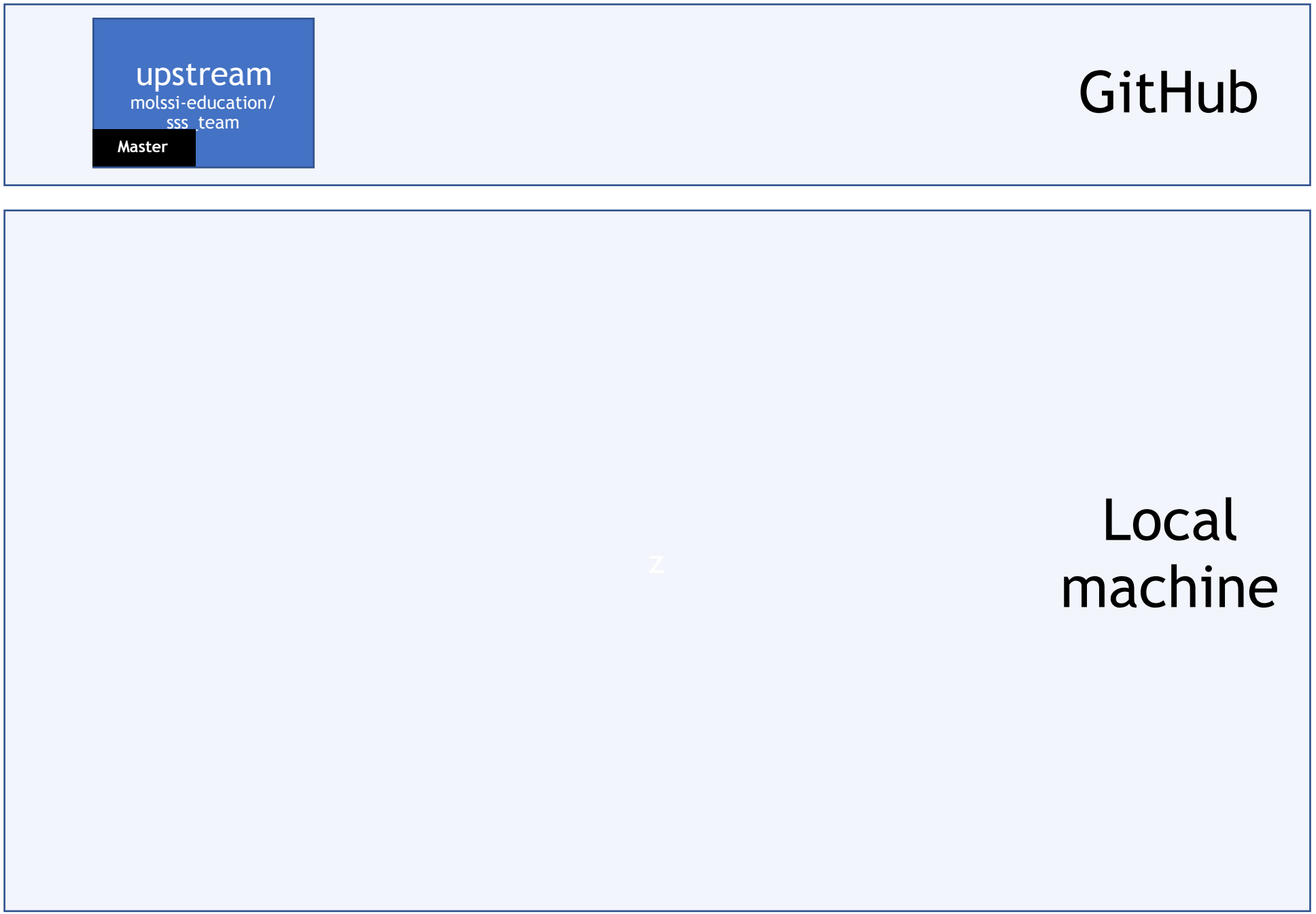
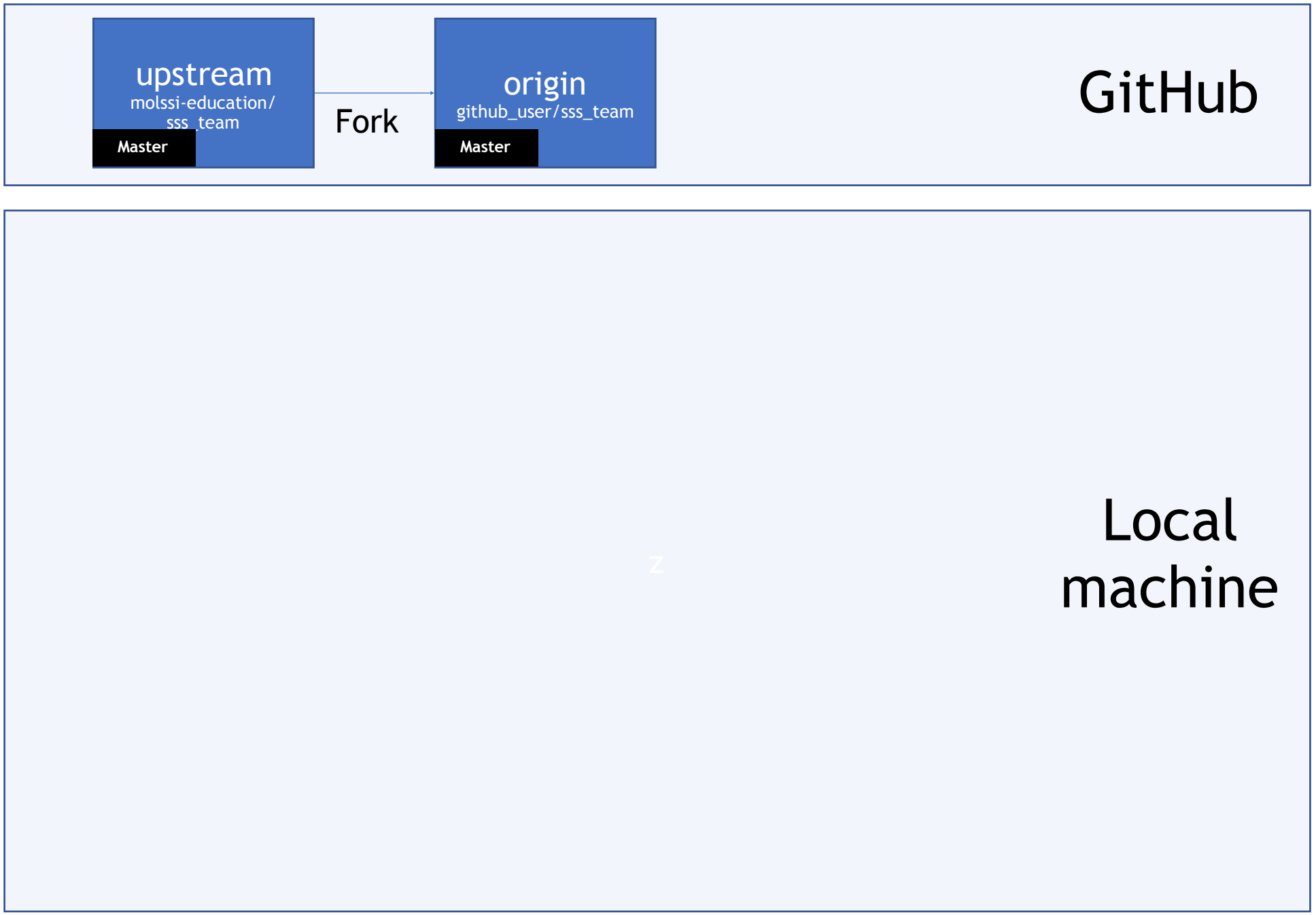


