My Journey Upstream with tlog Testing

Introduction

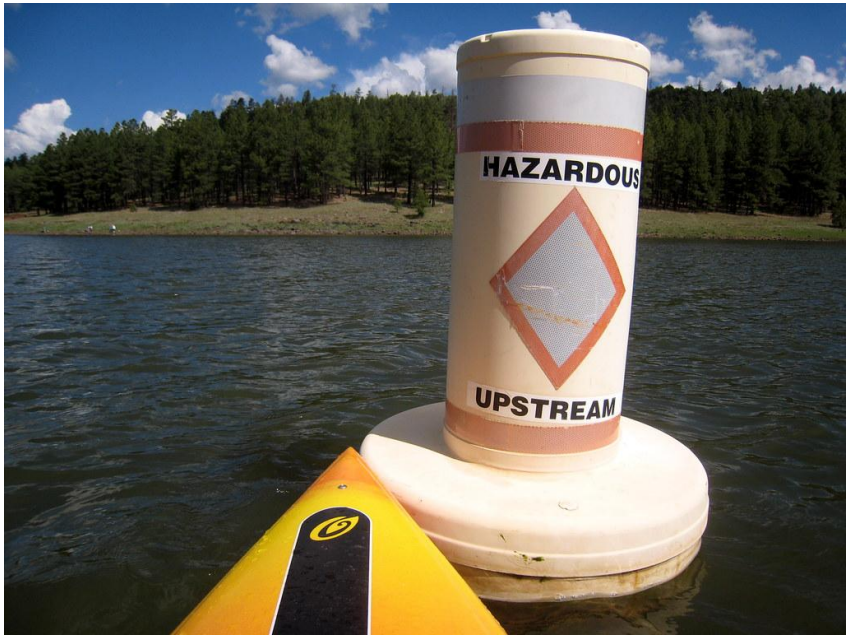
Scott Poore

- Quality Engineer at Red Hat
- Identity Management testing in RHEL

My Journey Upstream

My Journey Upstream

- Terminology
- Starting Point
- Challenges
- How to Contribute
- Plan Tests
- Write Tests
- Submit Upstream
- What I Learned



Terminology

What is Open Source?

- Originated with Open Source Software
  - Source code available
  - “anyone can inspect, modify, and enhance”
  - https://opensource.com/resources/what-open-source
- Applied more broadly today
  - Products
  - Processes
  - https://opensource.com/open-source-way
- Anything worked on publicly and shared openly

What is Open Source?

- Working "openly" in public
- Usually free to use, modify, share
  - according to the terms of the license
- Licensing varies
  - many to choose from
  - can charge money for the software/product
  - must provide source code
  - https://opensource.com/law/13/5/does-your-code-need-license
  - https://en.wikipedia.org/wiki/Comparison_of_free_and_open-source_software_licences

Upstream vs Downstream

- Upstream
  - public repositories
  - outside help
  - tests maintained with the source
- Downstream
  - private repositories
  - limited help
  - tests maintained separately
  - processes may differ greatly from upstream
    - type of git repo
    - workflow
    - people
- https://en.wikipedia.org/wiki/Downstream_(software_development)

From Upstream to Downstream

Linux Kernel

Fedora

Red Hat

Tux.svg

Fedora-logo.svg

Starting Point

Starting Point

- Little experience working upstream
- RHEL Identity Management testing
- Most of my tests were downstream
  - processes differed (at the time)
- tlog would be my first “real” upstream tests

Motivation

- I wanted to work upstream
- tlog was already upstream
- What is tlog?
  - terminal I/O recording and playback
  - also provides recording user shell activity
- What needs testing?
  - CLI tools, recording, and playback
  - user’s shell activity recorded from login
- Where to maintain tests?
  - Upstream!

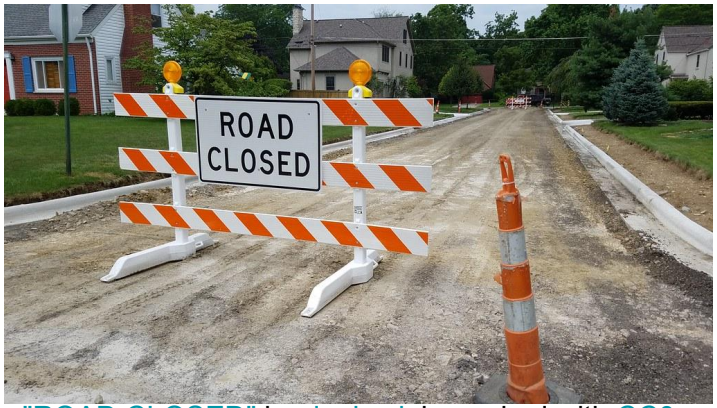
Challenges

Why am I here today?

