Einfach mal Nix machen

Eine kleine Einführung

Felix Breidenstein / fleaz
Seezeit
25.07.2025

fleaz

Lebt seit **1992**

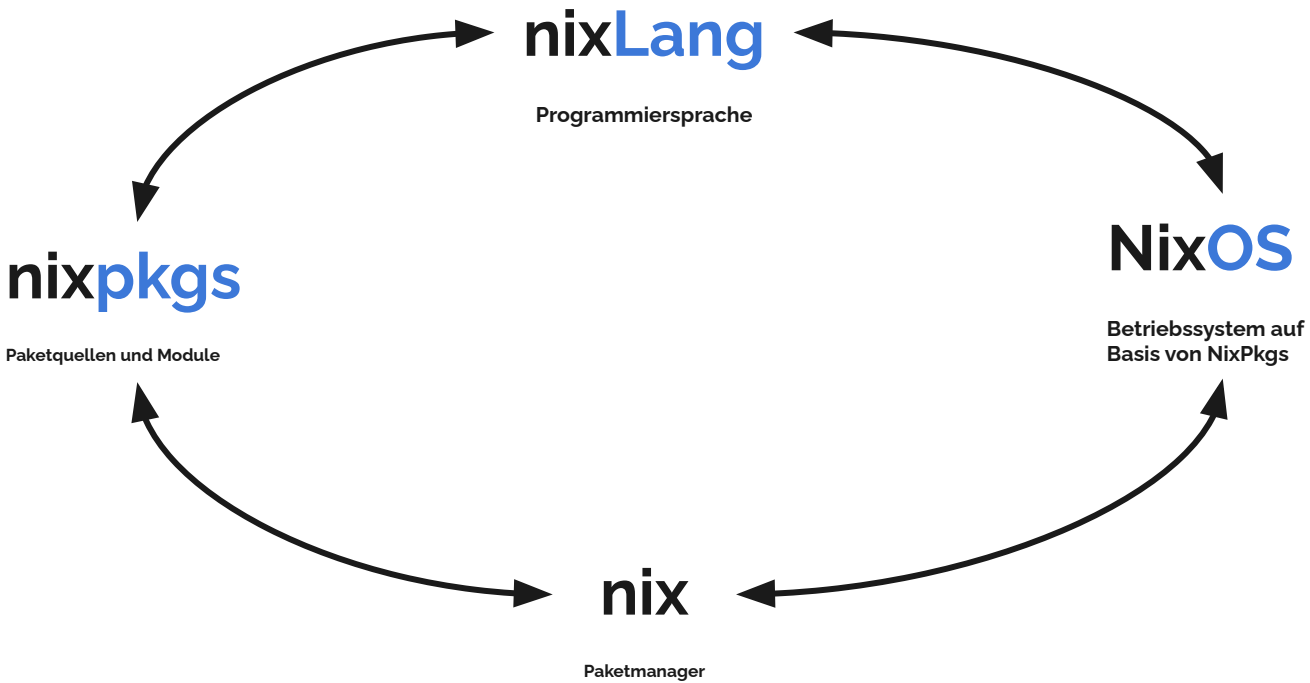
CCC seit **2012**

Nix seit **2019**

YAML Engineer seit **2023**

- Nix != Nix != Nix != Nix
- Historie
- Wieso will ich das?
- Wieso will ich das nicht?
- Security / Hardening
- Demo
- Community/Popularity
- nix ohne NixOS
- Home Manager
- Links
- Meetups

Begriffserklärung



Historie

2003: Eelco Dolstra fängt an Nix zu entwickeln

2005: Armijn Hemel entwickelt NixOS

2012: Nix 1.0

2013: NixOS 13.10 als erstes stable Release

2015: NixOS Foundation wird gegründet

2024: Lix wird geforked

Heute: NixOS Release 25.05, ~120.000 Pakete in nixpkgs

Problemstellung?

```
# apt install openssh-server

# vim  /etc/ssh/sshd.conf

# systemctl enable --now sshd
```

- Was hab ich in welcher Version bekommen?
- Welche Abhängigkeiten? Passt das zum Rest des Systems?
- Wo liegt die Config? Welche Settings hab ich verändert? Valide Config?
- Configchanges nach Updates? (.rpmnew files)
- Ist alles im Backup?
- Sauber deinstallieren?

