NIXOS

Malte Neuss

NIXOS

“Reproducible builds and deployments.”

By Tim Cuthbertson CC BY 4.0, https://nixos.org

CONTENT

Nix

- OS
- Package Manager
- Language

NIX PACKAGE MANAGER

- Adhoc environments

```
$ nix shell nixpkgs#python311
        nixpkgs#poetry
```

- Reproducible environments

```
mkDerivation {
  name = "myProject-1.0.0";
  src = ./src;
  buildInputs = [ python311 poetry
        ...];
}
```

DEMO

NIX LANGUAGE

Derivation

```
{
  "/nix/store/dvmidxj5...-myProject-1.0.0.drv": {
    "inputSrcs": [
      "/nix/store/9krlzvn...-default-builder.sh"
    ],
    "inputDrvs": {
      "/nix/store/1psqjc0l1..-bash-4.4-p23.drv": [
        "out"
      ],
      "/nix/store/8fbvqyxl9y...-myProject-1.0.0.tar.gz.drv": [
        "out"
      ],
      "/nix/store/m15naxf2...-stdenv-linux.drv": [
      ...
```

NIX LANGUAGE

≈Json + Functions

```
mkDerivation {
  name = "myProject-1.0.0";
  src = ./src;
  buildInputs = [ python311 poetry ...];
}
```

Benefits

- Functional
- Pure (almost)
- Declarative

NIX LANGUAGE

- Numbers `42`
- Strings `"hello"`
- List `[1 2 3]`
- Expressions `1+2+3`
- Attribute Sets `{ key = value; ... }`
- Let Bindings `let x = ... in ...`
- Functions `f x`

ATTRIBUTE SET

```
{
  name         = "myProject-1.0.0";
  src          = ./src;
  buildInputs = [ python311 poetry ...];
  ...
}
```

```
{
  myNumber   = 3;
  myAttrSet  = { name = "myProject-1.0.0"; src = ... };
  myString   = otherAttrSet.nestedSet.name;
  myFunction = x: 2*x;
}
```

LET-BINDING

Factor out common expression:

```
{
  name = otherAttrSet.a.b.c;
  src  = otherAttrSet.a.b.d;
  ...
}
```

```
let common = otherAttrSet.a.b;
in
  {
    name = common.c;
    src  = common.d;
    ...
  }
```

FUNCTIONS

Typescript

```
const f = x => 2x
f(3)
```

Nix

```
f = x: 2*x          # definition
f 3                 # usage
```

FUNCTIONS

Typescript

```
const g = (x,y) => x+y
g(3,4)
```

Nix

```
g = x: y: x+y
g 3 4
```

FUNCTIONS

Currying

```
g: (ℝ,ℝ) → ℝ
g: (x,y) ↦ x+y
```

```
g: ℝ → (ℝ → ℝ)
g: x ↦ (y ↦ x+y)
```

Usage

```
g (3,4)
```

```
g 3 4
((g 3) 4)
```

FUNCTIONS

```
# Definition
mkDerivation = overrideAttrs: ...

# Usage
mkDerivation {
  name = "myProject-1.0.0";
  src = ./src;
}
```

Resulting Derivation

```
{
  "/nix/store/dvmidxj5...-myProject-1.0.0.drv": {
    "inputSrcs": [
      "/nix/store/9krlzvn...-default-builder.sh"
    ],
    ...
```

DERIVATION

```
let
 pkgs = ...
in
pkgs.stdenv.mkDerivation {
  name = "myProject-1.0.0";
  src = ./src;
  buildInputs = [ pkgs.python311 pkgs.poetry ];
  buildPhase = "poetry build";
}
```

