HOW TO CONTRIBUTE TO OPEN SOURCE

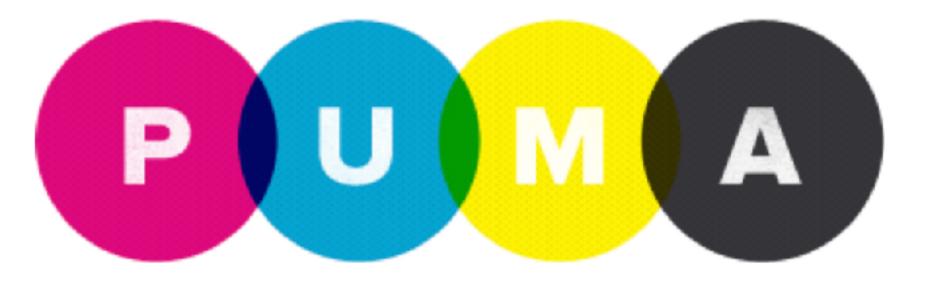
What maintaining a project with 280 million+ downloads taught me

Nate Berkopec for Tokyo Tech Meetup, Feb 2023





7,308



260,830,673

1,165,330

Puma: A Ruby Web Server Built For Parallelism



Puma is a simple, fast, multi-threaded, and highly parallel HTTP 1.1 server for Ruby/Rack applications.

Built For Speed & Parallelism

Puma processes requests using a C-optimized Ragel extension (inherited from Mongrel) that provides fast, accurate HTTP 1.1 protocol parsing in a portable way. Puma then serves the request using a thread pool. Each request is served in a separate thread, so truly parallel Ruby implementations (JRuby, Rubinius) will use all available CPU cores.

Originally designed as a server for Rubinius, Puma also works well with Ruby (MRI) and JRuby.

On MRI, there is a Global VM Lock (GVL) that ensures only one thread can run Ruby code at a time. But if you're doing a lot of blocking IO (such as HTTP calls to external APIs like Twitter), Puma still improves MRI's throughput by allowing IO waiting to be done in parallel.



Open source contribution is an excellent way to learn, usually from principallevel engineers



Midjourney: "16-bit puma cat video game character sprite"





A project you should contribute to will:

- Have regular releases (every 2-3 months)
 Actively seek new
 - 2. Actively seek new contributors (how many do they have?)
- 3. Be something you already use every week

CONTRIBUTING.md



Get the test suite working and passing

- 1. git clone 2. brew install fancy-stuff
- 3. bundle exec rake





It's not just about pushing new code.



Midjourney: "a puma cat cartoon character, chibi, kawaii, cute, sanrio style, low detail, anime, hacker, computers, programmer"

Clarifying the "next action" in an issue or PR is often more helpful than actually coding.



contrib-wanted label



needs-repro label



docs label

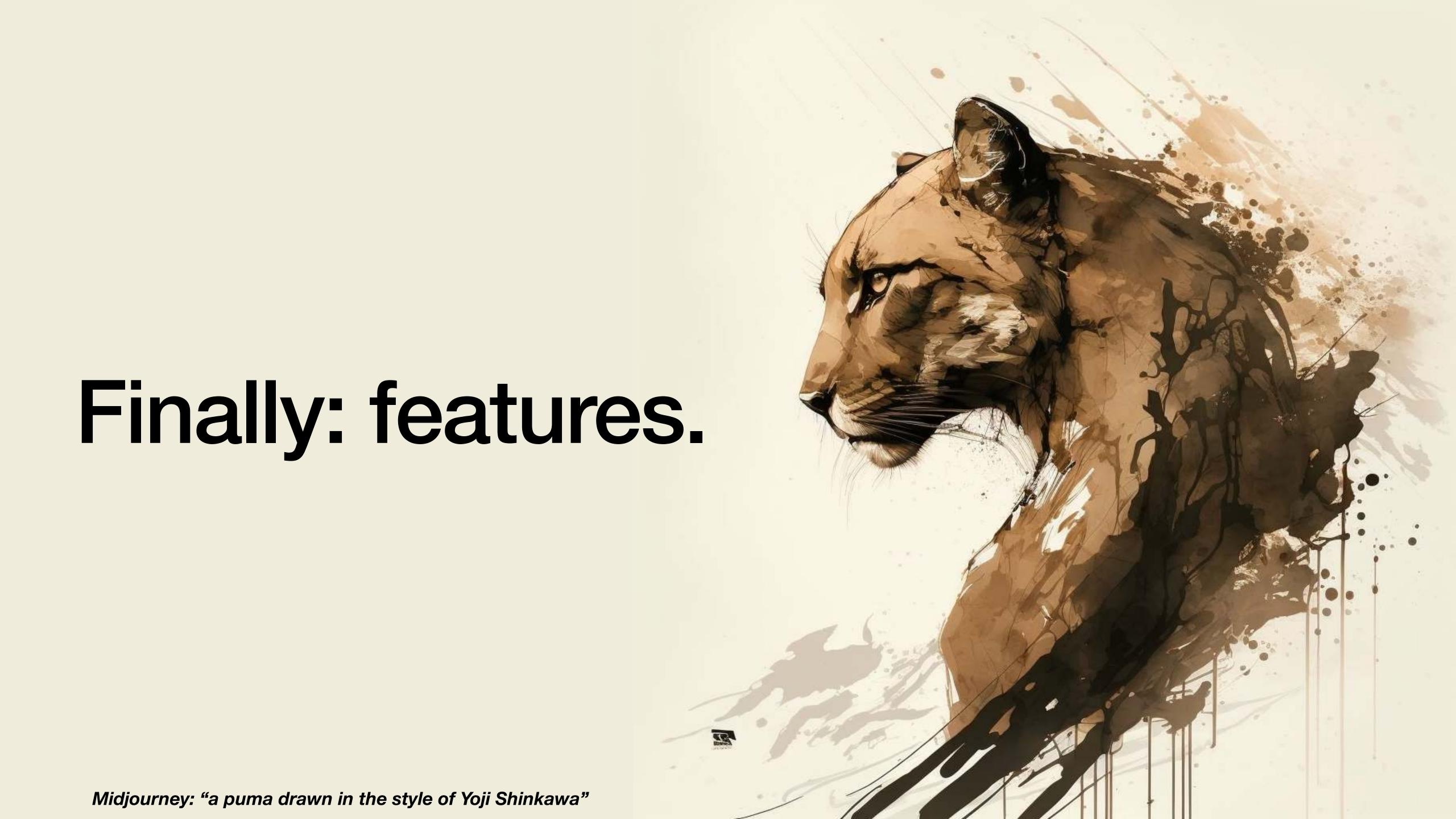


Code review isn't just for "maintainers"



You can fix bugs or backport bug fixes and features.





To start contributing to open source:

- 1. Don't worry about how much you know. You can learn while you contribute!
- 2. Pick the right project for you
- 3. Start contributing by:
 - 1. Reproducing bugs
 - 2. Adding documentation
 - 3. Reviewing PRs
 - 4. Fixing bugs, backport5. Write new features



spedshop

@nateberkopec https://speedshop.co