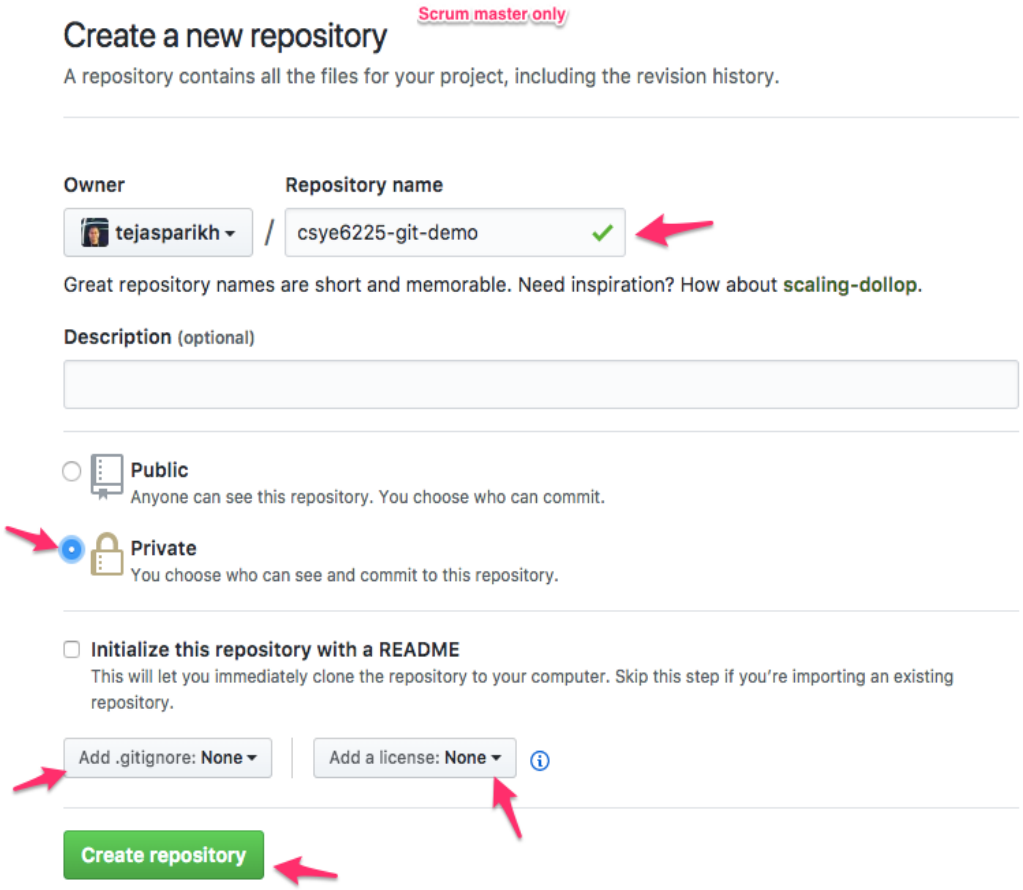
Git Forking Workflow Lab Session

Pre-requisites

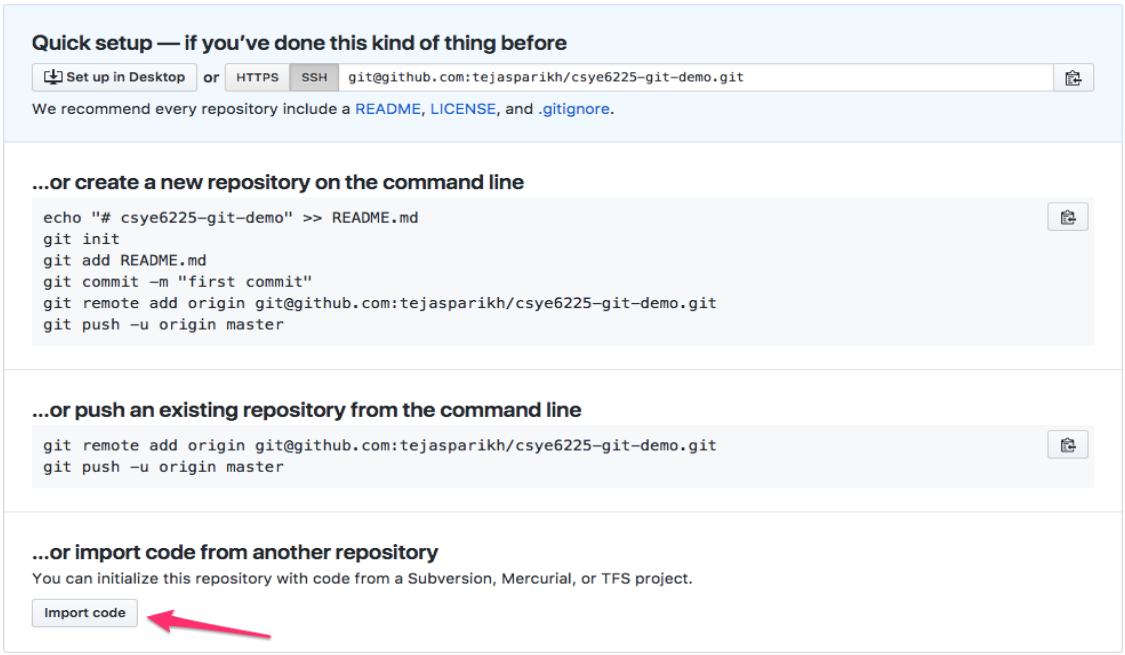
1. GitHub account.
2. Git client installed locally.
3. SSH Key Setup for GitHub. https://help.github.com/articles/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent/ and