Develop

QTIP Developer Guide

This guide is about how to develop QTIP.
If you just want to use it for performance benchmark, check the user guide instead.

Getting Started

Source code

~$ git clone https://gerrit.opnfv.org/gerrit/qtip

VirtualEnv

It is recommended to use virtualenv to isolate your development environment from system, especially when you are working on several different projects.

Testing

QTIP use tox to automate the testing tasks

$ pip install tox
$ tox

Undering macOS system, it will happen to a fatal error when installing package cryptography:

```
'openssl/opensslv.h' file not found
#include <openssl/opensslv.h>
^ 1 error generated.
```

It is for macOS uses TLS instead of OpenSSL and no header files supported. The solutions is:

```
# brew install openssl
# add these lines in to your shell profiles, such as .bash_profile, .zshrc
# export CPPFLAGS='-I $openssl_install_path/include'
# export LDFLAGS='-L $openssl_install_path/lib'
```

Third Party Code

QTIP includes a few third party code via subrepo. All third party code are stored in /third-party.

To pull the changes from remote repository, use

```
git subrepo pull third-party/<subdir>
```

It will create a new commit in parent repo, i.e. qtip. However, the auto generated commit message does not include mandatory tags such as Change-Id required by gerrit. You need to manually amend the commit to append those.

```
git commit --amend -s
```

Example of final commit message
Include third party script for license checking and amending
The following commit message are generated automatically by git-subrepo
----------------------------------------------------------------------------
git subrepo clone git@github.com:openzero-zte/License.git third-party/License

subrepo:
    subdir: "third-party/license"
    merged: "61489da"
upstream:
    origin: "git@github.com:openzero-zte/LICENSE.git"
    branch: "master"
    commit: "61489da"
git-subrepo:
    version: "0.3.0"
    origin: "https://github.com/ingydotnet/git-subrepo"
    commit: "988f8c8"
----------------------------------------------------------------------------
Change-Id: I8eab86a8ce3f26995af3e3535f31f361b4826a8b
Signed-off-by: Yujun Zhang <zhang.yujunz@zte.com.cn>

Sometimes you may modify the third-party code to adapt it in your repository. To push the changes to a remote repository, run:

```bash
git subrepo push third-party/<subdir>
```

If you want to include a new repository of third-party code, use:

```bash
git subrepo clone <remote-url> [<subdir>]
```