- Getting involved in Open Source can be intimidating at first

"wooden door" by zebulon.walton is marked with CC PDM 1.0

What are the barriers?



- It can be intimidating!
- New languages, toolsets, and processes
- Little to no documentation
- High skill level required
  - you may be interested in something that is beyond your current skills
- Tight knit communities
  - people for whatever reason aren't welcoming
  - expect only the best developers
  - no time to train someone new

My fears

- Not familiar with github/upstream workflow
  - Have to learn new tools and workflow to contribute
- Working with highly skilled developers
  - It can be intimidating
- Worried my code wouldn't be good enough
  - Have to re-write everything
  - Techniques would be considered wrong
  - Strict adherence to coding guidelines
  - All things that make OSS better but, still intimidating!

How to Contribute

How to contribute to Open Source?

- Dev -- write software
- Doc -- write howtos, faqs, etc
- Test -- write automated tests
- Help -- irc, mailing lists, forums

Why Tests?

- It helps people like me (and Dev!)
- Easy way to learn the project
- Software always needs more testing
- A lot of projects already have tests upstream
- Easier entry sometimes than development of the software
    - tests in python, software in c

Plan Tests

tlog

- git repo README
  - https://github.com/Scribery/tlog/blob/master/README.md
- list files
  - rpm -ql tlog
- man pages
  - man tlog-rec
- use it
  - tlog-rec -o record.out whoami
  - usermod nuser1 -s /usr/bin/tlog-rec-session*

* Note: for RHEL it is suggested to use SSSD not usermod

Planning

- Identify Test Cases
  - break into groups based on CLI tools and config changes
- Write up high level TCs
  - brief description of what I was going to test
- Review with Dev
  - share doc and get feedback

Review

- Planning tests went well!
- Dev reviewed plan doc and gave feedback
- Minimal changes necessary
- Straightforward and painless
- Nothing to worry about...so far so good

Write Tests

Time to Code

- Pick a language/framework
  - python/pytest
  - https://docs.pytest.org/en/stable/getting-started.html
  - already familiar so no learning curve
- How to handle interactive I/O
  - subprocess???
  - pexpect
  - https://pexpect.readthedocs.io
- Time to code
  - Developed tests offline
  - written initially to be run independent of the source
  - this was a mistake. I should have simply started with my fork.

Submit Upstream

The Pull Request

- Most stressful step
  - How will my PR be received?
- Had to break up my code into smaller PRs
  - Start with small subset of tests, setup, and libraries
  - https://github.com/Scribery/tlog/pull/231
- Review, fix, repeat
  - Dev suggested changes
    - code fixes
    - where to store things
    - shell tips for setup script
  - Submitted updates
  - We repeated until PR was accepted and merged

Outcome

- Was not as difficult as I thought it was going to be!
- The hardest part was breaking up my code

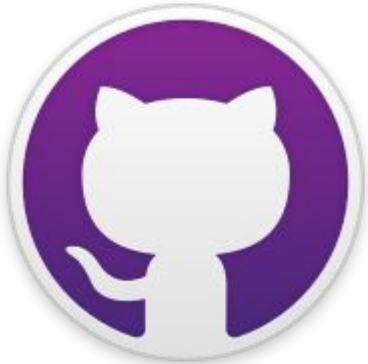
What I Learned

Upstream, Tools, and Processes

- Working upstream wasn’t that bad!
- pexpect
- shell tips
- github workflow
- wish I’d written tests upstream from the start

Github Flow

- github flow: https://opensource.com/article/19/7/create-pull-request-github
  - fork project
  - git clone your_fork
  - git checkout -b your_change
  - edit files
  - git add; git commit;
  - git push -u origin your_change
  - Submit PR
  - PR reviewed and changes suggested
  - make changes
  - git add; git commit --amend;
  - git push -f origin your_change
  - PR merged



What can you do?

Write Tests!

Get started with Open Source Software

- Picking a project
  - something you like
  - something you know
- Find Docs
  - README.md
    - how to install and start using
  - CONTRIBUTING.md
    - how to contribute
- Learn the software
  - official documentation
  - man pages

Start writing tests

- Review existing tests if they exist
- Add tests directory if none exists
  - you may be asked to change this location
- Start writing tests
  - python/pytest/java/bash/go/whatever
  - some projects use ansible
- Submit PR
  - review with Dev or project team
- Congratulations!
  - You've now successfully contributed to Open Source!

Useful resources

- https://opensource.com/resources/what-open-source
- https://opensource.com/law/13/5/does-your-code-need-license
- https://en.wikipedia.org/wiki/Comparison_of_free_and_open-source_software_licences
- https://github.com/Scribery/tlog/blob/master/README.md
- https://opensource.com/article/19/7/create-pull-request-github

Thank You