


2025-08-20

# 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

An example in-text citation ([Wickham 2016](#)).

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**

### **Start-up phase**

### **Technical delivery**

### **Other aspects**

## **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**

#### **Measuring success**

#### **Future work**

#### **Key risks**

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.  
<https://ggplot2.tidyverse.org>.