Forgejo improvements

A brief overview of our code infrastructure for those not in the Lix project:

- Forgejo https://git.lix.systems for issue tracking for Lix and other projects, as well as for project boards. It also provides a read-only git mirror of Lix, forks for WIP code in Lix, and hosts minor projects like infra and other things that use PR workflows.
- Bookstack https://wiki.lix.systems as wiki. It's a more normal wiki program, which has working search among other niceties. Also, it presents the whole wiki as one entity.
- Gerrit https://gerrit.lix.systems for major code review. We have seen the Forgejo gerrit-like patch workflow and it looks great, but the review UX is not at all like Gerrit and makes way different prioritization of space.
  Not being like Gerrit is largely fine for Forgejo's goals, since it is not aiming to be Gerrit-like, it is aiming to be GitHub like (and GitHub has quite bad review UX); we would suggest supporting Change-Ids to have unambiguous change associations, but we probably won't use that feature anyway as we have Gerrit available.
- Buildbot https://buildbot.lix.systems (private link) for CI. We need something quite laser focused on being good at CI'ing Nix expressions and buildbot-nix is generally the least broken one of those.
- Keycloak https://identity.lix.systems for SSO. It is comprehensive across our entire infrastructure, and it is the *only* login system in our infrastructure. We use GitHub as an upstream for anti-spam + moderation reasons and additionally issue a limited number of local accounts on Keycloak.

Stuff that works great

Forgejo does a lot of stuff better than GitHub and we love it very much for these things.

- The web interface is snappy!
- Syncing groups from Keycloak is a breeze.
- We can patch it. Gosh. This helps so much.
- We only rarely run into bugs.
- The devs have generally been quite responsive!
- The release notes are generally comprehensive and cover the issues we run into on upgrades (such as the theme rename and the UTF8 column type thing in mysql).

Stuff that makes us Very Sad

- Forgejo is a major source of sticky user metadata
  https://codeberg.org/forgejo/forgejo/issues/3657 and

https://codeberg.org/forgejo/forgejo/issues/3682. This requires admin intervention if the username *needs* to be changed, and causes confusion in administering the system in general, since usernames don't match with Keycloak so you have to find users in Keycloak by ID. We want selectable auto-synced user metadata from SSO combined with locking such synced metadata out on the Forgejo side to prevent confusion.

See also: https://wiki.lix.systems/link/11#bkmrk-git.lix.systems

- A mild sadness, but it is not great: You can't make branch protection rules that forbid all human pushes yet allow force pushes; branch protection always implies no force pushes

  https://codeberg.org/forgejo/forgejo/issues/7238

Stuff that would be Nice

UX

- Top simple request, which we are quite likely to do ourselves: something like Gerrit commentlinks: https://gerrit-review.googlesource.com/Documentation/config-gerrit.html#commentlink

  That is, arbitrary regexes that can inject links into the page, which are applied in comments and in commit messages. This use case is not served for us by the external issue tracker feature of Forgejo, since we want to link stuff like `Change-Id: I1360750d4181ce1ca2a3aa4dc0e97e131351c469`, which is *not* an external issue, it is external code review. Also, we want to have more than one of them, for things like `cl/123` which should go to the Gerrit changelist 123.
  Gerrit has these either installation or repo scoped, and we use both on our installation: Change-Id and cl/ gets linked everywhere since it is global, and issue refs like `#123` get linked just on the Lix project since it is project specific.
  Also, I think that the external issue tracker feature might not work well for its intended use case anyhow? On GitHub you can have autolinks of TEAMONE-123 and TEAMTWO-123 and I don't know if Forgejo knows how to do that?

  Hacked in with an evil local patch: https://gist.github.com/lf-/f2e31a329c3c48f09198c865e21618e6
- Ability to put arbitrary links at the top of repos next to Issues, etc.
  We want this for linking to our Gerrit reviews, so it shouldn't be called Pull Requests, but idk there might be another way to do this that's better.

  Hacked in with an evil local patch: https://gist.github.com/lf-/f2e31a329c3c48f09198c865e21618e6
- jade's personal complaint: I can't hit `t` to fuzzy-find files like I can on GitHub!
- jade's other personal complaint: issue comment fields aren't saved across page changes, and they *also* can get desynced such that the comment was posted but you get a warning on navigating off of the page.
- Issue categories display differently if you have a / in them if they are exclusive or not. This is probably on purpose, but it does make our issue labels look a little bit funny when they

are all hierarchical, just not all of the categories are exclusive.
- We would like to be able to replace the explore page with the Lix organization page, since we don't really want people enumerating our users, and it is the only thing that really matters significantly on our instance.

