



SUPERUSER™

Type to search

User Stories

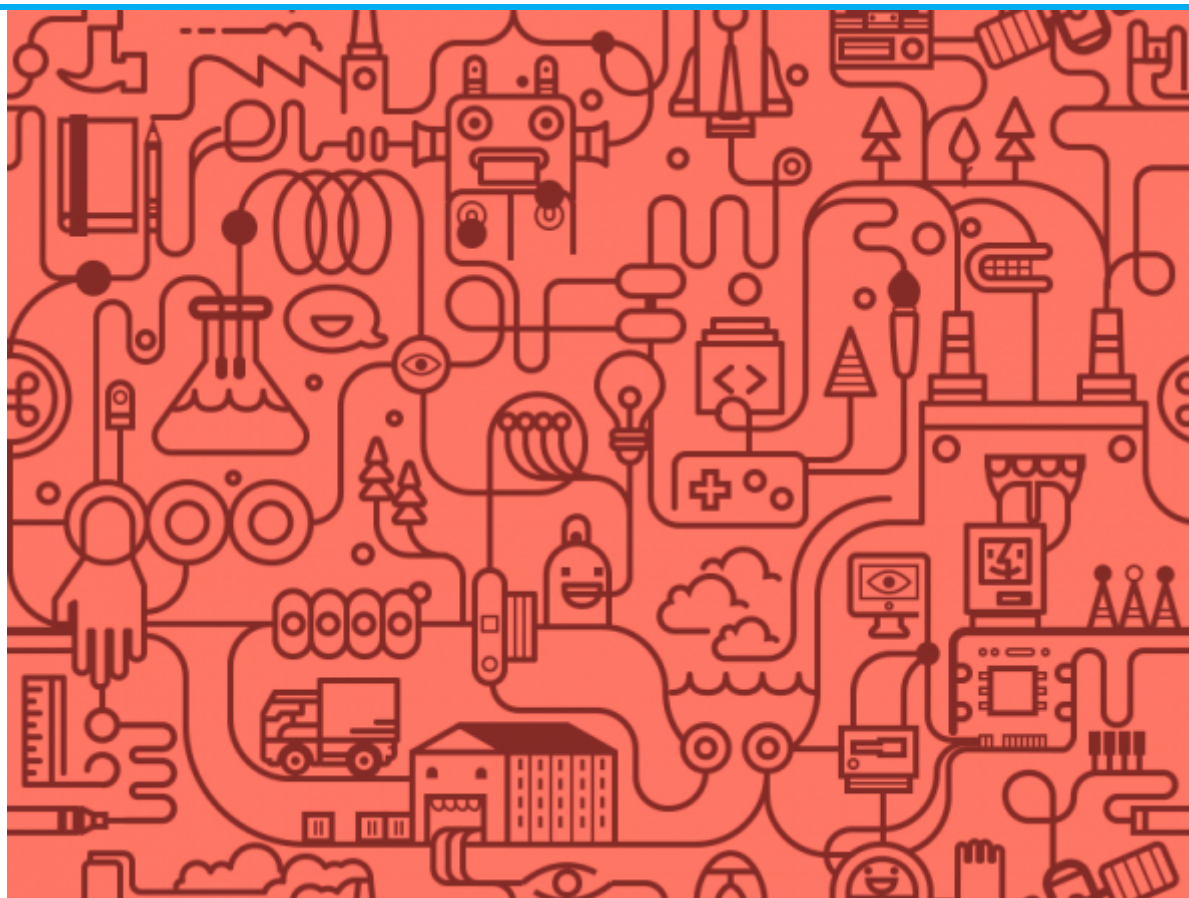
How-To

Superuser TV

Awards

About

Search



HOW-TO

How to Effectively Contribute to An Open Source Project Such As OpenStack Neutron

A look inside at what it takes to ensure a project has a successful Juno release.



Kyle Mestery
July 31, 2014

This post originally appeared on [Kyle Mestery's blog Silicon Loons](#). Kyle is the Project Technical Lead for OpenStack Neutron, and a Principal Engineer at Cisco Systems where he works on OpenStack, OpenDaylight, and Open vSwitch. He is an active speaker at Open Source conferences, and the founder of the Minnesota OpenStack User Group. You should [follow him on Twitter](#).

Since being [elected](#) as the OpenStack Neutron PTL, I've been mostly heads down working to ensure the Neutron project has a successful Juno release. Increasingly, and especially [near OpenStack Juno](#) milestone deadlines, I'm seeing frustration from new contributors around their contributions to Neutron. I sent an [email](#) to the openstack-dev mailing list this morning addressing this in a terse form, this blog is an attempt to expand upon that email.

An increasing concern I see from people who are new contributors is the perceived issues in getting their code merged into Juno. The common concerns come from one of the following possible ideas:

1. OpenStack Neutron is a "closed development" environment where only a select few are allowed to make changes.
2. OpenStack Neutron is part of some grand conspiracy by vendor X, and as I work for vendor Y, my changes never get merged.
3. OpenStack Neutron needs more core developers, this would solve all the velocity problems.

