


2025-09-03

Proposal Title

Annie Easley 

Lewis Research Center, National Aeronautics and Space Administration

Signatories

Project team

- Philipp Baumann
- Justin Bedő
- László Kupcsik
- Bruno Rodrigues

Contributors

- Philipp Baumann
- Justin Bedő
- László Kupcsik
- Bruno Rodrigues

Consulted

- Amanda L. Martin
- Josiah Parry

The Problem

Packaging R packages for different operating systems and architectures is a difficult and time-consuming task. Building binaries for various systems means dealing with different compiler settings, managing platform-specific dependencies, and ensuring everything works well across environments. This often leads to inconsistencies and build failures, especially when system-level dependencies are not clearly handled.

There have been efforts to ease this process. For example, organizations like Posit and community projects such as r2u have made progress in automating binary builds. However, these solutions have

some drawbacks. The Posit Public Package Manager does not manage system-level dependencies automatically, leaving that work to users, while r2u only packages the current state of CRAN, which means older package versions are not available.

Since R is used on many different operating systems and hardware platforms, solving these issues is essential. One solution that seemed to tackle these issues was to use Docker images in the build process. Docker can capture the entire software environment—including all necessary dependencies—so that R packages are built and run consistently on any system. This improves reproducibility and makes deployment easier. However, using Docker alone does not ensure full reproducibility, and its isolated containers can make interactive, explorative work more challenging.

Additionally, it can be tough to deploy the same environment across multiple computers, which is important for collaborative work and continuous integration. Addressing these challenges will benefit a wide community of R users by providing more consistent and reliable package installations, ultimately leading to better reproducibility and user experience.

The proposal

Overview

We propose to strengthen reproducibility for the R community by ensuring that all CRAN and Bioconductor packages are available as prebuilt binaries through Nix, a package manager with reproducibility as a design goal. While Nix already supports multiple package versions, per-project environments, and cross-platform compatibility, many R packages still require building from source. Our project addresses this gap by funding dedicated hardware that the NixOS Foundation will use to build and distribute R package binaries.

This effort will provide researchers, data scientists, and statisticians with reliable, reproducible environments, simplify collaborative workflows, and enhance continuous integration practices. By making this targeted investment, we deliver a concrete, practical improvement to the R ecosystem's reproducibility and stability, benefiting the community immediately without requiring major infrastructure changes.

Detail

Our proposal aims to extend the power of Nix, a package manager designed for reproducible builds, to fully support CRAN and Bioconductor for R. Nix already enables seamless installation of multiple package versions and per-project development environments across different operating systems and architectures. While CRAN and Bioconductor are partially available through Nix today, many packages still need to be compiled from source. Our goal is to cover the entire repositories—including historical versions—so that the R community can rely on complete reproducibility and long-term stability.

To achieve this, we will invest in dedicated hardware and produce clear documentation, lowering the barriers for R users and making reproducible environments more accessible. This initiative will benefit researchers, data scientists, and statisticians alike, while improving collaborative workflows and continuous integration practices across platforms.

The minimum viable product is straightforward. The NixOS Foundation already operates a build farm that compiles binaries for nearly all of the 120,000 packages available through Nix. Our proposal is to purchase additional hardware and donate it to the NixOS Foundation, which would then integrate it into their existing infrastructure to build R package binaries at scale.

We have already confirmed with the NixOS Foundation that they are open to this arrangement. The main question is whether we should purchase and donate the hardware directly, or transfer the funds to the NixOS Foundation so they can acquire the equipment themselves, which would be a simpler and more effective approach.

In short, this project delivers a practical, community-oriented solution to the persistent challenges of managing R packages across diverse environments. By ensuring consistency, reliability, and reproducibility, it strengthens the foundation for R's long-term growth and collaborative potential.

