



Contributing to Docker 101
(an external maintainer's view)

Phil Estes

IBM Cloud, Open Technology

Twitter: @estesp

About Me

Phil Estes

Senior Technical Staff Member
IBM Cloud, Open Technology & Strategy
[@estesp](#)

- > Docker core engine maintainer
- > Member of “Docker Captains” program
- > 10+ years involved in Linux/OSS

Community accomplishments:

- > User namespace support in the Docker engine
- > Helped design v2.2 image spec with multi-platform support
- > Implemented first tool to create multi-platform images in Docker v2.3 registry & DockerHub

The Fine Print

- **Contributing is more than code**
 - But, my **experience** is with code contributions and will lean heavily towards that aspect
- **The overall topic is quite broad**
 - So I'll focus on what I have (and will) personally share with an interested developer looking to contribute to a Docker project
- **There are personal aspects to workflow/focus areas/etc.**
 - So please don't blame me if your experiences don't match mine :)
- **There is a TL;DR if you find me {unconvincing,boring}!**
 - <https://github.com/docker/docker/blob/master/CONTRIBUTING.md>

Where do I Start?

(the basics)

The Basics before the Basics

- **GitHub account**
 - Set up and use TFA (two-factor authentication)
 - Add an SSH key
- **A `git` workspace**
 - Fork repos you plan to contribute to (GitHub UI)
 - Clone them on your local system for development
- **IRC client**
 - Join #docker-dev, listen to #docker-maintainers, join other project channels as interested (swarm, network, distribution)
 - Interact with other contributors and maintainers!
- **Go through the contributor's guide!**
 - <https://docs.docker.com/opensource/code/>
-

Good and Bad

(are some contributors better than others?)

Communication



GOOD

- Find and **discuss** large changes with the right maintainers
- **Actively ask** for input/**feedback**
- Respond in a **timely** manner
- **Join** the team on IRC



BAD

- **Silent** (large) PR submissions
- **Lack of response** to questions/feedback
- **Never** get on IRC and build **relationships**

Tone

✓ GOOD

- It's **not** personal
- If you are taking the response personally, a) **breathe**, b) ask for **clarification**
- Patience with busy maintainers!
- Understand “close” is **not** a personal **rejection**

✗ BAD

- Knee-jerk/**reactive** response to feedback
- **Lack** of patience to maintain “burden”
- **Ignore** maintainer **workflow**
- Respond **negatively** to a “close” response

Learn

✓ GOOD

- Read **other** PRs; **learn** what works/fails
- Follow IRC; see what's being **discussed**
- **Ask for help!**
- Know you aren't a Go language **expert** (unless you are); see what you can **learn**

✗ BAD

- “**Heads down**” mode; never look around
- **Ignore** the **direction** of the project
- **Assume** you know **everything**
- Consider yourself having **mastered** Go completely

Practice

✓ GOOD

- **Sharpen** your skills; understand what's “**idiomatic**” vs. not
- Add **tests!**
- Help **improve** docs
- **Improve** the codebase even if is **not related** to what you want to work on

✗ BAD

- Ignore that idiomatic Go is an **expectation**
- Add **function** with **no tests**
- **Hope** that someone else works on docs
- **Ignore** the codebase **outside** your interests

What can I do?

(find your niche)

Finding Work to Do

- **Search repo for beginner/easy issues**
 - Dependent on project's use of tags
 - Use #dibs to mark working
- **Ask on IRC**
 - Ask for help finding something to hack on
- **What is your area of expertise?**
 - Are you a networking expert? Do you know all about Linux namespace internals?
 - Learn the code for that area of the Docker ecosystem; what would you add/fix/help improve?
- **Look for roadmap items (and ask about ownership)**

Closing Thoughts

(A Lifetime Channel heartwarming episode
of “My Story: A Docker Maintainer”)

Two Years of docker / docker

- **Community is Extremely Welcoming**
 - Have **never** seen a new contributor treated poorly
 - Plenty of guidance for those that are interested in learning
- **No one Agrees on Everything**
 - It is not “Docker, Inc.” versus “everyone else”. Maintainers have personal perspectives, and Docker walks a tightrope between OSS and products
- **Docker is an Opinionated Container Ecosystem**
 - Developer-focused, developer experience has high precedence
 - Focus on UI simplicity/clarity means long discussions on UX changes
- **Docker OSS Projects are Imperfect**
 - Just like every other open source project
 - But it's a great one!

How do I Become a Maintainer?

- **Make sure you really want it!**
 - Maintainer role is a change of focus (away from pure contribution)
 - Comes with an expectation/burden; read this document first: <https://github.com/docker/opensource/blob/master/MAINTAINERS>
- **Become known**
 - A maintainer is a trusted position; you can't be a maintainer if the current maintainers don't know you (see "IRC")
- **Become known for working outside of your own PRs**
 - Your "LGTM" is not valid, but you can definitely provide helpful comments/feedback. This will be noticed!
- **Help out!**
 - Triage issues; debug problems

Thank you!

 @estesp

 github.com/estesp

 estesp@gmail.com

 <https://integratedcode.us>

 IRC: estesp