Perks of NixOS

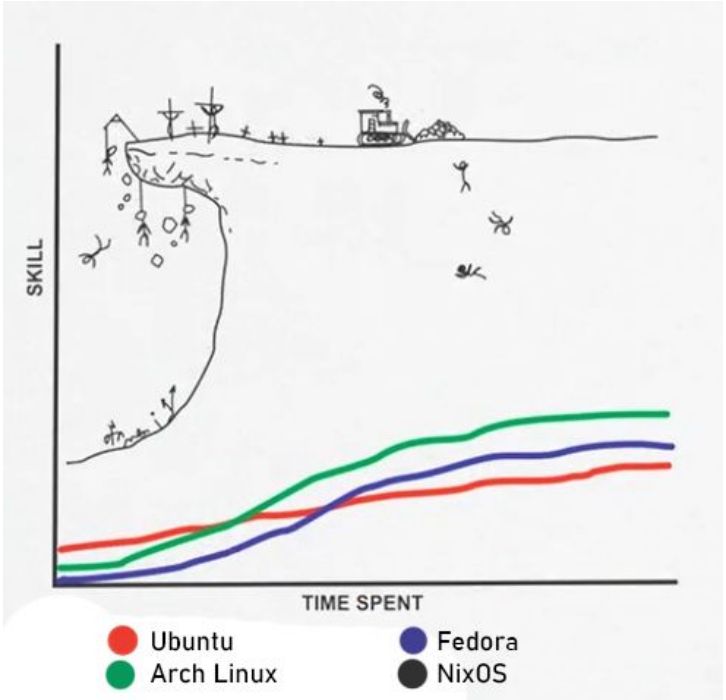
- Deklarative Beschreibung des Gesamtsystems
  - Funktionale DSL (kein ADSL oder VDSL)
  - Config-Management built-in
  - große Auswahl an Modulen und Paketen
- Realisierung als Systemgenerationen
  - atomare Migration in neue Generation
  - einfacher Rollback in alte Generationen
- Reproduzierbar
  - Abhängigkeiten sind gepinnt
- Reusable
  - Teilen von Konfiguration über mehrere Maschinen oder Setups
- Source-based Binary Distribution
- Caching

Yay!

- Vorteile durch Paket/Modul/Tests aus einer Hand
  - Integrationstest der eigenen Module und Pakete
- Keine Versionskonflikte mehr (Dependency hell)
  - ‘alternatives’ ist überholt, jede Anwendung hat ihre eigenen Dependencies
- Guter aarch64-Linux Support
- Mittelguter Darwin Support
- Ideal für Server und Desktop
- Konfiguration für das System ist autoritativ
  - Kein Drift
  - Keine undokumentierten Änderungen:w

Nay?

- Steile Lernkurve!
- Kein FHS (Filesystem Hierarchy Standard)
  - Vorkompilierte Binaries laufen nicht ohne weiteres
- Rolling Release Distro mit zwei Releases pro Jahr
  - Neues Release im Mai und November
  - 7 Monate Support
- Doku ist verstreut und Ausbaufähig
- Fehlermeldungen aus der Hölle
- Vendor Lock-In



