

ALLES NIX





REPRODUZIERBAR, DEKLARATIV, ZUVERLÄSSIG

FOLLOW THIS LIVE



<https://blog.thym.at/glt23>

WHO AM I?

- Student at Graz University of Technology 
- Backend Engineer / Linux SysAdmin 
- Linux Advocate and Avid User of FOSS 
- Daily Driving NixOS since 2019 

WHAT IS NIX?

Not to be confused with *nix

WHAT IS NIX?

- Nix Language
- Nix Package Manager
- Nix Package Library - Nixpkgs
- Nix Operating System - NixOS
- ...

NIX LANGUAGE

- Domain-specific
- Purely functional
- Dynamically typed
- Lazily evaluated

NIX LANGUAGE

LAZY EVALUATION

```
let
  data = {
    a = 1;
    b = functionThatTakesMinutesToRun 2;
  };
in data.a
```

NIX LANGUAGE

- Used to declare packages and configurations
- Build derivations

BUILDING A NIX DERIVATION

```
#!/bin/env bash
```



BUILDING A NIX DERIVATION

```
{ lib
, stdenv
}:

stdenv.mkDerivation {
  pname = "example";
  version = "1.0";

  dontUnpack = true;

  installPhase = ''
    mkdir -p $out/bin
    cat > $out/bin/linuxtage << EOF
    #!/usr/bin/env sh
    echo "Hello, world! Welcome to Grazer Linuxtage 2023!"
    EOF
```

BUILDING A NIX DERIVATION

```
nix-build -E 'with import <nixpkgs> {}; callPackage ./example.nix
```

BUILDING A NIX DERIVATION

```
λ /nix/store/zgnwkbmr528n6pjwt8h1zy3ps9190myl0-example-1.0/bin/lin  
Hello, welcome to Grazer Linuxtage 2023!
```



BUILDING A NIX DERIVATION

```
{ lib
, stdenv
, zsh
, hello
}:

stdenv.mkDerivation {
  pname = "example";
  version = "1.0";

  dontUnpack = true;

  installPhase = ''
    mkdir -p $out/bin
    cat > $out/bin/linuxtage << EOF
    "#!@(-=+<)/bin/+=<"
```

BUILDING A NIX DERIVATION

```
nix-build -E 'with import <nixpkgs> {}; callPackage ./a-very-soph
```

BUILDING A NIX DERIVATION

```
λ /nix/store/sg0g9xlhnb9z0m6yrdykjqz9dcdvb9b-example-1.0/bin/lin  
Hello, welcome to Grazer Linuxtage 2023!
```



