NDN's code review

By Zhiyi Zhang (zhiyi@cs.ucla.edu)

Contents:

Introduction to Gerrit

Different from directly using GitHub (or GitLab or others), NDN introduce a code review system before your code can finally go to the GitHub to ensure the high quality of the official code.

The code review system is called gerrit: https://gerrit.named-data.net To use gerrit, please follow the steps:

1. Go to gerrit website and register a new account on gerrit. You can simply use your google/github account to do that.

2. Using the nav bar, click the gear icon to go the user setting page.

3. In user setting page, set your username, e.g., your first name or any name you like

4. In user setting page, set your email address to be the one used in your git. To know your email used in git, check:

   ```
   git config --global --get user.email
   ```

5. In user setting page, set your SSH key. To set your SSH key, copy paste the result of:

   ```
   cat ~/.ssh/id_rsa.pub
   ```

   If you don't have this file, please google "github generate SSH key" and generate a key pair for your device

6. Using the nav bar, go to BROWSE -> Repositories and select the project that you want to work on. Using ndn-cxx as an example, find ndn-cxx among the project list and click its name to go into ndn-cxx's page (https://gerrit.named-data.net/admin/repos/ndn-cxx)

7. Use the SSH to download the project. Specifically, you need to use "Clone with commit-msg hook"

How to work with Gerrit

Let's still use ndn-cxx as the example. Now you have your ndn-cxx downloaded from gerrit. It's time to learn how to work with gerrit. Several important things to know:

- After you push your code to gerrit, reviewers will review your code and give you comments. After you update your code and address all the comments, you need to change the current commit instead of creating a new commit. In other words, you need to use "git add . && git commit --amend" to "change the history" instead of using "git add . && git commit" to create another commit. It's like you have different versions of a single commit.
- In the step 6 in the previous section, you need to clone with commit-msg hook. This commit-msg hook is a small tool that will generate a change-id whenever you create a new commit. What is this change-id used for? After your "git

add . && git commit --amend", the SHA value of your commit changes but the change-id won't change. Change-id will help gerrit to recognize all your versions of the same commit.

Follow the following step to push a commit to gerrit.

1. coding

2. commit your code.

   ```
   git add .
   git commit -m "let others know what you've done"
   ```

3. push your change to gerrit. Importantly, you should NOT use

   ```
   git push origin
   ```

   you should use:

   ```
   git push origin HEAD:refs/for/master
   ```

   After that, you should be able to see your commit on gerrit's webpage.

Okay. now we assume that there is a reviewer who gave a -1 to your code. for example, he asked you to change a variable name from "varname1" to "varName1".

4. coding: you change the var name to "varName1"

5. as mentioned, do commit --amend:

   ```
   git add .
   git commit --amend
   ```

6. push your change to gerrit. Importantly, still use:

   ```
   git push origin HEAD:refs/for/master
   ```

   After that, you will see you have a new version of the same commit on gerrit's webpage.

Some life hacks when using Gerrit

This is for MACOS ONLY

A nice GUI to show your git commit tree is so important for you to use git. Go to https://rowanj.github.io/gitx/ and download this awesome tool.

1. click the downloaded file

2. drag the gitx to your Applications dir

3. open gitx and using the nav bar: Gitx -> Enable Terminal Use

4. go back to your project (e.g., ndn-cxx) in terminal, type in:

   ```
   gitx
   ```

you will find everything is so clear :D