Deployment von NixOS im GHG

Samson

Contents

1. Einleitung 3
    1.1. NixOS 3
2. Nix 3
    2.1. Nix Language 3
    2.2. Nix Package Manager und Derivationen 3
    2.3. Nix Hashes 4
    2.4. Nix Store 4
3. Vorteile von NixOS als Grundlage der Client PC’s 4
4. Glossar 5
Quellen 6

1. Einleitung

In dieser Facharbeit geht es darum, wie man Linux im GHG als Betriebssystem der Arbeitsrechner verwenden koennte. Dabei wird speziell die Verwendung der Distribution NixOS in den Vordergrund gestellte.

1.1. NixOS

NixOS ist eine deklarative Linux Distribution, man bestimmt installierte Programme und alle Konfigurationen, beispielsweise Benutzer, durch die Nix Sprache typischer Weise im Folder `/etc/nixos`. Man kann Firefox beispielsweise so installieren:

configuration.nix Nix

```nix
programs.firefox = {
  enable = true;
  preferences = {
    "privacy.resistFingerprinting" = true;
  };
};
```

Nach jeder Veraenderung der Konfiguration kann man eine neue Generation des Systems bauen und aktivieren mit `sudo nixos-rebuild switch`. Allerdings werden alte Systembeschreibungen weiterhin behalten, dass ermoeglicht Rollbacks ohne viel Speicherplatz zu verwenden. Wenn man von Generation 1 zu Generation 2 ein Programm entfernt, bleibt dieses jedoch im Nix Store ausser man lehrt den store.

2. Nix

Die Grundlagen von Nix entstanden ab 2003 als Forschungsprojekt von Eelco Dolstra in seiner 2006 veroeffentlichten These [1].

2.1. Nix Language

Nix ist eine Turing Vollstaendige, Pure Funktionale, Domain Spezifizierte, Lazy Evaluierte und Dynamically typed Programmiersprache, in der die Beschreibung von Paketen und die Konfiguration von NixOS vorgenommen wird.

2.2. Nix Package Manager und Derivationen

Die Sprache Nix bietet Funktionen zur Beschreibung von Build-Anweisungen für Pakete. Diese werden zu Nix-Derivationen kompiliert, die (für den Nix-Builder) genau beschreiben, wie das Paket erstellt wird.

Den schritt des Kompilierens und des Bauens uebernimmt der Nix Package Manager. 140′000 Nix Pakete (nixpkgs) [2] werden momentan auf github:NixOS/nixpkgs gehostet.

2.3. Nix Hashes

Der Nix Hash ist wie ein Fingerabdruck einer Datei, er ist nicht nur von dem Output abhängig (z.B. einer Binary wie “vi”) sondern auch von den Dependencies mit denen diese Gebaut wurde, dieser wird Benutzt um zu Verifizieren ob der Build Output exact der Spezifikation entspricht.

2.4. Nix Store

Im Nix Store, der unter `/nix/store/` liegt, werden alle Packages und Dateien unveränderlich gespeichert. Das layout sieht wie folgt aus: `Hash + Name + Version` beispielsweise fuer das Package `hello` ist der Pfad “/nix/store/xl1h9i29pgq2q5cszjhm5wpfxfbbqwyi-hello-2.12.3/bin/hello”. Nur weil ein Package im Store ist, ist es allerdings nicht unbedingt im Pfad und kann somit nicht benutzt werden.

3. Vorteile von NixOS als Grundlage der Client PC’s

Da NixOS eine Linux Distribution ist bezahlt man keine License kosten. Auch sind die Systemanforderungen weitaus nicht so hoch wie bei Windows. NixOS ist sehr minimal, weshalb man ein System einfach und Modular von grundauf gestalten kann. Durch die Immutabilität NixOS’es wird ausserdem das Ausfuehren runtergeladener Binaries fast unmoeglich allerdings theoretisch machbar durch das Programm nix-ld.

4. Glossar

Domain Spezifizierte: Eine Programmiersprache entwickelt direkt fuer einen Spezifischen Zweck. Beispielsweise SQL fuer Datenbanken oder Regex fuer Expression Matching 3

Dynamically typed: Variablen muessen sich nicht explizit auf einen Datentyp festlegen. Der Datentyp eines Wertes wird zur Laufzeit bestimmt und kann sich während der Programmausführung ändern. 3

Lazy Evaluierte: Funktionen und Variablen werden erst berechnet wenn sie gebraucht werden und nicht wenn sie definiert werden. 3

Pure Funktionale: Ein Programmierparadigma das alle Berechnungen als Mathematische Funktionen betrachtet. 3

Turing Vollstaendige: Eigenschaft einer Programmiersprache, jede Aufgabe lösen zu können, die auch eine Turing Maschine berechnen kann. 3

Quellen

[1] E. Dolstra, “The Purely Functional Software Deployment Model,” 2006. Accessed: Oct. 07, 2026. [Online]. Available: https://edolstra.github.io/pubs/phd-thesis.pdf

[2] NixOS, “nixpkgs: Nix Packages collection & NixOS.” Accessed: Oct. 07, 2026. [Online]. Available: https://github.com/NixOS/nixpkgs/