Why

Why? Why self-host all your own infrastructure?

We tried not to, at the very beginning of the project. We agreed that Github-style code review wasn't really fit for our kind of project, and wanted to use Gerrit for code review. We started setting up Gerrithub for a repo hosted on Github, and we ran into so many problems with that approach that it was actually easier to just self-host Gerrit instead (for starters, some members could not log in at all).

Then there was little reason to use GitHub, since none of us are really happy with Github direction lately anyway, and Forgejo + GitHub-enabled SSO mean that contributors shouldn't have to jump through too many hoops to help out.

So now we have a fully independent and open source infrastructure stack, with (hopefully) a good onboarding path as well. And we're also in our own critical path: we run into Nix's papercuts and gashes alike every day, so we better fix it!

Here is our thought process from back then:

- GitHub?
  - Well-known
  - Easy to contribute to as everyone has an account
  - CI would need extra work due to VM tests, e.g.
    - Not a Nix CI system by itself, will be doing self-hosted engineering work for something extra anyway
  - Rubbish code review
  - Can use GerritHub for better review?
    - CI though?
    - Evaluated: found that login does not work and vendor is unresponsive to emails
    - Possible moderation difficulties
- Gitlab/Codeberg?
  - Basically GitHub but by different people
  - Same CI problems as GitHub but fewer off-the-shelf solutions to them
  - Rubbish code review
    - Cannot use GerritHub even assuming that it worked
    - Need self-hosted Gerrit, so need self-hosted Gerrit *auth* and well...
- (self-hosted) forgejo/gitlab
  - Much better control over integrations

    - Same poor code review situation as cloud Forgejo/GitLab
        - Might as well stand up a forgejo if we already have self-hosted Gerrit and the auth for it
- (self-hosted) Gerrit?
    - Easy to integrate with and add plugins to vs GerritHub (e.g. Gerrit does not come with Nix or Meson syntax highlighting)
    - Full control: can fix issues like moderation
    - Works at all
    - Good code review

The Lix CI is broken though!

Yes, our buildbot is a high maintenance service and it *is* janky. Multiple members of the Lix team have plans about writing entirely new Nix CI systems, but they are otherwise busy with another major project in the form of Lix. This is the matrix of extant alternatives:

- buildbot
    - buildbot-nix exists and we have hacked it to use nix-eval-jobs and speak Gerrit with (limited) caveats
        - Logs aren't intermixed
        - Our Gerrit integration is considerably janky auth-wise
        - There *are* bugs
        - UI makes it very non-discoverable which CL caused a build
    - The old Angular UI is mildly busted, and the new React UI has severe accessibility problems and is more busted
    - Knows how to speak Gerrit
- Hydra
    - Due to architectural flaws, it cannot deliver notifications of status to a code review system
        - This is a non-starter
        - Cannot be taught to speak Gerrit because it cannot speak to code review systems
    - Notoriously dubious code and DB schema
    - Logs aren't intermixed
    - Dubious UI
    - Unmaintained for non-hydra.nixos.org use cases (and hydra.nixos.org would rather not use it, but the infra team does not have the cycles to rewrite Hydra)
- Something GitHub Actions based
    - Reinstalls Nix every job
    - Needs a separate binary cache
    - Cloud runners are slow and can't run VM tests, need self-hosted runners
    - Need to solve `nix-eval-jobs` shaped problems
    - All the logs go into one stream, rather than being per-derivation, so error reporting is challenged
    - Tied to GitHub, impractical to attach to Gerrit

- Garnix
  - Tied to GitHub, impractical to attach to Gerrit
  - Unclear if it can run VM tests
  - Closed source, not observable
- Woodpecker
  - Build your own Nix CI!
  - FIXME: add more about this

Revision #5
Created 2024-04-09 12:54:22 UTC by Qyriad
Updated 2024-04-09 13:41:11 UTC by jade