Building Open Source Project Infrastructures

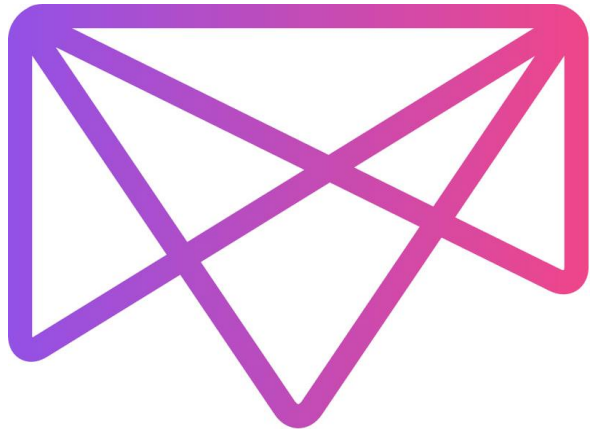
Open Community Conference
@ Open Source Summit
North America 2017

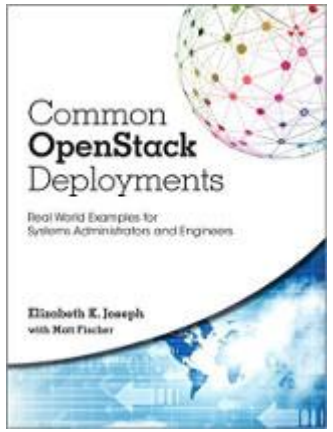
Elizabeth K. Joseph
@pleia2

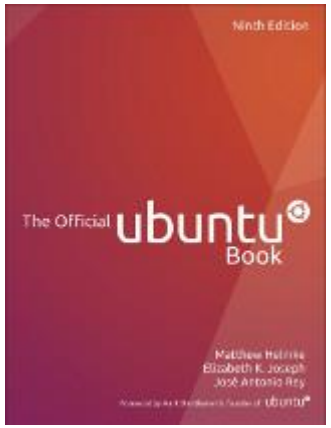
MESOSPHERE

Elizabeth K. Joseph, Developer Advocate

- Developer Advocate at Mesosphere
- 15+ years working in open source communities
- 10+ years in Linux systems administration and engineering roles
- Founder of OpenSourceInfra.org
- Author of The Official Ubuntu Book and Common OpenStack Deployments







Thank You

- FreeCode (Freshmeat)
- SourceForge
- Google Code
- GitHub

Benefits of Proprietary Hosting

- Low administrative overhead
- Familiar workflow for contributors
- Ecosystem of other tools that hook into it

But you also have...

- Vendor lock-in
- Potential for costs as your project grows
- Lack of flexibility

There is another way: Build an open source infrastructure.

Definition time!

What’s an open source infrastructure?

An infrastructure that is, in whole or in part, managed in the open by a team of community members.

Who is doing it?

debian

openstack.

fedora

OSL
OPEN SOURCE LAB

THE APACHE® SOFTWARE FOUNDATION

python SOFTWARE FOUNDATION

ubuntu

GlusterFS

moz://a

GNOME™

Jenkins

eclipse

What to open source in your infrastructure?

Everything! But that may be a bit ambitious. Start out with:

- Documentation
- Project website
- Continuous integration system
- Build system
- Services that developers interact with: chat bots, wikis, planet, ticketing system...

What you get: Documentation

Most projects are doing this already, well done! But if you're not…

- Drive-by contributions from users who found and fixed a piece of your documentation
- Documentation is often the first way a user interacts with your project, makes it welcoming and collaborative from the start
- Sense of ownership by your community instead of a black box that only special people have access to

What you get: Project website

Even if the specific web server configuration isn’t open source, open source the site.

- Help from less invested contributors regarding changes that can improve it and fix bugs, particularly across browsers and devices
- Community ownership of a resource that has historically been static and incredibly restricted to a chosen few

What you get: Continuous integration system

When you are running a CI system, it’s essential the developers know what’s being tested

- Creates a sense of ownership by developers in the QA pipeline
- Causes developers to be mindful about writing tests for their own cde
- Allows the community of users to provide input into what tests are important to how they’re using the system

What you get: Build system

How your software is being built matters

- Allows users and developers alike to recompile your code with the same environment that your project uses, reducing potential for further bugs
- Ability to more easily work with projects advocating for things like a series of best practices for Reproducible Builds https://reproducible-builds.org/

What you get: Services that developers interact with

- Allows for your project to address bugs and feature requests
- Empowering contributors to improve existing tooling, and fix things when they break or are not functioning optimally
- Gives your community the ability to add tooling within your project infrastructure when they see a need, which increases their sense of ownership