Notes

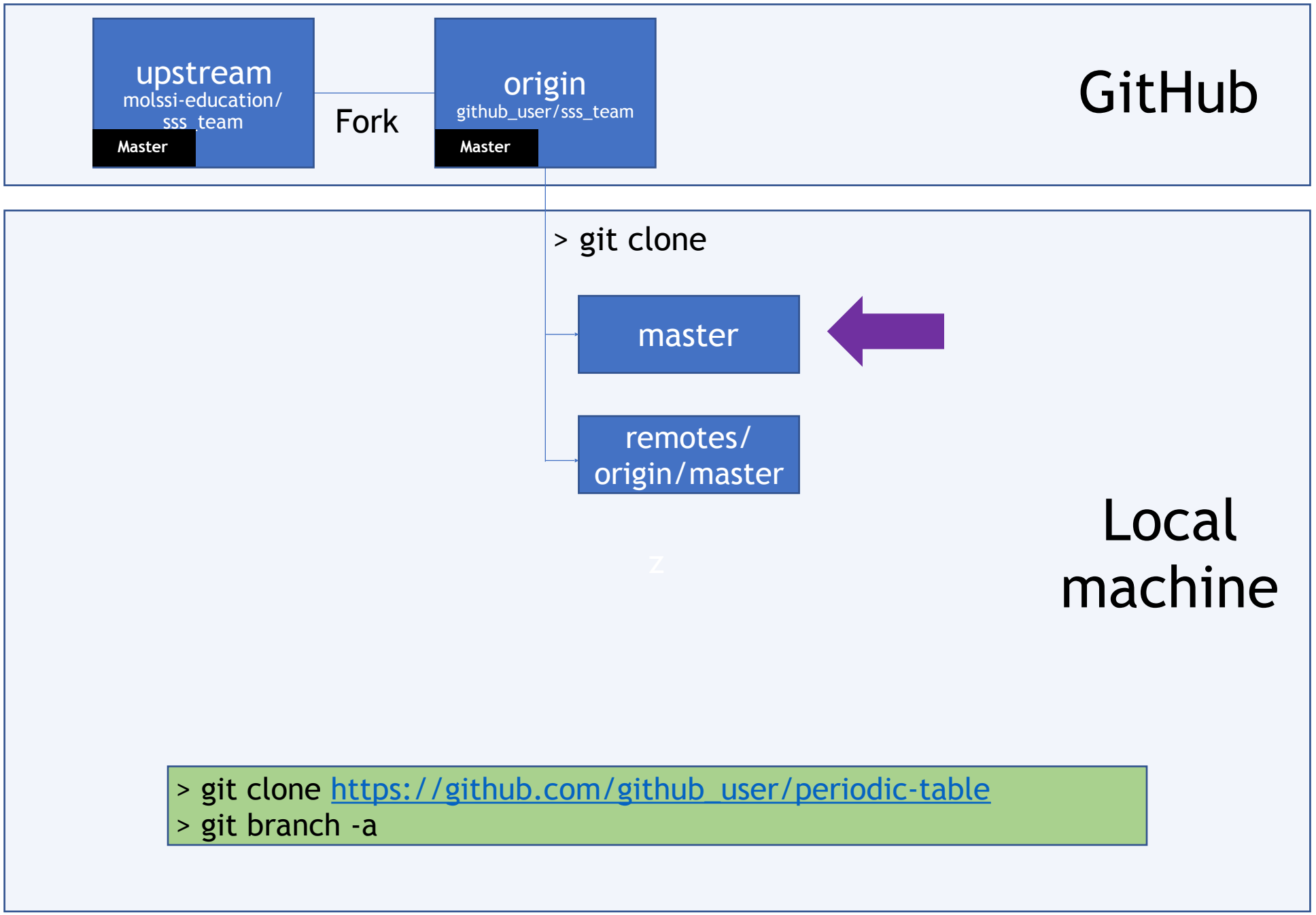
Go to the GitHub webpage of the central repository (i.e. upstream).

Upstream has only one branch, ***master***.



Notes

Click on the “Fork” button located at the top right of the sss_team GitHub webpage.



Notes

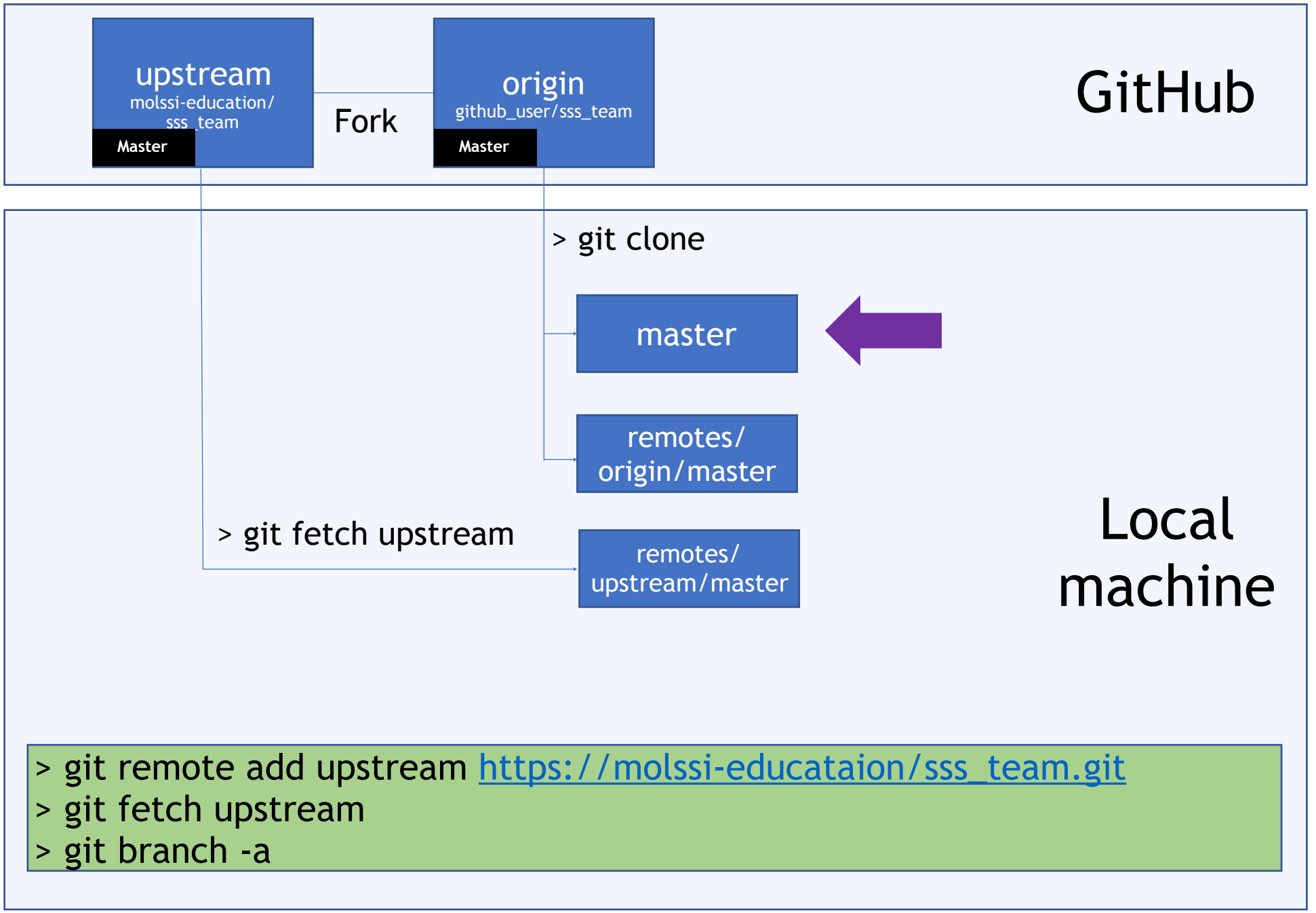
Clone your fork into your local machine. The URL should be consistent with your GitHub username.

You can see what branches you have locally by typing

> git branch -a

You will see a ***master*** branch, which is your local master branch. This is where you'll start your work from.

You will also see a branch named ***remotes/origin/master***. This is a remote tracking branch. It is used to track the changes that are in the GitHub repositories. It is a copy of the remote master branch.



Notes

Your local Git repository does not know about the central repository (i.e. upstream in molssi-education/sss_team). We need to add upstream as a new remote in our local Git.

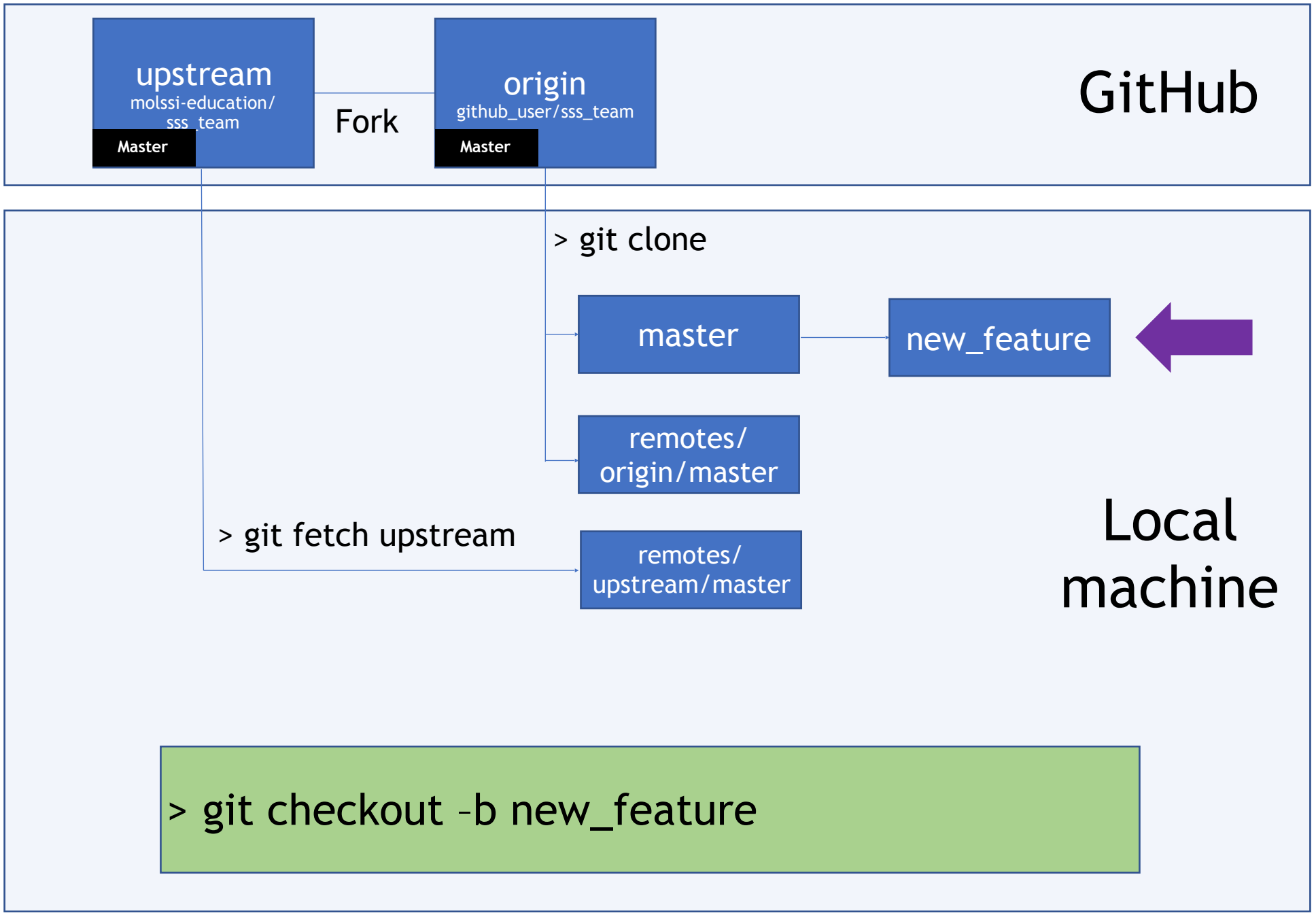
Then, get a copy of the upstream repository