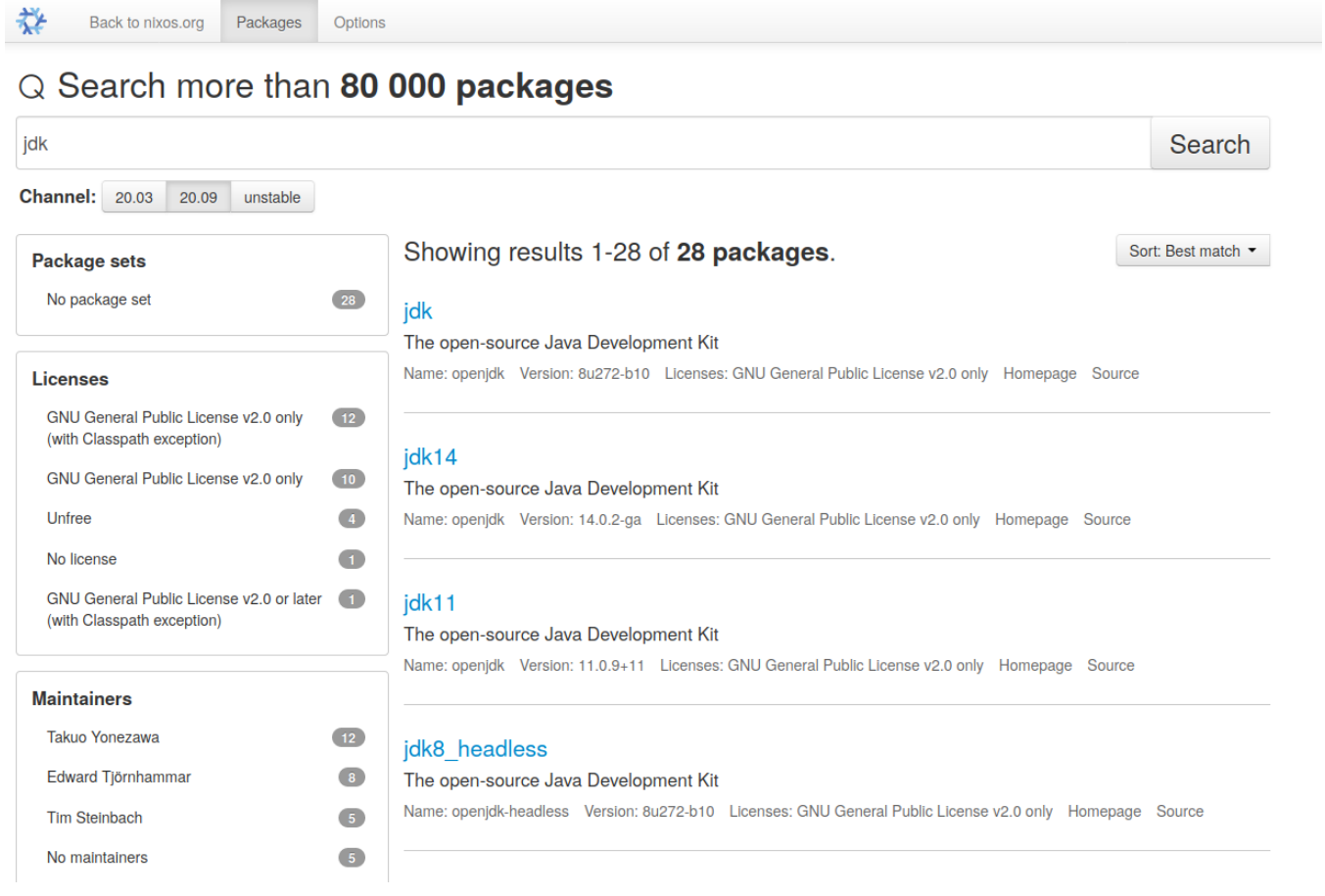
NIXPKGS

Just a huge attribute set:

```
# pkgs =
{
  stdenv = { mkDerivation = .. ; ... }
  ...
  poetry = stdenv.mkDerivation {... python311 ...}
  python39 = ..
  python311 = ..
  ...
  git = ....
  ...
  firefox = ...
  ...
}
```

https://github.com/NixOS/nixpkgs

SEARCH PACKAGES



https://search.nixos.org

REPRODUCIBLE BUILDS

```
$ nix build <derivation> # build artifact, put into store
```

Store

```
/nix/store/9234jfkdfj23j45r2102jfd-myProject-1.0.0
/nix/store/34234sdfjskdfj32j4kjdsf-python311-3.11.0
..
/nix/store/fsdkf234jdfdsfjkj0111df-poetry-1.6.1
/nix/store/sdf34dfkjlkj09u123123ss-poetry-1.6.1
..
```

```
$ nix develop flake.nix    # load environment with buildInputs
```

DEMO

NIXOS

```
{
  boot.kernelPackages = pkgs.linuxPackages_6_5;
  users.users.mneuss = { home = "/home/mneuss"; ...};

  environment.systemPackages = with pkgs; [
    jetbrains.idea-ultimate
    python311
  ];

  services.xserver.desktopManager.gnome3.enable = true;
  networking.firewall.allowedTCPPorts = [ 80 ];
  ...
}
```

```
nixos-rebuild switch ...
```

FURTHER STUDY

- **Main Page**
  https://nixos.org/
- **Nix packages**
  https://github.com
  /NixOS/nixpkgs
- **My Blog**
  https://lambdablob.cor
- **Switch Dev Environments**
  https://direnv.net/