NIX PACKAGE LIBRARY - NIXPKGS



nixpkgs

Nix Packages collection

 Nix



11.9k



9.6k

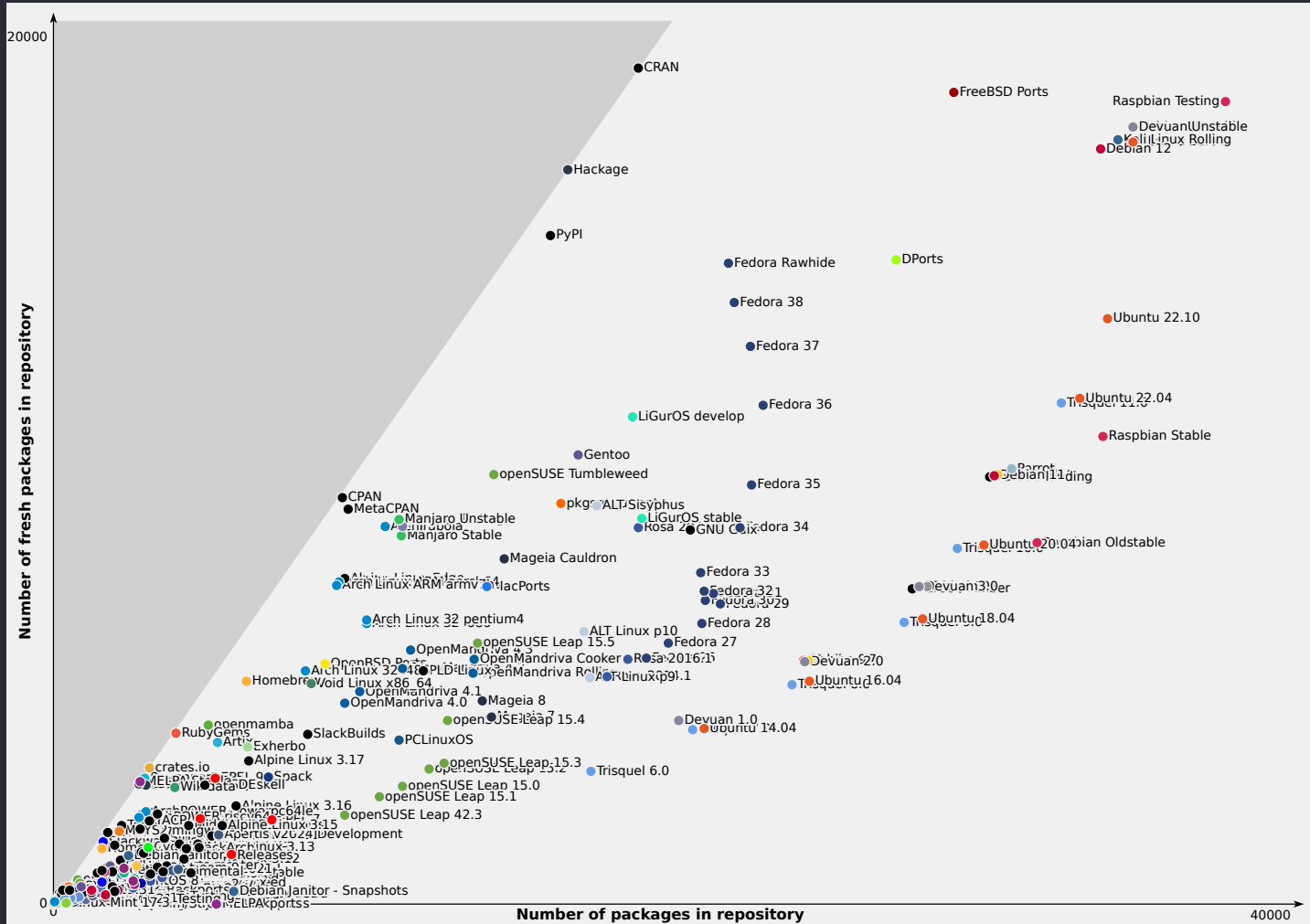
NIXPKGS

EXAMPLE PACKAGE

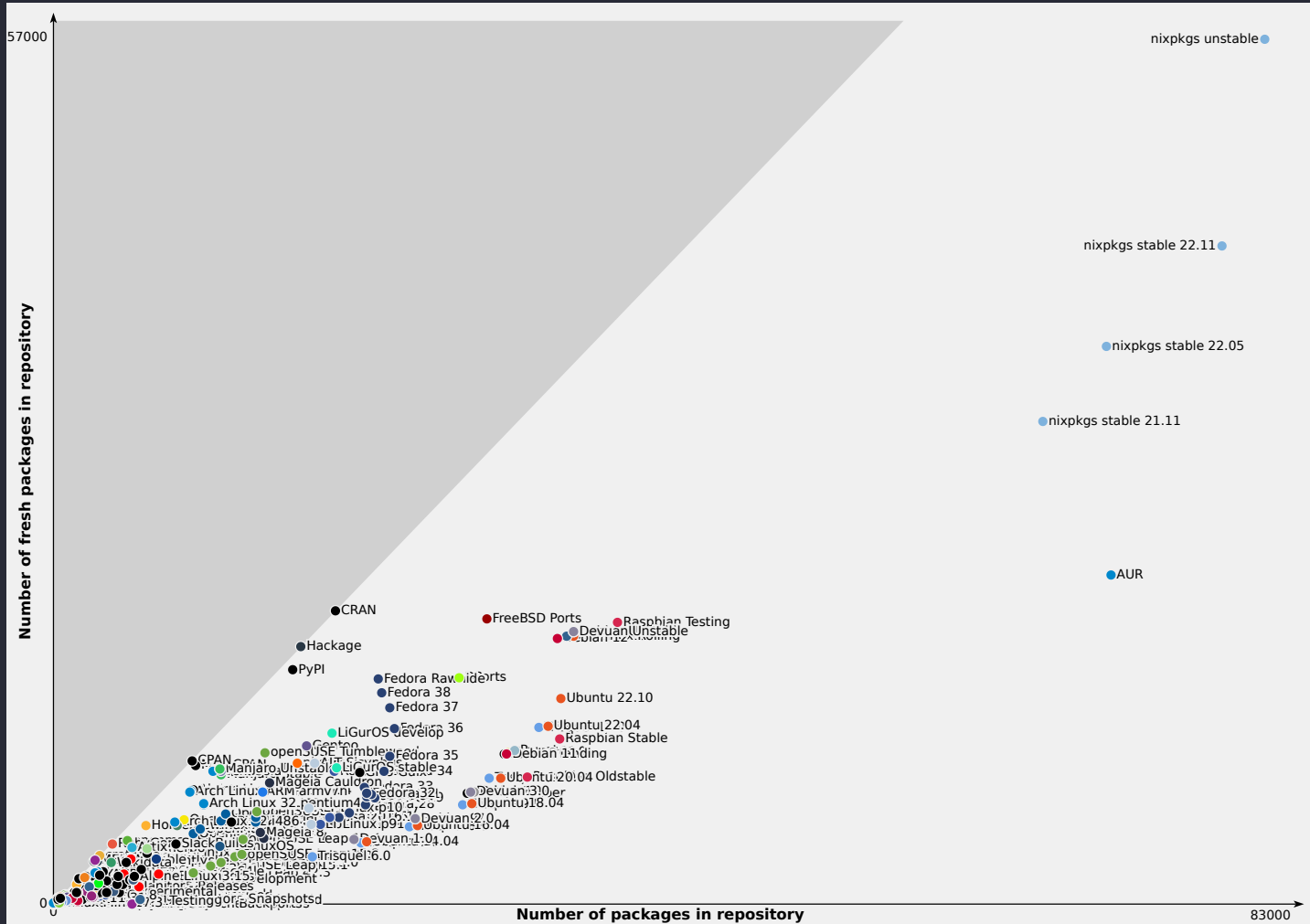
CURL'S DEFAULT.NIX

```
{ lib, stdenv, fetchurl, pkg-config, perl, nixosTests
, brotliSupport ? false, brotli
, c-aresSupport ? false, c-aresMinimal
, gnutlsSupport ? false, gnutls
, gssaslSupport ? false, gssasl
, gssSupport ? with stdenv.hostPlatform; (
  !isWindows &&
  # disable gss because of: undefined reference to `k5_bcmp'
  # a very sad story re static: https://bugs.debian.org/cgi-bin
  !isStatic &&
  # the "mig" tool does not configure its compiler correctly. T
  # fixed in mig, but losing gss support on cross compilation t
  # not worth the effort.
  !(isDarwin && (stdenv.buildPlatform != stdenv.hostPlatform))
), libkrb5
```

REPOSITORY SIZE AND FRESHNESS



IN ITS OWN CLASS



NIX PACKAGE MANAGER

- For Linux and other Unix systems
 - Yes, that includes macOS
- Makes package management reproducible and reliable

NIX PACKAGE MANAGER

```
nix-env -iA nixos	curl  
nix-env -iA unstable	curl
```

HOME MANAGER

```
programs.direnv = {
  enable = true;
  nix-direnv.enable = true;
};

home.packages = with pkgs; [
  coreutils
  curl
  wget
];

home.file = {
  ".zshrc".source = ../config/zsh/.zshrc;
};
```

```
home.pointerToMotd = ../config/motd/motd.nix;
```

HOME MANAGER

```
homebrew = {
  enable = true;
  onActivation = {
    autoUpdate = true;
    upgrade = true;
    # "zap" removes manually installed brews and casks
    cleanup = "zap";
  };
  brews = [
    "cliclick"
  ];
  casks = [
    "alt-tab"
  ];
  taps = [
    "
```

WHAT IS NIXOS?