Hardening

Systemd Unit von nginx als Beispiel

```
fleaz@vmhost01 ~ $ systemctl cat nginx | grep
[Unit]
Description=A high performance web server and
Documentation=man:nginx(8)
After=network.target nss-lookup.target

[Service]
Type=forking
PIDFile=/run/nginx.pid
ExecStartPre=/usr/sbin/nginx -t -q -g 'daemon
ExecStart=/usr/sbin/nginx -g 'daemon on; maste
ExecReload=/usr/sbin/nginx -g 'daemon on; mast
ExecStop=-/sbin/start-stop-daemon --quiet --st
TimeoutStopSec=5
KillMode=mixed

[Install]
WantedBy=multi-user.target
fleaz@vmhost01 ~ $
```

```
# /etc/systemd/system/nginx.service
[Unit]
After=network.target acme-selfsigned-alert.fleaz.me.service acme-selfsigned-bookmarks.fleaz.me.service acme-selfsigned-documents.fleaz.me.service acme-selfsigned-rainbowr
Before=acme-alert.fleaz.me.service acme-bookmarks.fleaz.me.service acme-documents.fleaz.me.service acme-rainbownerds.de.service acme-grafana.fleaz.me.service acme-headsca
Description=Nginx Web Server
StartLimitIntervalSec=60
Wants=acme-finished-alert.fleaz.me.target acme-finished-bookmarks.fleaz.me.target acme-finished-documents.fleaz.me.target acme-finished-rainbownerds.de.target acme-finish

[Service]
Environment="LOCALE_ARCHIVE=/nix/store/fzm1flvb7zmz3ij4sscn521shz2f76jh-glibc-locales-2.37-45/lib/locale/locale-archive"
Environment="PATH=/nix/store/w8vm09hri2zz7yacryzzzxvsapik4ps4-coreutils-9.1/bin:/nix/store/4cxs4cigh2zdxvma52ygm3mh2igq70iw-findutils-4.9.0/bin:/nix/store/b4in4hmq54h6l34
Environment="TZDIR=/nix/store/951696yxqlphz378fx126wjnrih08mz3-tzdata-2023c/share/zoneinfo"


X-StopIfChanged=false
AmbientCapabilities=CAP_NET_BIND_SERVICE
AmbientCapabilities=CAP_SYS_RESOURCE
CacheDirectory=nginx
CacheDirectoryMode=0750
CapabilityBoundingSet=CAP_NET_BIND_SERVICE
CapabilityBoundingSet=CAP_SYS_RESOURCE
ExecReload=/nix/store/m37r5v1p00ci0qfbwpz6f1j1imx0p5n1-nginx-1.24.0/bin/nginx -c '/nix/store/lqglcpsar7frjz4ljrwnvbhy02kqqhgk-nginx.conf' -t
ExecReload=/nix/store/w8vm09hri2zz7yacryzzzxvsapik4ps4-coreutils-9.1/bin/kill -HUP $MAINPID
ExecStart=/nix/store/m37r5v1p00ci0qfbwpz6f1j1imx0p5n1-nginx-1.24.0/bin/nginx -c '/nix/store/lqglcpsar7frjz4ljrwnvbhy02kqqhgk-nginx.conf'
ExecStartPre=/nix/store/3fdck55xmlpn3p9kzp22vanfjrpi7fzv-unit-script-nginx-pre-start/bin/nginx-pre-start
Group=nginx
LockPersonality=true
LogsDirectory=nginx
LogsDirectoryMode=0750
MemoryDenyWriteExecute=true
NoNewPrivileges=true
PrivateDevices=true
PrivateMounts=true
PrivateTmp=true
ProcSubset=pid
ProtectClock=true
ProtectControlGroups=true
ProtectHome=true
ProtectHostname=true
ProtectKernelLogs=true
ProtectKernelModules=true
ProtectKernelTunables=true
ProtectProc=invisible
ProtectSystem=strict
RemoveIPC=true
Restart=always
RestartSec=10s
RestrictAddressFamilies=AF_UNIX
RestrictAddressFamilies=AF_INET
RestrictAddressFamilies=AF_INET6
RestrictNamespaces=true
RestrictRealtime=true
RestrictSUIDSGID=true
RuntimeDirectory=nginx
RuntimeDirectoryMode=0750
SystemCallArchitectures=native
SystemCallFilter=~@cpu-emulation @debug @keyring @mount @obsolete @privileged @setuid
SystemCallFilter=~@ipc
UMask=0027
User=nginx
```