https://help.github.com/articles/adding-a-new-ssh-key-to-your-github-account/
4. Run thru initial setup for Git. See lecture slides.
5. Read thru https://www.atlassian.com/git/tutorials/comparing-workflows/forking-workflow

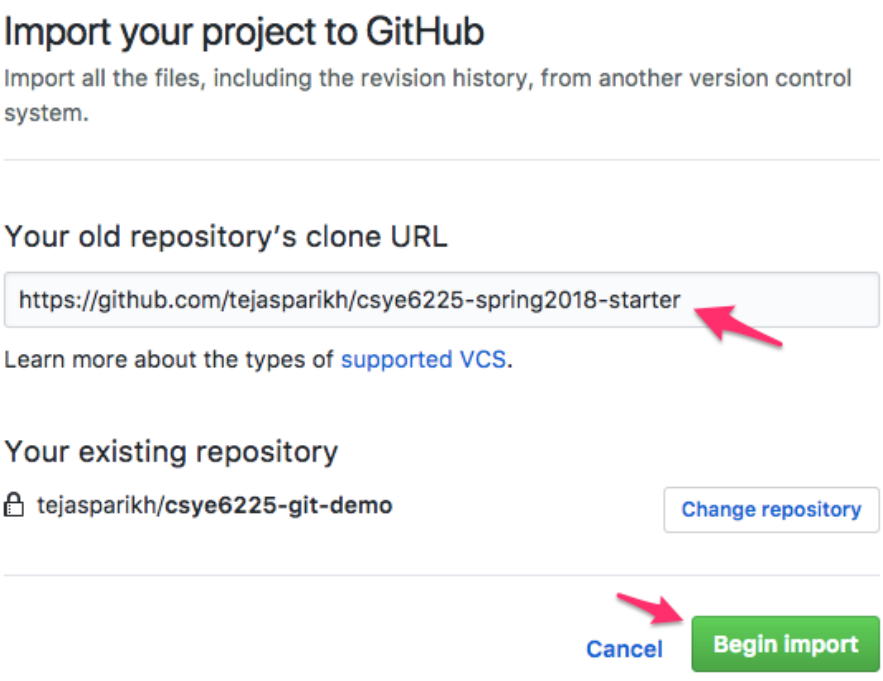
Steps to Setup GitHub Repositories for Collaboration