- Started as research project by Eelco Dolstra in 2003
- Took off after first stable release branch in 2013 (13.10)
- Stable releases in May and November (e.g. Raccoon - 22.11)
- Receive bugfixes and security updates for 7 months

NIXOS



```
environment.systemPackages = with pkgs; [  
  git  
  curl  
  zsh  
  # Modern and intuitive terminal-based text editor  
  micro  
];
```

NIXOS

```
# Find system disk by nixos label by default
fileSystems."/".device = "/dev/disk/by-label/nixos";

# Use the latest kernel by default
boot.kernelPackages = pkgs.unstable.linuxPackages_latest;
```

NIXOS



```
services = {  
  # Start a systemd service for each incoming SSH connection  
  openssh.startWhenNeeded = true;  
  # Enable periodic SSD TRIM to extend life of mounted SSDs  
  fstrim.enable = true;  
  # Suspend when power button is short-pressed  
  logind.extraConfig = ''  
    HandlePowerKey=suspend  
  '';  
};
```

NIXOS

```
# Take out the garbage every once in a while
nix.gc = {
  automatic = true;
  dates = "weekly";
  options = "--delete-older-than 30d";
};
```

COMMUNITIES

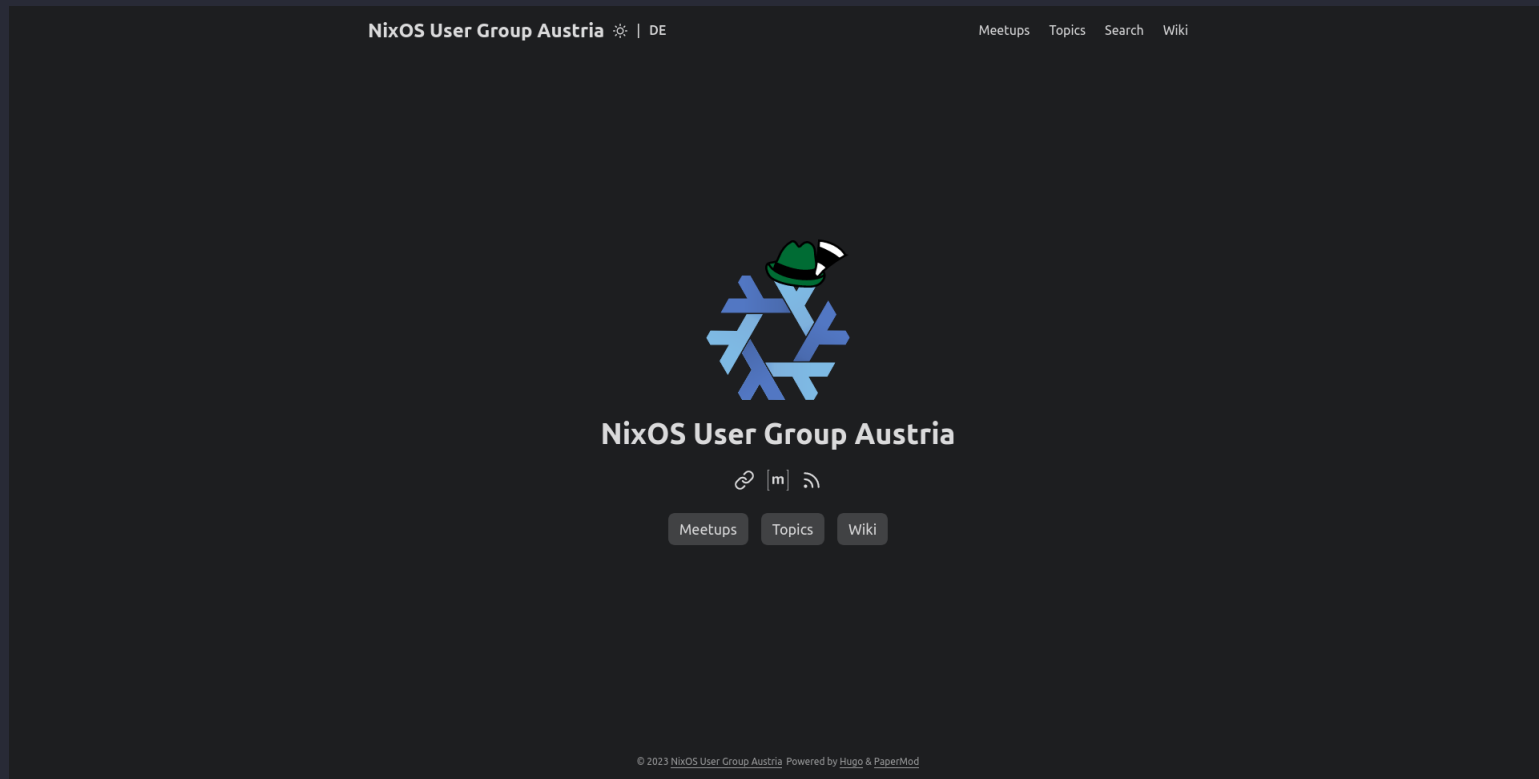
- Discourse
- Matrix
- GitHub
- Discord 🙄