Security

- Multi-User Setup (Installation meiner Applikation ohne root access)
- Sandboxed Builds ohne Internet
- Inputs und Outputs sind sha256-hashed
- Einfach selber patches auf Pakete anwenden

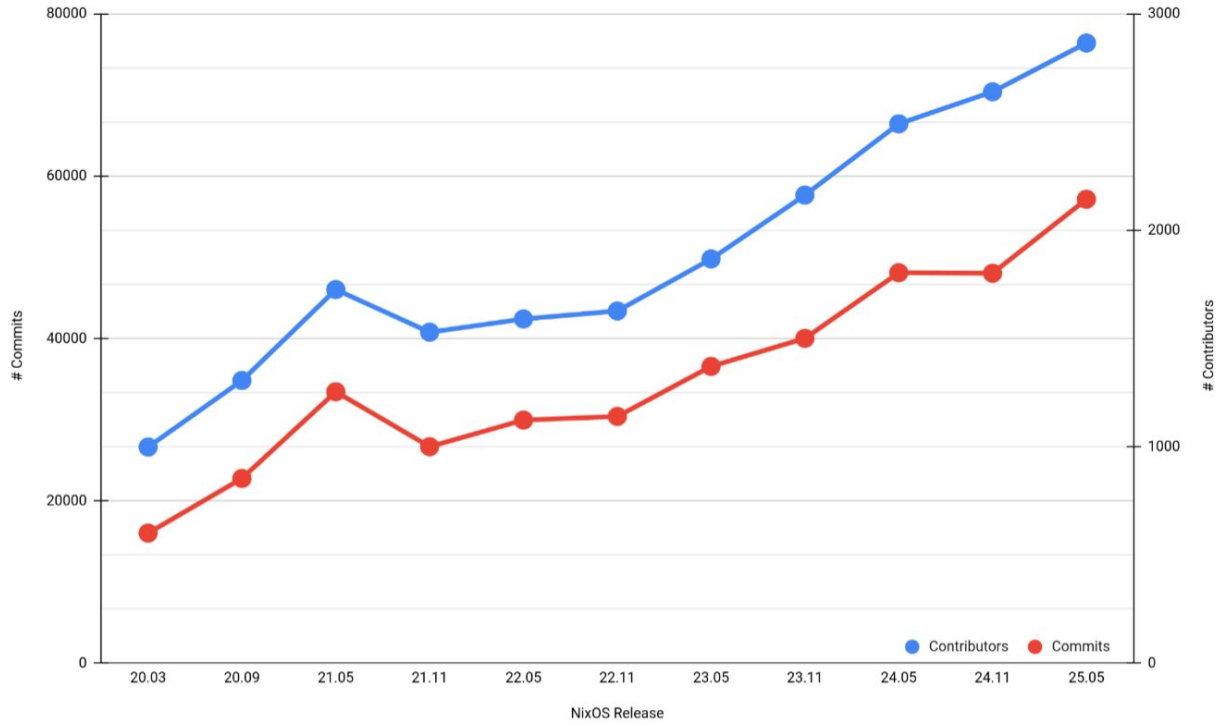
- Security through obscurity
  - Malware braucht bestimmt ein normales FHS