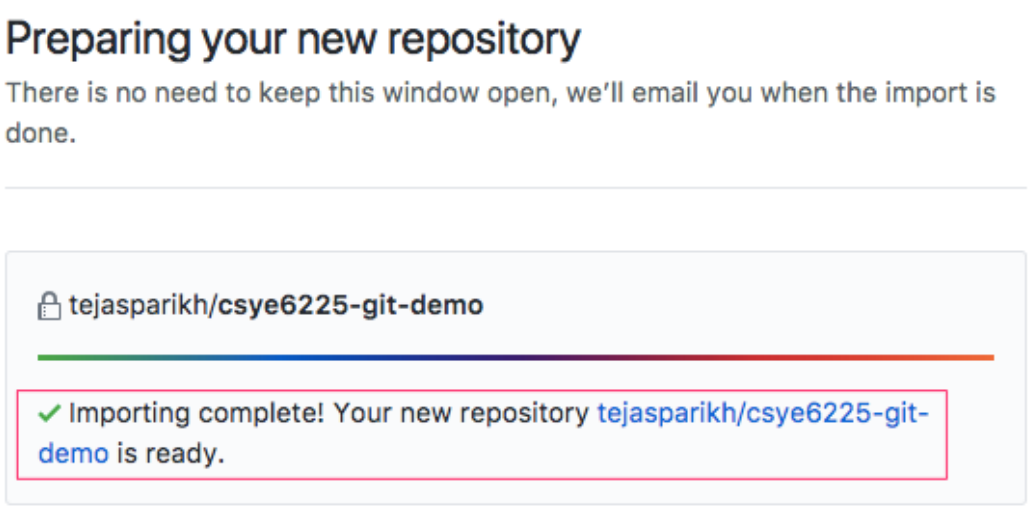
1. Choose one person in your team to be the “scrum master”.
2. Scrum Master should now create a repository in their GitHub called “csye6225-git-demo”



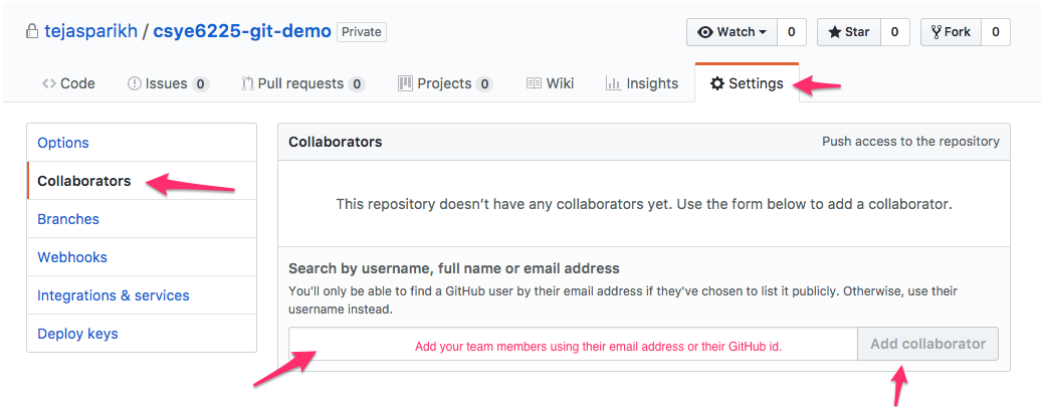
3. Import code from the starter application at https://github.com/tejasparikh/csye6225-spring2018-starter