> git fetch upstream

You can check that you have a new remote tracking branch with

> git branch -a

We'll use ***remotes/upstream/master*** to keep track of new changes that happen in upstream that we do not have yet in our local master.

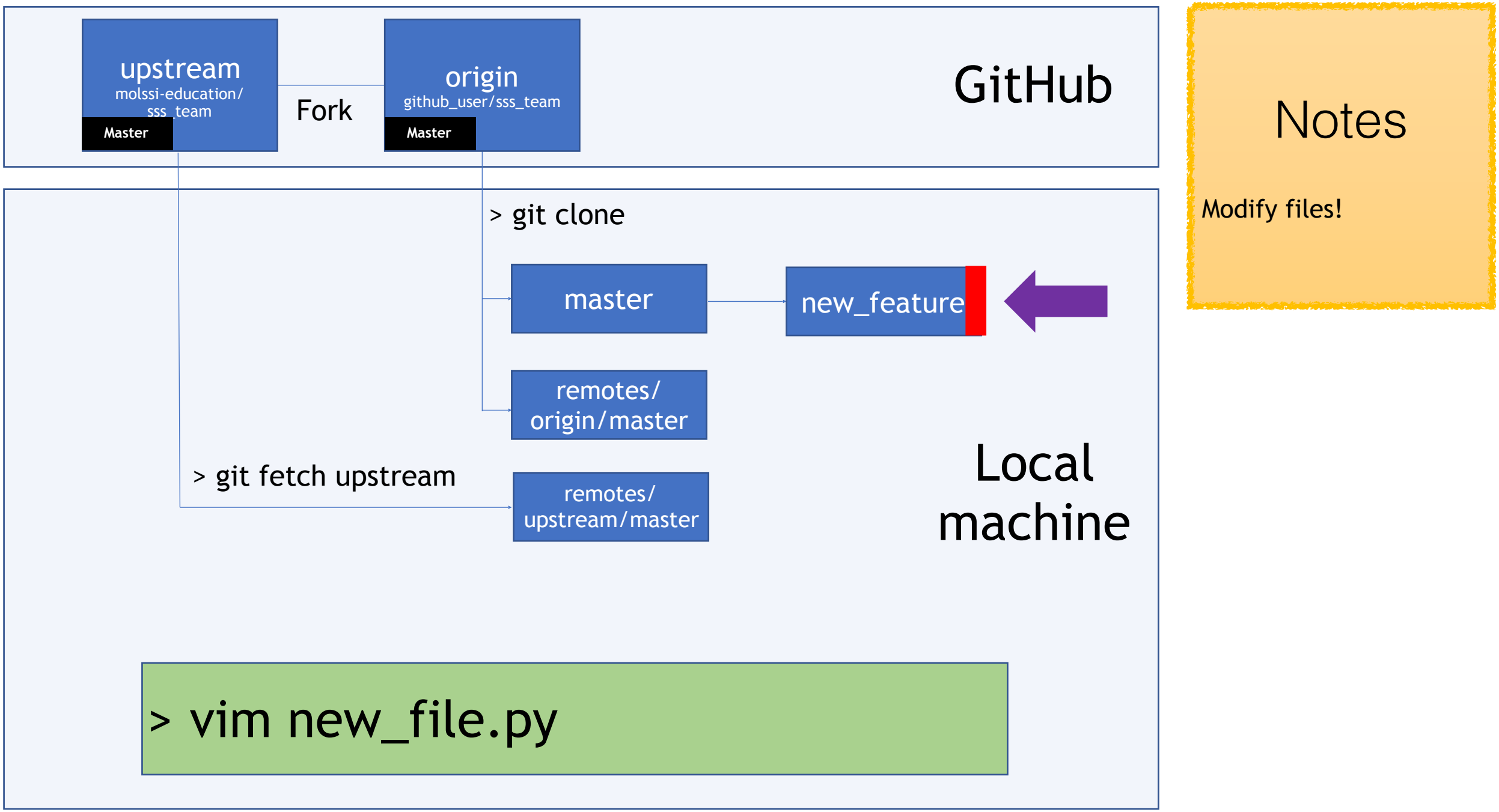


Notes

This step creates a new branch out of your local master branch.

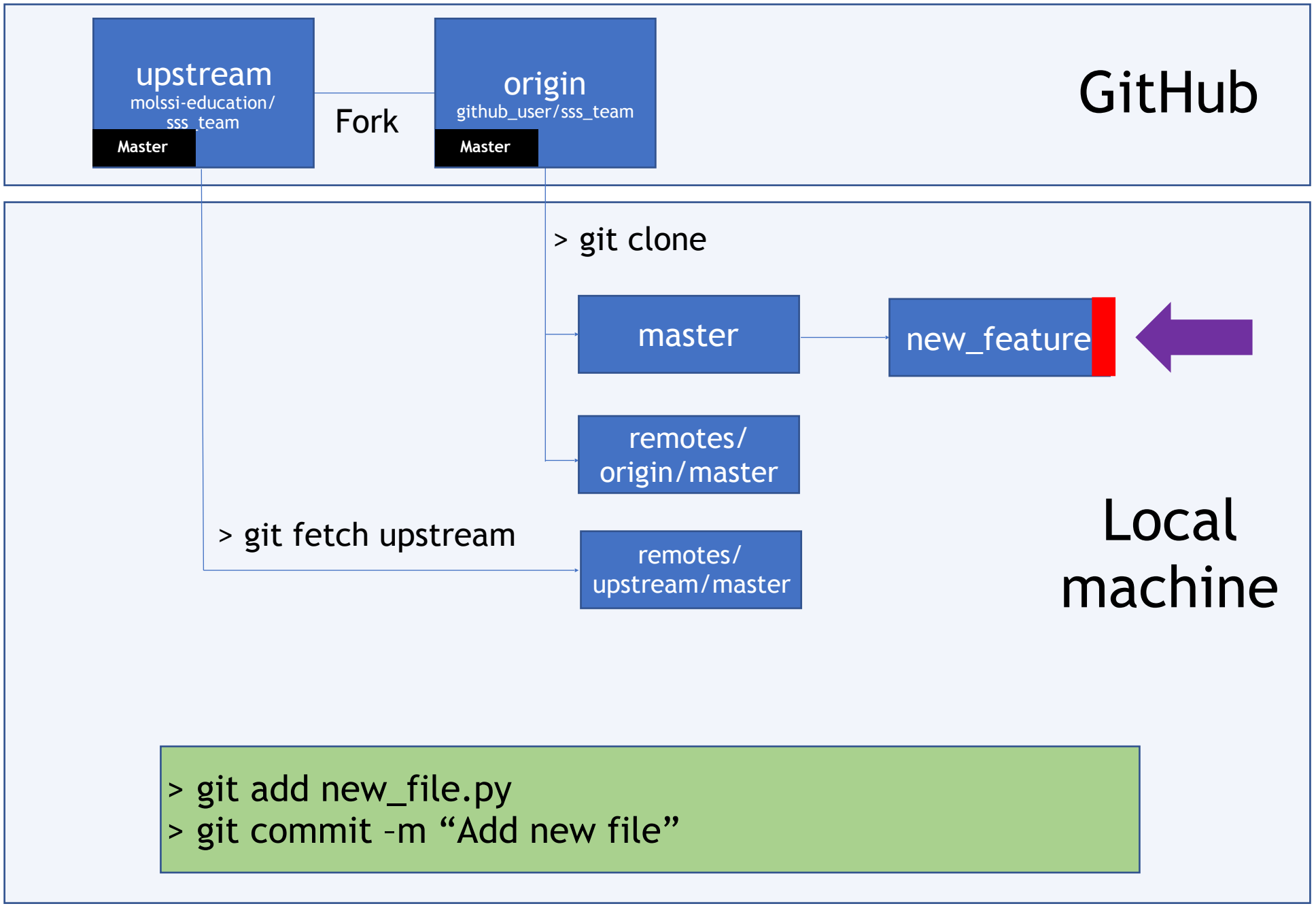
If this is the first time you forked a repository, then ***master*** and ***remotes/upstream/master*** will be synchronized.

