

# NixOS

**NixOS** is a [Linux distribution](#) built on top of the [Nix package manager](#). It uses [declarative configuration](#) and allows reliable system upgrades.<sup>[3]</sup> Two main branches are offered: current Stable release and Unstable following latest development. Although NixOS started as a research project,<sup>[4]</sup> it is a fully functional and usable operating system. NixOS has tools dedicated to [DevOps](#) and deployment tasks.<sup>[5][6]</sup>

## Contents

### History

#### [Versions](#)

### Features

#### [Declarative system configuration model](#)

#### [Reliable upgrades](#)

#### [Atomic upgrades](#)

#### [Rollbacks](#)

#### [Reproducible system configurations](#)

#### [Source-based model, with binaries](#)

#### [Consistency](#)

#### [Multi-user package management](#)

### Implementation

### Reception

### See also

### References

### External links

## History

NixOS started as a research project by Eelco Dolstra in 2003.<sup>[4][7]</sup>

The [Stichting NixOS](#) was founded in 2015 and aims to support projects like NixOS that implement the purely functional deployment model.<sup>[8]</sup>

## Versions

NixOS follows a cadenced releasing, twice a year.<sup>[9]</sup> This used to happen around March and September but, starting with 21.05, NixOS targets May and November instead.<sup>[10]</sup> Each version number has the following format: "YY.MM".

## NixOS



# NixOS

<b><a href="#">Written in</a></b>	Nix expression language
<b><a href="#">OS family</a></b>	<a href="#">Linux</a> ( <a href="#">Unix-like</a> )
<b><a href="#">Working state</a></b>	In development
<b><a href="#">Source model</a></b>	<a href="#">Open source</a>
<b><a href="#">Initial release</a></b>	2003
<b><a href="#">Latest release</a></b>	21.05 / May 31, 2021 <sup>[1]</sup>
<b><a href="#">Repository</a></b>	<a href="https://github.com/NixOS/nixpkgs">github.com/NixOS/nixpkgs</a> ( <a href="https://github.com/NixOS/nixpkgs">https://github.com/NixOS/nixpkgs</a> )
<b><a href="#">Marketing target</a></b>	General purpose
<b><a href="#">Package manager</a></b>	<a href="#">Nix</a>
<b><a href="#">Platforms</a></b>	i686, <a href="#">x86-64</a> , <a href="#">ARMv7</a> , <a href="#">AArch64</a>
<b><a href="#">Kernel type</a></b>	<a href="#">Monolithic</a> ( <a href="#">Linux kernel</a> )
<b><a href="#">License</a></b>	<a href="#">MIT</a> <sup>[2]</sup>
<b><a href="#">Official website</a></b>	<a href="https://nixos.org">nixos.org</a> ( <a href="https://nixos.org">http://nixos.org</a> )

For example: "20.03" is the version released around March 2020.

Besides that, each version is named, such as “Markhor” for the release 20.03.

## Versions History

List of NixOS versions<sup>[11]</sup>

Date	Version	Name	Notes
2013-10-31	13.10	Aardvark	First stable release
2014-04-30	14.04	Baboon	Linux kernel 3.12
2014-12-30	14.12	Caterpillar	Linux kernel 3.14
2015-09-30	15.09	Dingo	Linux kernel 3.18
2016-03-31	16.03	Emu	Linux kernel 4.4
2016-09-30	16.09	Flounder	Linux kernel 4.4
2017-03-31	17.03	Gorilla	Linux kernel 4.9
2017-09-29	17.09	Hummingbird	Linux kernel 4.9
2018-04-04	18.03	Impala	Linux kernel 4.14
2018-10-05	18.09	Jellyfish	Linux kernel 4.14
2019-04-11	19.03	Koi	Linux kernel 4.19
2019-10-09	19.09	Loris	Linux kernel 4.19
2020-04-20	20.03	Markhor	Linux kernel 5.4
2020-10-27	20.09	Nightingale	Linux kernel 5.4
2021-05-31	21.05	Okapi	Linux kernel 5.10

## Features

### Declarative system configuration model

In NixOS, the entire operating system – the kernel, applications, system packages, configuration files, and so on – is built by the Nix package manager from a description in a functional build language. This means that building a new configuration cannot overwrite previous configurations.<sup>[12]</sup>

A NixOS system is configured by writing a specification of the functionality that the user wants on their machine in a global configuration file. For instance, here is a minimal configuration of a machine running an SSH daemon:<sup>[13]</sup>

```
{
  boot.loader.grub.device = "/dev/sda";
  fileSystems."/" .device = "/dev/sda1";
  services.sshd.enable = true;
}
```

After changing the configuration file, the system can be updated using the `nixos-rebuild switch` command. This command does everything necessary to apply the new configuration, including downloading and compiling packages and generating configuration files.

