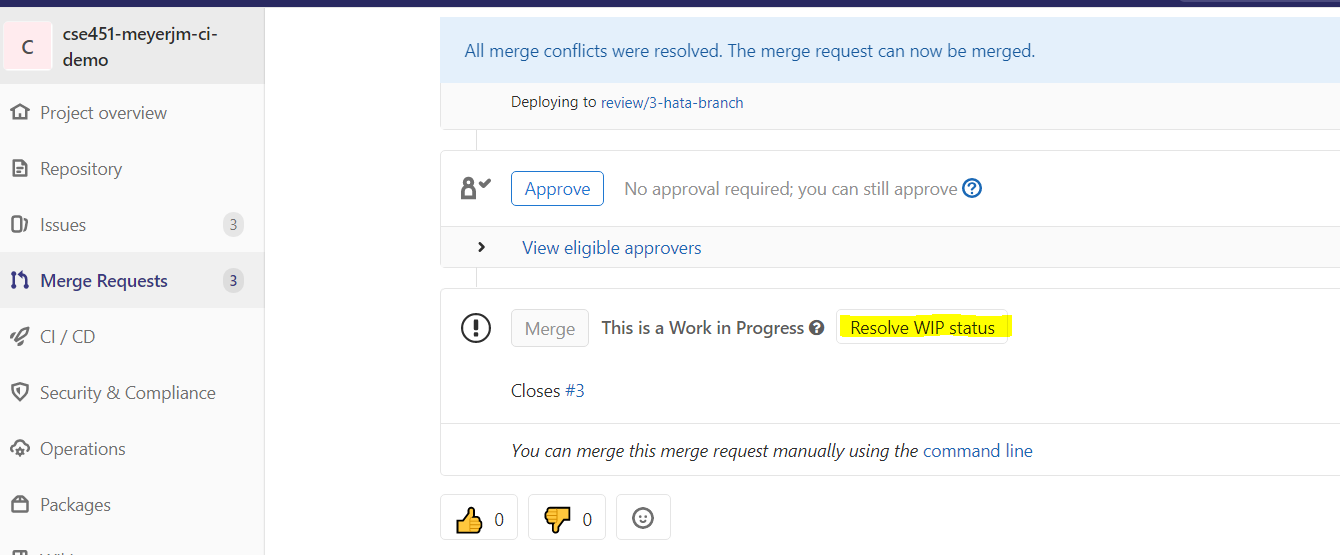
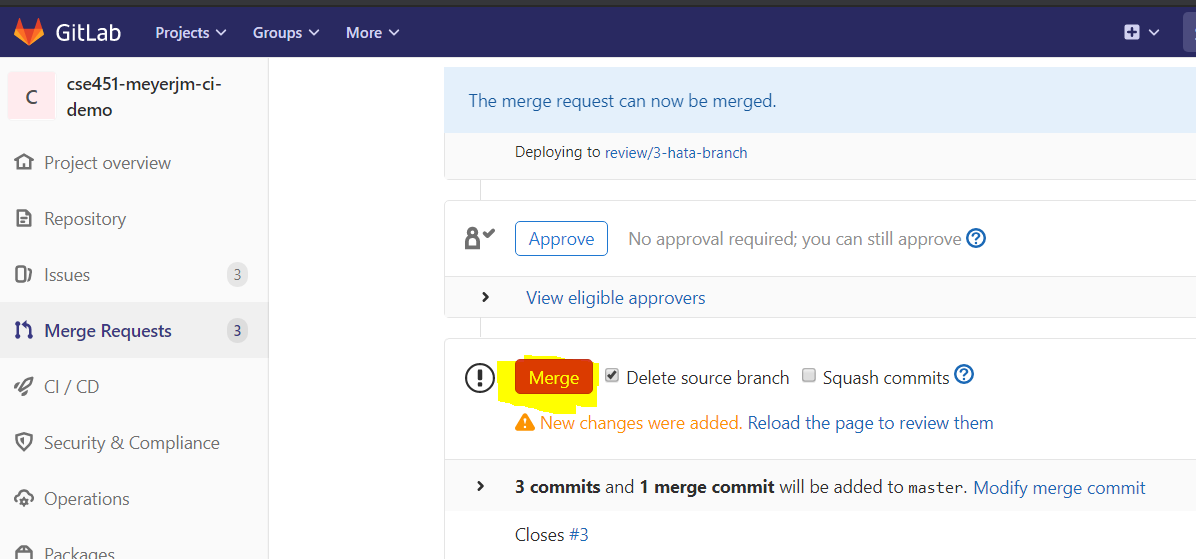
remote: https://gitlab.csi.miamioh.edu/meyerjm/cse451-meyerjm-ci-demo/merge\_requests/new?merge\_request%5Bsource\_branch%5D=6-test

remote:

$ git push --set-upstream origin branch-name

Click on <https://gitlab.csi.miamioh.edu/UNIQUEID/cse451-UNIQUEID-web/merge_requests/new>

13) Complete the Merge Request. In the case of conflict, git pull, resolve your conflict, and then attempt to re merge. Alternatively, make use of the GitLab UI to do all these steps for you.

---------------------------------------------------------------------------------------------------------------------------------------------

---------------------------------------------------------------------------------------------------------------------------------------------





How to fix merge conflict from terminal:

1. git status to determine list unmerged files
2. Files will be marked with “<<<<<”,”=====”,”>>>>>” to indicate differences
3. Make the intended change to one version and remove merge annotations
4. Double check for remaining annotations
5. Finish the merge using git commit

14) Continue Step 11 until all members successfully merge.

15) View the completed list of merges on the GitLab repo to validate success

Post a link to the repo, the website on Canvas

16) git clone from the repo located [here](https://gitlab.csi.miamioh.edu/meyerjm/cse451-meyerjm-ci-demo)

17) Create a directory using mkdir in the root folder of the repo

18) Copy your finished index.html file from step 14 into the newly created directory

19) git-add, git-commit, git-push the directory to master. Observe the pipeline at work [here.](https://gitlab.csi.miamioh.edu/meyerjm/cse451-meyerjm-ci-demo/pipelines)

20) Have a good Spring Break!

Note, Our particular YAML CI script runs anytime anyone pushes to master, and also when branches are merged into master. The linting process and file copy-over enables the merge curators have a directory that copies over master’s files into a new directory.

[**Here**](http://meyerjm.451.csi.miamioh.edu/cse451-meyerjm-ci-demo/branch/Feature383/)is the branched version of a previous version of the Master located [here](http://meyerjm.451.csi.miamioh.edu/cse451-meyerjm-ci-demo/).