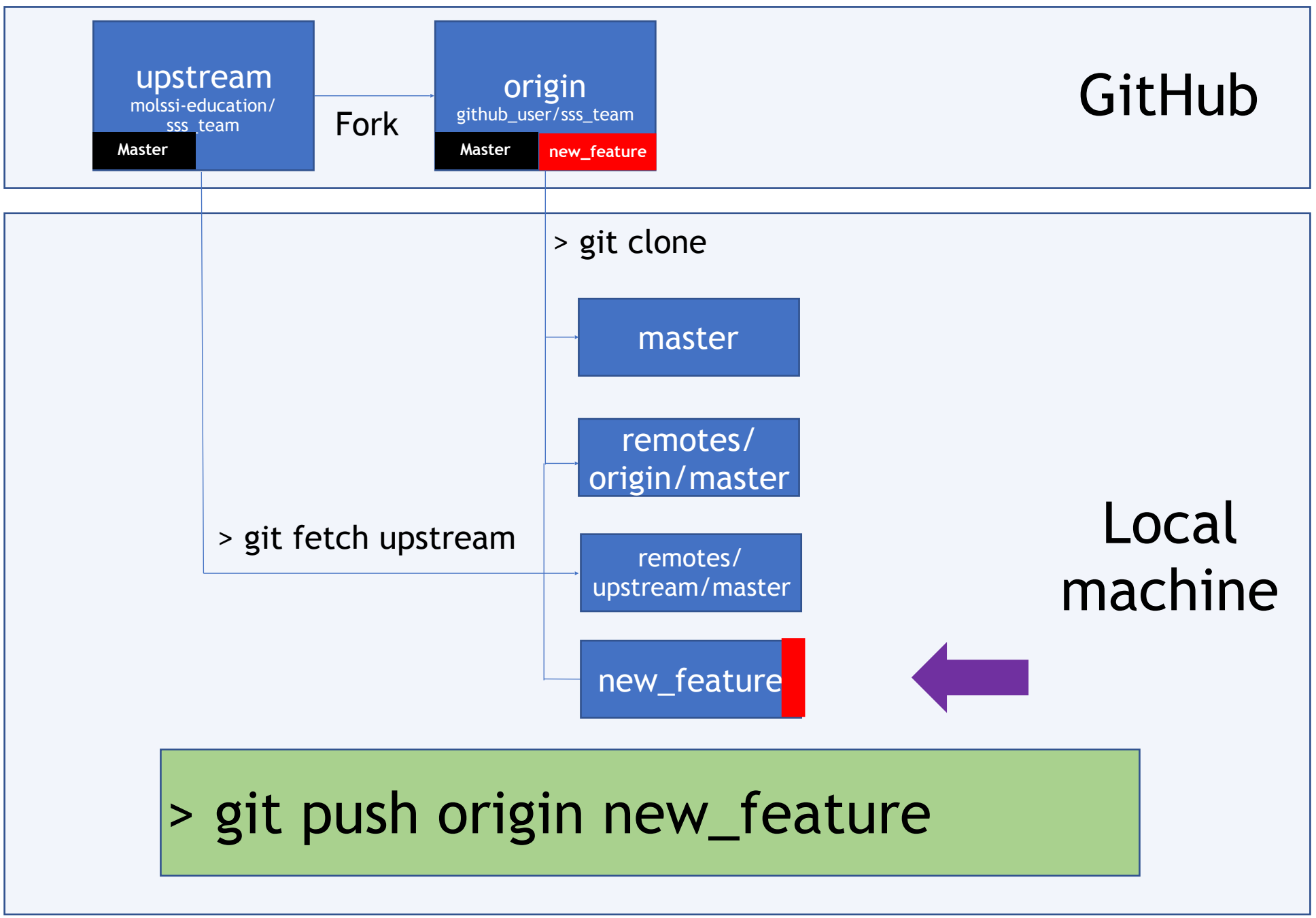
If you haven't updated your local ***master*** branch, make sure it is up to date with ***remotes/upstream/master***! (see next few slides to know how to do this)



Notes

Modify files!

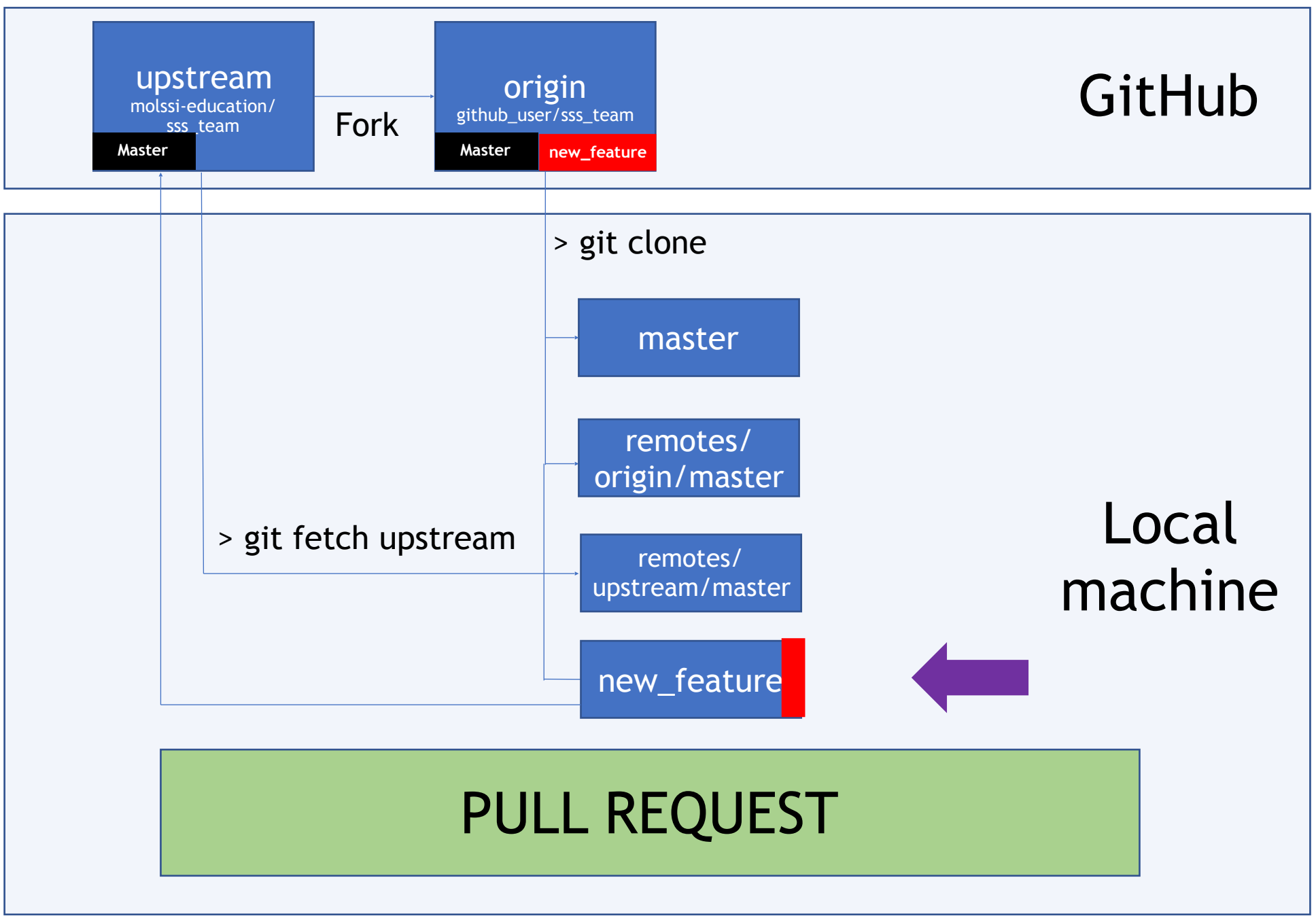

upstream
molssi-education/
sss_team
Master
Fork
origin
github_user/sss_team
Master
GitHub
> git clone
master
new_feature
remotes/
origin/master
> git fetch upstream
remotes/
upstream/master
Local
machine
> git add new_file.py
> git commit -m “Add new file”



