

services-flake

[services-flake](#) provides declarative, composable, and reproducible services for [Nix development environment](#), and is based on [flake-parts](#). Enabling users to have NixOS-like service on macOS and Linux.



Links to this page

[services-flake: Services simplified for Dev/CI workflows](#)

Shivaraj B H [will demonstrate](#) using services-flake to declaratively manage service dependencies (for example, databases) of your project in the Nix development environment and re-use the same configuration for testing in your CI workflow. This talk builds upon my lightning talk from NixCon 2024 [dive deeper](#) into the topic with additional insights.

[Replacing docker-compose with Nix for development](#)

How does services-flake solve them?

Now that we have all the answers, it's time to replace docker-compose in NY [with services-flake](#). We will only focus on a few services to keep it simple; for more details, refer to the PR [PR](#).

Sharing services-flake modules deserves a separate post, so we will delve into this topic more in the next post.

These were the exact problems services-flake was designed to solve. Along with running services natively, it also integrates with your project's `flake.nix` [flake.nix](#).