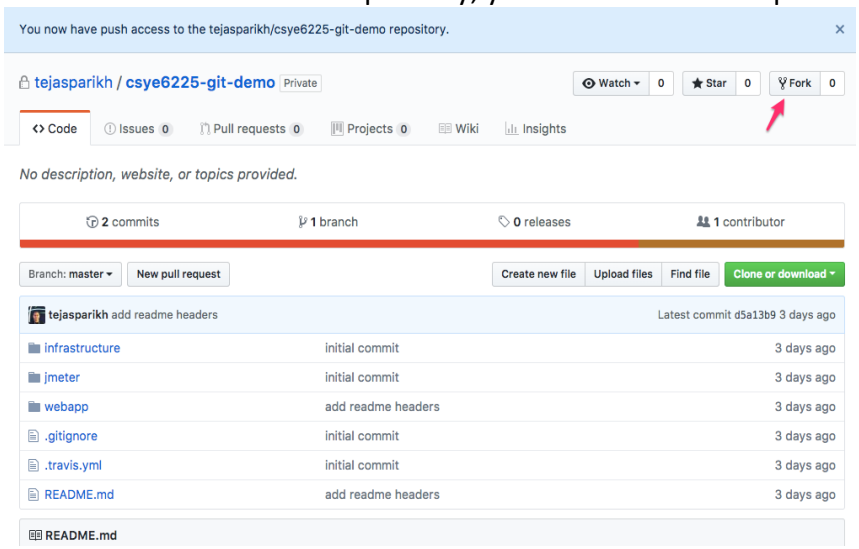




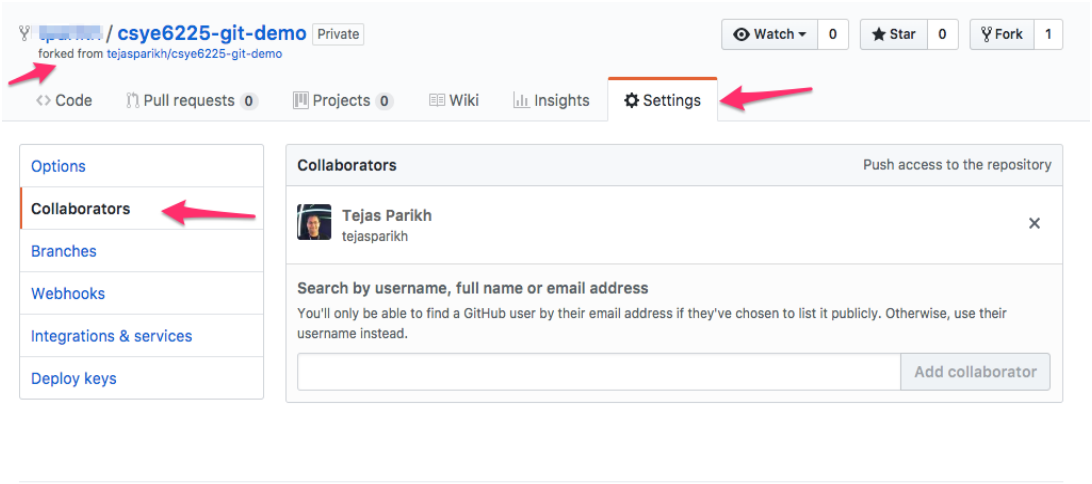
4. Scrum Master should add other team members to their repository as collaborators



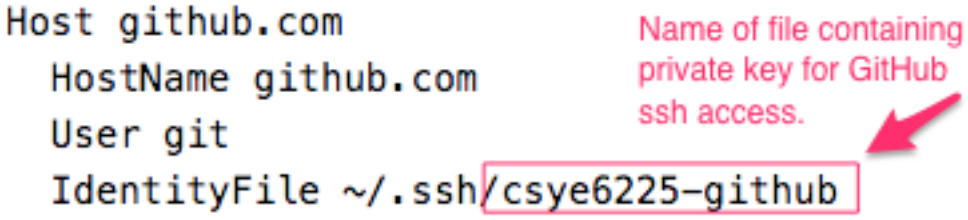
5. Once team members can see the repository, you should fork the repository.



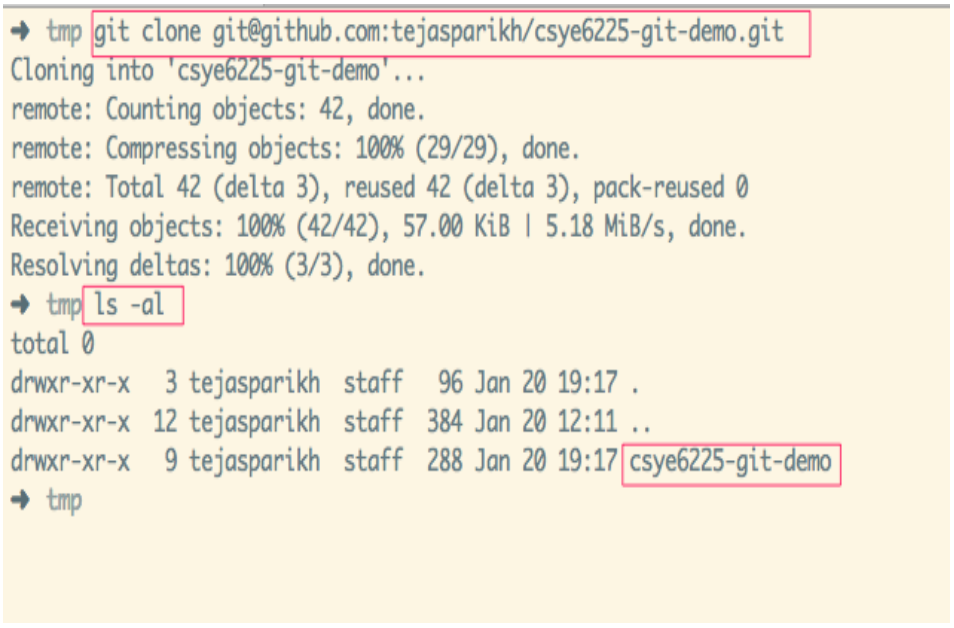
6. Now each team members should add rest of the team members including scrum master to their repository as collaborators.