## Reliable upgrades

Since Nix files are pure and declarative, evaluating them will always produce the same result, regardless of what packages or configuration files are on the system. Thus, upgrading a system is as reliable as reinstalling from scratch.

## Atomic upgrades

NixOS has a transactional approach to configuration management making configuration changes such as upgrades atomic. This means that if the upgrade to a new configuration is interrupted – say, the power fails half-way through – the system will still be in a consistent state: it will either boot in the old or the new configuration. In other systems, a machine might end up in an inconsistent state, and may not even boot anymore.<sup>[14]</sup>

## Rollbacks

If after a system update the new configuration is undesirable, it can be rolled back using a special command (`nixos-rebuild switch --rollback`). Every system configuration version automatically shows up at the system boot menu. If the new configuration crashes or does not boot properly, an older version can be selected. Rollbacks are lightweight operations that do not involve files being restored from copies.

## Reproducible system configurations

NixOS's declarative configuration model makes it easy to reproduce a system configuration on another machine. Copying the configuration file to the target machine and running the system update command generates the same system configuration (kernel, applications, system services, and so on) except for parts of the system not managed by the package manager such as user data.

## Source-based model, with binaries

The Nix build language used by NixOS specifies how to build packages from source. This makes it easy to adapt the system to user needs. However, building from source being a slow process, the package manager automatically downloads pre-built binaries from a cache server when they are available. This gives the flexibility of a source-based package management model with the efficiency of a binary model.<sup>[15]</sup>

## Consistency

The Nix package manager ensures that the running system is consistent with the logical specification of the system, meaning that it will rebuild all packages that need to be rebuilt. For instance, if the kernel is changed then the package manager will ensure that external kernel modules will be rebuilt. Similarly, when a library is updated it ensures that all the system packages use the new version, even packages statically linked to it.

## Multi-user package management

There is no need for special privileges to install software in NixOS. In addition to the system-wide profile, every user has a dedicated profile in which they can install packages. Nix also allows multiple versions of a package to coexist, so different users can have different versions of the same package installed in their respective profiles. If two users install the same version of a package, only one copy will be built or downloaded, and Nix's security model ensures that this is secure.

## Implementation

---

NixOS is based on the Nix package manager that stores all packages in isolation from each other in the package store.

Installed packages are identified by a cryptographic hash of all input used for their build. Changing the build instructions of a package modifies its hash and that will result in a different package installed in the package store. This system is also used to manage configuration files ensuring that newer configurations are not overwriting older ones.

An implication of this is that NixOS doesn't follow the Filesystem Hierarchy Standard. The only exceptions are a symlink `/bin/sh` to the version of `bash` in the Nix store (like this: `/nix/store/s/5rnfzla9kcx4mj5zdc7nlnv8na1najvg-bash-4.3.43/`); and while NixOS does have an `/etc` directory to keep system-wide configuration files, most files in that directory are symlinks to generated files in `/nix/store` such as `/nix/store/s2sjbl85xnrc18rl4fhn56irkxqxyk4p-sshd_config`. Not using global directories such as `/bin` is part of what allows multiple versions of a package to coexist.

## Reception

---

Jesse Smith reviewed NixOS 15.09 for DistroWatch Weekly.<sup>[16]</sup> Smith wrote:

I very much like the way NixOS takes the worry out of upgrading packages by placing each change in its own "generation" and I found, from the end user's point of view, NixOS worked just the same as any other Linux distribution. Setting up NixOS is not for beginners, and I do not think NixOS is intended to be used as a general purpose desktop operating system. But what NixOS does do is give us a useful playground in which to examine the Nix package manager and I think this is very interesting technology which deserves further exploration and adoption by additional distributions.