Project plan

A detailed, granular timeline for this project is challenging to define, as the core technical implementation (hardware procurement and integration) will be executed by the NixOS Foundation. Our role is to facilitate this process by securing the necessary agreement and funding. We have already established initial contact with key members of the NixOS Foundation, and they have agreed in principle to this collaboration. The timeline below is therefore a high-level estimate, contingent on their internal processes and hardware availability. However, we are confident that the project can be fully delivered by the end of 2026.

Start-up phase

(Q1 2026)

The initial phase focuses on formalizing the collaboration and preparing for the fund transfer.

- **Finalize Agreement:** Draft and sign a formal memorandum of understanding (MoU) with the NixOS Foundation. This document will outline the responsibilities of each party, confirm that the hardware will be dedicated to building R package binaries, and establish a framework for reporting.
- **Establish Communication Channels:** Set up a dedicated communication channel (e.g., a shared Matrix channel or regular video calls) between the project team and our contacts at the NixOS Foundation to ensure smooth coordination.
- **Reporting Framework:** Agree on a simple reporting mechanism where the NixOS Foundation can provide regular updates on hardware acquisition, integration, and build status. This will be used for our own tracking and for reporting to the ISC.
- **Funding Mechanism:** Finalize the decision on whether to purchase and donate the hardware directly or transfer the funds to the NixOS Foundation for them to acquire it. The latter is preferred for simplicity.

Technical delivery

(Q1 2026 - Q4 2026)

This phase covers the transfer of funds and the subsequent technical work performed by the NixOS Foundation. Our role will be one of monitoring.

- **Q1 2026: Funding Transfer:** Once the agreement is signed, the secured funds will be transferred to the NixOS Foundation.
- **Q2 2026: Hardware Procurement:** The NixOS Foundation will order and acquire the specified Linux server and Mac Minis. This timeline includes a buffer for potential supply chain delays.

- **Q3 2026: Hardware Integration:** The NixOS Foundation will install and integrate the new hardware into their existing build farm infrastructure (Hydra). This includes network setup, system configuration, and ensuring the new builders are recognized by their system.
- **Q3-Q4 2026: Initial Builds and Scaling:** The new builders will begin compiling R packages from CRAN and Bioconductor. The system will be monitored and scaled to work through the entire backlog of packages, aiming for comprehensive coverage.
- **End of 2026: Full Service Availability:** The build farm is fully operational, providing pre-compiled binaries for the vast majority of current CRAN and Bioconductor packages on x86_64-linux, aarch64-linux and aarch64-darwin. The project's primary technical goals are now met.

Other aspects

(Ongoing throughout 2026)

These activities will run in parallel to the technical delivery to ensure the R community is aware of the project and can benefit from its results.

- **Announcement Post (Q1 2026):** Once funding is secured and the agreement is in place, we will publish a blog post announcing the project to the R and Nix communities.
- **Progress Updates (Q2-Q3 2026):** We will provide periodic updates via blog posts or social media (e.g., Twitter/X, Mastodon, Bluesky) as major milestones are reached (e.g., hardware purchased, integration complete).
- **Delivery Blog Post (Q4 2026):** Upon successful completion and full service availability, a detailed blog post will be published. This post will serve as the official launch announcement and will include documentation on how R users can configure Nix to use the new binary cache.
- **useR! / Community Engagement (2027):** We plan to submit a talk or poster to a relevant conference (like useR! or NixCon) in the following year to share the project's success and encourage wider adoption.
- **ISC Meetings (Ongoing):** We will provide regular progress reports at ISC meetings as required, using the information from our agreed-upon reporting framework with the NixOS Foundation.

Requirements

People

The people involved are the current maintainers of the R programming language for Nix, namely:

- Philipp Baumann
- Justin Bedó
- László Kupcsik
- Bruno Rodrigues

From the NixOS Foundation's side, we have taken up contact with:

- Ron Efroni
- Ryan Trinkle

Processes

The process should:

- establish a formal commitment with the NixOS Foundation to ensure that any purchased hardware will be dedicated to building R package binaries,
- secure funding, and either forward it to the NixOS Foundation for hardware acquisition or purchase the hardware directly and donate it.