7. Setup your local file system. I would recommend you create a course folder and a dev folder inside it by running the command *"mkdir -p csye6225/app && mkdir-p csye6225/dev"* in your terminal from your home directory (/home/user).
8. Create a *config* file in your *.ssh* directory and add following configuration in it



9. Change into *csye6225/dev* directory and clone your repository using the *git clone* command.

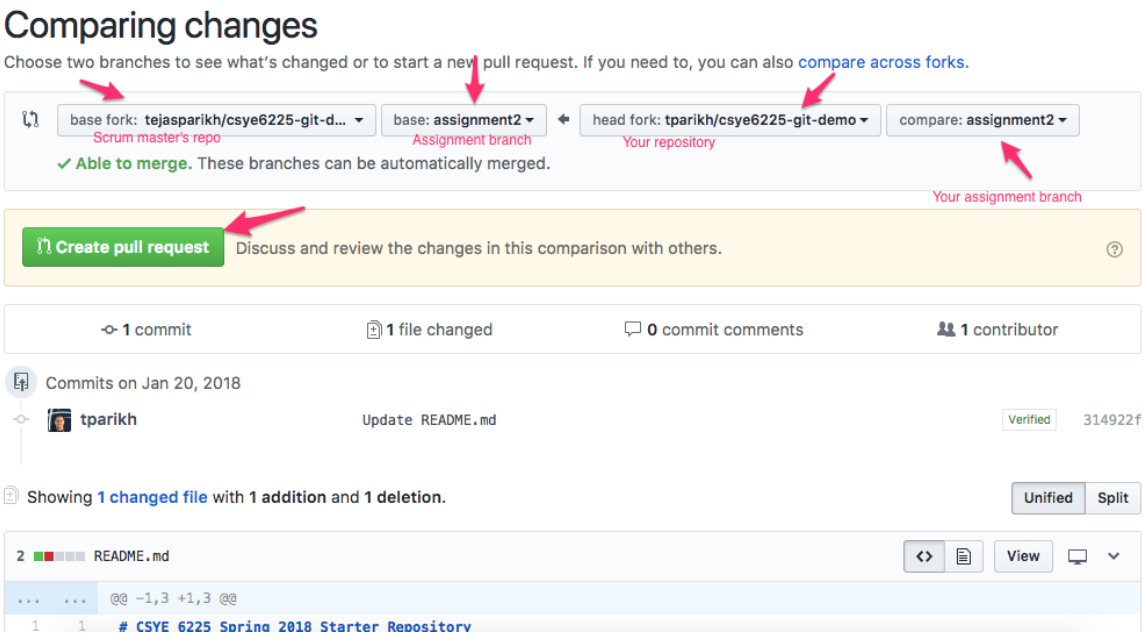


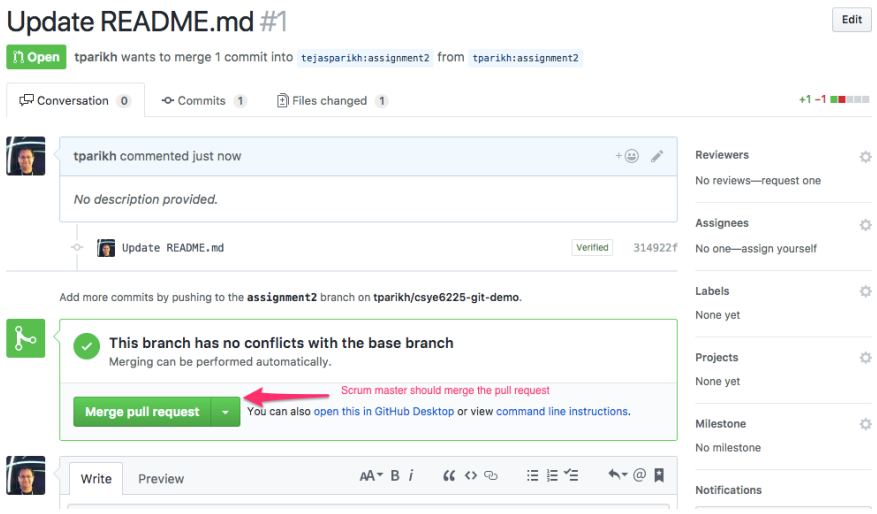
10. Now add aliases to your team members repository using the command
    git remote add <team_member_first_name> git@github.com:<TEAM_MEMBER_USERNAME>/csye6225-git-demo.git
    Repeat the above command for each one of your team members. Replace the placeholders surrounded by <>.
11. Fetch branches and/or from each one of your team members repository using the command *git fetch <team_member_first_name>* Replace the placeholders surrounded by <>.
12. At this point you are ready to start working with your team on the newly created GitHub repository.

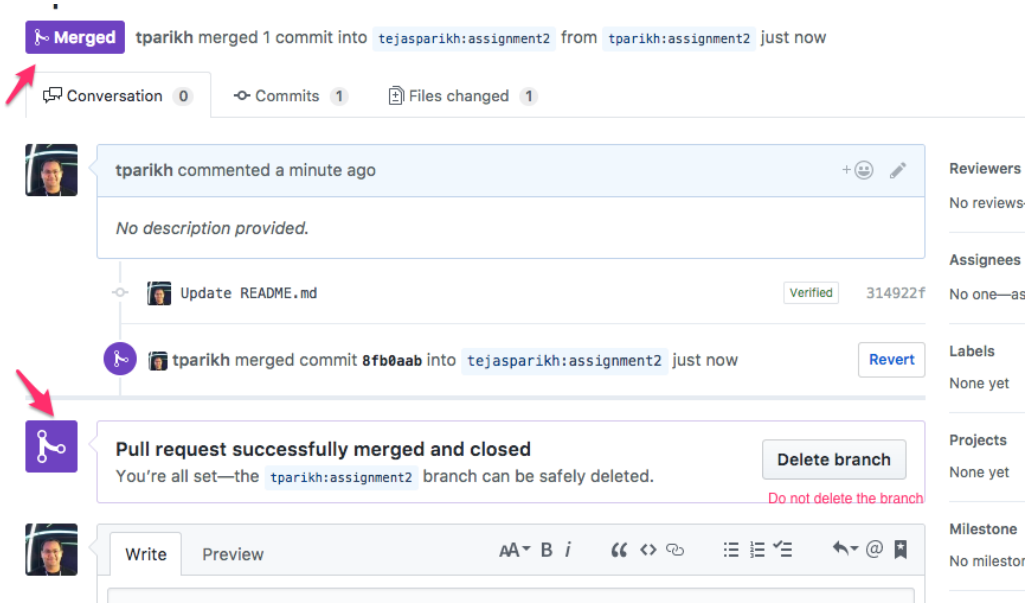
Working with Assignment Branches

All team members must execute following steps at the same time.

1. For each assignment you should create a new branch from the *master* branch using the command *git checkout -b <branch_name>*
2. Push this newly create branch to Github using the command *git push -u origin <branch_name>*
3. Now start working on code changes. For the lab session make changes to any of the existing files and stage the changes using the *git add -A* command.
4. Commit the staged changes using the *git commit -m "COMMIT MESSAGE"* command.
5. Push the changes to the server using the command *git push -u origin <branch_name>*
6. Now that everyone has pushed changes to their own "origin" repository, you should create a pull request to your scrum master's repository.
   https://help.github.com/articles/creating-a-pull-request/







7. Now that the code is merged sync your code with team member's repository. To pull latest changes from scrum master's repository, run the command *git pull <team_member_first_name> <branch_name>* in your repository. Replace the placeholders surrounded by <>.
8. Push the changes you pulled in step 7 to your repository using the command *git push -u origin <branch_name>*