DistroWatch Weekly also has a review of NixOS 17.03, written by Ivan Sanders.<sup>[17]</sup>

## See also

---

- Nix package manager – The package manager upon which NixOS is based
- GNU Guix System – An operating system built on GNU Guix that is inspired by Nix<sup>[18]</sup>

## References

---

1. "Release 21.05 ("Okapi", 2021-05/31)" (<https://nixos.org/manual/nixos/stable/release-notes.htm>)

- [/#sec-release-21.05](#)). 2021-05-31. Retrieved 2021-06-16.
2. "nixpkgs/COPYING at master · NixOS/nixpkgs · GitHub" (<https://github.com/NixOS/nixpkgs/blob/master/nixos/COPYING>). Github.com. Retrieved 2015-09-19.
  3. "DistroWatch.com: NixOS" (<http://distrowatch.com/table.php?distribution=nixos>). Distrowatch.com. Retrieved 2015-09-19.
  4. Dolstra, Eelco. "Integrating Software Construction and Software Deployment" (<https://web.archive.org/web/20190421081837/https://nixos.org/~eelco/pubs/iscsd-scm11-final.pdf>) (PDF). *Lecture Notes in Computer Science*. **2649**: 102–117. Archived from the original (<https://nixos.org/~eelco/pubs/iscsd-scm11-final.pdf>) (PDF) on 2019-04-21.
  5. "NixOps - The NixOS Cloud Deployment Tool" (<https://nixos.org/nixops/>). Nixos.org. Retrieved 2015-09-19.
  6. "Disnix" (<https://nixos.org/disnix/>). Nixos.org. Retrieved 2015-09-19.
  7. Dolstra, Eelco (2006). *The Purely Functional Software Deployment Model* (<https://web.archive.org/web/20190609111633/https://nixos.org/~eelco/pubs/phd-thesis.pdf>) (PDF) (Ph.D.). Archived from the original (<https://nixos.org/~eelco/pubs/phd-thesis.pdf>) (PDF) on 2019-06-09.
  8. "Stichting NixOS Foundation" (<http://nixos.org/nixos/foundation.html>). Nixos.org. Retrieved 2015-09-19.
  9. "Governance" (<https://web.archive.org/web/20200816224244/https://nixos.org/governance.html>). Nixos.org. Retrieved 2020-08-28.
  10. <https://github.com/NixOS/rfcs/blob/master/rfcs/0080-nixos-release-schedule.md>
  11. "Appendix B. Release Notes" (<https://web.archive.org/web/20200728175754/https://nixos.org/nixos/manual/release-notes.html>). Nixos.org. Retrieved 2020-08-28.
  12. Dolstra, Eelco; Hemel, Armijn (2007). "Purely Functional System Configuration Management" (<https://web.archive.org/web/20190706084138/http://nixos.org/~eelco/pubs/hotos-final.pdf>) (PDF). Archived from the original (<http://nixos.org/~eelco/pubs/hotos-final.pdf>) (PDF) on 2019-07-06.
  13. "About NixOS" (<http://nixos.org/nixos/about.html>). Nixos.org. Retrieved 2015-09-19.
  14. van der Burg, Sander; Dolstra, Eelco; de Jonge, Merijn (2008). "Atomic Upgrading of Distributed Systems" (<https://web.archive.org/web/20190115184012/https://nixos.org/~eelco/pubs/atomic-hotswup2008-final.pdf>) (PDF). Archived from the original (<https://nixos.org/~eelco/pubs/atomic-hotswup2008-final.pdf>) (PDF) on 2019-01-15.
  15. Dolstra, Eelco (2005). "Secure Sharing Between Untrusted Users in a Transparent Source/Binary Deployment Model" (<https://web.archive.org/web/20190926081048/https://nixos.org/~eelco/pubs/secsharing-ase2005-final.pdf>) (PDF). Archived from the original (<https://nixos.org/~eelco/pubs/secsharing-ase2005-final.pdf>) (PDF) on 2019-09-26.
  16. DistroWatch Weekly, Issue 637, 23 November 2015 (<https://distrowatch.com/weekly.php?issue=20151123#nixos>)
  17. DistroWatch Weekly, Issue 712, 15 May 2017 (<https://distrowatch.com/weekly.php?issue=20170515#nixos>)
  18. "About — GuixSD" (<https://www.gnu.org/software/guix/about/>). *www.gnu.org*. Retrieved 2018-05-03.

---

## External links

- [Official website \(https://nixos.org/\)](https://nixos.org/)
  - [NixOS \(https://www.distrowatch.com/table.php?distribution=nixos\)](https://www.distrowatch.com/table.php?distribution=nixos) at [DistroWatch](#)
  - [Nix packages repository \(https://github.com/NixOS/nixpkgs\)](https://github.com/NixOS/nixpkgs)
  - [Unofficial NixOS Wiki \(https://nixos.wiki/\)](https://nixos.wiki/)
-

---

**This page was last edited on 16 June 2021, at 17:42 (UTC).**

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.