Tools & Tech

x86_64 Linux Servers

To build and serve binaries at scale, we will require at least one high-performance x86_64 server. A typical configuration would be:

- CPU: Dual Intel Xeon Gold or AMD EPYC processors, 32–64 cores total
- RAM: 256–512 GB ECC memory
- Storage: 4–8 TB NVMe SSD
- Networking: 10 Gbit Ethernet or better (to sync with the build farm)
- Redundancy: Hot-swappable drives, redundant PSUs for reliability

Such a machine provides the compute and memory required to build thousands of R packages quickly, while integrating smoothly with the existing NixOS build infrastructure.

macOS (aarch64-darwin) Builders

For macOS binary coverage, we will provide two Apple Mac Minis (M4 generation or latest available at purchase). A Mac Mini (as of 2025) typically includes:

- CPU: Apple M4 Pro or M4 Max SoC
- GPU: Integrated 16–40 core GPU
- RAM: 128 GB unified memory
- Storage: 1 TB SSD (or more, but 1 TB is likely enough)
- Networking: Built-in 10 Gbit Ethernet option

Two Mac Minis will be deployed in parallel to provide sufficient capacity and redundancy for macOS builds, ensuring timely binary availability for aarch64-darwin users.

Funding

Estimated cost: ~10'000 EUR: - 2500 EUR for each Mac mini - 5000 EUR for the Linux server

Summary

This project requires modest investment in hardware and formal coordination with the NixOS Foundation. By dedicating a small cluster of Linux servers and Mac Minis to building R package binaries, we can provide full binary coverage for CRAN and Bioconductor within Nix. The total cost of around 10'000 EUR covers both the purchase and donation of this hardware. Once integrated into the existing NixOS build farm, the infrastructure will serve the entire R community by ensuring reproducibility, faster package installation, and reliable support across Linux and macOS platforms.

Success

Definition of done

The project will be considered complete when the following conditions are met: - A formal agreement with the NixOS Foundation is signed, dedicating the new hardware to the task of building R package binaries. - The specified hardware (one x86_64 Linux server and two aarch64-darwin Mac Minis) has been successfully procured and officially transferred to the NixOS Foundation. - The hardware is fully integrated into the NixOS build farm infrastructure and is actively building packages. - Comprehensive binary coverage for the entirety of current CRAN and Bioconductor packages is achieved and publicly available for x86_64-linux, aarch64-linux and aarch64-darwin architectures.

Measuring success

Success will be measured through the following key markers: - **Integration Confirmation:** Official confirmation from the NixOS Foundation that the donated hardware is online and successfully integrated into their build farm (Hydra). - **Binary Coverage:** A publicly accessible metric (e.g., a dashboard) tracking the percentage of CRAN and Bioconductor packages with available pre-built binaries. The primary success marker is reaching and maintaining near 100% coverage.

Future work

Upon successful completion, this project can be extended in several ways: - **Historical Coverage:** A dedicated effort could be launched to build historical versions of all CRAN and Bioconductor packages, providing a complete archive for long-term reproducibility.

Key risks

- **People:** The project relies on a small group of volunteer maintainers from both the R-on-Nix community and the NixOS Foundation. The departure or unavailability of key individuals could create delays in integration or long-term maintenance.
- **Processes:** Delays in formalizing the agreement with the NixOS Foundation or unforeseen logistical complexities in purchasing and transferring hardware internationally could slow down the project's start.
- **Tooling & Technology:** There may be unforeseen technical challenges in integrating the specific hardware into the NixOS build farm. Furthermore, a small subset of R packages with exceptionally complex or non-standard build requirements may not be built successfully for NixOS.
- **Costs:** The estimated hardware cost of ~10,000 EUR may fluctuate due to market price changes or unexpected shipping and import duties.