Notes

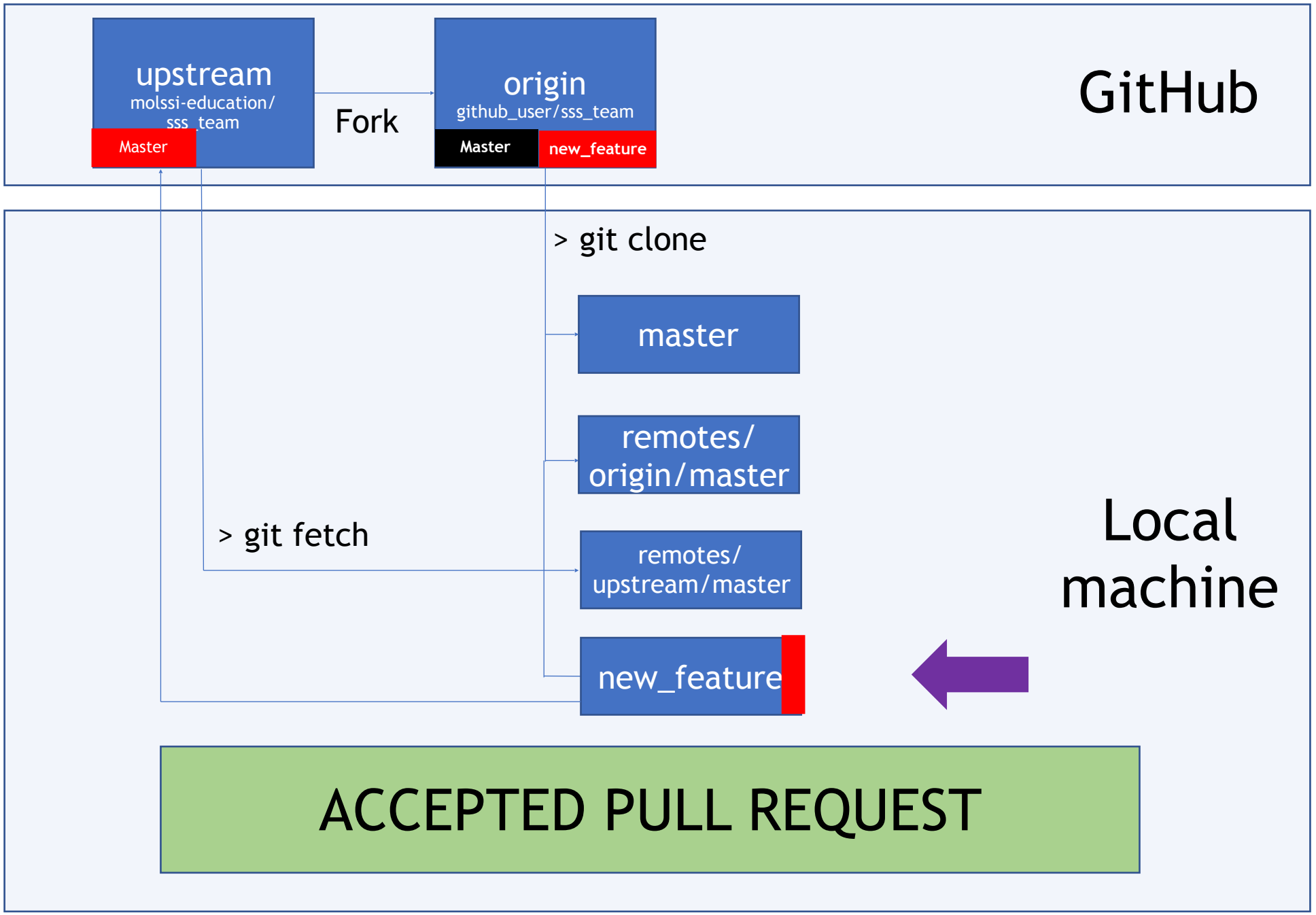
Push your changes to origin. Note that your ***master*** and ***remotes/origin/master*** do not contain your newest change.

If you go your fork in GitHub, you will find a new branch named ***new_feature***.



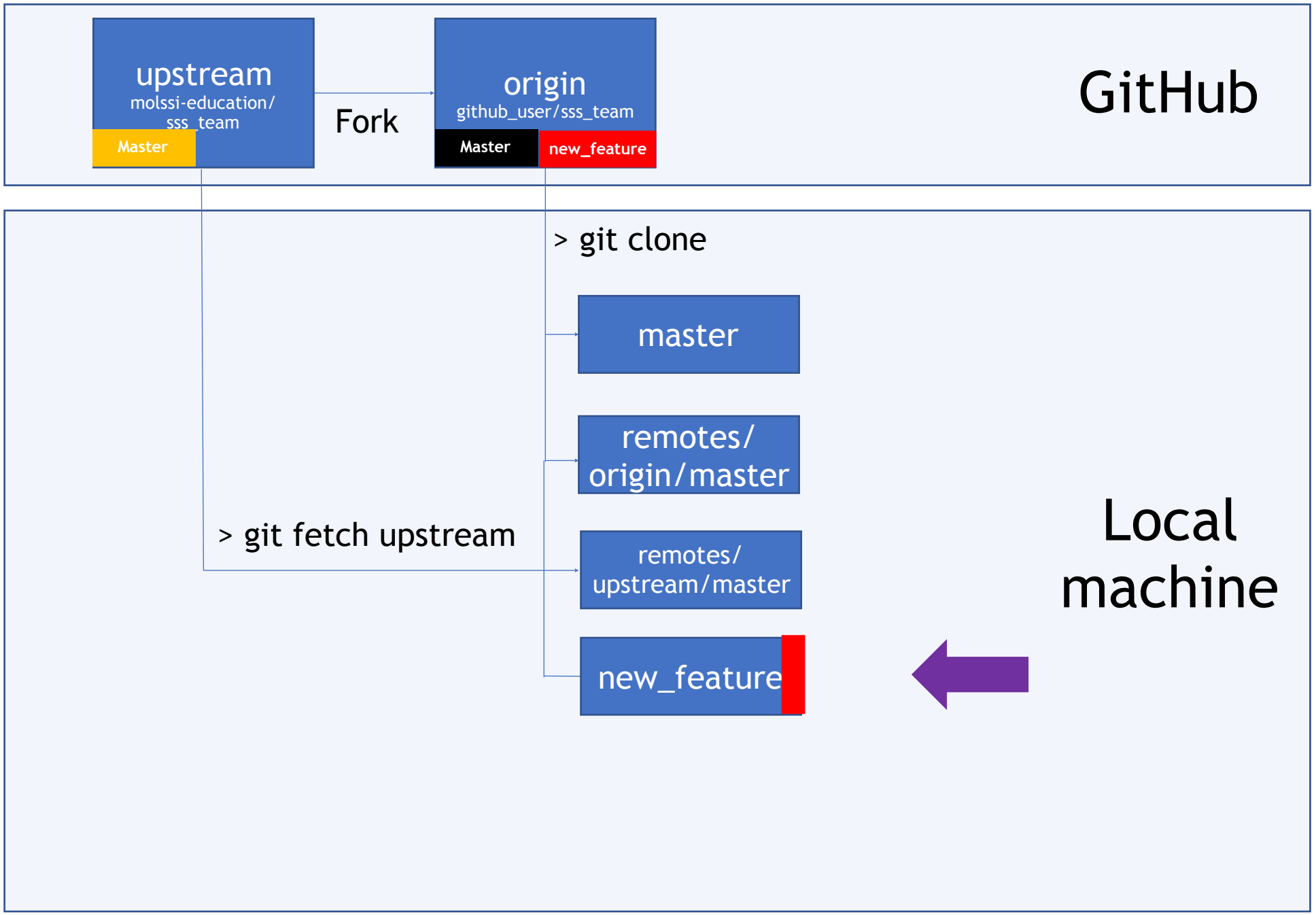
Notes

Issue a Pull Request from GitHub to let the core development team that you're ready to incorporate some changes to upstream.



