Getting Started with Contribution to Openstack

Created by:
Yatin Karel
IRC Nick: ykarel

Openstack Project Management that you need to know

Source Code for Services/Libraries/Clients:-
https://github.com/openstack
https://github.com/openstack/<project-name>
Bugs are tracked at:-
https://bugs.launchpad.net/
https://bugs.launchpad.net/<project-name>
Code Review is done at:-
https://review.openstack.org/
https://review.openstack.org/#/q/project:openstack/<project-name>

Communication

- IRC:- Each project has his own channel for discussion and meetings on freenode server https://wiki.openstack.org/wiki/IRCpage
- IRC Logs http://eavesdrop.openstack.org/irclogs

- Mailing lists: For open discussion/announcement across teams https://wiki.openstack.org/wiki/Mailing_Lists

Setup Accounts

- Launchpad: This will create single sign-on for review.openstack.org also https://login.launchpad.net/3nNMopjfRFPZ7i5t/+login
- Join as a Foundation member: https://www.openstack.org/join/register/?membership-type=foundation
- Login to review.openstack.org and set a username(can use same as launchpad username), sign ICLA and add your public ssh key.

Contribution to openstack

Contribution can be done in many ways:-

- Filing bugs on launchpad, but remember to give details for the bugs.
- Triaging bugs on launchpad if you know the fix
- Resolving queries on IRC/Mailing lists
- Submitting patches: functional or non-functional
- Reviewing patches
- Drafting/Implementing features

Not tried devstack yet, try it you will learn more. Reference from our last meetup:-
https://review.rdoproject.org/etherpad/p/devstack_vm

Setup your development environment

$ sudo yum install -y python-devel openssl-devel mysql-devel libffi-devel git git-review gcc python-pip python-tox

$ git config --global user.name "Firstname Lastname"
$ git config --global user.email "your_email@youremail.com"
$ git config --global gitreview.username “your gerrit username”

Clone and push source code for review

# Clone project code locally and start contributing
$ git clone https://github.com/openstack/<project-name>
$ git checkout -b <topic-branch>
# Update source code and run unit tests locally as defined in project guide, mostly tox is used to run tests in a virtualenv
$ tox -epy27 #can check tox.ini for available options like py27,py35, pep8 etc
# Fix failures and push your code for review
$ git commit -a -m “Your commit message one line”
$ git review -s
# Add description to the commit message
$ git commit --amend
$ git review
You will get review link in the output of above command

Post source code push

The source code is tested against predefined test jobs(check pipeline: see example [0]) for a project, jobs, events, triggers all are automated and are defined in following repository: https://github.com/openstack-infra/project-config (see example [1] and [2]).

Status of jobs can be checked at: http://zuul.openstack.org/

Examples:-
[0] https://github.com/openstack-infra/project-config/blob/master/zuul/layout.yaml#L4
[1]https://github.com/openstack-infra/project-config/blob/master/zuul/layout.yaml#L12742-L12804
[2]https://github.com/openstack-infra/project-config/blob/master/zuul/layout.yaml#L11257,L11293

continue...

- Jenkins review (-1, +1)
- Peer review (-1, 0, +1)
- Core review (-2, -1, 0, +1, +2) and Workflow(-1, 0, +1)
- Gate jobs defined in openstack-infra/project-config, see examples [0]

Examples:-
[0]
https://github.com/openstack-infra/project-config/blob/master/zuul/layout.yaml#L11284-L11284

continue...

- Check logs for failures and fix the issues. Logs are stored at logs.openstack.org, the result of jobs contains the link to logs.
- During the test run also logs can be checked at logs.openstack.org/<last 2 digit of review id>/<review id>

Some sample reviews:-

Submit your first patch:-

Thank You….