

Lix organisation

Information related to the Lix project organisation.

- [Code of Conduct](#)
- [Community Standards](#)
- [Matrix rooms](#)
- [Related projects](#)

Code of Conduct

Everyone in the Lix community is expected to abide by the [Lix Community Standards](#).

Community Standards

Moved to <https://lix.systems/community-standards/>

Matrix rooms

Since Matrix is haunted sometimes, here is a list of rooms in the public Lix space:

- [Public space #space:lix.systems](#)
- [#community-team:lix.systems](#), for moderation concerns
- [#announcements:lix.systems](#), for announcements relating to the Lix beta
- [#discuss:lix.systems](#), for general discussion
- [#dev:lix.systems](#), for development discussion
- [#open-beta:lix.systems](#), for running HEAD

Related projects

This page lists community projects that are built with, optimized for, or make specific use of Lix features. While these projects are not officially affiliated with the Lix project, we maintain this curated list to highlight tools, systems, and initiatives that align well with Lix's design goals and ecosystem.

The projects featured here often explore advanced or experimental use cases, provide integrations, or offer enhancements that take advantage of Lix-specific capabilities. We aim to keep this page up to date as the ecosystem evolves.

If you maintain or know of a project that fits this description, feel free to suggest it for inclusion.

CI automation

- [lix-gha-installer-action](#)
- [nix-build-action](#) – defaults to the Lix GHA installer
- [lix-quick-install-action](#) – based on <https://github.com/nixbuild/nix-quick-install-action>

Lix sub-commands

This requires the `lix-custom-sub-commands` experimental feature as of May 2025.

- [lix-diff](#) similar to <https://khumba.net/projects/nvd/>

Misc

- [Xil](#), an experimental alternate CLI for Nix implementations by Qyriad, which will potentially slowly merge with the Lix CLI.