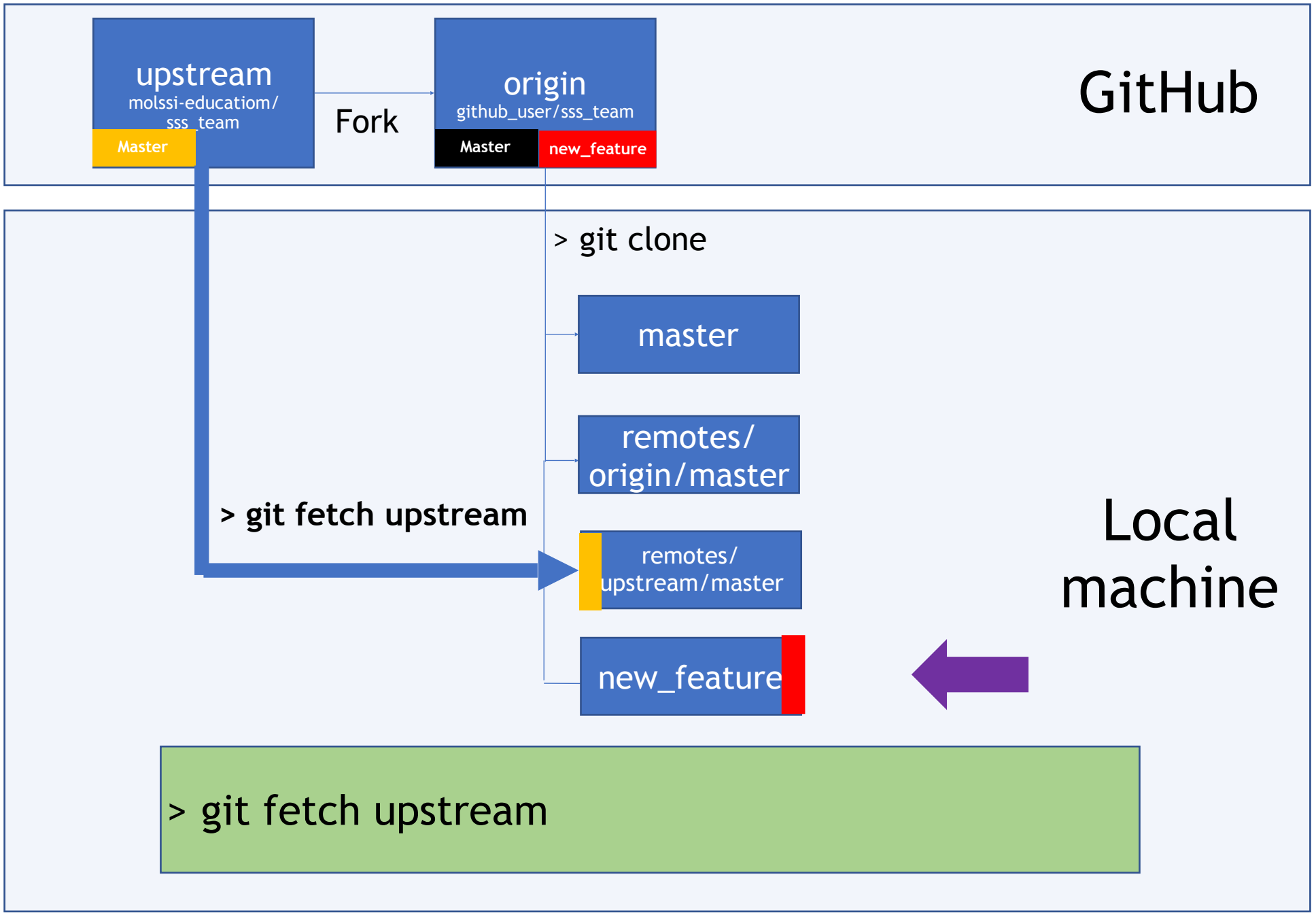
Notes

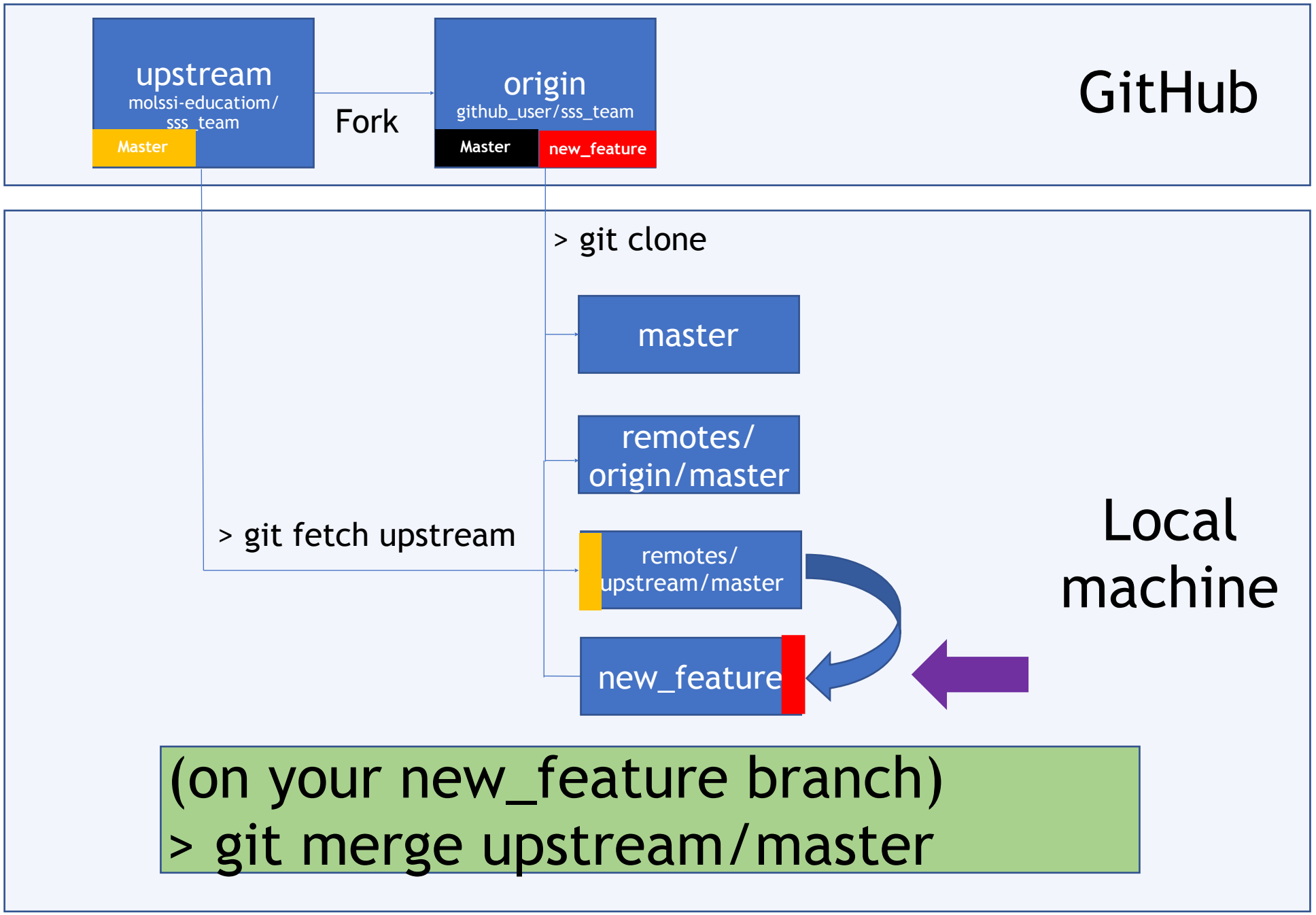
Pass tests and code reviews. Your code is accepted!



Notes

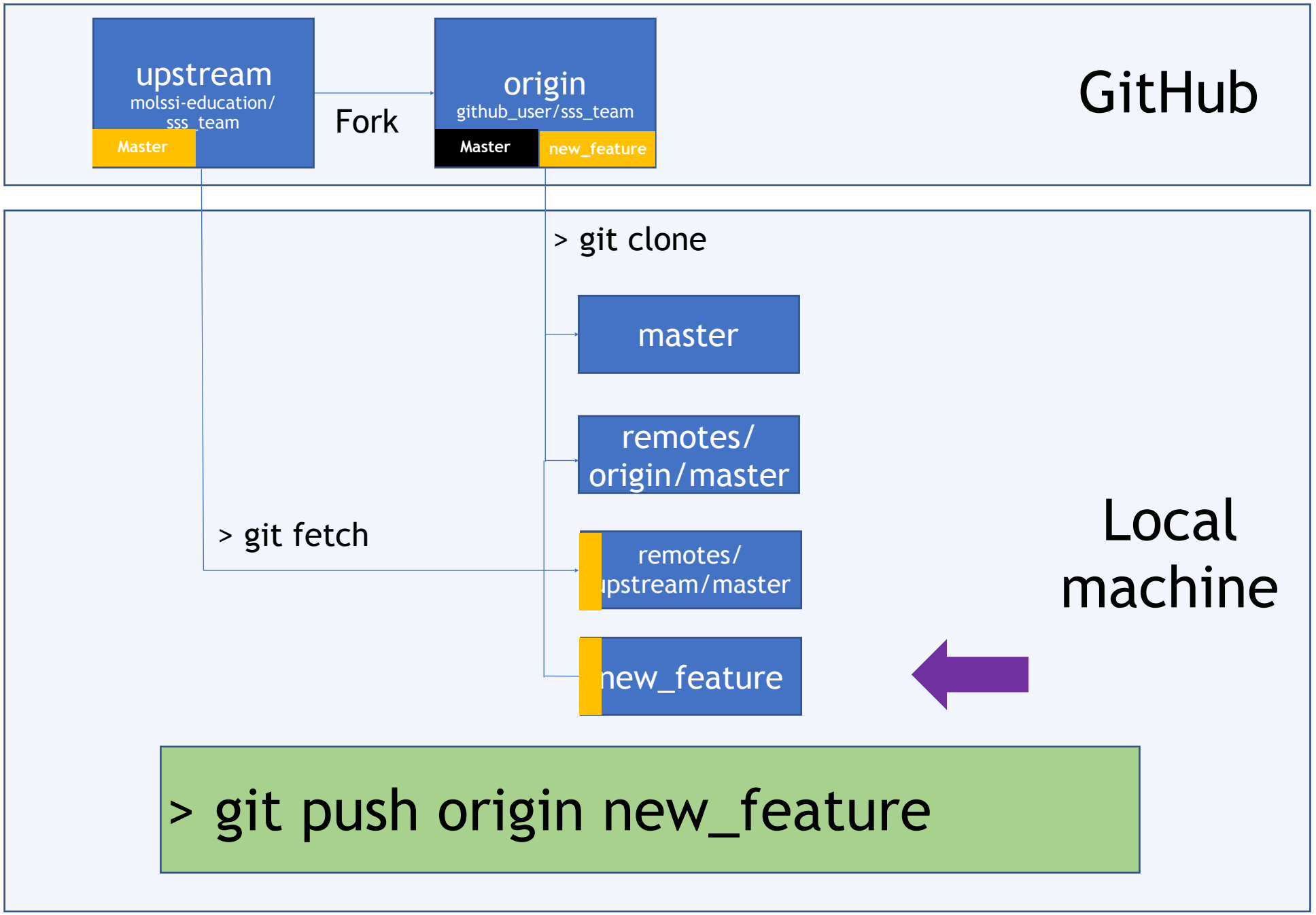
Imagine there is a new change in ***remotes/upstream/master***. How do you get it locally?