Addressing the above concerns in order, OpenStack Neutron is not a "closed development" environment. We do all of our work in the open by using mailing lists and IRC (for both meetings and discussions). Anyone can submit a patch to OpenStack Neutron once you've signed the CLA and followed the process [here](#). We welcome new contributors and do our best to work with them! My advice for new contributors, echoed below, is to start small, listen and learn.

The second concern around conspiracy theories comes from people who are frustrated that their specification or blueprint isn't prioritized for inclusion into a specific Juno release. They may also work for a competing company of some existing Neutron core developer, and think there is some political machinations going on here. While it's true upstream developers work for corporations (we like to get paid and feed our families, it's true!), it's also true the core team in particular works together as a team. This means that people who work for Cisco do in fact work with people from VMware, as an example. We're all working to drive stability and innovation upstream, but we do it in an open and collaborative manner.



OpenStack Community Weekly Newsletter July 18-25

Highlights from the weekly community newsletter.



Stefano Maffulli



OpenStack Community Weekly Newsletter Aug 15-22

Highlights from the weekly community newsletter.



Superuser

The third concern is a tricky one. While adding more core reviewers would certainly help spread the load, we're generally picky about adding new ones and have a [policy](#) in place around how we do this. The reason we're picky is because being a core reviewer is a big responsibility, and we want dedicated developers who can contribute time upstream for the role. We're in the process of coming up with a mentoring program to groom Neutron core developers, which will hopefully help here.

I'd like to reiterate from the email a list of effective ways to contribute to an upstream Open Source project:

1. Get involved in the ways the project uses to communicate, including IRC, mailing lists, Hangouts, phone calls, whatever they use.
2. Work upstream to fix some bugs. Documentation bugs and low-hanging fruit bugs are a good place to start.
3. Understand how the existing team works. It's best to understand this before suggesting any changes.
4. Come to weekly meetings. Make sure you spend a little while listening to understand how the existing team works before jumping in with your own agenda.
5. Build relationships. Doing the 4 points above will naturally lead you to this point.

An Open Source project is really a large development project being done in the open. To effectively join this project, you need to learn how the project operates and contribute in small ways initially. As in close sourced development, gaining the trust of the existing contributors is key to having influence over the direction of the project. The existing developers for a project such as Neutron (both core and non-core developers) all spend varying amount of time upstream working. But the key point is that they have taken the time to develop their contributions as well as grow relationships with existing upstream developers.

Upstream effectiveness is really about the time you put into your contributions. As existing developers upstream, we're responsible for the current and future state of a project like OpenStack Neutron. While we have [policies](#) in place dictating how we work, we're also responsible for things like gate failures, bug regressions, packaging issues, broad community interaction, and core developer grooming, among other things.

This post isn't meant to discourage anyone from being an upstream developer. Quite the contrary, actually. I'm hoping to highlight effective ways to work upstream, which is beneficial for both new



What you need to know about Astara

Henrik Rosendahl, CEO of Akanda, introduces OpenStack's newest project, an open-source network orchestration platform built by OpenStack operators for



How developers from four continents delivered an OpenStack feature



The future of OpenStack and Kubernetes



How Burton Snowboards is carving down the OpenStack trail



effective ways to work upstream, which is beneficial for both new contributors as well as existing contributors. At the end of the day, a project like OpenStack Neutron involves people from across a wide spectrum of companies and interests. As the PTL, it's my job to lead these people to a common goal. Growing the base of contributors is part of this responsibility as well. I look forward to seeing new contributors in Neutron and OpenStack in general!



FACEBOOK



TWITTER



GOOGLE +



LINKEDIN



SUPERUSER AWARDS

Superuser Awards nominations open for OpenStack Summit



Tips for getting a travel grant to the next OpenStack Summit



Inside WalmartLabs
and its OpenStack
core



European public
clouds take hold



Innovate,
collaborate,
replicate



Working to bridge
the developer/end-
user divide in
OpenStack



How to introduce
OpenStack in your
organization

SUPERUSER

[User Stories](#)[How-To](#)[Superuser TV](#)[Awards](#)[About](#)

COMMUNITY

[User Groups](#)[Events](#)[Jobs](#)[Contribute](#)

USER RESOURCES

[OpenStack Marketplace](#)[OpenStack Training Courses](#)[OpenStack Documentation](#)[OpenStack Operations Guide](#)[Application Developer Resources](#)[OpenStack Security Guide](#)[OpenStack Questions](#)

OpenStack Superuser is a publication built to chronicle the work of superusers, and their many accomplishments personally, professionally, and organizationally. The emphasis is on a blend of original journalism and user-generated content, ranging from technical to business-level issues with feature stories, case studies, tips and videos for OpenStack cloud architects and administrators.