Demotime! 🥳

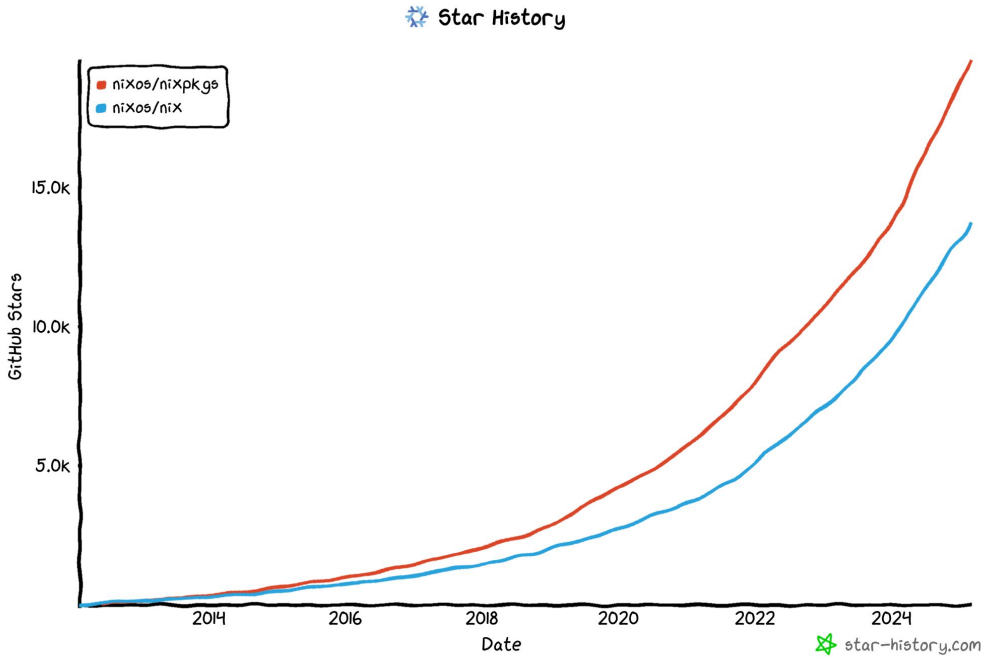
Community & Popularität

- Direkte Contribution zur PKGs über öffentliches Github Repo
- bekannte Softwaretools nutzen direnv/flakes
  - Shopware
- große Projekte bauen auf nix/nixpkgs auf
  - Firebase Studio (Google IDX)
  - Cern LHCb
- Alle im Hackspace nutzen es!

Contributor & Commits



Github Sterne

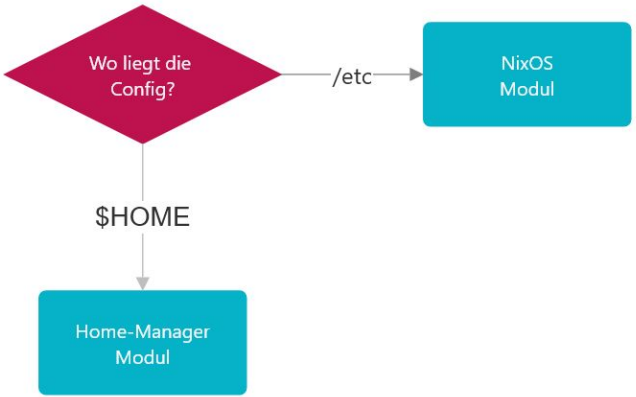


nix ohne NixOS

- Nix Paketmanager kann unter jedem Unix genutzt werden
  - Alle Pakete aber keine Module
- Das bessere homebrew unter macOS
- Als Ergänzung zu apt um Snaps zu umgehen

HomeManager

- Verwaltung von User-Environments mit Nix/NixPkgs
- Bringt Module auf nicht NixOS-Systeme
-



Slides gibts unter https://slides.fleaz.me

Meine private NixOS-Config: https://git.rainbownerds.de/felix/nixos-config

Server Config der letzten MRMCD: https://git.darmstadt.ccc.de/mrmcd/infra/nixos-config

Podcastfolge über NixOS: https://focus.sva.de/podcast/focus-on-linux-die-fabelhafte-welt-von-nix/

Einführung in NixLang: https://nix.dev

Einführung in Nix und Flakes: https://zero-to-nix.com/

Community Wiki: https://wiki.nixos.org

Talks der letzten NixCON: https://media.ccc.de/c/nixcon2023

Lokale Meetups

Im CCCDA etwa alle 6 Wochen

Im CCCFFM jeden 1. Mittwoch

Ausblick

- envrc/direnv für DevEnvironments
- Packaging
- Module bauen
- Server Deployment
- Flakes