Getting started

1. Talent
2. Hosting
3. Funding

Step 1: Talent

- Systems administrators exist in open source communities (I’m one of them!)
- It can be difficult to break into the practice of systems administration, there are people eager to participate and learn
- Create tasks for your budding administrators
  - Evaluate appropriate open source software for the project
  - Develop maintenance plan
  - Create a ticket queue of administrative tasks
- Maintain a focus on *maintainability*, *openness* and *documentation* so future administrators can learn quickly

Step 2: Hosting

Once you have determined what you wish to host for your project, evaluate hosting options:

- Will a hosted version of an open source platform work for you? (GitLab, Jenkins)
- Do you need a full Virtual Private Server?
- Where will you host your code? (both project and infrastructure)

Step 3: Funding

- Ask for resources from companies you know are using your project
- Talk to your employer, and encourage fellow project participants to do the same
- Reach out to infrastructure vendors who have been friendly to open source projects in the past
- If you're OK with using hosted open source software, research and reach out to them about their support of open source projects

Fiscal sponsorship and other options

Non-profit status is frequently not required to accept in-kind contributions, but if it is:

- Work with an existing non-profit in your space for fiscal sponsorship
- Talk with tangential projects and your community about joining their organization
- Apply to join the Software Freedom Conservancy https://sfconservancy.org/
- Apply to join the Apache Software Foundation http://apache.org/

Funding tips from Cat Allman, Google OSPO

Scope your needs

Do a project budget

Set your goals: quarterly, annual, long term

Develop materials

Have a cushion

Decide what you're willing to sell

Consider your resources

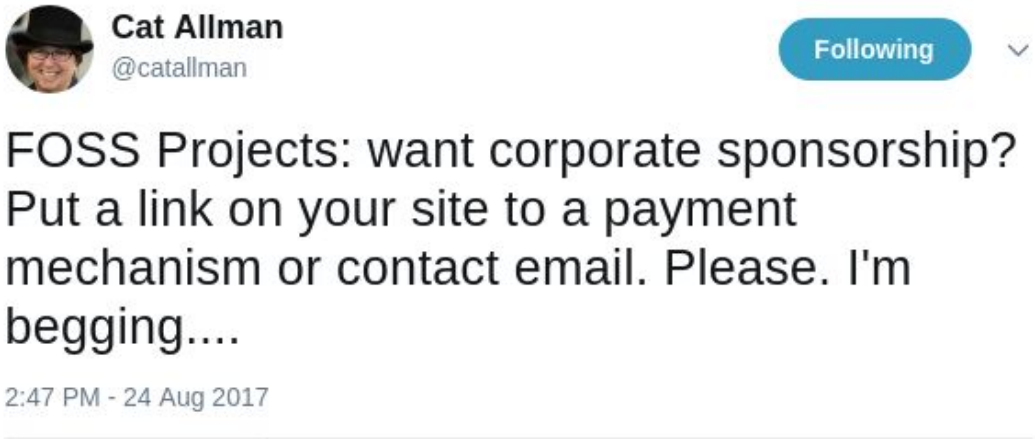
Assign responsibilities

Prearrange how the money will be accepted and managed

Decide on record keeping and activities tracker

And more in Fundraising 101 (or "Free as in Freedom So Who Pays for the Beer?"), OSCON 2011, Cat Allman
https://conferences.oreilly.com/oscon/oscon2011/public/schedule/detail/18886

Some Final Wisdom from Cat



Case Study

OpenStack

OpenStack: Open sourced resources

- Continuous Integration System (Gerrit, Jenkins, Zuul, Nodepool)
- Git cluster (git server and cgit)
- Metrics (Cacti, Grafana)
- ELK stack
- Mailing lists, chat bots
- Wiki, Etherpad
- And much more at https://docs.openstack.org/infra/system-config/

Case Study

Ubuntu and Xubuntu

Ubuntu: Open sourced resources

- Launchpad.net: code hosting, mailing lists, SSO, etc
- Configuration for various community projects hosted, including WordPress themes, proxies, Drupal: https://code.launchpad.net/~canonical-sysadmins
- A collection of open source Juju Charms which are also used internally
- Continuous Delivery system for Juju: https://mojo.canonical.com/

Xubuntu: Open sourced resources

- Collaboratively maintained server at https://dev.xubuntu.org/
- Fully open sourced WordPress theme so that the site can be replicated
- Collaboratively maintained documentation, along with releasing of that documentation

@dcos

chat.dcos.io

users@dcos.io

/dcos
/dcos/examples
/dcos/demos

Questions?

Elizabeth K. Joseph
Twitter: @pleia2
Email: lyz@princessleia.com