







NixOS

Vortrag im Linux-Cafe am 11.9.2024

von Aldo Brießmann

Lizenz: **CC BY-NC-SA 4.0** (<https://creativecommons.org/licenses/by-nc-sa/4.0/>)

Einführung

- wer bin ich:     
- persönlicher Bezug zu NixOS
 -  NixOS ist momentan für *mich* nicht die bevorzugte Linux-Distro (aber für einige andere Leute schon)
 - 2018 mal kurz genauer ausprobiert
 - länger (seit 2017) geplantes Projekt: Bürorechner in der Fachschaft an der Uni mit NixOS (statt Gentoo) -> letztes Jahr wiederaufgenommen

Ziel(e) des Vortrags

- Einblick in NixOS vermitteln (Setup/Konfiguration, Betrieb/Updates, ...)
- Darstellung von Chancen und Herausforderungen der NixOS-Herangehensweise
- Vermittlung von Denkansätzen und Quellen für die persönliche weitere Recherche
- ich lern gern auch noch was :-)

Was ist NixOS?

- **NixOS** : (eigenständige) Linux-Distribution
 - **nixpkgs**: eigenes Paket-Format, >100.000 Pakete
 - **Channels**: Fixed-/Point-Releases (ca. 2x pro Jahr), aktuell *24.05*, und Rolling Release mit “unstable”-Channel
- **nix** : eingebaute *deklarative* Konfigurationssprache des Systems

Einfache Alltags-Aufgaben

- Updaten (mit vorhandener Config): `nixos-rebuild switch` und dann ggf. rebooten
- ohne Root-Rechte als User Pakete installieren: `nix-env --install $PACKAGE`
- ohne Root-Rechte Programme runterladen und in temporärer(!) abgekapselter Umgebung/
Shell starten: `nix-shell -p $package`
- interaktive temporäre(!) Shell basierend auf Nix-Expression (Config-File) starten: `nix-shell $NIX_FILE`
- nach Paketen suchen: `nix-env -qaP '.*$package.*'`

Nix(OS)-Config

```
1 { config, pkgs, ... }:
2 {
3     environment.systemPackages = [
4         vim
5         bash
6         firefox
7     ];
8     environment.variables = {
9         EDITOR = "vim";
10    };
11 }
```

Konfiguration und Anpassung des Systems

- Standard-Config-Speicherort: `/etc/nixos/configuration.nix`
- Optionen: siehe <https://search.nixos.org/options> (<https://search.nixos.org/options>)
- Pakete: siehe <https://search.nixos.org/packages> (<https://search.nixos.org/packages>)

Demo

- Übersicht über Config mit einigen Beispielen
- Veränderung des Systems (zusätzliches Programm, veränderter Hostname)
- Reboot
- Rollback

Link zu Demo-Ressourcen: <https://gitlab.com/Supersonic112/linux-cafe-vortrag-nixos> (<https://gitlab.com/Supersonic112/linux-cafe-vortrag-nixos>)

Wann/Warum NixOS?

- sehr nützlich für klar strukturiertes System
- reproduzierbares Setup leicht umsetzbar^[1]
- einfache Rollbacks möglich
- keine zusätzlichen Konfigurations-Tools nötig
- seeehr viele Pakete verfügbar
- einige fortgeschrittene Funktionen sehr einfach (z.B. SSH-Server)


1. <https://reproducible.nixos.org/> (<https://reproducible.nixos.org/>) ↩

Wann/Warum eher nicht NixOS?


- Umgewöhnung auf deklarative Konfigurationssprache und Denkweise erforderlich
- sehr umfangreiches Ökosystem (zahlreiche Tools und Projekte)
- viel verborgene Logik
- eigenes Paketformat, Pakete eher aus NixOS-Community als von Herstellern

- manche Pakete sind stark veraltet und haben unsichere Dependencies; Gegenmaßnahmen sind in Arbeit, siehe Vortrag von der Fosdem 2024 (<https://fosdem.org/2024/schedule/event/fosdem-2024-1983-remediating-thousands-of-untracked-security-vulnerabilities-in-nixpkgs/>)
- Dokumentation umfangreich (z.B. zwei Wikis), aber für einige Pakete nicht ausreichend, zusätzlich fortgeschrittenes Wissen nötig
- NixOS folgt nicht dem Filesystem Hierarchy Standard, z.B. standardmäßig kein /bin/bash verfügbar
- Rollback-Feature verbraucht viel Festplatten-Platz

Interessante fortgeschrittene Themen

- Config im Git versionieren
- `nix` in anderen Linux-Distributionen nutzen
- Home-Manager: Nix-Config für Home-Verzeichnisse
-  Nix-Flakes^[1]: experimentelles (!) Feature des Nix-Package-Managers, erlaubt es Nix-Config-Segmente zu erstellen und wiederzuverwenden und mit anderen Leuten zu teilen

1. <https://nixos.wiki/wiki/flakes> (<https://nixos.wiki/wiki/flakes>) ↩

- CI-Testing mit Hydra ^[1]
- `nixos-rebuild build-vm` : VM mit momentaner Config erstellen; geht auch für einzelnen Service ^[2]
-  NixOps^[3]: Cloud-Deployment-Tool
- Secrets (Passwörter etc.) in Config z.B. mit `nix-sops` verschlüsselt speicherbar

1. <https://nixos.wiki/wiki/Hydra> (<https://nixos.wiki/wiki/Hydra>) ↩

2. https://nixos.wiki/wiki/Cheatsheet#Building_a_service_as_a_VM_for_testing.29 (https://nixos.wiki/wiki/Cheatsheet#Building_a_service_as_a_VM_for_testing.29) ↩


3. <https://nixos.wiki/wiki/NixOps> (<https://nixos.wiki/wiki/NixOps>) ↩

Weiterführende Links

-  *offizielles* NixOS-Wiki: <https://wiki.nixos.org> (<https://wiki.nixos.org>)
- *inoffizielles* User-Wiki: <https://nixos.wiki/> (<https://nixos.wiki/>)
- NixOS-Doku: <https://nix.dev/> (<https://nix.dev/>)
- NixOS-Manual: <https://nixos.org/manual/nixos/stable/> (<https://nixos.org/manual/nixos/stable/>)
- NixOS-Intro: <https://nix.dev/tutorials/first-steps/> (<https://nix.dev/tutorials/first-steps/>)
- Nix-Doku: <https://nix.dev/manual/nix> (<https://nix.dev/manual/nix>)
- Nixpkgs-Doku: <https://nixos.org/manual/nixpkgs/stable/> (<https://nixos.org/manual/nixpkgs/stable/>)

- ☆ NixOS- und Flakes-Buch: <https://nixos-and-flakes.thiscute.world/> (<https://nixos-and-flakes.thiscute.world/>)
- Einführungs-Seite zu Nix, nixpkgs und Flakes: <https://zero-to-nix.com/> (<https://zero-to-nix.com/>)
- Sammlung interessanter Web-Ressourcen: <https://nixos.wiki/wiki/Resources> (<https://nixos.wiki/wiki/Resources>)

Community-Anlaufstellen

- NixOS Matrix: #users:nixos.org (<https://matrix.to/#/#users:nixos.org>)
- NixOS Discourse-Forum: <https://discourse.nixos.org/> (<https://discourse.nixos.org/>)
-  Konferenz: NixCon^[1] **2024 am 25.-27.10. in Berlin**

1. <https://nixcon.org/> (<https://nixcon.org/>) ↩