Operations

- Easier management of bot accounts, perhaps by attaching them to an org or something; we would like to issue tokens for bot use cases at an org level rather than having to make weird local accounts and put them in a password manager.
- Git operations are strangely slow on our instance, taking about 5 seconds to push on seemingly every repo. We aren't 100% sure why this is, and don't know how to profile it. The web interface is snappy.
- We have a hacky prototype to use SCIP code databases to provide perfect go-to-definition in the code browser, but we haven't operationalized it with CI builds etc. It would be nice if Forgejo knew how to handle this natively.
  - puck> A big issue with this is that this would preferably hook into the syntax highlighter, to ensure spans don't pass SCIP token boundaries. The prototype I put together does some trickery to try and figure out exactly which character you clicked on. This means it works terribly on mobile due to click target sizes. (And it's missing a nice popup.)
- You can't move issues between repos.
- Repo-scoped project boards are somewhat useless since you can't include issues from multiple repos. I am not sure if you can move repo-scoped boards to org-scoped either.
- It isn't possible to pause the mail queue by setting workers to 0. Or to flush the mail queue when the mailer is disabled.
  The reason we are writing this is that there was such an operational incident where we were about to send 900 emails that nobody wanted because of an issue-copying script (which later had a dont_notify API param added to not try to send the emails in the first place). I think we deleted the queue directory or broke the mail config on purpose and then hit the flush button or something.
  We have worked around this ourselves but we wanted to mention it.
- We have noticed that moderation tools are somewhat lacking in Forgejo compared to GitHub. For example you can't lock non-member participation in issues if some heated external event happens (but we would probably just hack our Keycloak allow/ban plugin to reject registrations).
  - Although it's not something we need as we can do it with SSO+dumping the session cache, I don't think you can ban people on Forgejo?
  - We would like to be able to time people out
- We can't easily redact/replace commit IDs from the activity log since they are stored as HTML. The purpose of this is to be able to perform Git history rewriting in case we want to wipe someone's deadname from history and fix it. This is not urgent, as in, we don't know when we will need this, but it will plausibly happen one day; I think we were considering writing tooling to fix it up in the DB though.

    - puck> The event history of a repository will keep old commits IDs, and their summaries, even if the commit no longer exists.
    - puck> The tooling to rewrite commit IDs has been written, but is a massive hack (see how to use it): it takes in a `mysqldump` and rewrites any mention of the commit ID. It works, though ^_^
    - puck> This might be a big ask, but it'd be great to have an equivalent to Gerrit's "banned commits" - commit hashes that, if encountered on any push, reject the push instantly.
- Teams don't have a check box to auto-add them to new public repos. We use teams to grant wide reaching issue triage permissions on everything (this works *fantastically* for us as a community, and nixpkgs also does this), but we have some private repos we don't wish to grant access on, so we had to set the team to not all repos.
- There are no org-scoped milestones, which is somewhat annoying as we are using forgejo for *planning*-related issues that aren't in the Lix repo itself, but do block release. We have used projects for now and it's generally fine.
- It isn't possible to continuously mirror new issues on a repo from GitHub.
    - Relatedly, it is not possible to *limit creating new issues on a repo* and thus messing up the numbering of such a mirror
    - We don't need this that bad since we seem to be going on a very-hard fork course of not worrying about the upstream issue tracker anymore.

Stuff we patched that could probably be done Better upstream

- We added a Nix tarball link feature, which is now upstream! □□
- In routers/web/auth/auth.go we have patched a redirect in to just unconditionally go to OAuth2 when you hit the login button. This is kind of a hack, and it prevents logging in with local accounts, which we have an extremely limited use of for a bot account. Probably the good implementation here is this but with a `?local` url parameter or so.
- We have an issue mirror of nixos/nix on GitHub (which is likely to be deprecated soon, to be fair! so don't worry too much about it) and we did this to add a button to get to upstream issues; in general adding navigation in the tool would be good:

```
diff --git a/templates/repo/issue/view_title.tmpl
b/templates/repo/issue/view_title.tmpl
index c1dd265..9a573ce 100644
--- a/templates/repo/issue/view_title.tmpl
+++ b/templates/repo/issue/view_title.tmpl
@@ -16,6 +16,9 @@
 □□□{{if and (or .HasIssuesOrPullsWritePermission .IsIssuePoster) (not
.Repository.IsArchived)}}
 □□□□<button id="edit-title" class="ui small basic button edit-button not-in-edit{{if
```

```
.Issue.IsPull}} gt-mr-0{{end}}">{{.locale.Tr "repo.issues.edit"}}</button>
 □□□{{end}}
+□□□{{if and (eq .Repository.Name "nix") (eq .Repository.OwnerName "NixOS")}}
+□□□□<a role="button" class="ui small basic button"
href="https://github.com/NixOS/nix/issues/{{.Issue.Index}}">{{svg "octicon-mark-
github"}} Open upstream</a>
+□□□{{end}}
 □□□{{if not .Issue.IsPull}}
 □□□□<a role="button" class="ui small green button new-issue-button gt-mr-0"
href="{{.RepoLink}}/issues/new{{if
.NewIssueChooseTemplate}}/choose{{end}}">{{.locale.Tr "repo.issues.new"}}</a>
 □□□{{end}}
```

- We have an extremely scuffed patch to add a `dont_notify` parameter on the issue creation endpoint. This is so you can build automation that creates a giant pile of issues that you want in the tracker but which don't need to notify anyone.

Revision #6
Created 2024-05-12 09:03:22 UTC by jade
Updated 2026-09-21 16:45:56 UTC by jade