The 'gerrit' code review system

Jan-Simon Möller
training.linuxfoundation.org
Topics

- What is 'Gerrit'
- Gerrit workflow
- OPNFVs use of Gerrit
- First steps, Submitting code, Reviewing
- Hands-on labs
What is 'Gerrit'
Gerrit

- Web-based code review tool
- Maintained by Google
  - https://www.gerritcodereview.com/
- Framework for reviewing every change
  - changes can be uploaded/proposed
  - but do not get applied until reviewed
  - combination with CI tool possible
Gerrit

- Serves/hosts git repositories
- Provides review workflow
- Access control for repositories
Gerrit workflow
Gerrit workflow
Gerrit workflow

- Developers fetch from the central repository
- Pushed changes get staged until they are reviewed
- This process can be combined with CI builds
OPNFVs use of Gerrit
OPNFV & Gerrit

• Every project has a repository
• Changes are submitted through gerrit (using 'git review')
• Review mandatory
• Automated testing supports review process
First steps,
Submitting code,
Reviewing
Topics

- Create ssh key *(if not present already)*
- Log-in to gerrit, upload ssh key, sign agreements
- Install git-review *(if not installed already)*
- Clone repository
- Submit change
- Review
Create ssh key (if not already present)

- `ssh-keygen` is used to create a ssh private/public keypair

```bash
$~/> ssh-keygen -t rsa -C "youremail@yourdomain.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/me/.ssh/id_rsa): ↵
Created directory '/home/me/.ssh'.
Enter passphrase (empty for no passphrase): <secretpass> ↵
Enter same passphrase again: <secretpass> ↵
Your identification has been saved in /home/me/.ssh/id_rsa .
Your public key has been saved in /home/me/.ssh/id_rsa.pub .
[...]```
Create ssh key  (if not already present)

- `ssh-add` can be used to avoid re-entering passwords

```shell
$~/> ssh-add ~/.ssh/id_rsa
Enter passphrase for /home/me/.ssh/id_rsa: <secretpass> ↵
Identity added: /home/me/.ssh/id_rsa (/home/me/.ssh/id_rsa)
$~/>
```

# Note: You'll have to do this once before you use ssh or git
# every time you log in to your system. It works like
# a keyring and will stay in the background.
Log in to gerrit.opnfv.org

- Open a browser and navigate to gerrit.opnfv.org
- Use your Linux Foundation ID to log in.
gerrit.opnfv.org

- The main screen of gerrit has
  - a few tabs on the left, a search bar
  - a personal settings menu on the right
Upload ssh key

- Click on your name
- Click on 'Settings'
- Select 'SSH Public Keys'
Upload ssh key

- Paste the content of .ssh/id_rsa.pub here
- Click on 'Add'
Contributor Agreement

- Clock on 'Agreements'
- You must accept the adopted agreements of the OPNFV project before you can submit code for review!
- See https://gerrit.opnfv.org/gerrit/static/CLA_individual.html
Install git-review *(if not installed already)*

- Install 'git-review' using your distro package manager
- alternatively install 'pip' and call 'pip install git-review'
Clone repository

- Clone the sandbox repository and initialize git review

```
$~/test/> rm -rf sandbox
$~/test/> git clone \\n    ssh://<username>@gerrit.opnfv.org:29418/sandbox

$~/test/> cd sandbox
$~/test/sandbox/>
$~/test/sandbox/> git review -s
```

Creating a git remote called "gerrit" that maps to:
```
    ssh://<username>@gerrit.opnfv.org:29418/sandbox.git
```

$~/test/sandbox/>
Submit change

• create/checkout a branch starting from gerrit/master

• Create a file `<yourfirstname>.txt`
  – content: 'Hello World'

• stage the file

• commit the file

• submit change using git review
Submit change

$~/test/sandbox/> git checkout -b mybranch gerrit/master
    # => shortcut for branch+checkout

$~/test/sandbox/> echo 'Hello World!' > yourfirstname.txt

$~/test/sandbox/> git add yourfirstname.txt
$~/test/sandbox/> git commit -s
    # Enter a proper commit message -
    # see git tutorial
$~/test/sandbox/> git review
remote: New file detected, please ensure that it has the correct license header
remote: Processing changes: new: 1, refs: 1, done
remote:
remote: New Changes:
remote:   https://gerrit.opnfv.org/gerrit/6745
remote:
To ssh://<username>@gerrit.opnfv.org:29418/sandbox.git
  * [new branch]   HEAD -> refs/publish/master/mybranch
Review

Change 6745 - Needs Code-Review

First commit of tutorial:

This adds `Hello World!` to a file.

Signed-off-by: Jan-Simon Moeller <jsmoeller@linuxfoundation.org>

Author: Jan-Simon Moeller <jsmoeller@linuxfoundation.org>
Commit: 932c997a6975d669cfa172bf21c6c51f46a0815c
Commit-time: 2016-01-14 14:21

Files:

- Commit Message

  File Path | Comments |
  --------- | -------- |
  /jensimon.txt | 1        |

History:

Jan-Simon Moeller  Uploaded patch set 1.  2016-01-14 21:24
Review

- Add reviewers
- Rebase (rare)
- Abandon
Review (by project members/reviewers/owners)

- Review the changeset
- Comment here
- Rate
  - +1/0/-1
    - approve/reject
  - +2/-2
    - force accept/deny (only for committers/project owners/admins)
- Verified (e.g. by CI builder)
Hands-on labs
Task 1

- Set up your system for gerrit
  - ssh-key
  - upload key to gerrit.opnfv.org
  - install git-review
- Clone the sandbox repository
- Initialize git review ('git review -s')
Task 2

- Create a branch 'mytopic' from gerrit/master
- Create a file <yourfirstname>.txt with the current date/time
- Stage and commit the file with a proper message
- Submit the change for review
Task 3

• You can amend existing changesets and resubmit
• For this you need to keep the Change-Id:  
in the commit message intact
• Extend <yourfirstname>.txt with 'Hello World',  
  stage, commit and (re-)submit for review
• (Note: one local branch per changeset helps a lot )