upstream
molssi-education/
sss_team
Master
Fork
origin
github_user/sss_team
Master
new_feature
GitHub
> git clone
master
remotes/
origin/master
> git fetch upstream
remotes/
upstream/master
Local
machine
new_feature
> git fetch upstream



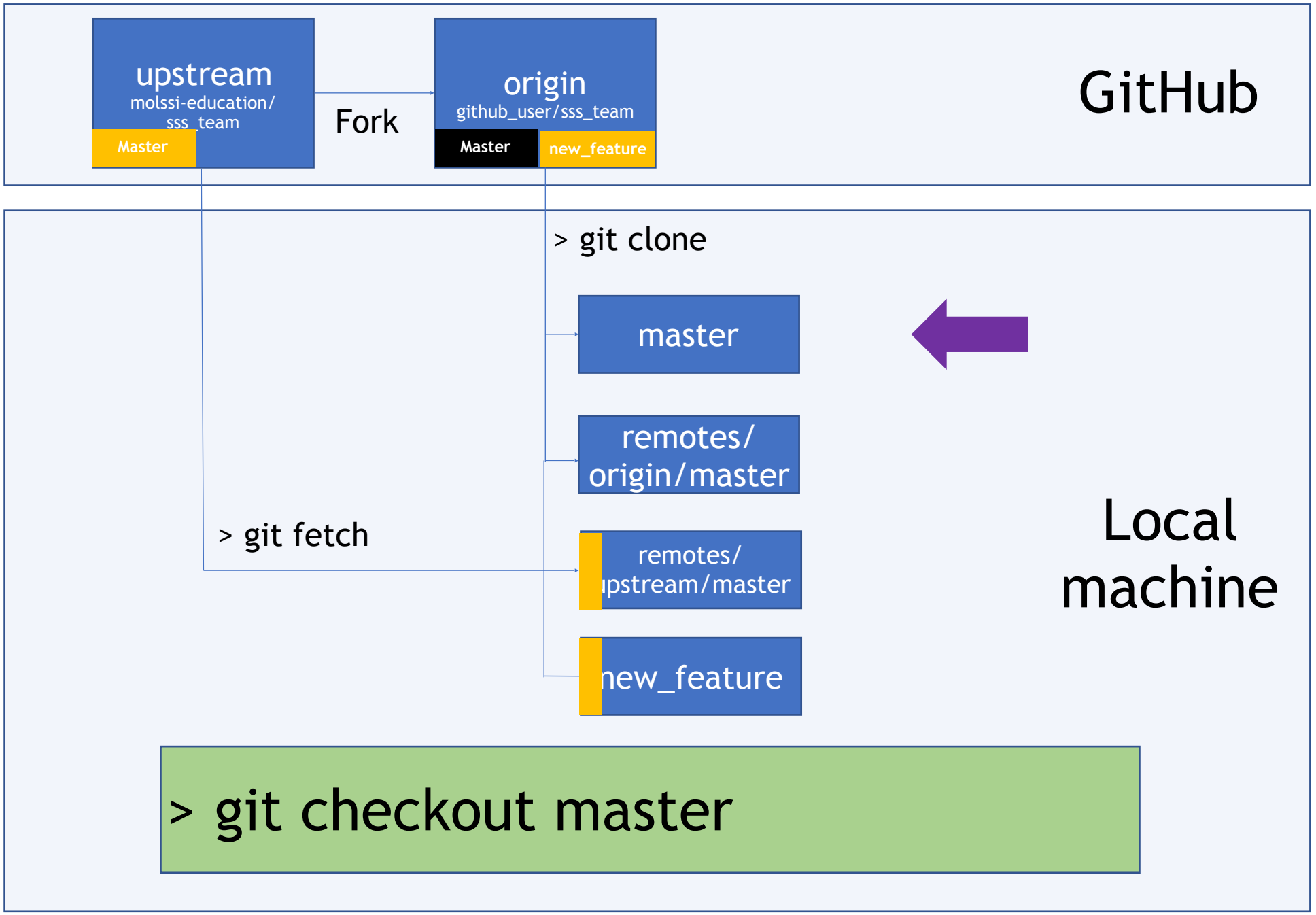
Notes

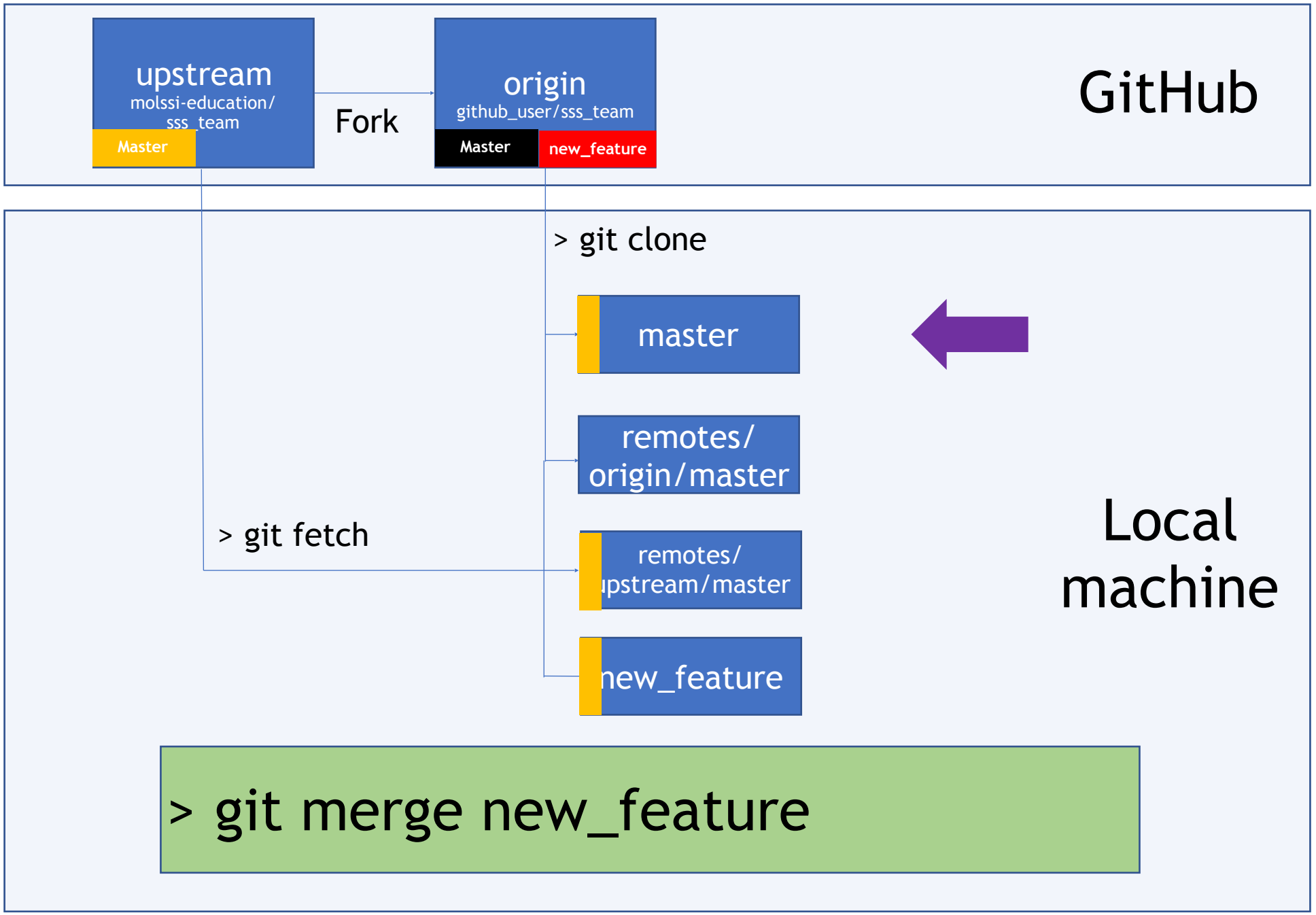
Updates your local new_feature branch to keep up with upstream/master

