ChatGPT

1. Pixi (prefix-dev/pixi) – Primary: Pixi (Nix-like package manager). Last Update: Active as of late 2025 (v0.49 in Jun 2025) [1]. Use Case: Robotics and reproducible dev environments. Pixi provides fast, cross-platform (Linux, macOS, Windows) reproducible environments for robotics (e.g. ROS 2), without Docker or OS lock-in [2]. It combines Nix-style guarantees with the Conda ecosystem, using a simple `pixi.toml` to pin dependencies and define tasks. Platforms: Laptop/desktop (all OS) and CI; used for ROS 2 development on Windows (endorsed by Open Robotics) and multi-OS projects [3]. Noteworthy: Supports multi-environment workspaces (e.g. CPU vs CUDA), lockfiles for consistency, and CLI task runner (`pixi run`). It enables sharing identical dev setups across teammates and CI pipelines [3], making ROS/robotics projects *reproducible, portable, and accessible* [2].

2. Botnix (nervosys/Botnix) – Primary: NixOS (custom distro). Last Update: Active development in 2025 (pre-1.0 release) [4]. Use Case: *Agentic robotics OS* – a Linux distribution for multi-agent autonomous systems and AI-enabled robots. Built on Nix's declarative model, Botnix aims to be a hardware-agnostic, production-grade OS for robotics [5]. Platforms: Intended for robots at all scales (edge devices to larger systems). Noteworthy: Provides specialized "flavors" (modes) like *Agent*, *Orchestrator*, *Simulator*, etc., each with tailored NixOS modules [6] [7]. It includes a curated "Botpkgs" package set (a domain-specific subset of Nixpkgs for robotics) and emphasizes multi-host and AI workflow integration [5]. *(Stars: low (~30) but project is new and ambitious.)*

3. Robotnix (nix-community/robotnix) – Primary: NixOS Flakes. Last Update: Dec 27, 2025 [8]. Use Case: *Reproducible Android builds* – A declarative, Nix-based build system for Android (AOSP) images [8]. Platforms: Mobile devices (builds device-targeted Android/LineageOS/GrapheneOS images), with build environments on Linux. Noteworthy: Highly advanced Nix flake configuration supporting multiple devices and flavors (e.g. LineageOS, GrapheneOS) and OTA updates [9]. Robotnix uses NixOS-like modules to allow enabling features (MicroG, F-Droid, etc.) and signing keys in a reproducible way [10]. It's popular (☆740) and recently revived by new maintainers, enabling hardware-agnostic Android builds that are consistent across developers and CI [8].

4. Mobile NixOS (mobile-nixos/mobile-nixos) – Primary: NixOS (Linux distro variant). Last Update: Nov 13, 2025 [11]. Use Case: *NixOS on smartphones* – A framework to run NixOS on mobile devices (e.g. PinePhone, Android-based phones) [11]. Platforms: Mobile/embedded hardware (ARM phones, tablets). Noteworthy: Provides device-specific bootstrapping (U-Boot, drivers) so the same NixOS configurations can be deployed on phones. Emphasizes hardware-agnostic configuration – you can manage a phone like any NixOS system. Mobile NixOS (☆1.2k) includes Home Manager and cross-compilation support to accommodate the mobile form factor, extending "install-anywhere" ideals to pocket devices [11].

5. Nix-on-Droid (nix-community/nix-on-droid) – Primary: Nix (user-land environment). Last Update: Dec 06, 2025 [12]. Use Case: *Portable Nix environments on Android* – Enables a full Nix package manager experience on Android devices without replacing Android [12]. Platforms: Android phones/tablets (runs alongside Android OS). Noteworthy: Uses a sandbox (proot) to create a Nix store on Android, allowing installation of Nix packages on an Android device [13]. This gives a reproducible, declarative environment on mobile – effectively an "install-anywhere" solution. Popular (☆1.9k) among developers, it supports cross-shell usage via Termux, so one can spawn a Nix shell on Android with the same packages and configuration as on a PC [12]. Great for edge computing or on-device builds.

6. deploy-rs – Primary: NixOS (deployment tool). Last Update: Dec 18, 2025 [14]. Use Case: *DevOps & distributed deployments* – A widely-used tool (☆1.9k) to reproducibly deploy NixOS configurations to many hosts via flakes [14]. Platforms: Clusters, servers, or heterogeneous fleets running NixOS (or using it for config). Noteworthy: Supports multi-host and multi-profile flakes – you can declaratively define deployments for say, a laptop, a Raspberry Pi, and a cloud VM in one flake, then push updates in one command. It is stateless and agentless (uses SSH), integrating well in CI/CD pipelines for infrastructure. Advanced patterns include "profiles" (e.g. staging vs production configs in one repo) and atomic upgrades on clusters. This makes deploy-rs a cornerstone for reproducible ops and consistent environments across diverse hardware [14].

7. **ryan4yin/nix-config – Primary: NixOS + Home Manager (personal config). Last Update: Jan 04, 2026 [15]. Use Case: *Cross-shell, cross-platform dev environments* – A top community dotfiles repository (☆1.7k) showcasing unified NixOS configurations for both macOS and NixOS systems [15]. Platforms: Developer laptops (macOS), desktops/servers (NixOS), with seamless consistency between them. Noteworthy: Demonstrates an advanced flake setup with nix-darwin (for macOS) and Home Manager, so the same config manages packages, OS settings, and dotfiles on multiple OSes. It explicitly targets multiple shells: for example, the author uses Nushell with Starship prompt as the shell environment [16], indicating the config is Nushell-compatible. The repo employs multi-host flake outputs and shared modules for laptop vs homelab, showing how to maintain portable workflows (editor, tools, environment variables) across heterogeneous systems [15]. This curated config is often referenced as a template for reproducible dev setups that work anywhere.**

Sources: GitHub repositories and documentation for the projects mentioned, as well as curated community lists and blogs for star counts, recency, and feature highlights [2] [5] [8] [11] [12] [14] [15]. Each resource is chosen for its popularity and recent updates in demonstrating advanced NixOS/Pixi configurations in the domains of robotics, DevOps, ubiquitous deployment, and cross-shell developer environments.

---

[1] Tags · prefix-dev/pixi · GitHub
https://github.com/prefix-dev/pixi/tags

[2] [3] Pixi: Modern package management for Robotics | prefix.dev
https://prefix.dev/blog/reproducible-package-management-for-robotics

[4] [5] [6] [7] GitHub - nervosys/Botnix: The operating system for robotic general intelligence™
https://github.com/nervosys/Botnix

[8] [11] [14] [15] GitHub - tolkonepiu/best-of-nix: A ranked list of the best resources in the Nix community. Updated weekly.
https://github.com/tolkonepiu/best-of-nix

[9] [10] GitHub - nix-community/robotnix: Build Android (AOSP) using Nix [maintainer=@danielfullmer,@Atemu,@cyclic-pentane]
https://github.com/nix-community/robotnix

[12] [13] GitHub - nix-community/nix-on-droid: Nix-enabled environment for your Android device. [maintainers=@t184256,@Gerschtli]
https://github.com/nix-community/nix-on-droid

[16] GitHub - ryan4yin/nix-config: ❄️ My nix config for both desktops(NixOS+macOS) and homelab servers(NixOS).
https://github.com/ryan4yin/nix-config