NIXOS USER GROUP AUSTRIA



<https://nixos.at>

NIXOS USER GROUP AUSTRIA



NIXOS USER GROUP AUSTRIA

[Home](#) » [Posts](#)

Software Development on NixOS

April 24, 2023 · Paul Härle · Vienna · Metalab | [Translations: DE](#) | [Suggest Changes](#)

Event

- Date: **2023-04-24**
- Time: **from 18:00**
- Location: **Metalab, Rathausstraße 6, 1010 Vienna** (in the Library)
- Presenters:
 - **Herwig Hochleitner** [@bendlas:matrix.org](mailto:bendlas@matrix.org)

The meetup will be held on **Monday the 24th of April**. We will start at about **18:00** with an introductory talk, with the opportunity for discussion and planning of further meetups later on.

Software Development on NixOS

NixOS is awesome for packaging existing software, with helpers for many common build systems.

From an operators's perspective, it really shines with composing software. How about the developer's perspective?

Nix is a programming language so it should be able to adapt to many use cases.

A developer wanting to use Nix from the ground floor up, may care about various questions:

- What tools are there to support software development?

DEVENV

```
{ pkgs, ... }: {
  env.GREET = "Let's get started with";

  enterShell = "echo $GREET Rust";

  languages.rust = {
    enable = true;
    version = "nightly";
  };

  pre-commit.hooks = {
    clippy.enable = true;
    rustfmt.enable = true;
  };
}
```

MENTIONED RESOURCES

NIX LANGUAGE

- [Nix Language Reference](#) - Nix Reference Manual
- [Nix Language Basics](#) - NixOS Guide
- [Nix REPL](#) - Nix Reference Manual

MENTIONED RESOURCES

NIX PACKAGE MANAGER

- [Homebrew Bundle](#) - GitHub Repository
- [Brewfile Manager](#) - Homebrew-file's Documentation
- [Home Manager Manual](#)
- [Home Manager Option Search](#)

MENTIONED RESOURCES

NIXPKGS

- Tools in `stdenv` - Nixpkgs Manual
- Repository Size and Freshness Graphs - Repology
- Nix User Repository - NUR - GitHub Repository

MENTIONED RESOURCES

NIXOS

- [NixOS](#) - Project Website
- [History of Nix/NixOS](#) - Nix Blog Announcements
- [Nix Guide](#) - Opinionated Guide for Developers
- [NixOS Wiki](#) - Unofficial User's Wiki
- [My Personal Dotfiles](#) - Codeberg Repository

MENTIONED RESOURCES

COMMUNITIES

- Discourse
- Matrix
- IRC
- GitHub
- Discord

MENTIONED RESOURCES

- [devenv](#) - Developer Environments using Nix
- [Awesome Nix](#) - Community-maintained link list with helpful resources

ATTRIBUTIONS

- Example for lazy evaluation taken from [What is Nix](#)
- Shopify

THANKS!

curl -sL <https://matthias.thym.at/card>

<https://blog.thym.at/p/glt23>

