

NixOS Spickzettel

Kurzreferenz für die häufigsten NixOS-Kommandos: System updaten, `nix shell` nutzen, Pakete einmalig ausführen und nach Paketen suchen.

System updaten

System-Konfiguration neu bauen und aktivieren (nach Änderungen an `/etc/nixos/configuration.nix` oder dem Flake):

```
sudo nixos-rebuild switch
```

Nur testen (aktiviert erst beim nächsten Boot als Default, aber sofort für die aktuelle Session):

```
sudo nixos-rebuild test
```

Wenn mit Flakes gearbeitet wird:

```
sudo nixos-rebuild switch --flake /etc/nixos#hostname
```

Channels aktualisieren (klassisches, nicht-Flake-Setup):

```
sudo nix-channel --update
```

Alte Generationen aufräumen (Speicherplatz freigeben):

```
sudo nix-collect-garbage -d
```

`nix shell` nutzen

Startet eine temporäre Shell mit den angegebenen Paketen, ohne sie dauerhaft zu installieren:

```
nix shell nixpkgs#ripgrep nixpkgs#fd
```

Danach sind `rg` und `fd` in der aktuellen Shell verfügbar, verschwinden aber nach dem Verlassen der Shell wieder. Für das klassische (nicht-Flake) Äquivalent:

```
nix-shell -p ripgrep fd
```

Ein Paket einmalig ausführen (ohne Config-Eintrag)

Ideal, um ein Tool kurz auszuprobieren, ohne es in `configuration.nix` einzutragen:

```
nix run nixpkgs#hello
```

Klassisches Äquivalent:

```
nix-shell -p hello --run hello
```

Nach Paketen suchen

Über die Kommandozeile (Flakes/neues CLI):

```
nix search nixpkgs firefox
```

Alternativ über die Weboberfläche: <https://search.nixos.org/packages>

Nach NixOS-Optionen suchen (z. B. für `configuration.nix`):

```
man configuration.nix
```

Weboberfläche für Optionen: <https://search.nixos.org/options>

Need to know

- **Deklarativ statt manuell:** Runtime-Änderungen wie `sysctl foo=bar` von Hand oder ein manuell installiertes Paket überleben keinen Reboot bzw. werden beim nächsten `switch` überschrieben. Alles Dauerhafte gehört in `configuration.nix` / `flake.nix`.
- **switch vs. boot vs. test:** `switch` baut, aktiviert sofort und setzt den Eintrag als Boot-Default. `boot` baut und setzt den Boot-Default, aktiviert aber erst beim nächsten Neustart. `test` aktiviert sofort, ohne den Boot-Default zu ändern (praktisch zum Ausprobieren, ein

Absturz wird beim nächsten Reboot automatisch rückgängig gemacht).

- **Flakes sehen nur, was Git kennt:** Neue Dateien müssen mindestens per `git add` getrackt sein (nicht zwingend committet), sonst ignoriert der Flake-Build sie stillschweigend, mit teils verwirrenden Fehlermeldungen.
- **Home-Manager ist hier als NixOS-Modul eingebunden**, nicht standalone. Ein normaler `nixos-rebuild switch` übernimmt daher auch Änderungen an `home.nix`, ein separates `home-manager switch` ist nicht nötig.

Best Practices

- Vor jedem `switch` die Syntax prüfen, ohne gleich zu bauen:

```
nix-instantiate --parse configuration.nix
```

- Bei riskanteren Änderungen erst nur bauen, ohne zu aktivieren, um Eval-/Build-Fehler ohne Downtime abzufangen:

```
sudo nixos-rebuild build
```

- Config-Repo ist git-versioniert: vor jedem `switch` committen, damit jede funktionierende (oder kaputte) Version nachvollziehbar bleibt und sich per `git revert` zurückrollen lässt.
- Keine Secrets direkt in `configuration.nix` ablegen. Der Wert landet im Klartext im Git-Repo und im Nix-Store, wo er für alle lokalen Nutzer lesbar ist. Für Passwörter/Tokens eignen sich `sops-nix` oder `agenix`.

Nützliche Hinweise

- **Rollback bei kaputtem Rebuild:** `sudo nixos-rebuild switch --rollback`, oder im Boot-Menü eine ältere Generation auswählen.

- **Generationen auflisten:**

```
sudo nix-env --list-generations --profile /nix/var/nix/profiles/system
```

- **Lokale Options-/Manpage-Doku**, passend zur eigenen installierten Version und offline verfügbar: `nixos-help` bzw. `man configuration.nix`.
- **Flake-Inputs aktualisieren:** `nix flake update` aktualisiert alle Inputs in `flake.lock`, `nix flake lock --update-input nixpkgs` nur einen einzelnen.
- **Vorsicht mit `nix-collect-garbage -d`:** löscht alte Generationen inklusive der Möglichkeit, dorthin zurückzurollen. Vorher sicherstellen, dass die aktuelle Generation stabil läuft.