

Version Controlling with Git - II

Introduction

Last week, we went through the basic commands in Git. This week, we are going to see one of the most important and useful features in Git: branching.

Along with branching, we will also see how to use a simple workflow (branching strategy) as well. You will learn a new concept called Pull Requests (PRs) in this practical session.

Since we have set up our git clients on our computers and GitHub accounts last week, we are just going to use them in this session.

Step 1 – Individual work

1. Create a Public GitHub Repository.
2. Then **Clone** it to your computer using the Git Clone command (*git clone <https://repo_url>*)
3. Once the cloning is complete, create a feature branch in the local repository.
 - a. There are two options for creating a branch.
 - i. *git branch <meaningful_branch_name>* - just creates the branch based on the current branch. Use *git checkout <new_branch>* to check out the new branch.
 - ii. *git checkout -b <meaningful_branch_name>* - creates the branch and immediately switches to the created branch.
 - iii. Make sure to give a meaningful branch name and follow a good naming convention. Here is an example.
E.g.: *git checkout -b feature/vishan.j/awesome-feature-x-to-our-app*
4. Again, add some text files/ html files (content doesn't matter) to this branch.
5. Commit changes in this branch locally and then push this branch to the remote.
6. Ensure your changes are available in the remote repository on GitHub if all went well. The changes should be available under your feature branch name.
7. Use [the GitHub official documentation](#) to create a Pull Request on Github for your branch.
8. Then merge the pull request to the main by referring to [this official documentation](#).

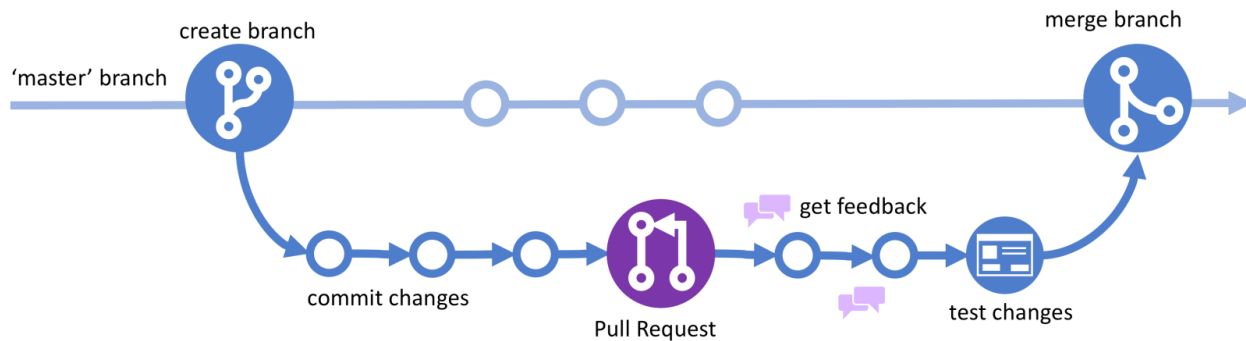
Step 2 – Collaborating with another colleague (Pair work)

1. Now pair up with a colleague (preferably sitting next to you) and add them to your repository as a collaborator by referring to [this documentation](#).
2. Now clone their repository to your computer.
3. Then as with step 1, create a branch and introduce a small change.
4. Push the branch to the remote repository.
5. Use [the GitHub official documentation](#) to create a Pull Request on Github for your branch.
6. Then merge the pull request to the main by referring to [this official documentation](#).

7. Once you complete this, reverse the roles and let your colleague practice too.

Step 3 – GitHub Flow (Pair work)

GitHub Flow



Copyright © 2018 Build Azure LLC

<http://buildazure.com>

Source: <https://build5nines.com/introduction-to-git-version-control-workflow/>

Next, practice implementing this simple workflow with your colleague.

1. Create a new feature branch from the main (master) branch. Name it properly in the format *feature/<your name>/<feature name>*.
2. Make several changes to the feature branch.
3. Once the feature is done, create a Pull Request (PR) for this to the main branch. Ask your colleague to review it.
4. If the PR is acceptable, the colleague will accept it.
5. Once it is accepted, delete your feature branch.
6. Once you complete this, reverse the roles and let your colleague practice too.

References

1. [Git Reference - Git Official Documentation](#)
2. [GitHub Git Cheat Sheet](#)