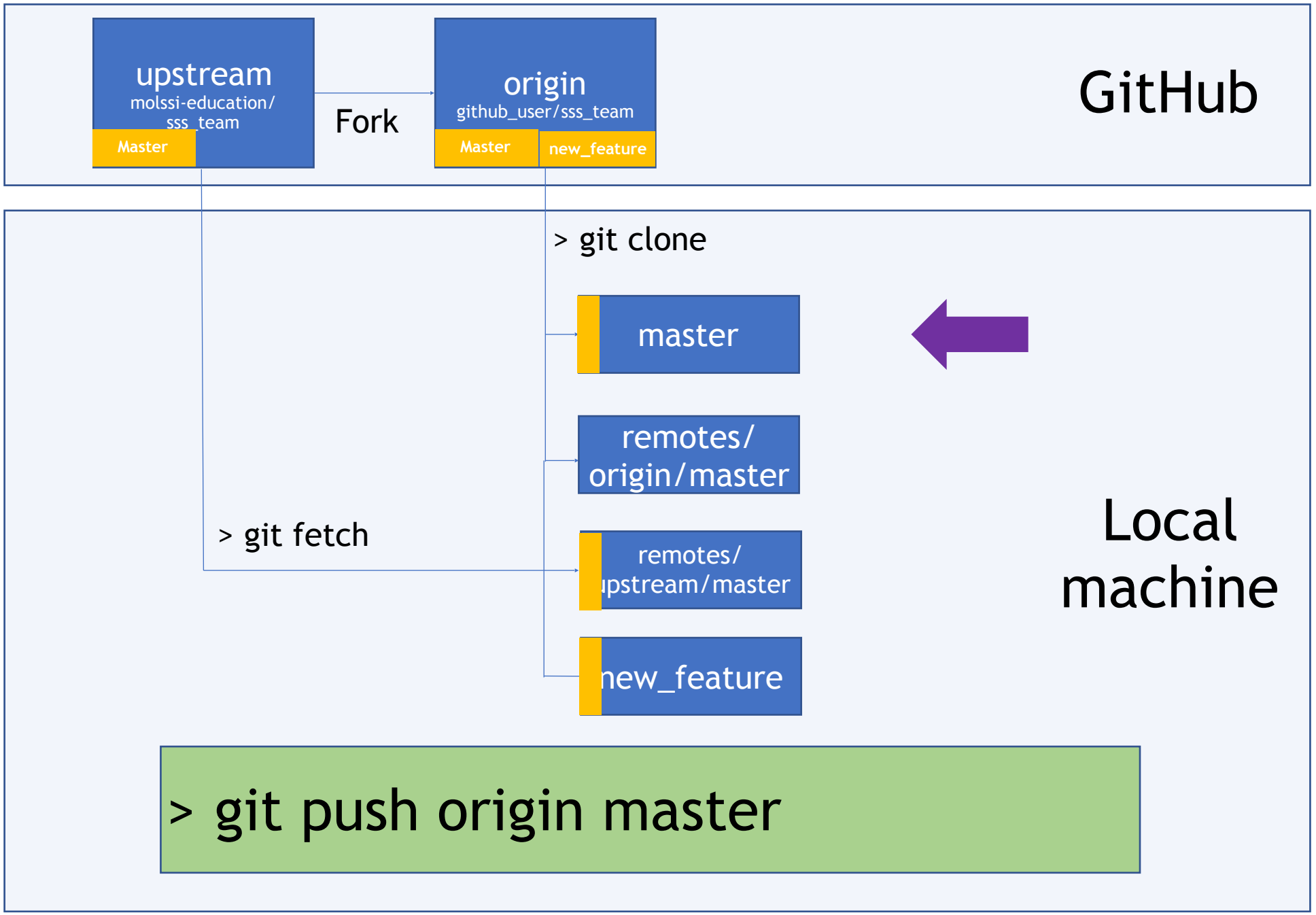


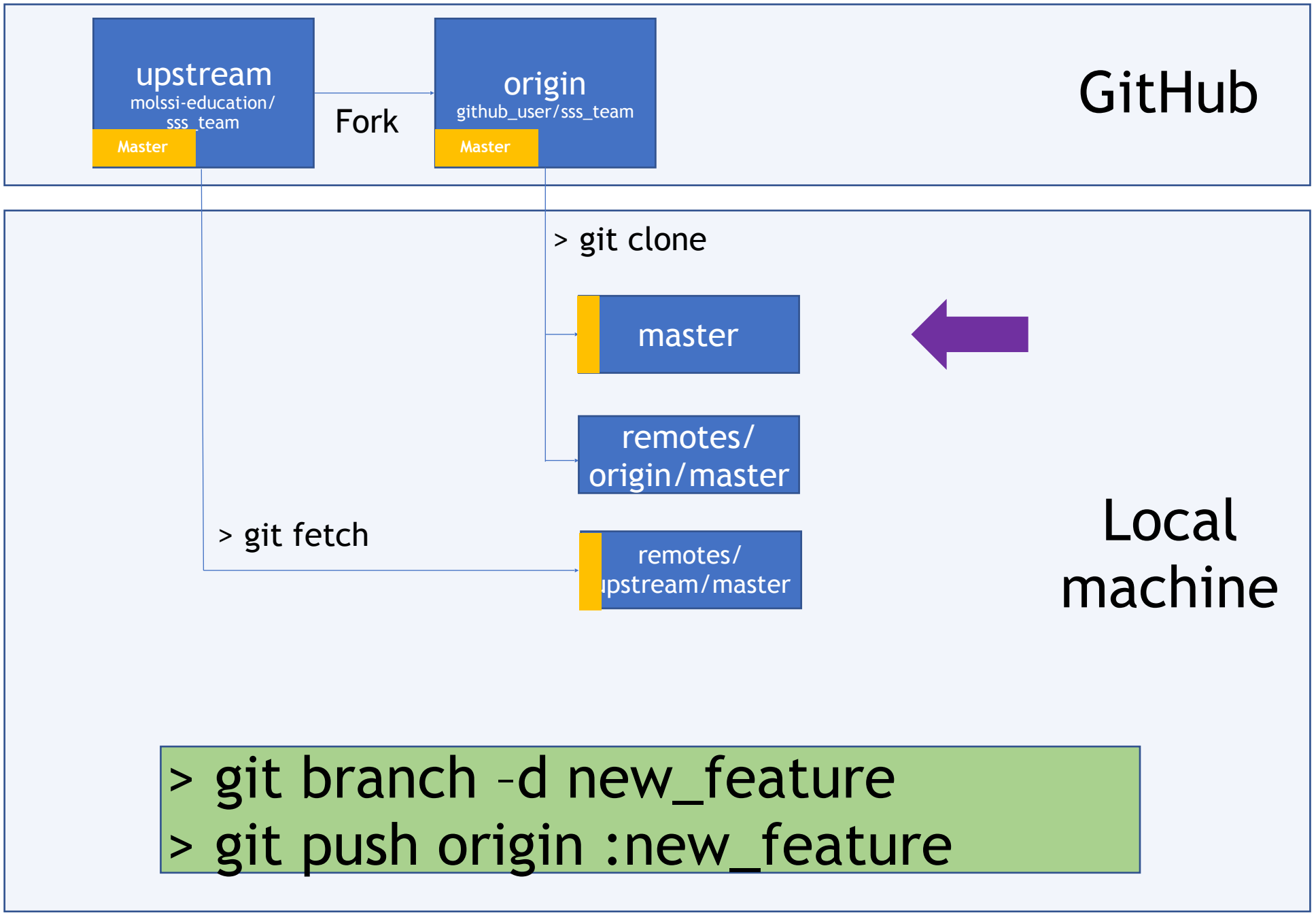
Notes

Updates your remote new_feature branch to keep up with upstream/master


upstream
molssi-education/
sss_team
Master
Fork
origin
github_user/sss_team
Master
new_feature
GitHub
> git clone
master
remotes/
origin/master
> git fetch
remotes/
upstream/master
new_feature
Local
machine
> git checkout master


upstream
molssi-education/
sss_team
Master
Fork
origin
github_user/sss_team
Master
new_feature
GitHub
> git clone
master
remotes/
origin/master
> git fetch
remotes/
upstream/master
new_feature
Local
machine
> git merge new_feature


upstream
molssi-education/
sss_team
Master
Fork
origin
github_user/sss_team
Master
new_feature
GitHub
> git clone
master
remotes/
origin/master
> git fetch
remotes/
upstream/master
new_feature
Local
machine
> git push origin master



Notes

Now that you have finished with the new feature, and your code has been incorporated into origin/master and upstream/master delete the branch.

Advantages and Features of this workflow

- Pull request is a way of doing code review on GitHub. Everyone can comment on the changes before merging any piece of work. Can get feedback from more experienced developers.
- The pull requests are (hopefully) open for a small amount of type to get enough reviews and merged as quickly as possible.
- Pull requests should be as concise as possible
- Multiple pull requests for bigger projects
- Each developer is responsible to address the feedback given in the pull request
- Master is more stable and always production-ready (In addition to code reviews, we need to set the execution of our test suite automatically with every pull request)

Disadvantages of this workflow

- Slower merging due to pull request review
- More commands to memorize – (more complex workflow)